TP22

Processus de Markov

Un processus de Markov est une suite d'expériences aléatoires où la distribution conditionnelle de probabilités des états futurs ne dépend que de l'état présent et non des états passés. On parle de processus sans mémoire. En d'autres termes, pour de tels processus, si on connait le présent, la connaissance du passé n'apporte pas d'information supplémentaire utile pour la prédiction du futur.

Exercice 1 Doudou, un hamster paresseux, ne connait que trois endroits dans sa cage : les copeaux où il dort, la mangeoire où il mange et la roue où il fait de l'exercice. Ses journées sont assez semblables les unes des autres. Toutes les minutes, il peut soit changer d'activité, soit continuer celle qu'il était en train de faire. L'appellation de processus sans mémoire n'est pas du tout exagérée pour parler de Doudou.

- A l'instant 0, il dort.
- Quand il dort, il a 9 chances sur 10 de ne pas se réveiller la minute suivante.
- Quand il se réveille, il y a 1 chance sur 2 qu'il aille manger et 1 chance sur 2 qu'il parte faire de l'exercice.
- Le repas ne dure qu'une minute. Après avoir mangé, il y a 3 chances sur 10 qu'il parte courir dans sa roue et 7 chances sur 10 qu'il retourne dormir.
- Courir est fatiguant, il y a donc 8 chances sur 10 qu'il retourne dormir au bout d'une minute. Sinon il continue en oubliant qu'il est déjà fatigué.

Pour tout $n \in \mathbb{N}$, on note D_n l'événement "Doudou dort à l'instant n" et $d_n = P(D_n)$, M_n l'événement "Doudou mange à l'instant n" et $m_n = P(M_n)$, et E_n l'événement "Doudou fait de l'exercice" et $e_n = P(E_n)$.

1. (a) Exprimer en notations mathématiques les probabilités conditionnelles données par l'énoncé.

   (b) Démontrer les relations suivantes :

$$\begin{cases} d_{n+1} &= \dfrac{9}{10}d_n + \dfrac{7}{10}m_n + \dfrac{8}{10}c_n \\ m_{n+1} &= \dfrac{1}{20}d_n \\ e_{n+1} &= \dfrac{1}{20}d_n + \dfrac{3}{10}m_n + \dfrac{2}{10}e_n \end{cases}$$

   (c) On définit le vecteur colonne $U_n = \begin{pmatrix} d_n \\ m_n \\ e_n \end{pmatrix}$.

   Expliciter une matrice A telle que $U_{n+1} = AU_n$.

   On dit que A est la matrice de transition associée au processus de Markov.

   (d) Montrer que, pour tout entier naturel n, $U_n = A^n U_0$. Préciser U_0.

2. (a) Écrire un programme Scilab qui, étant donné un entier naturel n, calcule le vecteur U_n.

   (b) Tester pour différentes valeurs de n. Que remarque-t-on ?

3. (a) Nous allons représenter sur Scilab l'état D par 1, l'état M par 2 et l'état E par 3.

   Compléter le programme suivant pour qu'il simule la suite des états $X_0, X_1, \ldots, X_n$ et les mémorise dans un vecteur X :

```
n=input('Entrer n: ')
X=zeros(1,n+1)
X(1)=1
for k=1:n do
    if X(k)==1 then
        if rand()<9/10 then
            X(k+1)=***
        else
            if rand()<1/2 then
                X(k+1)=***
            else
                X(k+1)=***
            end
        end
    elseif X(k)==2 then
        if *** then
            X(k+1)=***
        else
            X(k+1)=***
        end
    else
        if *** then
            X(k+1)=***
        else
            X(k+1)=***
        end
    end
end
disp(X)
```

(b) Comme on peut le voir à la question précédente, la simulation de processus de Markov est rapidement très lourde (et il y a seulement 3 états).

Scilab propose une fonction qui permet d'éviter ce problème : Si X_0 est l'état initial et si A est la matrice de transition associée au processus de Markov, l'instruction `X=grand(n,'markov',A',X0)` donne un vecteur ligne X représentant la simulation des états $(X_1, X_2, \ldots, X_n)$.

Simuler l'expérience pour $n = 10$, pour $n = 20$, pour $n = 50$.

Attention : On doit mettre A' la transposée de A car la définition d'une matrice de transition en Scilab est la transposée de notre définition.

4. (a) Entrer les instructions suivantes dans l'éditeur de Scilab :

```
n=input('Entrer n: ')
p=zeros(1,n)
A=[0.9,0.7,0.8;0.05,0,0;0.05,0.3,0.2]
for k=1:n do
    X=grand(60,'markov',A',1)
    p(k)=X(60)
end
disp(p)
```

Tester pour différentes valeurs de n. Que représente le vecteur p ?

(b) Taper l'instruction `M=tabul(p,'i')`. Que fait cette instruction ? En déduire la fréquence de chacun des états de Doudou au bout d'une heure avec 1000 simulations.

Comparer avec le résultat obtenu à la question 2.

Exercice 2 Un étudiant décide de réviser ses cours en vue d'un concours, et dispose pour cela de trois choix : réviser les mathématiques, l'économie ou l'anglais.
On observe pendant un certain temps les efforts de l'étudiant et on constate les phénomènes suivants :

- A l'heure numéro 0, l'étudiant travaille l'économie.

- S'il révise les mathématiques à l'heure n, alors à l'heure $n+1$ il révisera encore des mathématiques avec une probabilité $\frac{1}{2}$ et sinon il révisera l'économie (mais pas l'anglais).
- S'il révise l'économie à l'heure n, il aura trois chances sur quatre de réviser les mathématiques l'heure suivante et une chance sur quatre de réviser l'anglais.
- S'il révise l'anglais à l'heure n, il se consacrera l'heure suivante à l'économie ou à nouveau à l'anglais, avec une probabilité de $\frac{1}{2}$ pour chaque.

On note A_n l'événement "l'étudiant révise les mathématiques à l'heure n", B_n l'événement "l'étudiant révise l'économie à l'heure n" et C_n l'événement "l'étudiant révise l'anglais à l'heure n".

Par ailleurs, on pose $a_n = P(A_n)$, $b_n = P(B_n)$, $c_n = P(C_n)$ et $U_n = \begin{pmatrix} a_n \\ b_n \\ c_n \end{pmatrix}$.

1. (a) Exprimer, pour tout entier naturel n, a_{n+1}, b_{n+1}, c_{n+1} en fonction de a_n, b_n, c_n.
   (b) Expliciter une matrice A telle que $U_{n+1} = AU_n$.
   (c) Montrer que, pour tout entier naturel n, $U_n = A^n U_0$. Préciser U_0.
2. Loi empirique de X_{10} :
   (a) Écrire un programme Scilab qui simule 100 chaînes de Markov $(X_1, X_2, \ldots, X_{10})$, enregistre dans un vecteur p la liste des états X_{10} constatés à la 10-ième heure et affiche p.
   (b) En utilisant l'instruction `tabul(p,'i')`, en déduire la fréquence de chacun des états au bout de 10 heures de travail.
3. Loi théorique de X_{10} :
   (a) Entrer les instructions suivantes dans l'éditeur de Scilab :

```
U=[0;1;0]
E=zeros(3,11)
E(:,1)=U
A=[1/2,3/4,0;1/2,0,1/2;0,1/4,1/2]
for k=2:11 do
    U=A*U
    E(:,k)=U
end
disp(E)
```

   Après avoir bien compris, étapes par étapes, ces instructions, exécuter le programme. En déduire le tableau de la loi théorique de X_{10}. Ce tableau est-il cohérent avec les résultats de la question 2 ?
   (b) Dans l'éditeur, à la suite des instructions précédentes, taper :

```
scf(1)
n=0:10
plot2d(n',[E(1,:)',E(2,:)',E(3,:)'])
title('les suites (an), (bn) et (cn)')
legend(['(an)';'(bn)';'(cn)'])
```

   Que peut-on en déduire ? Comparer avec les résultats de la question 2.
4. (a) Considérons la matrice $P = \begin{pmatrix} 3 & 3 & -1 \\ 2 & -4 & 0 \\ 1 & 1 & 1 \end{pmatrix}$. Calculer l'inverse de P. Vérifier le résultat avec Scilab.
   (b) Déterminer la matrice D telle que $D = P^{-1}AP$. Vérifier le résultat avec Scilab.
   (c) Montrer que, pour tout entier naturel n, $D^n = P^{-1}A^nP$.
   (d) En déduire A^n, puis a_n, b_n et c_n en fonction de n.
   (e) Calculer les limites lorsque n tend vers $+\infty$ de a_n, b_n et c_n.
   Comparer les résultats obtenus avec ceux des questions 2 et 3.