TP3_SIDM1

March 9, 2017

0.0.1 Université Paul Sabatier 2016-2017, M1 SID MES

Modélisation Stochastique - TP3

En tête pour charger les fonctions nécessaires au TP.

```
In [ ]: %matplotlib inline
        # coding: utf8
        from matplotlib.pyplot import *
        from math import *
        from numpy import *
        from numpy.random import *
        from numpy.linalg import *
        from scipy.misc import *
        #from scipy import *
```

Marche aléatoire simple sur $\mathbb{R}$ Soit $(\varepsilon_n)_{n\geq 0}$ une suite de variables aléatoires indépendantes et de même loi de Rademacher $\mathcal{R}(p)$ avec $0 < p < 1$ et, pour tout $n \geq 0$, soit

$$X_n := \varepsilon_0 + \varepsilon_1 + \cdots + \varepsilon_n.$$

Montrer que $(X_n)_{n\geq 0}$ est une chaîne de Markov homogène dont on précisera la matrice de transition Π. Le code Python permettant de simuler cette chaîne de Markov suit. Les valeurs du paramètre p et de l'horizon de temps sont affectées par l'utilisateur.

Pour quelles valeurs de p la chaîne est-elle irréductible?

Pour quelles valeurs de p la chaîne est-elle récurrente? Transiente?

Lorsque la chaîne est transiente, quel est son comportement asymptotique? Même question quand la chaîne est récurrente.

```
In [ ]: p=float(input('quelle est la valeur de p?'));
        N=int(input('Quel est l\'horizon de simulation?'));
        Z=rand(N,1);
        eps=2*(Z<p)-1;
        X=cumsum(eps);
        xx=linspace(1,N,N);
        plot(xx,X)
        title(u'marche aléatoire simple')
```

Transmission de l'information Une information sous la forme de oui ou non est transmise à travers n individus. On suppose que chaque intermédiaire transmet l'information avec la probabilité p et son contraire avec la probabilité $1-p$ où $0<p<1$. De plus, on suppose que les intermédiaires sont indépendants.

Modéliser cette situation par une chaîne de Markov $(X_n)_{n\geq 0}$ à deux états $E=\{-1,1\}$ et déterminer sa matrice de transition Π.

Calculer de deux manières différentes la probabilité q_n que le n-ème individu transmette fidèlement l'information initiale.

La question précédente est traitée numériquement dans le code qui suit. Expliquer précisément ce que fait ce programme.

Calculer la limite de q_n lorsque n tend vers l'infini.

Créer un code Python permettant d'illustrer cette convergence, où la valeur du paramètre p est affectée par l'utilisateur.

```
In [ ]: p=float(input('quelle est la valeur de p?'));
        N=int(input('Quel est l\'horizon?'));
        n=linspace(1,N,N);
        l=2*p-1;
        qn1=0.5*l**n+0.5;
        #print('première méthode');
        figure(1);
        plot(n,qn1,'r');
        title(u'méthode 1')
        PI=np.matrix([[p,1-p],[1-p,p]]);
        Q=np.matrix([1,0])
        qn2=zeros([1,N])
        for i in range(1, N+1):
            Q=Q*PI
            qn2[0,i-1]=Q[0,0]
        qn2=qn2[0,:];
        figure(2)
        title(u'méthode 2')
        plot(n,qn2,'b');
```

Chaîne à deux états On considère la chaîne de Markov $(X_n)_{n\geq 0}$ d'espace d'états $E=\{0,1\}$ et de matrice de transition

$$P=\begin{pmatrix} 1-a & a \\ b & 1-b \end{pmatrix}$$

avec $0<a,b<1$. Montrer que

$$P^n=\frac{1}{a+b}\begin{pmatrix} b & a \\ b & a \end{pmatrix}+\frac{(1-a-b)^n}{a+b}\begin{pmatrix} a & -a \\ -b & b \end{pmatrix}$$

puis déterminer la limite de P^n lorsque n tend vers l'infini. Calculer l'unique probabilit'e invariante μ de la chaîne de Markov $(X_n)_{n\geq 0}$. Si $S_n=X_1+X_2+\cdots+X_n$, montrer que

$$\lim_{n\to\infty}\frac{S_n}{n}=\frac{a}{a+b}.$$

Créer un code Python permettant de simuler cette chaîne de Markov et d'illustrer ce résultat de convergence, où les paramètres a et b sont affectés par l'utilisateur.

Une chaîne à quatre états On considère la chaîne de Markov $(X_n)_{n\geq 0}$ d'espace d'états $E = \{1,2,3,4\}$ et de matrice de transition

$$\Pi = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Calculer l'unique probabilité invariante μ de la chaîne de Markov $(X_n)_{n\geq 0}$. A partir du théorème ergodique, montrer que

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n} X_k = a \text{ et } \lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n} X_k^2 = b \qquad \text{p.s.}$$

avec a et b à déterminer. Expliquer et interpréter le code Python suivant.

```
In [ ]: Pi=np.matrix([[0,1,0,0], [0.5,0,0.25,0.25],[0.5,0.5,0,0],[0,0,1,0]]);
        Pi1=Pi.transpose();
        propre=eig(Pi1)[1];
        propre1=propre[:,0]
        mu=abs(propre1)/sum(abs(propre1));
        print('Probabilité invariante')
        print(mu.transpose());
        print('Transition puissance 100')
        print(Pi**100);
        N=int(input('Quel est l\'horizon?'));
        X0=int(input('Quel est le point initial?'));
        X=zeros([1,N+1]);
        X[0,0]=X0;
        for i in range(1, N+1):
            p1=Pi[X0-1,0];
            p2=p1+Pi[X0-1,1];
            p3=p2+Pi[X0-1,2];
            u=rand(1,1);
            X0=4
            if u<p1:
                X0=1;
            elif  u<p2:
                X0=2
            elif  u<p3:
                X0=3
            X[0,i]=X0
        X=X[0,:];
        n=linspace(0,N,N+1);
        figure(1);
        plot(n,X,'g');
        title('Une trajectoire')
        m1=mean(X)
        m2=mean(X*X)
```

```
m1theo=mu[0]+2*mu[1]+3*mu[2]+4*mu[3];
m2theo=mu[0]+4*mu[1]+9*mu[2]+16*mu[3];
print('valeur empirique de la moyenne')
print(m1)
print(u'valeur théorique de la moyenne')
print(m1theo[0,0])
print('valeur empirique du moment d\'ordre 2' )
print(m2)
print(u'valeur théorique du moment d\'ordre 2')
print(m2theo[0,0])
```

Chaîne d'Ehrenfest Soit d boules numérotées de 1 à d avec $d > 1$, réparties dans deux urnes A et B. On tire un nombre i au hasard entre 1 et d et la boule numéro i est changée d'urne. On note X_n le nombre de boules dans l'urne A après n tirages indépendants. Montrer que $(X_n)_{n\geq 0}$ est une chaîne de Markov homogène d'espace d'états fini $E = \{0, 1, \cdots, d\}$, appelée chaîne d'Ehrenfest. Déterminer sa matrice de transition Π ainsi que son unique probabilité invariante μ. Montrer qu'il existe deux constantes $a, b \in \mathbb{R}$ telles que, $\forall x \in E$,

$$\sum_{y\in E} yP(x,y) = ax + b.$$

En déduire que $\mathbb{E}[X_n|X_0] \to d/2$. Créer un code Python permettant de simuler une chaîne d'Ehrenfest et d'illustrer ce résultat de convergence.

Algorithme de Metropolis-Hastings Soit E un espace d'états fini et μ une mesure de probabilité sur E qui charge tous les états. L'algorithme de Metropolis-Hastings pour un noyau auxiliaire de transition Q (tel que $Q(x, y) = 0$ si et seulement si $Q(y, x) = 0$) s'écrit:

Etape 0:
Initialiser X_0 ;
Etape n+1 :
Choisir y *selon la loi* $Q(X_n, y)$;
Poser $X_{n+1} = y$ *avec proba* $\min\left(1, \frac{\mu(y)Q(y,X_n)}{\mu(X_n)Q(X_n,y)}\right)$, *sinon poser* $X_{n+1} = X_n$

Montrer que si la probabilité d'acceptation-rejet $\min\left(1, \frac{\mu(y)Q(y,x)}{\mu(x)Q(x,y)}\right)$ est remplacée par

$$\frac{\mu(y)Q(y,x)}{\mu(y)Q(y,x) - \mu(x)Q(x,y)},$$

la chaîne de Markov définie par l'algorithme de Metropolis-Hastings aura encore μ pour mesure invariante.

On remplace plus généralement la probabilité d'acceptation-rejet $\min\left(1, \frac{\mu(y)Q(y,x)}{\mu(x)Q(x,y)}\right)$ par

$$\alpha\left(\frac{\mu(y)Q(y,x)}{\mu(x)Q(x,y)}\right),$$

avec $\alpha : \mathbb{R}_+ \to]0, 1]$. Donner une condition suffisante sur la fonction α pour que la mesure μ soit la mesure invariante de la chaîne de Markov définie par l'algorithme de Metropolis-Hastings.

Recuit simulé - le voyageur de commerce On considère le problème du voyageur de commerce suivant. Un commerçant doit visiter N clients dans N villes différentes puis revenir à son point de départ (en ne visitant qu'une seule fois chaque ville). On cherche à minimiser la distance totale parcourue.

On représente les villes par N points $v_1, \ldots, v_N$ dans $[0,1] \times [0,1] \subset \mathbb{R}^2$, un trajet possible par une suite $(v_{i(1)}, \ldots, v_{i(N)})$ de points et la distance à minimiser est

$$d\Big((v_{i(1)}, \ldots, v_{i(N)})\Big) = \sum_{k=1}^{N} |v_{i(k)} - v_{i(k+1)}|,$$

où $v_{i(N+1)}$ est $v_{i(1)}$.

Justifier que $N!$ est le nombre de trajets possibles.

Simuler la position de $N = 10$ villes au hasard dans $[0,1] \times [0,1]$.

] Ecrire l'algorithme de recuit simulé pour le noyau d'exploration Q qui consiste à échanger 2 villes au hasard dans un trajet $(v_{i_1}, \ldots, v_{i_N})$.

Représenter graphiquement les étapes de l'algorithme en reliant les villes des parcours.

```
In [ ]:
```