Probabilité
QFM 2023 _UM6P

Probabiliés appliquées

Elhoucine AIT BOUGNSA
Elhoucine.AITBOUGNSA@um6p.ma

1 Simulation des V.A à valeurs dans un ensemble fini et les chaînes de Markov

1.1 généralités sur les chaînes de Markov

Soient $E = \{1, 2, 3, ..., m\}$ et (X_n) une chaîne de Morkov homogène et de matrice de transition P

$$P_{i,j} = \mathbb{P}\left(X_{n+1} = j / X_n = i\right)$$

Autrement dit : la i^{eme} ligne de la matrice P représente la loi conditionnelle de X_{n+1} sachant que $X_n = i$

1. Réaliser une fonction python $accessible_N(i, j, P, N)$ qui demande comme arguments l'état i, l'état j , la matrice de passage P et un entier N. la fonction $accessible_N(i, j, P, N)$ doit retourner 1 s'il existe un chemin de longueur N de i vers j et 0 sinon. (1 si j est accessible à partir de i et 0 sinon)
2. Réaliser une fonction python $accessible(i, j, P)$ qui demande comme arguments l'état i, l'état j et la matrice de passage P. la fonction $accessible(i, j, P)$ doit retourner 1 s'il existe un chemin de i vers j et 0 sinon.(1 si j est accessible à partir de i et 0 sinon)
   Indication : si il existe un chemin de i vers j alors il existe au moins un chemin de longueur inférieur à m (avec m le nombre d'états , $cardinal(E)$)
3. Réaliser une fonction $recurrente(i, P)$ qui demande comme argument un état i et la matrice de passage P et qui retourne 1 si l'état i est récurrente et 0 sinon
   Indication : Il faut trouver la liste A des états accessibles à partir de i. Pour que i soit récurrente il faudrait que i soit accessible à partir de chaque état dans la liste A
4. Réaliser une fonction python $classes(P)$ qui demande comme argument la matrice de passage P et qui retourne les classes de la chaîne (X_n)
5. Créer une fonction python $recurrente(P)$ qui demande comme argument la matrice de transition P et qui retourne les états récurrents

1.2 Exemple de simulation et vérification du théorème Ergodic :

On donne dans cette partie :

$$P = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0.2 & 0.3 & 0 & 0.5 \\ 0.1 & 0 & 0 & 0.9 \\ 0 & 0.3 & 0.7 & 0 \end{bmatrix}$$

Soit $\pi_0 = [0.2, 0.1, 0.4, 0.3]$ la loi de X_0

1. Créer une fonction python $simulationX(S)$ qui demande comme argument une liste $S = [s_1, ..., s_m]$ et qui caractérise (comme retour) la simulation d'une variable aléatoire définie par : $X(\Omega) = \{1, 2, ..., m = card(S)\}$ et $\mathbb{P}(X = i) = s_i$
   indication : si U suit la loi uniforme sur $]0, 1[$, la V.A $\min\{1 \leq k \leq m / \sum_{i=1}^{k} s_i > U\}$ suit même loi que X

2. En utilisant la fonction précédente , réaliser 100000 observation selon X_0. tester la loi des grandes nombres (comparer entre $\mathbb{P}(X_0 = i)$ et la fréquence de i dans l'enchantant obtenu).(L'utilisation de l'histogramme est largement suffisant)
3. Créer une fonction python $Trajectoire(l)$ qui demande comme argument un entier l et qui retourne une liste $[X_0, X_1, ...X_l]$ (cette fonction nous donne une trajectoire la la chaîne de Markov (X_n) de 0 jusqu'à l
4. l'objectif de cette question est de vérifier le théorème Ergodique

   Créer une fonction python qui donne une approximation de la mesure invariante π. indication : Soit $\pi = [\pi_1, ..., \pi_m]$, on a :

$$\frac{1}{n}\sum_{k=1}^{n} \mathbb{1}_j(X_k) \longrightarrow \pi_j$$

5. Créer une fonction python $T(i)$ qui demande comme argument un état i et qui caractérise le temps de premier retour à l'état i :

$$T(i) = \min(n > 0 / X_n = i \textit{ sachant que } X_0 = i)$$

6. En utilisant la loi des grandes nombres, donner une approximation de $E\left(T(i)\right)$. Comparer entre π_i et $\frac{1}{E(T(i))}$ pour chaque $1 \leq i \leq 4$

2 Simulation des lois continues, application de Monte Carlo

2.1 Simulation d'un couple

Soit (X, Y) un couple qui suit la loi uniforme sur B avec :

$$B = \{(x, y) \in \mathbb{R}^2 / -\frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} \leq y \leq e^{-|x|}\}$$

(a) On vous donne que :

$$f_X(t) = \frac{1}{3}\left(\frac{e^{-\frac{t^2}{2}}}{\sqrt{2\pi}} + e^{-|t|}\right)$$

. En utilisant la méthode de loi inverse, réaliser la simulation de la loi de densité $f(t) = \frac{1}{2}e^{-|t|}$.
En utilisant la méthode mixte et Box-muller réaliser la simulation de X puis la simulation de (X, Y)

(b) En utilisant la méthode de Monte Carlo, donner une approximation de :

$$\iint_B e^{-|x.y|} dx.dy$$

2.2 La croissance Bactérienne :Escherichia coli (EC)

Dans un milieu qui contient une réserve de substance, chaque cellule bactérienne se devise en 2 après une durée aléatoire d'une moyenne de 20 minutes. On suppose que ce temps d'attente suit la loi exponentielle de paramètre $\frac{1}{20}$. à l'instant 0 le milieu contient une seule cellule.

1. Réaliser la simulation de la loi exponentielle de paramètre $\frac{1}{20}$. (Donner l'instant de premier division)
   densité de la loi exponentielle de paramètre λ

$$f_X(t) = \lambda e^{-\lambda.t} \mathbb{1}_{t>0}$$

2. Après la première division, nous aurons deux cellules bactériennes, 1 et 2. Il faudra attendre une durée aléatoire pour que la première se divise en deux et une autre durée aléatoire pour la deuxième. Après la naissance de chaque cellule, il faudra attendre une durée aléatoire $X()$.
   Donner l'instant auquel la cellule 1 va se deviser et l'instant correspondante à la deuxième cellule.
3. Donner tous les instants, entre 0 et 1000, auxquels une division cellulaire aura lieu.
4. Dans chaque division on aura une augmentation de nombre de cellules dans le milieu par 1. Soit $N(t)$ le nombre de cellules dans le milieu à l'instant t, tracer la courbe $(t, N(t))$ avec $0 \leq t \leq 1000$