Feuille de travaux pratiques # 4

Ce document a pour but de rappeler quelques éléments de base concernant l'implémentation des chaînes de Markov dans Scilab et l'illustration de certaines de leurs propriétés, qualitatives ou quantitatives. Par chaîne de Markov, on entend ici chaîne de Markov homogène à temps discret et à espace d'états également discret, au plus dénombrable.

1 Simulation des chaînes de Markov en Scilab

Étant donnés un espace d'états au plus dénombrable E et une matrice stochastique Q indexée par E, il existe essentiellement deux façons de générer à l'aide de `Scilab` des trajectoires d'une chaîne de Markov $(X_n)_{n\geq 0}$, à valeurs dans E et de matrice de transition Q.

1.1 Espace d'états fini de petit cardinal

Si l'espace d'états E est fini, de petit cardinal $d := \#E$ et si la matrice Q est facilement implémentable dans `Scilab`, la façon la plus simple et économique de simuler des trajectoires de la chaîne $(X_n)_{n\geq 0}$ consiste à utiliser le générateur `grand(`n`,'markov',`Q`,`x_0`)`. Si l'on fait appel à cette commande avec une matrice stochastique $Q \in \mathcal{M}_d(\mathbb{R})$, `Scilab` assimile automatiquement l'espace E à l'espace $E' := \{1, 2, \ldots, d\}$ et renvoie directement n états successifs $(X_1, \ldots, X_n)$ d'une chaîne de Markov à valeurs dans E', de matrice de transition Q et issue de $X_0 = x_0 \in E'$.

Par la même commande, on peut simuler simultanément m trajectoires de la même chaîne de Markov en prenant pour $x_0 = (x_0^1, \ldots, x_0^m)'$ un vecteur (colonne) de longueur m. La commande `grand(`n`,'markov',`Q`,`x_0`)` génère alors une matrice à m lignes et n colonnes, la $i-$ième ligne de cette matrice représentant une trajectoire de la chaîne issue de x_0^i.

Remarque 1 *La sortie de `grand(`n`,'markov',`Q`,`x_0`)` est la suite $(X_1, \ldots, X_n)$ de longueur n. Si l'on souhaite afficher toute la trajectoire $(X_0, X_1, \ldots, X_n)$, on pourra considérer la commande* `res= [x_0, grand(n,'markov',Q,x_0)]`

Considérons par exemple le cas où l'espace d'états $E = \{\text{oui}, \text{non}, \text{ne se prononce pas}\}$ est de cardinal 3 et où la matrice de transition est donnée par

$$Q = \begin{pmatrix} 7/10 & 2/10 & 1/10 \\ 1/10 & 8/10 & 1/10 \\ 1/10 & 1/10 & 8/10 \end{pmatrix}.$$

Dans cet exemple, on a alors $E' := \{1, 2, 3\}$ avec les correspondances naturelles entre états $1 = $ oui, $2 = $ non et $3 = $ ne se prononce pas. Dans ce cadre, la réponse `ans = 1. 1. 3. 2.` à la commande `traj=[1 grand(3,'markov',Q,1)]` correspond à la trajectoire (oui, oui, ne se prononce pas, non) dans E qui a pour probabilité

$$\mathbb{P}_{\text{oui}}\left(X_1 = \text{oui}, X_2 = \text{ne se prononce pas}, X_3 = \text{non}\right) = Q_{11}Q_{13}Q_{32} = \frac{7}{10} \times \frac{1}{10} \times \frac{1}{10} = \frac{7}{1000}.$$

1.2 Espace d'états de grand cardinal ou infini

Lorsque l'espace d'états E est de très grand cardinal (resp. s'il est infini), il est souvent difficile (resp. impossible) d'implémenter la matrice de transition Q dans `Scilab` sinon à travailler coefficient par coefficient. Dans ce cas, il est toujours possible de construire de "proche en proche" une trajectoire de la chaîne de Markov $(X_n)_{n\geq 0}$ en utilisant une formule de récurrence explicite du type

$$X_{n+1} = f(X_n, U_{n+1}),$$

où f est une fonction mesurable explicite et (U_n) une suite d'innovations indépendantes et identiquement distribuées.

Remarque 2 *Cette méthode de "proche en proche" permet par ailleurs de prendre en compte les chaînes de Markov inhomogènes, vérifiant des relations de récurrences du type*

$$X_{n+1} = f_n(X_n, U_{n+1}),$$

où les fonction f_n sont des fonctions mesurables explicites. Une telle chaîne inhomogène à valeurs dans E (fini ou infini) peut en effet toujours être vue comme une chaîne homogène à valeurs dans l'espace d'états élargi $\mathbb{N} \times E$.

Considérons l'exemple d'une marche aléatoire symétrique dans $E = \mathbb{N}$, réfléchie en zéro. La matrice de transition Q, bien que très simple, est infinie, ce qui rend impossible l'utilisation de la fonction `grand(`n`,'markov',`Q`,`x_0`)`. Néanmoins, on a la relation

$$X_{n+1} = \begin{cases} X_n + U_{n+1}, & \text{si } X_n > 0 \\ X_n + 1, & \text{si } X_n = 0, \end{cases}$$

où les variables (U_n) sont indépendantes de loi $\mathbb{P}(U_n = 1) = \mathbb{P}(U_n = -1) = 1/2$.

Exercice 1 *Modèle de Wright-Fisher*
Soient k et N deux entiers tels que $0 < k < N$. On considère la chaîne de Markov $(X_n^N)_{n\geq 0}$ à valeurs dans $E := \{0, \ldots, N\}$, issue de $X_0^N := k$ et dont la matrice de transition est donnée par :

$$Q_{ij} = \mathbb{P}(X_{n+1}^N = j | X_n^N = i) = \binom{N}{j} \left(\frac{i}{N}\right)^j \left(1 - \frac{i}{N}\right)^{N-j}.$$

1. Écrire un programme qui étant donné un entier N, renvoie la matrice de transition Q de la chaîne ci-dessus.
2. En utilisant la fonction `grand`, écrire un programme qui prend en entrée les entiers k, N, n et qui génère et trace une trajectoire $(X_m^N)_{m=0\ldots n}$.
3. En se passant de la fonction `grand`, écrire un programme qui prend en entrée les entiers k, N, n et qui génère et trace une trajectoire $(X_m^N)_{m=0\ldots n}$.
4. Pour des trajectoires de longueur $n = 100$, à l'aide des functions `tic` et `toc`, comparer le temps de calcul des deux programmes précédents, ce pour différentes valeurs de N.

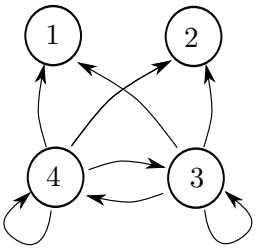
1.3 Générateur de matrices de transition

Pour tester des fonctions/programmes qui sont censés s'appliquer à des chaînes de Markov à espace d'états finis et de matrice de transition "quelconques", il est parfois commode d'avoir à sa disposition une matrice de transition générique. `Scilab` possède un générateur de telles matrices, auquel on fait appel via la commande `genmarkov`.

Les arguments d'entrée de `genmarkov` sont : un vecteur d'entiers strictement positifs contenant le nombre d'éléments de chacune des classes récurrentes, et entier représentant le nombre d'états transients de la chaîne. Ainsi `genmarkov([`$n_1, n_2, \ldots, n_p$`],`n_t`)` renvoie la matrice de transition d'une chaîne de Markov ayant p classes récurrentes, de cardinaux respectifs $n_1, n_2, \ldots, n_p$ et n_t états transients, communiquant tous entre eux. En particulier, la commande `genmarkov(d,0)` renvoie la matrice de transition d'une chaîne irréductible d'espace d'états $\{1, \ldots, d\}$.

Dans l'exemple ci-dessous, on a choisit $[n_1, n_2] = [1, 1]$ autrement dit, il y deux singletons formant deux classes de récurrence distinctes, i.e. deux états absorbants. De plus $n_t = 2$, donc il y a deux états transients (i.e. qui mènent aux états absorbants).

```
Q=genmarkov([1,1],2)
 Q  =
    1.          0.          0.          0.
    0.          1.          0.          0.
    0.0136343   0.3357779   0.3966634   0.2539243
    0.1902398   0.0966147   0.2539705   0.459175
```



2 Mise en évidence de propriétés qualitatives/quantitatives

L'étude des chaînes de Markov à espace d'états fini relève essentiellement de l'algèbre linéaire élémentaire. Les objets de base de `Scilab` étant des matrices, de nombreuses fonctions utiles pour l'études des propriétés qualitatives des chaînes de Markov sont implémentées par défaut dans le logiciel.

2.1 Récurrence et transience

Les propriétés de récurrence et de transience des états d'une chaîne de Markov sont des propriétés qualitatives fondamentales, qu'**il faut savoir déterminer et illustrer**. Étant donnée une matrice de transition $Q \in \mathcal{M}_d(\mathbb{R})$, la fonction `classmarkov(Q)` permet précisément d'effectuer automatiquement la classification des états de la chaîne de Markov associée.

La commande `[perm,rec,tr,indsRec,indsT]=classmarkov(P)` renvoie ainsi
- un entier `tr` $\leq d$, le nombre d'états transients
- un vecteur d'entiers strictement positifs `rec` les tailles des classes récurrentes
- un vecteur d'indices `indsRec` de $\{1, \ldots, d\}$, les indices des états récurrents
- un vecteur d'indices `indsT` de $\{1, \ldots, d\}$, les indices des états transients
- ainsi qu'une permutation `perm` $\in \mathfrak{S}_d$ permettant de mettre la matrice de transition Q sous forme canonique

Remarque 3 *Sur certaines versions récentes de `Scilab`, postérieures à la version 5.3, les fonctions `classmarkov` et `eigenmarkov` ne fonctionnent pas. Le bug est connu, Cf ce lien.*

Pour mettre en évidence, à l'aide de simulations, les états récurrents et transitoires d'une chaîne à espace d'états fini, on peut par exemple simuler une longue trajectoire de la chaîne et représenter la proportion de temps passée en chaque état. Cette proportion sera négligeable pour les états transitoires, contrairement aux états récurrents.

Exercice 2 À l'aide de la fonction `genmarkov`, considérer une matrice de transition Q d'une chaîne de Markov sur un espace d'états à 6 éléments dont deux sont transitoires et les quatre autres forment une unique classe de récurrence. À l'aide de la fonction `tabul`, représenter les fréquence empiriques associées à une longue trajectoire.

2.2 Mesure invariante et convergence à l'équilibre

Sur un espace d'états fini, lorsque qu'une chaîne de matrice de transition Q est irréductible, elle est automatiquement récurrente positive et admet ainsi une unique probabilité invariante π. Lorsque la chaîne est de plus apériodique, on sait qu'elle converge en loi vers la mesure invariante. Il existe différentes manières de mettre en évidence ce phénomène.

Exercice 3 À l'aide de la fonction `genmarkov`, considérer une matrice de transition Q d'une chaîne de Markov irréductible, récurrente positive sur un espace d'états à 5 éléments.

1. Écrire une fonction qui à partir de Q, détermine la probabilité invariante π. On pourra par exemple utiliser les fonctions `linsolve` ou `spec`.
2. Écrire une fonction qui mesure la distance d_n entre la matrice Q^n et la matrice dont les lignes sont égales à π. Tracer $\log(d_n)/n$ en fonction de n. Faire le lien avec la décomposition spectrale de Q obtenue grâce à la fonction `spec`.
3. À partir de la simulation d'une longue trajectoire de la chaîne, illustrer le théorème ergodique qui assure que presque-sûrement, pour toute fonction intégrable f

$$\lim_{n\to+\infty} \frac{1}{n}\sum_{k=1}^{n} f(X_k) = \int_E f(x)\pi(dx).$$

   Quel est l'ordre des fluctuations dans la limite ci-dessous ? Illustrer.
4. En utilisant la méthode de Monte-Carlo, mettre en évidence la convergence en loi

$$\lim_{n\to+\infty} \mathbb{P}(X_n = x) = \pi(x), \quad \forall x \in E.$$

   On pourra par exemple tracer sur le même graphique l'histogramme empirique et celui associé à la loi π. Illustrer de même la convergence en variation totale :

$$\lim_{n\to+\infty} \sum_{x\in E} |\mathbb{P}(X_n = x) - \pi(x)| = 0.$$

5. En utilisant la méthode de Monte-Carlo, estimer l'espérance du temps de retour en $x \in E$. Comparer à l'inverse de $\pi(x)$.

Prenez l'habitude de commenter vos codes et de légender vos dessins.