



N° Attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--



Année univ.: 2018/2019

Processus semi Markovien à temps discret

Mémoire présenté en vue de l'obtention du diplôme de

Master Académique

Université de Saida - Dr Moulay Tahar

Discipline : MATHÉMATIQUES

Spécialité : Analyse Stochastique, Statistique
des Processus et Applications

par

Farida Berga¹

Sous la direction de

Dr/Mme F. Mokhtari

Soutenu le 16/07/2019 devant le jury composé de

Mr. M. Laouni	Université Dr Tahar Moulay - Saïda	Président
Dr. F. Mokhtari	Université Dr Tahar Moulay - Saïda	Encadreur
Dr. F. Benziadi	Université Dr Tahar Moulay - Saïda	Examinatrice
Dr. S. Rahmani	Université Dr Tahar Moulay - Saïda	Examinatrice

1. e-mail : farida5berga@gmail.com



Remerciements

Je tiens à remercier **ALLAH** le tout puissant de m'avoir donné la foi et de m'avoir permis d'en arriver là. Je tiens à exprimer toute ma reconnaissance à Mme **Fatiha Mokhtari** ma directrice de mémoire qui est à l'origine de ce travail. C'est un honneur pour moi de travailler avec elle, parce qu'elle a partagé ses idées avec moi et, surtout, elle m'a transmis sa passion de la recherche et la motivation nécessaire pour mener à bien ce travail. Je la remercie aussi de son aide morale et scientifique durant les moments cruciaux de ce mémoire. Je remercie chaleureusement *M^{me}* Fatima Benziadi, *M^{elle}* Saadia Rahmani, *M^r* M.Laouni pour avoir accepté d'être les examinateurs et de faire partie de mon jury. Je tiens à remercier à titre individuel mes enseignants : *M^r* M.Kadi, *M^r* A.Kandouci, *M^{me}* F.Benziadi, *M^{elle}* L.Bousmaha, *M^{elle}* F. Benziadi, *M^{elle}* S.Rahmani, *M^{elle}* S.Idriss et *M^{me}* R.Rouane. Merci à mes chers amis pour les moments inoubliables qu'on a passé ensemble et à toute ma promotion 2014/2015. Merci à la flamme qui m'a toujours éclairée, à la plus parfaite des sœurs, qui m'a toujours orientée, soutenue, supportée, ma grande sœur Hassiba.

Merci à toute ma famille et tous ceux qui ont cru en moi et m'ont encouragée de près où de loin, je suis

honorée de vous voir fières

de moi.



♡ Farida

Résumé

Nous considérons des processus semi Markoviens à temps discret avec un espace d'état fini qui sont une généralisation des chaînes de Markov et des processus de renouvellement à temps discret. Un intérêt particulier est porté sur les problèmes de l'inférence statistique de ces processus. Nous présentons ensuite le package SMM du langage R, consacré à la simulation et à l'estimation des modèles semi Markoviens et Markoviens multi états à temps discret.

Title : Discret time semi Markov process

Abstract

We consider a discrete time semi Markov processes with a finite state space which are generalisation of Markov chains and discrete time renewal processes. An particular interest is focused on the statistical inference problems for this processes. Then we present the R package SMM, devoted to the simulation and estimation of discrete time multi state semi Markov and Markov models.

Mots clés :

Chaîne de Markov, matrice de transition, processus semi Markovien, distributions du temps de séjour, estimation, noyau semi Markovien.

Key words :

Markov chains, transition matrix, semi-Markov processes, sojourn time distributions, estimation, semi-Markov kernel.

Table des matières

Introduction	7
Notations	10
1 Chaîne de Markov	12
1.1 Généralités	13
1.1.1 Définitions et exemples	13
1.1.2 Graphe associé à une matrice de transition	15
1.2 Caractérisations d'une chaîne de Markov homogène	16
1.2.1 Résultats principaux	16
1.2.2 Classification d'états	18
1.3 Loi stationnaire	19
1.3.1 Définition	19
1.3.2 Existence et unicité	19
1.3.3 Convergence vers la loi stationnaire	20
1.4 Statistique d'une chaîne de Markov	22
2 Processus semi Markovien	24
2.1 Exemple	25
2.2 Modèle	27
2.3 Matrice de transition semi markovienne	28
2.4 Récurrence, irréductibilité	29
2.5 Temps de séjour	30
2.6 Estimation empirique	32
2.7 Equations de renouvellement semi-Markovien	36

3	Package SMM pour l'estimation et la simulation d'un modèle semi Markovien	37
3.1	Introduction	37
3.2	Simulation d'un modèle semi Markovien	39
3.2.1	Simulation selon les distributions classiques	39
3.2.2	Simulation selon les distributions données par l'utilisateur	42
3.3	Estimation d'un modèle semi Markovien	45
3.3.1	Estimation paramétrique d'un modèle semi Markovien	45
3.3.2	Estimation non paramétrique d'un modèle semi Markovien	50
3.4	Estimation d'une chaîne de Markov d'ordre k-th	53
3.5	Simulation d'une chaîne de Markov d'ordre k-th	56
3.6	Application sur des données réelles	58
3.6.1	Code génétique du virus HPV	58
3.6.2	Données sur l'insuffisance rénale	64
	Bibliographie	68

Introduction

Dans divers secteurs, tels que les centrales nucléaires et électriques, les réseaux de communication, les systèmes biologiques, la fiabilité des logiciels, les systèmes militaires, etc, les systèmes deviennent de plus en plus complexes. Au cours des dernières années, on constate un intérêt croissant pour l'évaluation de la performance des systèmes. L'évolution d'un système est modélisée par un processus stochastique à temps continu ou discret.

Parmi les modèles qui sont utilisés largement comme outil standard pour décrire l'évolution d'un système, nous avons les modèles Markoviens et les modèles semi-Markoviens. Ces derniers ont été introduits simultanément par Lévy [17] et Smith [24]. Les fondements de la théorie des processus semi-Markoviens ont été introduits par Pyke ([23], [22]) où le deuxième travail est consacré aux processus d'espace d'état fini. Les modèles semi-Markoviens sont plus généraux que les modèles Markoviens car ils ne sont pas limités par l'hypothèse Markovienne. L'hypothèse Markovienne est très forte puisque les lois des temps entre deux sauts successifs (dits temps de séjour) du système sont supposées exponentielles en temps continu (resp. géométriques en temps discret), ce qui caractérise l'absence de mémoire. En revanche, les modèles semi-Markoviens permettent de s'affranchir de l'hypothèse Markovienne. Pour les systèmes semi-Markoviens, les lois des temps de séjour peuvent être quelconques lois et l'évolution future dépend du temps passé au travers de la dernière transition.

Les processus semi-Markoviens constituent une famille de processus stochastiques englobant comme cas particuliers les processus de Markov et les processus de renouvellement ordinaires.

Un processus semi-Markovien peut être défini, également, par un processus bidimensionnel dont la première composante représente les états successivement visités du processus et la deuxième décrit les instants de changement d'état du processus. Ce processus bidimensionnel est appelé processus de renouvellement Markovien.

Comme nous l'avons mentionné précédemment, la transition vers le prochain état ne dépend pas de l'histoire du processus avant son entrée dans l'état actuel. Par conséquent, la première composante du processus de renouvellement Markovien forme une chaîne de Markov, dite chaîne de Markov immergée. Étant donné l'état du système, les temps de séjour, qui déterminent la deuxième composante de ce processus, sont indépendants puisque la loi du temps de séjour dans l'état actuel ne dépend pas de l'histoire du processus avant d'entrer dans cet état.

Par leur flexibilité, les modèles semi-Markoviens se sont avérés très fidèles aux systèmes réels, et cela dans de nombreux domaines (fiabilité des systèmes, évolution des populations, traitements médicaux, files d'attente, sismologie, etc.). Cependant, les systèmes semi-Markoviens présentent plusieurs inconvénients dont le majeur est la complexité des calculs de performances du système.

Outre la modélisation stochastique de l'évolution aléatoire d'un système, il est également important d'évaluer statistiquement la performance du système. L'inférence statistique est une problématique de l'étude des processus stochastiques en général, et des processus semi-Markoviens en particulier. Il existe une bibliographie riche sur l'estimation paramétrique et non-paramétrique des noyaux et des mesures concernant un système semi-Markovien.

La construction des estimateurs et l'étude de leurs propriétés asymptotiques, principalement la convergence (ou consistance) et la normalité asymptotique, font l'objet de plusieurs travaux de recherche. Les processus semi-Markoviens en temps discret ont été moins étudiés que ceux en temps continu mais, récemment, un intérêt croissant s'est porté sur leur inférence statistique.

Dans la littérature récente, les modèles semi-Markoviens à temps dis-

cret ont été largement utilisés pour la modélisation probabiliste et statistique, et en particulier ceux à espace d'état fini. D'abord, la théorie ergodique des chaînes semi-Markoviennes a été présentée par Anselone [1]. Gerontidis[14] a examiné les chaînes semi-Markoviennes dans les problèmes de remplacement. Puis, Barbu et al [6] ont fixé un cadre semi-Markovien à temps discret, ont proposé une méthode de calcul pour résoudre l'équation de renouvellement Markovien correspondante et, ensuite, ont déterminé quelques mesures de fiabilité. Divers aspects du modèle semi-Markovien à temps discret ont été présentés par Howard [16], Janssen et Manca [18] ainsi que leurs références. Par ailleurs, Vassiliou et Papadopoulou [22] ont étudié les systèmes semi-Markoviens non-homogènes à temps discret. Une présentation approfondie des chaînes semi-Markoviens homogènes d'espace d'état fini a été donnée par Barbu et Limnios [8], orientée vers les applications.

Dans ce mémoire, on considère uniquement des estimateurs non-paramétriques et on étudie leur comportement asymptotique.

Ce mémoire est organisé comme suit :

Dans le premier chapitre on commence par un rappel sur les chaînes de Markov. On passe ensuite à la classification d'états et on termine par l'inférence statistique de ces processus.

Le deuxième chapitre est dédié à l'étude des processus semi Markovien à temps discret. On présente quelques définitions puis on considère les estimateurs empiriques pour : la matrice de transition de la chaîne de Markov immergée, le noyau semi-markovien, la loi conditionnelle et la fonction de répartition conditionnelle du temps de séjour, enfin une solution de l'équation de renouvellement Markovien est donnée.

Le troisième chapitre présente des simulations numériques des processus semi Markoviens et Markoviens d'espace d'état fini en utilisant les packages SMM et MK du langage R. Des applications sur des données réelles (code génétique du virus HPV et données sur l'insuffisance rénale) sont fournies à la fin de ce chapitre.

Notations

Abréviations

CM	Chaîne de Markov.
CSM	Chaîne Semi-Markovienne.
CRM	Chaîne de Renouvellement Markovien.
CMI	Chaîne de Markov Immergée.
NSM	Noyau Semi-Markovien.
ERM	Équation de Renouvellement Markovien.
$v.a.$	variable aléatoire.

Probabilistes

$(\Omega, \mathcal{F}, \mathbb{P})$	espace probabilisé.
$\mathbb{P}$	probabilité.
$\mathbb{P}_i(\cdot)$	probabilité conditionnelle.
$\mathbb{E}$	espérance.
$Z := (Z_k)_{k \in \mathbb{N}}$	chaîne semi-Markovienne.
$(J, S) := (J_n, S_n)_{n \in \mathbb{N}}$	chaîne de renouvellement markovien.
$J := (J_n)_{n \in \mathbb{N}}$	chaîne de Markov immergée.
$S := (S_n)_{n \in \mathbb{N}}$	chaîne des instants de sauts.
$X := (X_n)_{n \in \mathbb{N}}$	temps entre deux sauts successifs (temps de séjour).
$\pi_0 := (\mathbb{P}_i; i \in \mathbb{E})$	loi initiale de la CSM et de la CMI.
$\pi := (\pi_i; i \in \mathbb{E})$	loi stationnaire de la CSM.
$Q := (Q(i, j); i, j \in \mathbb{E})$	matrice de transition de la CMI.

$q(\cdot) := (q_{ij}(\cdot); i, j \in \mathbb{E})$	noyau semi-markovien.
$N(M)$	nombre de sauts dans l'intervalle $[1, M]$.
$N_i(M)$	nombre de transitions vers l'état i dans l'intervalle $[1, M]$.
$N_{ij}(M)$	nombre de transitions de i vers j de la CSM dans l'intervalle $[1, M]$.
$N_{ij}(k, M)$	nombre de transitions de i vers j de la CSM dans, $[1, M]$ avec temps de séjour k unités dans l'état i .

Symboles divers

δ_{ij}	symbole de Kronecker.
$\mathbb{I}_A$	fonction indicatrice d'un sous-ensemble A .
$*$	produit de convolution.

Chapitre 1

Chaine de Markov

Lorsque deux systèmes identiques à un instant donné peuvent avoir des comportements différents dans le futur, on est amené à introduire une suite de variables aléatoires $(X_t)_t$ pour décrire leurs évolutions : X_t servent à définir l'état du système étudié à l'instant t . Si le système étudié est une population, l'état du système à un instant donné peut être décrit simplement par un nombre lorsqu'on s'intéresse uniquement à la taille de cette population, ou par un ensemble de nombres tels que l'ensemble des positions de chaque individu lorsqu'on s'intéresse à la répartition spatiale de la population.

On va ici se limiter à des systèmes dont l'état peut être décrit par une variable aléatoire ou un vecteur aléatoire discret.

En général l'évolution future d'un système dépendant au moins de son état présent, les variables aléatoires décrivant l'état du système à chaque instant ne pourront pas être considérées comme indépendantes. On va s'intéresser aux situations où l'évolution future d'un système ne dépend du passé qu'au travers de son état présent et pour simplifier, on n'étudiera pas l'évolution du système en temps continu, mais son évolution en une suite infinie d'instant $0 = t_0 < t_1 < \dots < t_n < \dots$

On travaillera donc avec une suite de variables aléatoires discrètes $(X_n)_{n \in \mathbb{N}}$ chaque variable aléatoire étant à valeurs dans un ensemble fini ou infini dénombrable noté E .

1.1 Généralités

1.1.1 Définitions et exemples

Définition 1.1.1. (Chaine de Markov)

Soit $X = \{X_n; n \geq 0\}$ une suite de variables aléatoires définies sur un même espace de probabilité $(\Omega, A, \mathbb{P})$ à valeurs dans $\mathbb{E}$ (**espace d'état**) un espace fini ou dénombrable. On dit que X est une chaine de Markov (CM) si, pour tout $i_1, i_2, \dots, i_{n+1} \in \mathbb{E}$, on a :

$$\mathbb{P}(\underbrace{X_{n+1} = i_{n+1}}_{\text{Le futur}} \mid \underbrace{X_1 = i_1, \dots, X_n = i_n}_{\text{Le passé et le présent}}) = \mathbb{P}(\underbrace{X_{n+1} = i_{n+1}}_{\text{Le futur}} \mid \underbrace{X_n = i_n}_{\text{Le présent}})$$

Définition 1.1.2. (Chaine de Markov homogène)

Une chaine de Markov est homogène si pour tout $n \geq 0$, i et j dans $\mathbb{E}$

$$\mathbb{P}(X_{n+1} = i \mid X_n = j) = \mathbb{P}(X_1 = i \mid X_0 = j).$$

Dans ce cas, on pose

$$Q(i, j) = \mathbb{P}(X_1 = i \mid X_0 = j), i, j \in \mathbb{E}.$$

$Q(i, j)$ est appelée la **probabilité de transition**.

Définition 1.1.3. (Matrice de transition)

La matrice $Q = (Q(i, j))_{i, j \in \mathbb{E}}$ est une matrice stochastique, c'est-à-dire :

pour tout $i, j \in \mathbb{E}$, $Q(i, j) \geq 0$ et pour tout $i \in \mathbb{E}$, $\sum_{j \in \mathbb{E}} Q(i, j) = 1$.

N.B. La loi de X_0 notée π_0 est appelée la **loi initiale** de la chaine de Markov.

Théorème 1.1. $\forall m \geq 0$ la probabilité de passage de i à j en m étapes est égale à l'élément (i, j) de la matrice Q^m (produit matriciel de Q m fois).

Exemple 1. (La souris dans le labyrinthe)

Une souris se déplace dans le labyrinthe de la Figure 1.1. Initialement, elle se trouve dans la case 1. A chaque minute, elle change de case choisissant, de manière équiprobable, l'une des cases adjacentes. Dès qu'elle atteint soit la nourriture (case 4), soit sa tanière (case 5), elle y reste.

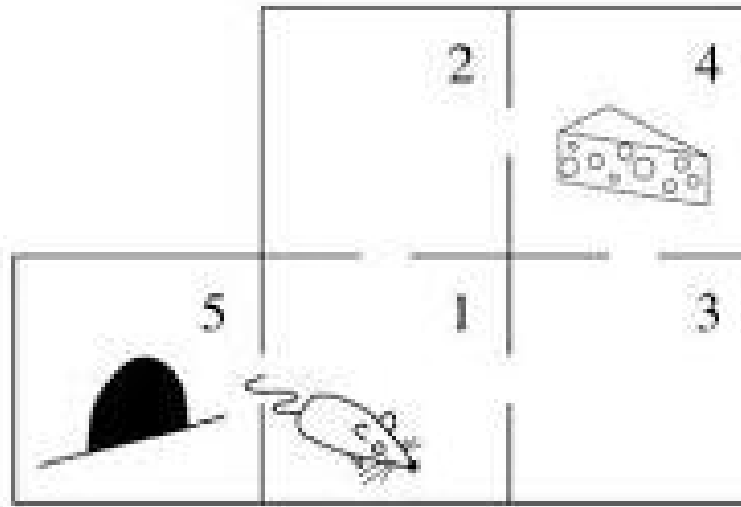


Figure 1.1 : Le labyrinthe dans lequel vit la souris.

Sa matrice de transition est comme suit :

$$\begin{pmatrix} 0 & 1/3 & 1/3 & 0 & 1/3 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Il est clair que la souris se retrouve dans sa tanière au bout d'une minute avec probabilité $1/3$. Sinon, elle passe soit dans la case 2, soit dans la case 3, et depuis chacune de ces cases elle a une chance sur deux de trouver la nourriture.

Il y a donc une probabilité de $1/6$ que la souris trouve la nourriture au bout de deux minutes. Dans les autres cas, elle se retrouve dans la case de départ, ce qui permet d'établir une formule de récurrence pour les probabilités cherchées.

Exemple 2. (Chaîne de Markov homogène)

Une grenouille monte sur une échelle. Chaque minute, elle peut monter d'un barreau avec probabilité $1/2$, ou descendre d'un barreau avec probabilité $1/2$. L'échelle a 5 barreaux. Si la grenouille arrive tout en haut, elle saute immédiatement en bas de l'échelle et recommence.

On appelle X_n la position de la grenouille sur l'échelle. L'espace d'état est donc $\mathbb{E} = \{0,1,2,..5\}$. Si à un instant n la grenouille est au niveau $x \in \{1,2,3,4\}$ de l'échelle, alors à l'instant $n+1$ elle sera sur barreau $x+1$ avec probabilité $1/2$, où sur barreau $x-1$ avec la même probabilité, ce qui se traduit :

$$\mathbb{P}(X_{n+1} = x + 1 | X_n = x) = 1/2 (= \mathbb{P}(X_1 = x + 1 | X_0 = x))$$

$$\mathbb{P}(X_{n+1} = x - 1 | X_n = x) = 1/2 (= \mathbb{P}(X_1 = x - 1 | X_0 = x))$$

les probabilités ne dépend pas de n , la matrice de transition est donnée par :

$$Q = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 & 0 & 0 \end{pmatrix}$$

Car si la grenouille se retrouve à l'état 5, alors elle peut soit passer à l'état 4, soit passer à l'état 0. La dernière ligne de la matrice est donc $(1/2, 0, 0, 1/2, 0)$ (encore une fois cela ne dépend pas de l'instant n).

Si la grenouille est à l'état 0, elle ne peut que passer à l'état 1. La première ligne de la matrice est donc $(0, 1, 0, 0, 0)$.

X_n est donc bien une chaîne de Markov homogène, avec matrice de transition Q .

1.1.2 Graphe associé à une matrice de transition

Définition 1.1.4. (Graphe associé à une matrice de transition)

Pour visualiser l'évolution d'une chaîne de Markov homogène, il est souvent utile de représenter la matrice de transition Q de la chaîne de Markov par un graphe orienté :

Les nœuds du graphe sont les états possibles pour la chaîne de Markov, une flèche allant de l'état i à l'état j indique qu'il y a une probabilité strictement positif que le prochain état de la chaîne soit l'état j si elle est actuellement dans l'état i . On met le poids $Q(i, j)$ à la flèche allant de l'état i à l'état j (Figure 1.2).

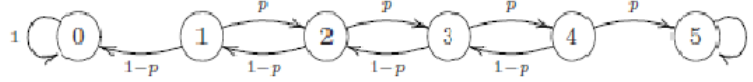


Figure 1.2 : graphe à 6 états.

1.2 Caractérisations d'une chaîne de Markov homogène

Soit $(X_n)_{n \in \mathbb{N}}$ une chaîne de Markov homogène, alors on a les propositions suivantes :

1.2.1 Résultats principaux

Proposition 1.2.1. Soit $i_0, \dots, i_n \in \mathbb{E}$. Pour tout $k \in \mathbb{N}$.

$$P(X_{n+k} = i_n, \dots, X_{k+1} = i_1 | X_k = i_0) = Q(i_0, i_1)Q(i_1, i_2)Q(i_2, i_3), \dots, Q(i_{n-1}, i_n).$$

Preuve 1. Posons $A_j = \{X_{k+j} = i_j\}$ pour tout $j \in 0, \dots, n$. On a :

$$P(A_n \cap \dots \cap A_1 | A_0) = P(A_n | A_{n-1} \cap \dots \cap A_0)P(A_{n-1} | A_{n-2} \cap \dots \cap A_0), \dots, P(A_1 | A_0).$$

Comme $(X_n)_n$ est une chaîne de Markov homogène de matrice de transition Q ,

$P(A_j | A_{j-1} \cap \dots \cap A_0) = Q(i_{j-1}, i_j)$ pour tout $j \in 1, \dots, n$, ce qui permet de conclure.

Proposition 1.2.2. La suite $(X_n)_{n \in \mathbb{N}}$ est une chaîne de Markov homogène si et seulement si il existe une matrice Q ayant la propriété suivante :

pour tout $n \in \mathbb{N}$ et pour tout $i_0, \dots, i_n \in \mathbb{E}$,

$$P(X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_0 = i_0)Q(i_0, i_1) \dots Q(i_{n-1}, i_n).$$

Dans ce cas, Q est la matrice de transition de la chaîne de Markov homogène $(X_n)_{n \in \mathbb{N}}$.

Preuve 2. - Supposons que $(X_n)_n$ soit une chaîne de Markov homogène. Notons

Q sa matrice de transition. D'après la proposition (1.2.1) avec $k = 0$,

$$\begin{aligned} P(X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) &= P(X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_1 = i_1 | X_0 = i_0) \\ &\times P(X_0 = i_0) \\ &= P(X_0 = i_0) Q(i_0, i_1) \dots Q(i_{n-1}, i_n). \end{aligned}$$

- Réciproquement, supposons que pour tout $n \in \mathbb{N}$ et pour tout $i_0, \dots, i_n \in \mathbb{E}$,

$$P(X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_0 = i_0) Q(i_0, i_1) \dots Q(i_{n-1}, i_n).$$

Pour tout $n \in \mathbb{N}$ et pour tout $i_0, \dots, i_{n+1} \in \mathbb{E}$ tels que

$$P(X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) > 0,$$

$$\begin{aligned} P(X_{n+1} = i_{n+1} | X_n = i_n, \dots, X_0 = i_0) &= \frac{P(X_{n+1} = i_{n+1}, X_n = i_n, \dots, X_0 = i_0)}{P(X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_0 = i_0)} \\ &= Q(i_n, i_{n+1}). \end{aligned}$$

Définition 1.2.1. (Temps de séjour)

Soit une chaîne de Markov homogène de matrice de transition $Q(i, j)$, on désigne par R_i la variable aléatoire égale au temps de séjour à l'état i . Pour tout $k \in \mathbb{N}^*$, $P(R_i = k) = P(X_{n+1} = i, \dots, X_{n+k} = i, X_{n+k+1} \neq i | X_n = i)$.

Proposition 1.2.3. Soit $(X_n)_{n \geq 0}$ une chaîne de Markov de fonction de transition Q . Alors :

$$P(R_i = k) = (1 - Q(i, i)) Q^k(i, i), \text{ (loi géométrique)}$$

et d'autre part, si $Q \neq 1$, nous avons, pour $j \neq i$:

$$P(X_{n+1} = j | X_n = i, X_{n+1} \neq i) = \frac{Q(i, j)}{1 - Q(i, i)}.$$

Supposons $Q(i, i) \neq 1$. Si à l'instant n la chaîne est en i , elle y reste encore une durée de loi géométrique de paramètre $1 - Q(i, i)$. Ici n est un instant fixé, non aléatoire. On ne peut en déduire pour l'instant que le temps de séjour dans l'état i est de loi géométrique de paramètre $1 - Q(i, i)$ car si l'état initial n'est pas i , l'instant d'entrée dans i est aléatoire.

1.2.2 Classification d'états

Définition 1.2.2. (Classe finale) Pour $(i, j) \in E^2$. On dit que j est accessible à partir de i si et seulement s'il existe $n \geq 0$ tel que : $\mathbb{P}(X_n = j | X_0 = i) > 0$. On dit que i et j communiquent si et seulement s'il existe $(n, m) \in \mathbb{N}^2$ tels que : $\mathbb{P}(X_n = j | X_0 = i) > 0$ et $\mathbb{P}(X_m = i | X_0 = j) > 0$.

La relation communiquer, notée $\longleftrightarrow$, est une relation d'équivalence. La relation être accessible, notée $\longleftarrow$ s'étend aux classes d'équivalence. Pour deux classes d'équivalence $\mathcal{C}$ et $\mathcal{C}'$ on a

$$\mathcal{C} \longleftarrow \mathcal{C}' \iff \{\exists (i, j) \in \mathcal{C} \times \mathcal{C}', i \longleftarrow j\} \iff \{\forall (i, j) \in \mathcal{C} \times \mathcal{C}', i \longleftarrow j\}.$$

Une classe est dite finale si elle ne conduit à aucune autre.

Définition 1.2.3. (Chaine de Markov absorbantes) Un état $i \in \mathbb{E}$ est dit **absorbant** si $\mathbf{Q}(i, i) = 1$ (et donc nécessairement $\mathbf{Q}(i, j) = 0$ pour tout $j \neq i$, si la chaîne entre dans cet état, elle y reste avec probabilité 1).

Une chaîne de Markov est dite **absorbante** s'il existe, pour tout état de $\mathbb{E}$, un état absorbant accessible depuis cet état. Dans une chaîne absorbante, tout état non absorbant est appelé **transient**.

Définition 1.2.4. (Chaine de Markov irréductible)

Une chaîne de Markov est dite **irréductible** si pour tout couple d'états différents i et j , i conduit à j c'est-à-dire s'il existe un chemin de longueur finie dans le graphe associé à la matrice de transition de la chaîne permettant de passer de i à j .

Définition 1.2.5. (Période, état apériodique)

La **période** d d'un état i d'une chaîne de Markov est égale au plus grand diviseur commun de tous les n tels que $\mathbf{Q}^n(i, i) > 0$ (la probabilité de passage de i à j en n étapes). Si $d > 1$, alors l'état i est **périodique**, sinon il est **apériodique**.

Définition 1.2.6. (Chaine Markovienne récurrente positive) Un état est dit **récurrent positif** si l'espérance du temps de premier retour en cet état, partant de cet état est fini. On parle de chaîne Markovienne **récurrente** si tous les états sont récurrents.

1.3 Loi stationnaire

1.3.1 Définition

Définition 1.3.1. (Loi stationnaire) Il peut exister une ou plusieurs mesures $\pi = (\pi_i)_{i \in E}$, sur l'espace d'état E , telles que : $\pi = \pi Q$, ou bien encore

$$\forall j \in E, \pi_j = \sum_{i \in E} \pi_i Q(i, j).$$

Une telle mesure π est appelée une **mesure stationnaire**. Une mesure stationnaire est une fonction propre de la transposée de la matrice de transition, associée à la **valeur propre** 1. Elle est appelée probabilité stationnaire ou loi stationnaire si elle remplit les conditions supplémentaires :

- $\forall i \in E, \pi_i \geq 0$,
- $\sum_{i \in E} \pi_i = 1$,

Le terme « stationnaire » est justifié par la proposition suivante :

Proposition 1.3.1. Si la loi initiale de la chaîne de Markov (i.e. la loi de X_0) est une probabilité stationnaire π , alors pour tout $n \geq 0$, la loi de X_n est encore π .

1.3.2 Existence et unicité

Théorème 1.3.2.1. Si une chaîne de Markov possède une seule classe finale C , alors il existe au plus une probabilité stationnaire. On a alors équivalence entre les 3 propositions :

- il existe une unique probabilité stationnaire, notée π ,
- il existe un état récurrent positif,
- tous les états de la classe finale sont récurrents positifs.

On a de plus l'équivalence $\{\pi_i > 0\} \Leftrightarrow \{i \in C\} \Leftrightarrow \{i \text{ est récurrent positif}\}$.

1.3.3 Convergence vers la loi stationnaire

Si la chaîne de Markov est irréductible, récurrente positive et apériodique, alors Q^k converge vers une matrice dont chaque ligne est l'unique distribution stationnaire π . En particulier, la loi μ_n de X_n converge vers π indépendamment de la loi initiale π_0 . Dans le cas d'un espace d'état fini, cela se prouve par le théorème de Perron-Frobenius. Notons qu'il est naturel que la suite μ_n , définie par récurrence par la relation $\mu_{n+1} = \mu_n Q$, ait, éventuellement, pour limite un point fixe de cette transformation, i.e. une solution de l'équation $\pi = \pi Q$.

Proposition 1.3.2. *Si une chaîne de Markov est irréductible et si son espace d'état est fini, tous ses états sont récurrents positifs.*

Exemple 3. (Doudou le hamster)

Doudou le hamster ne connaît 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 aux autres, et son activité se représente aisément par une chaîne de Markov. Toutes les minutes, il peut soit changer d'activité, soit continuer celle qu'il était en train de faire. L'appellation processus sans mémoire n'est pas du tout exagérée pour parler de Doudou.

- 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 il fait autre chose.
- Après avoir mangé, il y a 3 chances sur 10 qu'il parte courir dans sa roue, mais surtout 7 chances sur 10 qu'il retourne dormir.
- Courir est fatigant pour Doudou ; il y a 8 chances sur 10 qu'il retourne dormir au bout d'une minute. Sinon il continue en oubliant qu'il est déjà un peu fatigué.

Diagrammes. Les diagrammes peuvent montrer toutes les flèches, chacune représentant une probabilité de transition. Cependant, c'est plus lisible si :

- on ne dessine pas les flèches de probabilité zéro (transition impossible) ;

- on ne dessine pas les boucles (flèche d'un état vers lui-même). Cependant elles existent ; leur probabilité est sous-entendue car on sait que la somme des probabilités des flèches partant de chaque état doit être égale à 1.

Matrice de transition. La matrice de transition de ce système est la suivante (les lignes et les colonnes correspondent dans l'ordre aux états représentés sur le graphe par copeaux, mangeoire, roue) :

$$Q = \begin{bmatrix} 0,9 & 0,05 & 0,05 \\ 0,7 & 0 & 0,3 \\ 0,8 & 0 & 0,2 \end{bmatrix}$$

Prévisions. Prenons l'hypothèse que Doudou dort lors de la première minute de l'étude.

$$\mathbf{x}^{(0)} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

Au bout d'une minute, on peut prédire :

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)}Q = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0,9 & 0,05 & 0,05 \\ 0,7 & 0 & 0,3 \\ 0,8 & 0 & 0,2 \end{bmatrix} = \begin{bmatrix} 0,9 & 0,05 & 0,05 \end{bmatrix} \text{ Ainsi,}$$

après une minute, on a 90% de chances que Doudou dorme encore, 5% qu'il mange et 5% qu'il coure.

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)}Q = \mathbf{x}^{(0)}Q^2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0,9 & 0,05 & 0,05 \\ 0,7 & 0 & 0,3 \\ 0,8 & 0 & 0,2 \end{bmatrix}^2 = \begin{bmatrix} 0,885 & 0,045 & 0,07 \end{bmatrix}$$

Après 2 minutes, il y a 4,5 % de chances que le hamster mange.

De manière générale, pour n minutes :

$$\mathbf{x}^{(n)} = \mathbf{x}^{(n-1)}Q \text{ et } \mathbf{x}^{(n)} = \mathbf{x}^{(0)}Q^n.$$

La théorie montre qu'au bout d'un certain temps, la loi de probabilité est indépendante de la loi initiale.

Notons-la π : $\pi = \lim_{n \rightarrow \infty} \mathbf{x}^{(n)}$

On obtient la convergence si et seulement si la chaîne est **apériodique** et **ir-réductible**. C'est le cas dans notre exemple, on peut donc écrire :

$$Q = \begin{bmatrix} 0,9 & 0,05 & 0,05 \\ 0,7 & 0 & 0,3 \\ 0,8 & 0 & 0,2 \end{bmatrix}$$

$$\begin{aligned} \pi Q &= \pi \quad (\pi \text{ est la loi invariante par rapport } Q) \\ &= \pi I \\ \pi(I - Q) &= 0 \end{aligned}$$

$$\begin{aligned} \pi(I - Q) &= \pi \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0,9 & 0,05 & 0,05 \\ 0,7 & 0 & 0,3 \\ 0,8 & 0 & 0,2 \end{bmatrix} \right) \\ &= \pi \begin{bmatrix} 0,1 & -0,05 & -0,05 \\ -0,7 & 1 & -0,3 \\ -0,8 & 0 & 0,8 \end{bmatrix} \\ &= [\pi_1 \quad \pi_2 \quad \pi_3] \begin{bmatrix} 0,1 & -0,05 & -0,05 \\ -0,7 & 1 & -0,3 \\ -0,8 & 0 & 0,8 \end{bmatrix} \end{aligned}$$

Sachant que $\pi_1 + \pi_2 + \pi_3 = 1$, on obtient : $[\pi_1 \quad \pi_2 \quad \pi_3] = [0.884 \quad 0.0442 \quad 0.0718]$.

Doudou donc passe 88.4% de son temps à dormir !

1.4 Statistique d'une chaîne de Markov

On a vu que pour tout $n > 0$, la loi du vecteur aléatoire $(X_0, X_1, \dots, X_n)$ ne dépend que de la loi initiale π_0 et de la matrice de transition Q . La première question que l'on peut se poser en statistique des chaînes de Markov est : peut-on estimer le couple (π_0, Q) au vu de l'observation de la

suite $(X_0, X_1, \dots, X_n)$, d'une telle façon que l'erreur d'estimation soit "petite" quand n est "grand". Comme on va le voir ci-dessous, on peut estimer Q . Par contre, on ne peut raisonnablement estimer la loi initiale π_0 que si celle-ci coïncide avec la probabilité invariante, et que l'on est dans le cas irréductible et récurrent positif, ce que nous supposerons dans toute la suite de cette section.

Commençons par l'estimation de la probabilité invariante π_0 . Pour tout $i \in E$,

$$\mu_i^n = \frac{1}{n+1} \sum_{l=0}^n \mathbb{1}_{\{X_l=i\}}$$

est un estimateur consistant de π_0 .

Proposition 1.4.1. *Pour tout $i \in E$, $\hat{\mu}_i^n \rightarrow \pi_i$ p.s, quand $n \rightarrow \infty$.*

Passons maintenant à l'estimation de $Q(i, j)$, $i, j \in E$. On choisit l'estimateur

$$\hat{P}_{ij}^n = \frac{\sum_{l=0}^{n-1} \mathbb{1}_{\{X_l=i, X_{l+1}=j\}}}{\sum_{l=0}^{n-1} \mathbb{1}_{\{X_l=i\}}}.$$

Proposition 1.4.2. *Pour tout $i, j \in E$. $\hat{P}_{ij}^n \rightarrow Q(i, j)$ p.s quand $n \rightarrow \infty$.*

Chapitre 2

Processus semi Markovien

Les processus semi-Markoviens sont une classe de processus stochastiques qui généralisent les processus Markoviens et les processus de renouvellement. La flexibilité des processus semi-Markoviens par rapport aux processus Markoviens est due au fait que le temps de séjour en tout état peut être une variable aléatoire (v.a.) qui suit une loi continue (resp. une loi discrète) arbitraire, contrairement au cas Markovien où cette dernière est une loi exponentielle en temps continu (resp. la loi géométrique en temps discret).

En pratique, il y a des cas où l'approche Markovienne ne peut être acceptée. L'extension à un modèle semi-Markovien est alors plus naturelle puisqu'elle relaxe l'hypothèse Markovienne de base. Ces dernières années, l'intérêt pour les modèles semi-Markoviens s'est considérablement accru en partie dû à la diversité de ses applications.

Dans ce qui suit, nous considérons le modèle semi-Markovien à temps discret. On utilisera le terme de chaîne pour un processus à temps discret.

La première raison de notre intérêt pour le temps discret provient de certaines applications de Markov où l'échelle est intrinsèquement discrète. Par exemple, en théorie de la fiabilité, on pourrait s'intéresser au nombre de cycles effectués par un système où le nombre de fois (heures, jours, ect) qu'un évènement spécifique se produit. Notez également qu'en analyse de l'ADN, tout semi Markov naturel approchée est basée sur le temps discret parce que nous avons affaire à des séquences discrètes des quatre bases A,

C, G, T.

La deuxième raison repose sur la simplicité de la modélisation et du calcul dans le temps discret. Un processus semi Markovien à temps discret fait seulement un nombre limité de sauts dans un intervalle de temps fini. Pour cette raison, toute quantité d'intérêt dans un modèle semi Markovien à temps discret peut être exprimé comme une série finie de produits de convolution de noyau semi Markoviens au lieu d'une série infinie comme dans le cas continu.

2.1 Exemple

Usine de textiles

Soit $\mathcal{F}$ une usine de textiles, dont les déchets sont traités par l'unité de traitement $\mathcal{U}$ avant d'être jetés dans la rivière $\mathcal{R}$.

En cas de panne de $\mathcal{U}$, pour ne pas devoir arrêter la production, on a créé un dépôt de stockage des déchets $\mathcal{D}$ qui sert de zone tampon entre $\mathcal{F}$ et $\mathcal{U}$.

Si une panne intervient en $\mathcal{U}$ et qu'elle est réparée avant que $\mathcal{D}$ ne soit plein, alors $\mathcal{D}$ continue à tourner normalement, et on suppose que $\mathcal{D}$ se vide instantanément quand $\mathcal{U}$ est réparée. Si la panne n'est pas réparée à temps, alors $\mathcal{F}$ doit s'arrêter et la production cesse jusqu'à ce que $\mathcal{U}$ soit réparée.

Modélisons à présent le système décrit ci-dessus.

Soit $\mathbb{E} = \{1, 2, 3\}$, l'ensemble des états possibles du système :

- 1 : tout va bien ($\mathcal{F}$ et $\mathcal{U}$ marchent, $\mathcal{D}$ est vide).
- 2 : panne de $\mathcal{U}$ mais $\mathcal{D}$ n'est pas encore plein (donc $\mathcal{F}$ marche normalement).
- 3 : panne de $\mathcal{U}$, $\mathcal{D}$ est plein et $\mathcal{F}$ ne marche pas.

On voit que :

- $1 \rightsquigarrow 2$ (si panne de $\mathcal{U}$).
- $2 \rightsquigarrow 3$ (si $\mathcal{D}$ plein).

- $2 \rightsquigarrow 1$ (si panne de $\mathcal{U}$ résolue avant que $\mathcal{D}$ ne soit plein).
- $3 \rightsquigarrow 1$ (quand $\mathcal{U}$ est réparée).

On a donc un processus J_n , qui représente les états successifs du système, et qui est définie par :

- une loi initiale $\pi_0 = (\alpha_1, \alpha_2, \alpha_3)$, où $0 < \alpha_1, \alpha_2, \alpha_3$ et $\alpha_1 + \alpha_2 + \alpha_3 = 1$.
- une matrice de transition (comme 1 ne communique qu'avec 2, 2 avec 1 et 3, et 3 avec 1) :

$$P = \begin{pmatrix} 0 & 1 & 0 \\ 1/2 & 0 & 1/2 \\ 1 & 0 & 0 \end{pmatrix}$$

Toutefois, pour bien modéliser le système qui nous intéresse, les changements d'état ne suffisent pas : il nous faut en effet considérer le temps X_n que le système passe dans chaque état (d'où découlera par exemple une estimation de la production).

Supposons donc que l'on peut discrétiser le temps (par exemple en prenant pour unité de temps une heure), et que les changements d'état ne peuvent intervenir qu'à ces moments là.

Introduisons alors, $\forall k \in \mathbb{N}$:

$$\begin{pmatrix} 0 & f_{12}(k) & 0 \\ f_{21}(k) & 0 & f_{23}(k) \\ f_{31}(k) & 0 & 0 \end{pmatrix}$$

où :

- $f_{12}(\cdot)$ est la loi du temps que met $\mathcal{U}$ à tomber en panne.
- $f_{21}(\cdot)$ est la loi du temps que met $\mathcal{U}$ à être réparée.
- $f_{23}(\cdot)$ est la loi du temps de remplissage de $\mathcal{D}$.
- $f_{31}(\cdot)$ est la loi du temps nécessaire pour réparer $\mathcal{U}$, et donc que $\mathcal{F}$ fonctionne à nouveau, une fois $\mathcal{D}$ plein.

Le temps de séjour dans un état dépend donc de l'état du système. On peut alors choisir les lois discrètes (Poisson, géométrique,...) qui s'adaptent le mieux à l'expérience réelle (via des estimations statistiques, par exemple).

On peut alors poser $S_n = \sum_{k=0}^n X_k$, qui est le temps du n-ième changement d'état.

Le système est donc défini par deux données :

- Un processus Markovien J_n , de loi initiale π_0 et de matrice de transition P , qui représente l'état du système après le n-ième changement d'état.
- Un processus S_n qui représente le temps du n-ième changement d'état.

Sous certaines hypothèses, que nous verrons ci-après, **le couple (J_n, S_n)** forme ce que l'on appelle **une chaîne de renouvellement Markovien**. Le processus $Z_n = J_{N(n)}$, où $N(n) = \max\{k \in \mathbb{N}, S_k \leq n\}$, qui représente l'état du système au temps n , sera alors ce que l'on appelle **une chaîne semi-Markovienne**.

2.2 Modèle

On reprend notre ensemble des états $\mathbb{E} = \{1, \dots, s\}$.

Soit Z_n l'état du processus au temps n .

On note S_n le temps du n-ième état, et J_n l'état du processus après le n-ième changement d'état, $X_n = S_{n+1} - S_n$ le temps de séjour du processus dans le n-ième état, et J_n l'état du processus après le n-ième changement d'état.

On voit tout d'abord que $J_n = Z_{S_n}$, i.e. $Z_n = J_{N(n)}$, où $N(n) = \max\{k \in \mathbb{N}, S_k \leq n\}$.

Définition 2.2.1. (Chaines semi Markoviennes)

On dit que (J_n, S_n) est une chaîne de renouvellement Markovien (ou Markov renewal chain, notée MRC) et que Z_n est une chaîne semi-Markovienne (ou semi Markov chain, notée SMC) si, $\forall n \in \mathbb{N}$:

$$\mathbb{P}(J_{n+1} = j, X_n = k / J_0, S_0, \dots, J_n, S_n) = \mathbb{P}(J_{n+1} = j, X_n = k / J_n). \quad (2.1)$$

Cette condition signifie que l'évolution future d'un processus semi-Markovien (temps de séjour dans l'état présent et état suivant) ne dépend que de l'état présent de celui-ci, et non des états antérieurs ou temps de séjour en ces derniers (Figure 1.3).

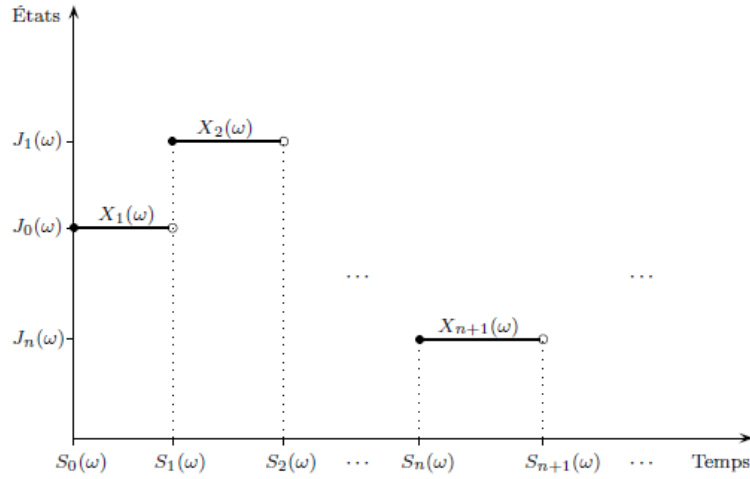


Figure 1.3 : La trajectoire d'un processus semi Markovien.

Si le terme de droite de (2.1) ne dépend pas de n , on parle de chaînes homogènes, et on peut alors introduire la matrice de transition semi-Markovienne suivante :

$$q_{ij}(k) = \mathbb{P}(J_{n+1} = j, X_n = k / J_n = i). \quad (2.2)$$

2.3 Matrice de transition semi markovienne

Soit $\mathcal{M}_{\mathbb{E}}$ l'ensemble des matrices $\mathbb{E} \times \mathbb{E}$, et $\mathcal{M}_{\mathbb{E}}(\mathbb{N})$ l'ensemble des applications de $\mathbb{N}$ dans $\mathcal{M}_{\mathbb{E}}$.

Définition 2.3.1. (Matrices de transition semi-Markoviennes)

On dit que $q \in \mathcal{M}_{\mathbb{E}}(\mathbb{N})$ est une matrice de transition semi markovienne si :

- i) $q_{ij}(k) \geq 0, \forall i, j \in \mathbb{E}, \forall k \in \mathbb{E}$.
- ii) $q_{ij}(0) = 0, \forall i, j \in \mathbb{E}$.

$$\text{iii)} \sum_{k \in \mathbb{N}} \sum_{j \in \mathbb{E}} q_{ij}(k) = 1, \forall i \in \mathbb{E}.$$

On voit clairement, dans (2.2), que $q_{ij}(k)$ s'interprète comme la probabilité de passer de i en j après un temps de séjour de k unités de temps en i , $q(\cdot) := (q_{ij}(\cdot); i, j \in \mathbb{E})$ est appelée **noyau semi-Markovienne** (NSM) à temps discret.

Le NSM représente un outil fondamental dans l'étude probabiliste et statistique de la CSM car la plupart des mesures relatives peuvent être exprimées directement en fonction de celui-ci. Par conséquent, l'étude approfondie du NSM est de première importance dans l'analyse d'une CSM.

2.4 Récurrence, irréductibilité

On note $(S_n^j)_{n \geq 0}$ les temps de retour successifs du processus en j , où S_0^j est le temps du premier passage en j .

$\forall i \neq j$, on note :

- $g_{ij}(n) = \mathbb{P}_i(S_0^j = n)$ la probabilité, partant de i , que le processus passe en j au temps n .
- $g_{jj}(n) = \mathbb{P}_j(S_1^j = n)$ la probabilité, partant de j , que le processus repasse en j au temps n .
- $G_{jj}(n) = \sum_{k=0}^n g_{jj}(k)$ la probabilité, partant de j , que le processus repasse en j au temps n au plus tard.

On remarque alors que $G_{ij}(\infty)$ est la probabilité que la chaîne passe de i en j .

Définition 2.4.1. (Chaines semi-Markoviennes irréductibles)

Si $G_{ij}(\infty) \times G_{ji}(\infty) > 0$, on dit que i et j communiquent. Cette relation, notée $i \longleftrightarrow j$, est une relation d'équivalence sur $\mathbb{E}$.

S'il n'existe qu'une seule classe, i.e. si tous les états communiquent, on dira que la chaîne semi-Markovienne est **irréductible**.

Définition 2.4.2. (Chaines semi-Markoviennes récurrentes)

Si $G_{jj}(\infty) = 1$, on dit que l'état j est récurrent, sinon qu'il est transient.

On parle de chaîne semi-Markovienne **récurrente** (resp. **transiente**) si tous les états sont récurrents (resp. transients).

2.5 Temps de séjour

Soit (J_n, S_n) une MRC homogène et J_n une chaîne de Markov de matrice de transition $p_{ij} = \mathbb{P}(J_{n+1} = j | J_n = i)$.

Introduisons à présent les objets suivants :

Définition 2.5.1. (Lois du temps de séjour)

$\forall i, j \in \mathbb{E}$, on appelle :

i) loi conditionnelle de X_n : la suite $f_{ij}(\cdot)$ telle que, $\forall k \in \mathbb{N}$

$$f_{ij}(k) = \mathbb{P}(X_{n+1} = k | J_n = i, J_{n+1} = j) = \begin{cases} \frac{q_{ij}(k)}{p_{ij}} & \text{si } p_{ij} \neq 0, \\ \mathbb{1}_{\{k=\infty\}} & \text{sinon.} \end{cases}$$

ii) fonction de répartition conditionnelle de X_n : la suite $F_{ij}(\cdot)$ telle que,

$$\forall k \in \mathbb{N}, F_{ij}(k) = \sum_{l=0}^k f_{ij}(l).$$

iii) loi du temps de séjour en i : la suite $h_i(\cdot)$ telle que, $\forall k \in \mathbb{N}$,

$$h_i(k) = \mathbb{P}(X_{n+1} = k | J_n = i) = \sum_{j \in \mathbb{E}} q_{ij}(k).$$

iv) fonction de répartition du temps de séjour en i : la suite $H_i(\cdot)$ telle que,

$$\forall k \in \mathbb{N}, H_i(k) = \mathbb{P}(X_{n+1} \leq k | J_n = i) = \sum_{l=0}^k h_i(l).$$

ivv) $\overline{H}_i(\cdot)$, la fonction de survie dans l'état i

$$\overline{H}_i(k) := \mathbb{P}(X_{n+1} > k | J_n = i) = 1 - H_i(k).$$

Remarques.

- On note que :

$$q_{ij}(k) = p_{ij} f_{ij}(k). \quad (2.3)$$

On montre qu'une chaîne semi Markovienne est entièrement déterminée par la distribution initiale $(\mu_i)_{i \in \mathbb{E}}$ et le noyau semi Markovien

$(q_{ij}(k))_{i,j \in E, k \in \mathbb{N}}$ où de manière équivalente, par une distribution initiale $(\mu_i)_{i \in E}$, une matrice de transition Markovienne p_{ij} et une distribution conditionnelle de temps de séjour $(f_{ij}(k))_{i,j \in E, k \in \mathbb{N}}$.

- Pour la distribution conditionnelle de temps de séjour, on peut considérer la fonction associée à la distribution cumulative.

$$F_{ij}(k) := P(X_{n+1} \leq k | J_n = i, J_{n+1} = j) = \sum_{t=1}^k f_{ij}(t).$$

- Pour la probabilité de transition on a : $p_{ij} = \sum_{k \in \mathbb{N}} q_{ij}(k)$.

2.6 Estimation empirique

Dans la suite, nous supposons que les hypothèses suivantes soient satisfaites :

- A_1 La CM $(J_n)_{n \in \mathbb{N}}$ est irréductible ;
- A_2 Le temps de séjour moyen dans chaque état est fini ;
- A_3 La CRM $(J_n, S_n)_{n \in \mathbb{N}}$ est apériodique.

Définition 2.6.1. *Considérons un exemple d'une trajectoire du DTMRP $(J_n, S_n)_{n \in \mathbb{N}}$, censuré au temps $M \in \mathbb{N}$*

$$H(M) := (J_0, X_1, \dots, J_{N(M)-1}, X_{N(M)}, J_{N(M)}, u_M),$$

Rappelons que $N(M) := \max\{n | S_n \leq M\}$, est le processus de comptage à temps discret du nombre de sauts dans $[1, M] \subset \mathbb{N}$ et $u_M = M - S_{N(M)}$,

pour tout $i, j \in E$ et $k \leq M$, on définit :

- (i) $N_i(M) := \sum_{n=0}^{N(M)-1} \mathbb{1}_{\{J_n=i\}}$: le nombre de visites dans l'état i , jusqu'au temps M ;
- (ii) $N_{ij}(M) := \sum_{n=1}^{N(M)} \mathbb{1}_{\{J_{n-1}=i, J_n=j\}}$: le nombre de transitions de i à j , jusqu'au temps M ;
- (iii) $N_{ij}(k, M) := \sum_{n=1}^{N(M)} \mathbb{1}_{\{J_{n-1}=i, J_n=j, X_n=k\}}$: le nombre de transitions de i à j , jusqu'au temps M , avec le temps de séjour dans l'état i égale k , $1 \leq k \leq M$.

Pour un échantillon de trajectoire $H(M)$ d'un DTMRP, $\forall i, j \in E$ et $k \in \mathbb{N}$, $k \leq M$, nous définissons les estimateurs empiriques de la probabilité de transition p_{ij} , du temps de séjour conditionnel $f_{ij}(k)$ et du noyau semi Markovien discret $q_{ij}(k)$ par :

$$\hat{Q}_{ij}(M) := \frac{N_{ij}(M)}{N_i(M)}, \quad \hat{f}_{ij}(k, M) := \frac{N_{ij}(k, M)}{N_{ij}(M)}, \quad \hat{q}_{ij}(k, M) := \frac{N_{ij}(k, M)}{N_i(M)}. \quad (2.4)$$

La fonction de vraisemblance correspondant à l'historique $H(M)$ est donnée par :

$$L(M) = \prod_{k=1}^{N(M)} p_{j_{k-1}j_k} f_{j_{k-1}j_k}(X_k) \bar{H}_{J_{N(M)}}(u_M)$$

Puisque $u_M < \infty$, on a $\lim_{M \rightarrow +\infty} (u_M/M) = 0$; par conséquent, nous pouvons négliger u_M , dans l'expression de $L(M)$. On considère la fonction de vraisemblance approchée

$$L_1(M) = \prod_{k=1}^{N(M)} p_{j_{k-1}j_k} f_{j_{k-1}j_k}(X_k) \quad (2.5)$$

Proposition 2.6.1. Pour un échantillon de trajectoire d'un DTMRP $(J_n, S_n)_{n \in \mathbb{N}}$, censuré à l'instant $M \in \mathbb{N}$, les estimateurs empiriques proposés dans l'équation (2.4), sont des approximations par la fonction de vraisemblance non paramétrique, ils maximisent la fonction de vraisemblance approchée L_1 , donnée dans l'équation (2.5).

Pour étudier la convergence des estimateurs proposés, nous avons besoin du résultat suivant.

Proposition 2.6.2. Soit μ_{ii} le temps de récurrence moyen de l'état i pour le DTMRP $(J_n, S_n)_{n \in \mathbb{N}}$. on a

$$\frac{N_i(M)}{M} \longrightarrow \frac{1}{\mu_{ii}}, \quad \frac{N_{ij}(M)}{M} \longrightarrow \frac{p_{ij}}{\mu_{ii}}$$

Nous pouvons maintenant donner les propriétés asymptotiques des estimateurs proposés.

Proposition 2.6.3. L'estimateur empirique $\hat{f}_{ij}(k, M)$ proposé dans l'équation (2.4) est uniformément fortement consistant i.e

$$\max_{i,j \in E} \max_{0 \leq k \leq M} |\hat{f}_{ij}(k, M) - f_{ij}(k)| \xrightarrow[M \rightarrow \infty]{a.s.} 0$$

Théorème 2.1. L'estimateur empirique du noyau semi Markovien proposé dans l'équation (2.4) est uniformément fortement consistant, i.e.

$$\max_{i,j \in E} \max_{0 \leq k \leq M} |\hat{q}_{ij}(k, M) - q_{ij}(k)| \xrightarrow[M \rightarrow \infty]{a.s.} 0$$

Pour $i, j \in E$ et $k \in \mathbb{N}$, $\sqrt{M}[\hat{q}_{ij}(k, M) - q_{ij}(k)]$ converge en loi, lorsque $M \rightarrow \infty$, vers une variable aléatoire normale de moyenne nulle et de variance $\mu_{ii}q_{ij}(k)[1 - q_{ij}(k)]$.

Théorème 2.2. Notons $\mathbf{Q} = (p_{11}, \dots, p_{1s}, \dots, p_{s1}, \dots, p_{ss})^t$ où $s = |E|$ et $\hat{\mathbf{Q}}_{ij}(M) = (\hat{Q}_{11}, \dots, \hat{Q}_{1s}, \dots, \hat{Q}_{s1}, \dots, \hat{Q}_{ss})^t$. Alors $\hat{\mathbf{Q}}_{ij}(M)$ est fortement consistant,

$$\widehat{Q}_{ij}(M) \xrightarrow[M \rightarrow \infty]{p.s.} \mathbf{Q}$$

Définition 2.6.2. (Produit de convolution sur $M_E(\mathbb{N})$)

$\forall A, B \in M_E(\mathbb{N})$, on pose :

$$(A * B)_{ij}(k) = \sum_{r \in E} \sum_{l=0}^k A_{ir}(k-l) B_{rj}(l).$$

On note δI l'élément neutre de ce produit de convolution, tel que :

$$\delta I(k) = \begin{cases} I_E & \text{si } k = 0, \\ 0_E & \text{sinon.} \end{cases}$$

Définition 2.6.3. (Puissance de matrices semi-Markoviennes)

Si $q \in \mathcal{M}_E(\mathbb{N})$ est une matrice de transition semi-Markovienne, on définit par récurrence les puissances matricielles suivantes : $\forall i, j \in E, \forall k \in \mathbb{N}$

$$\begin{aligned} q_{ij}^{(0)}(k) &= \begin{cases} 1 & \text{si } i = j \text{ et } k = 0, \\ 0 & \text{sinon.} \end{cases} \\ q_{ij}^{(1)}(k) &= q_{ij}(k) \text{ (i.e. } q^1 = q). \\ \forall n \in \mathbb{N}, q_{ij}^{(n+1)}(k) &= (q^{(n)} * q)_{ij}(k) (= \sum_{r \in E} \sum_{l=0}^k q_{ir}^{(n)}(k-l) q_{rj}(l)). \end{aligned}$$

On vérifie que, si q est une matrice de transition semi-Markovienne, il en est de même pour $q^{(n)}$, $\forall n \in \mathbb{N}$.

De plus, on voit que $q_{ij}^{(2)}(k) = \sum_{r \in E} \sum_{l=0}^k q_{ir}(k-l) q_{rj}(l)$ est la probabilité de passer de i en j en deux changements d'état, et en un temps de séjour cumulé de k unités de temps : $(k-l)$ passées en i et l passées en r .

De même, $q_{ij}^{(n)}(k)$ est la probabilité de passer de i en j en n changements d'état et en un temps de séjour cumulé de k unités de temps, i.e.

$$q_{ij}^{(n)}(k) = \mathbb{P}(J_n = j, S_n = k \mid J_0 = i).$$

Il en découle que, $\forall n \geq k+1, q_{ij}^{(n)}(k) = 0$.

Définition 2.6.4. (Inverse de gauche pour le produit de convolution) Soit $A \in \mathcal{M}_{\mathbb{E}}$. S'il existe $B \in \mathcal{M}_{\mathbb{E}}$ tel que $B * A = \delta I$, on dit que B est l'inverse de gauche de A (pour le produit de convolution), noté $A^{(-1)}$

Considérons les fonctions de la matrices $W = (W(k); k \in \mathbb{N}) \in M_E(\mathbb{N})$, telles que :

$$W_{ij}(k) := \mathbb{P}(J_{n+1} = j, X_{n+1} \leq k | J_n = i) = \sum_{l=1}^k q_{ij}(l), \quad i, j \in E, k \in \mathbb{N}.$$

et $\psi = (\psi(k); k \in \mathbb{N}) \in M_E(\mathbb{N})$, définie par

$$\psi_{ij}(k) := \sum_{n=0}^{\infty} q_{ij}^{(n)}(k) = \lim_{k \rightarrow \infty} \sum_{n=0}^k q_{ij}^{(n)}(k), \quad i, j \in E, k \in \mathbb{N}.$$

telle que $q_{ij}^{(n)}(k) = \mathbb{P}(J_n = j, X_{n+1} = k | J_0 = i)$. La série infinie qui apparaît dans la définition de ψ est une série finie du fait $q_{ij}^{(n)}(k) = 0, \forall n$ et $k \in \mathbb{N}$ tel que $n > k$. Notez que cette propriété est spécifique à un processus semi Markovien à temps discret. Notons $\hat{W}(k, M)$, $\hat{\psi}(k, M)$ les estimateurs de $W(k)$, $\psi(k)$, définis respectivement par :

$$\hat{W}(k, M) := \sum_{l=1}^k \hat{q}(l), \quad \hat{\psi}(k, M) := \sum_{n=0}^k \hat{q}^{(n)}(k, M) \quad (2.6)$$

On peut montrer que $\hat{W}$ et $\hat{\psi}^{(n)}$ sont des estimateurs uniformément consistant.

Théorème 2.3. L'estimateur de la fonction de matrice

$\psi = (\psi(k); k \in \mathbb{N}) \in M_E(\mathbb{N})$ est fortement consistant et pour tout $k \in E$, on a :

$$\sqrt{M} [\hat{\psi}_{ij}(k, M) - \psi_{ij}(k)] \xrightarrow[M \rightarrow \infty]{D} \mathcal{N}(0, \sigma_{ij}^2(k))$$

où

$$\sigma_{ij}^2(k) = \sum_{m=1}^s \mu_{mm} \left(\sum_{r=1}^s [(\psi_{im} * \psi_{rj})^2 * q_{mr}](k) - \left[\sum_{r=1}^s (\psi_{im} * q_{mr} * \psi_{rj})(k) \right]^2 \right)$$

2.7 Equations de renouvellement semi-Markovien

Définition 2.7.1. La fonction de transition de la chaine semi Markovienne Z est la fonction de la matrice $P \in M_E(\mathbb{N})$ définie par :

$$P_{ij}(k) = \mathbb{P}(Z_k = j | Z_0 = i), i, j \in E, k \in \mathbb{N}.$$

Le résultat suivant consiste en une formule récursive de calcul de la fonction de transition P de la chaine semi markovienne Z .

Proposition 2.7.1. Pour tout $i, j \in E$ et pour tout $k \in \mathbb{N}$, on a :

$$P_{ij}(k) = \mathbb{I}_{\{i=j\}}(k)[1 - H_i(k)] + \sum_{r \in E} \sum_{l=0}^k q_{ir}(l) P_{rj}(k-l) \quad (2.7)$$

où

$$\mathbb{I}_{\{i=j\}}(k) = \begin{cases} 1 & \text{si } i = j \text{ et } k \geq 0, \\ 0 & \text{sinon.} \end{cases}$$

Définissons pour tout $k \in \mathbb{N}$:

- $I(k) := (\mathbb{I}_{\{i=j\}}(k); i, j \in E)$, $I := (I(k); k \in \mathbb{N})$;
- $H(k) := \text{diag}(H_i(k); k \in \mathbb{N})$, $H = (H_k, k \in \mathbb{N})$.

En notation de fonction matricielle, l'équation (2.7) devient

$$P = I - H + q * P. \quad (2.8)$$

L'équation (2.8) est un exemple de ce qu'on appelle l'équation de renouvellement de Markov à temps discret, nous savons que la solution de cette équation existe est unique (voir[4]) et a la forme suivante :

$$P(k) = (\delta I - q)^{(-1)} * (I - H)(k) = (\delta I - q)^{(-1)} * (I - \text{diag}(W.1))(k), k \in \mathbb{N} \quad (2.9)$$

où 1 note le vecteur à s colonnes dont tous les éléments valent 1.

On propose pour P l'estimateur suivant :

$$\hat{P}(k) = [(\delta I - \hat{q}(\cdot, M))^{(-1)} * (I - \text{diag}(\hat{W}(\cdot, M) \cdot 1))(k)]$$

Théorème 2.4. L'estimateur de la matrice de transition semi Markovienne $\hat{P}(k, M)$ est un estimateur fortement consistant de $P(k)$.

Chapitre 3

Package SMM pour l'estimation et la simulation d'un modèle semi Markovien

3.1 Introduction

Dans ce chapitre, on va présenter le paquet R appelé SMM développé par Vlad et al[?], ce paquet permet d'effectuer une estimation et une simulation paramétrique et non paramétrique pour un processus Markovien et semi Markovien multi états à temps discret. Pour l'estimation paramétrique plusieurs distributions discrètes sont prises en compte pour les temps de séjour : Uniforme, Géométrique, Poisson, Weibull discret et binôme négatif. L'estimation peut être faite sur la base d'une ou plusieurs trajectoires avec ou sans censure.

En effet, le **paquet SMM** traite plusieurs problèmes :

- Estimation paramétrique pour les distributions du temps de séjour (uniforme ou géométrique ; poisson ; weibull discret et négatif binomial) ou estimation non paramétrique.
- Une ou plusieurs trajectoires.
- Quatre types du temps de séjour différents, un temps de séjour générale en fonction de l'état actuel et de l'état suivant à visiter, l'un

ne dépend que de l'état suivant, l'autre dépend uniquement de l'état actuel, l'autre ne dépend ni de l'état actuel ni de l'état suivant.

- Quatre types de censure différents : censure au début des trajectoires, censure à la fin des trajectoires, censure au début et à la fin ou aucune censure du tout. Plusieurs remarques doivent être faite ici.

Premièrement, En ce qui concerne la censure, la situation la plus simple est celle où tous les temps de séjour sont complètement observés (non censurés).

Un cadre plus compliqué et réaliste est lorsque le dernier temps de séjour n'est pas complètement observé, donc censuré à droite ; dans la plupart des situations pratiques c'est ce cas qui se produit. Une situation analogue est lorsque le premier temps de séjour n'est pas complètement observé, ainsi étant également censuré à gauche. En pratique, ce cas se produit quand on ne connaît pas le début d'un phénomène modélisé par une chaine semi Markovienne.

Deuxièmement, lorsqu'on envisage une estimation à partir de plusieurs échantillons de trajectoires indépendants d'une chaine semi Markovienne, on suppose que toutes les trajectoires sont censurées de la même manière.

Troisièmement, il est important de prendre en compte les quatre types de modèles (de temps de séjour) séparément parce que :

- (i) Dans les situations pratiques, un modèle pourrait être plus adapté que certains.
- (ii) Différents modèles donnent des estimateurs de paramètres spécifiques, il est donc important de les étudier séparément.

Nous considérons quatre modèles semi Markoviens correspondant aux quatre types du temps de séjour suivant :

- Temps de séjour en fonction de l'état actuel et de **l'état suivant**

$$f_{ij}(k) = P(X_m = k | J_m = i, J_{m+1} = j) ;$$

- Les durées de séjour ne dépendent que de **l'état actuel**

$$f_i(k) = P(X_m = k | J_m = i) ;$$

- Les temps de séjour dépendent **uniquement du prochain état visité**

$$f_{.j}(k) = P(X_m = k | J_{m+1} = j) ;$$

- Les temps de séjour ne dépendent ni de l'état actuel ni de l'état suivant

$$f(k) = P(X_m = k).$$

- Notez que les temps de séjour du type $f_{i.}(\cdot)$, $f_{.j}(\cdot)$ ou $f(\cdot)$ sont des cas particuliers de la règle générale de type $f_{ij}(\cdot)$.
 - Dans tout ce qui suit, nous supposons que le MRC ou le SMC sont homogènes.

Le package SMM de R est principalement consacré à la simulation et à l'estimation des modèles semi Markovien à temps discret dans différents cas par les deux fonctions suivantes :

- **simulSM()**, Pour la simulation d'une séquence à partir d'un modèle semi Markovien.
- **estimSM()**, Pour l'estimation des différentes quantités liées au chaîne semi Markovienne.

Le package SMM de R est également consacré à la simulation et à l'estimation de modèle de Markov à temps discret par les deux fonctions suivantes :

- **simulMK()**, Pour la simulation d'une séquences d'un modèle de Markov d'ordre k.
- **estimMK()**, Pour l'estimation des paramètres du modèle.

3.2 Simulation d'un modèle semi Markovien

3.2.1 Simulation selon les distributions classiques

Paramètres : Cette simulation est réalisée par la fonction **simulSM()**, les différents paramètres de la fonction sont :

- **E** : vecteur d'espace d'états de longueur S.
- **NbSeq** : nombre de séquences simulées.
- **lengthSeq** : vecteur contenant les longueurs de chaque séquence simulée.

- **TypeSojournTime** : type de temps de séjour ; il peut être " f_{ij} ", " $f_{i.}$ ", " $f_{.j}$ " où "f" selon les quatre cas discutés précédemment.
- **init** : distribution initiale de longueur S.
- **Ptrans** : matrice des probabilités de transition de la chaîne de Markov immergée $J = (J_m)_m$ de taille $S \times S$.
- **distr** : distribution du temps de séjour :
 - est une matrice de distribution de taille $S \times S$ si le type du temps de séjour est égale à " f_{ij} ",
 - est un vecteur de distribution de taille S si le type du temps de séjour est égale à " $f_{i.}$ " où " $f_{.j}$ ",
 - est une distribution si le type du temps de séjour est égale à "f".

où les distributions à utiliser peuvent être l'une des distributions suivantes : "uniforme", "geom", "pois", "weibull" ou "nbino".
- **param** : paramètres des distributions du temps de séjour :
 - est un tableau de paramètres de taille $S \times S \times 2$ si le type du temps de séjour est égale à " f_{ij} ",
 - est une matrice de paramètres de taille $S \times 2$ si le type du temps de séjour est égale à " $f_{i.}$ " où " $f_{.j}$ ",
 - est un vecteur de paramètres si le temps de séjour est égale à "f".
- **cens.beg** : type de censure au début d'échantillon ; 1 (si le premier temps de séjour est censuré) où 0 (sinon). Toutes les séquences doivent être censurées de la même manière.
- **cens.end** : type de censure à la fin des trajectoires ; 1 (si le dernier temps de séjour est censurée) où 0 (sinon).
- **File.out** : Nom du fichier pour la sauvegarde des séquences ; si File.out = NULL, aucun fichier n'est créé.

Les commandes R ci-dessous génèrent trois séquences de tailles 1000 ; 10000 et 2000 respectivement avec espace d'état fini $E = \{a, c, g, t\}$ et

les temps de séjour dépendent de l'état actuel et de l'état suivant.

Exemple 1

```
# Espace d'état
E = c("a","c","g","t")
S = length(E)

# length de séquence
lengthSeq3 = c(1000, 10000, 2000)

# Création de la distribution initiale
vect.init = c(1/4,1/4,1/4,1/4)

# Création de la matrice de transition
Pij = matrix(c(0,0.2,0.3,0.4,0.2,0,0.5,0.2,0.5,0.3,0,0.4,0.3,0.5,0.2,0),
ncol=4)

# Création de la matrice de distribution
distr.matrix = matrix(c("dweibull", "pois", "geom", "nbinom", "geom",
"nbinom", "pois",
"dweibull", "pois", "pois", "dweibull", "geom", "pois", "geom", "geom",
"nbinom")

## Création d'un tableau contenant les paramètres
param1.matrix = matrix(c(0.6,2,0.4,4,0.7,2,5,0.6, 2,3,0.6,0.6,4,0.3,0.4,4),
nrow = S, ncol = S, byrow = TRUE)
param2.matrix = matrix(c(0.8,0,0,2,0,5,0,0.8, 0,0,0.8,0,4,0,0,4),
nrow = S, ncol = S, byrow = TRUE)
param.array = array(c(param1.matrix , param2.matrix), c(S,S,2))

## Simulation de 3 séquence
seq3 = simulSM(E = E, NbSeq = 3, lengthSeq = lengthSeq3 ,
TypeSojournTime = "fij", init = vect.init , Ptrans = Pij, distr = distr.matrix ,
param = param.array , File.out = "seq3.txt")

## Pour
seq3 = read.fasta("seq3.txt").
```

Tout d'abord, notez que dans cette simulation, les paramètres cens.beg et cens.end sont égaux à 0, c'est-à-dire que les séquences simulées ne sont pas censurées.

Deuxièmement, notez également que les paramètres des distributions sont donnés de la manière suivante : pour par exemple, $f_{13}(\cdot)$ est une distribution géométrique avec le paramètre 0.4, tandis que $f_{14}(\cdot)$ est une binomial négative avec les paramètres 4 et 2. En d'autres termes, les paramètres de $f_{13}(\cdot)$ sont donnés dans le vecteur `param.array[1,3,]` égale à (0.4, 0) et les paramètres de $f_{14}(\cdot)$ sont donnés dans le vecteur `param.array[1,4,]` égale à (4, 2) ; cela signifie que si une distribution a seulement 1 paramètre, le vecteur de paramètres correspondant aura 0 sur la deuxième position.

Valeurs : La fonction **simulSM()** renvoie une liste de séquences qui peuvent être sauvegardé dans un fichier à l'aide du paramètre `File.out`.

```
seq3[[1]][1:15]
[1] "c" "t" "g" "a" "g" "t" "t" "a" "a" "a" "g" "g"
```

3.2.2 Simulation selon les distributions données par l'utilisateur

Nous allons maintenant considérer la simulation en fonction des distributions données par l'utilisateur

Paramètres : Cette simulation est réalisé par la fonction **simulSM()**. Les différents paramètres de la fonction sont :

- **E** : vecteur d'espace d'états de longueur S.
- **NbSeq** : nombre de séquences simulées.
- **lengthSeq** : vecteur contenant les longueurs de chaque séquence simulée.
- **TypeSojournTime** : type du temps de séjour ; il peut être " f_{ij} ", " f_i ", " $f_{.j}$ " ou "f" selon les quatre cas discutés précédemment.
- **init** : distribution initiale de longueur S.
- **Ptrans** : matrice des probabilités de transition de la chaîne de Markov immergée $J = (J_m)_m$ de taille $S \times S$.

- **distr** : distribution du temps de séjour :
- **Loi** : distribution du temps de séjour introduite par l'utilisateur :
 - est une matrice de taille $S \times S \times K_{\max}$ si le type de temps de séjour est égale à " f_{ij} ",
 - est une matrice de taille $S \times K_{\max}$ si le type de temps de séjour est égale à " f_i " ou " f_j ",
 - est un vecteur de longueur $K_{\max}$ si le type de temps de séjour est égale à " f ".
- **cens.end** : type de censure à la fin des trajectoires ; 1 (si le dernier temps de séjour est censurée) ou 0 (sinon). Toutes les séquences doivent être censurées de la même manière.
- **cens.beg** : type de censure au début d'échantillon ; 1 (si le premier temps de séjour est censuré) ou 0 (sinon). Toutes les séquences doivent être censurées de la même manière.
- **File.out** : Nom du fichier pour la sauvegarde des séquences ; si `File.out = NULL`, aucun fichier n'est créé.

Les commandes R ci-dessous génèrent trois séquences de taille 1000 ; 10000 et 2000 respectivement avec l'espace d'état fini $E = \{a, c, g, t\}$; où les temps de séjour dépendent uniquement de l'état suivant.

Exemple 2

```
### Espace d'état
E = c("a","c","g","t").
S = length(E).

### La taille de la séquence
lengthSeq3 = c(1000, 10000, 2000).

### Création de la distribution initiale
vect.init = c(1/4,1/4,1/4,1/4).

### Création de la matrice de transition
Pij = matrix(c(0,0.2,0.3,0.4,0.2,0,0.5,0.2,0.5,0.3,0,0.4,0.3,0.5,0.2,0),
ncol=4).

### Création de la matrice de distribution
### conditionnel de temps de séjour
Kmax = 6
nparam.matrix = matrix(c(0.2,0.1,0.3,0.2,0.2,0,0.4,0.2,0.1,
0,0.2,0.1,0.5,0.3,0.15,0.05,0,0, 0.3,0.2,0.1,0.2,0.2,0),
nrow = S, ncol = Kmax , byrow = TRUE).

### Simulation de 3 séquences avec censure au début
seqNP3-begin = simulSM(E = E, NbSeq = 3, lengthSeq = lengthSeq3 ,
TypeSojournTime = "fj", init = vect.init , Ptrans = Pij,
laws = nparam.matrix , File.out = "seqNP3-begin.txt", cens.beg = 1,
cens.end = 0).

### Simulation de 3 séquences avec censure à la fin
seqNP3-end = simulSM(E = E, NbSeq = 3, lengthSeq = lengthSeq3 ,
TypeSojournTime = "fj", init = vect.init , Ptrans = Pij,
laws = nparam.matrix , File.out = "seqNP3-end.txt", cens.beg = 0,
cens.end = 1).

### Simulation de 3 séquences censurées au début et à la fin
seqNP3-begin-end = simulSM(E = E, NbSeq = 3, lengthSeq = lengthSeq3 ,
TypeSojournTime = "fj", init = vect.init , Ptrans = Pij,
laws = nparam.matrix , File.out = "seqNP3-begin-end.txt", cens.beg = 1,
cens.end = 1).

### Simulation de 3 séquences sans censure
seqNP3-no = simulSM(E = E, NbSeq = 3, lengthSeq = lengthSeq3 ,
TypeSojournTime = "fj", init = vect.init , Ptrans = Pij,
laws = nparam.matrix , File.out = "seqNP3-no.txt").

### Pour la reproductibilité des résultats
seqNP3-begin = read.fasta("seqNP3-begin.txt").
seqNP3-end = read.fasta("seqNP3-end.txt").
```

```
seqNP3-begin-end = read.fasta("seqNP3-begin-end.txt").
seqNP3-no = read.fasta("seqNP3-no.txt").
seqNP3-begin[[1]][1 :15].
```

Notez que dans cette simulation, toutes les différentes censures sont considérées.

Valeur : La fonction **simulSM()** renvoie une liste de séquences simulées. Ces séquences peuvent être sauvegardé dans un fichier fasta à l'aide du paramètre **File.out**.

```
seqNP3-begin[[1]][1 :15].
[1] "t" "g" "a" "g" "g" "g" "g" "a" "a" "a" "a" "t" "a" "g" "g"
```

3.3 Estimation d'un modèle semi Markovien

Dans cette sous-section, nous expliquons et illustrons l'estimation d'un modèle semi Markovien par le modèle paramétrique.

3.3.1 Estimation paramétrique d'un modèle semi Markovien

Les détails sur l'estimation paramétrique des chaînes semi Markoviennes se trouvent dans Barbu, Bérard, Cellier, Sautreuil et Vergne [5]. On considère la distribution $f_{ij}(k) = f_{ij}(k; \theta_{ij})$ en fonction de paramètre inconnu $\theta_{ij} \in \mathbb{R}^{m_{ij}}$, où la dimension du paramètre m_{ij} est connu, aucune hypothèse n'est faite sur $(p_{ij})_{ij}$. À partir des données, nous voulons estimer p_{ij} et θ_{ij} , $i, j \in E$.

Paramètres : L'estimation est réalisée par la fonction **estimSM()**. Les différents paramètres de la fonction sont :

- **file** : chemin qui contient les séquences à partir desquelles, on désire effectuer une estimation.
- **E** : vecteur d'espace d'états de longueur S.

- **seq** : liste des séquences à partir desquelles on veut effectuer une estimation.
- **TypeSojournTime** : type de temps de séjour ; il peut être " f_{ij} ", " f_i ", " $f_{.j}$ " ou "f" selon les quatre cas discutés précédemment.
- **distr** : distributions du temps de séjour :
 - est une matrice de distributions de taille $S \times S$ si type de temps de séjour est égale à " f_{ij} ",
 - est un vecteur de distributions de taille S si type de temps de séjour est égale à " f_i " ou " $f_{.j}$ ",
 - est une distribution si type de temps de séjour est égale à "f".
où les distributions à utiliser peuvent être l'une de "uniforme", "geom", "pois", "weibull" ou "nbinom".
- **cens.end** : type de censure à la fin des trajectoires ; 1 (si le dernier temps de séjour est censurée) ou 0 (sinon). Toutes les séquences doivent être censurées de la même manière.
- **cens.beg** : type de censure au début d'échantillon ; 1 (si le premier temps de séjour est censuré) ou 0 (sinon). Toutes les séquences doivent être censurées de la même manière.

Notez que les séquences à partir desquelles nous estimons peuvent être données soit sous forme de liste R ou en tant que fichier au format fasta.

```
## data
seq3 = read.fasta("seq3.txt").
E = c("a","c","g","t").
## Création de matrice de transition
distr.matrix = matrix(c("dweibull", "pois", "geom", "nbinom",
"geom", "nbinom", "pois", "dweibull", "pois", "pois", "dweibull", "geom",
"pois", "geom", "geom", "nbinom"), nrow = S, ncol = S, byrow = TRUE).
## Estimation des séquences simulées
estSeq3 = estimSM(seq = seq3 , E = E, TypeSojournTime = "fij",
distr = distr.matrix , cens.end = 0, cens.beg = 0).
```

Ici, nous estimons des séquences simulées sans censure. L'estimation effectuée correspondra à la probabilité donnée en (6), sans les troisièmes et quatrièmes termes.

Valeur : La fonction **estimSM()** renvoi une liste contenant :

- **init** : vecteur de taille S avec probabilités initiales estimées de la chaîne semi Markovienne.

```
estSeq3$init
[1] 0.3333333 0.6666667 0.0000000 0.0000000
```

- **Ptrans** : matrice de taille $S \times S$ des probabilités de transition estimées de la chaîne de Markov immergée $J = (J_m)_m$.

```
estSeq3$Ptrans
      [,1]      [,2]      [,3]      [,4]
[1,] 0.0000000 0.2097826 0.2968750 0.2913043
[2,] 0.2062500 0.0000000 0.2968750 0.4968750
[3,] 0.3046944 0.5110717 0.0000000 0.1842338
[4,] 0.3960084 0.1995798 0.4044118 0.0000000
```

- **param** : tableau des paramètres estimés des distributions du temps de séjour.

```

estSeq3$param
,, 1
      [,1]      [,2]      [,3]      [,4]
[1,] 0.0000000 1.979275 0.4250000 3.6891763
[2,] 0.6711864 0.0000000 4.9578947 0.6117130
[3,] 1.8488372 3.043328 0.0000000 0.5859155
[4,] 4.0450928 0.296875 0.4221491 0.0000000
,, 2
      [,1] [,2] [,3] [,4]
[1,] 0     0     0     1.9257056
[2,] 0     0     0     0.8603673
[3,] 0     0     0     0.0000000
[4,] 0     0     0     0.0000000

```

Notez que, par exemple, `estSeq3$param[1,3,]` est le vecteur contenant les paramètres de la distribution $f_{13}(\cdot)$.

- **q** : tableau de taille $S \times S \times K_{\max}$.


```

estSeq3$q[,1:3]
,,1
      [,1]      [,2]      [,3]      [,4]
[1,] 0.00000000 0.02898554 0.212038043 0.06185857
[2,] 0.138432203 0.00000000 0.002086351 0.19293011
[3,] 0.047965028 0.02436585 0.000000000 0.10794546
[4,] 0.006933346 0.05925026 0.170722072 0.00000000
,,2
      [,1]      [,2]      [,3]      [,4]
[1,] 0.000000000 0.02898554 0.212038043 0.06185857
[2,] 0.138432203 0.00000000 0.002086351 0.19293011
[3,] 0.047965028 0.02436585 0.000000000 0.10794546
[4,] 0.006933346 0.05925026 0.170722072 0.00000000
,,3
      [,1]      [,2]      [,3]      [,4]
[1,] 0.000000000 0.05677584 0.07010508 0.06293531
[2,] 0.01496706 0.00000000 0.02564200 0.06330691
[3,] 0.08197701 0.11283632 0.000000000 0.01850897
[4,] 0.05672440 0.05925026 0.05700609 0.00000000

```

Notez par exemple, $q_{13}(2) = P(J_{m+1} = 3, S_{m+1} - S_m = 2 | J_m = 1) = 0.12192187$.

3.3.2 Estimation non paramétrique d'un modèle semi Markovien

Soit $\{Y_1, Y_2, \dots, Y_n\}$ une trajectoire d'une chaîne semi Markovienne $Y = (Y_n)_{n \in \mathbb{N}}$; censuré à un temps arbitraire fixé n

- Le cas $q_{ij}(k) = p_{ij}f_{ij}$ les estimateurs empiriques sont :

$$\hat{p}_{ij}(n) = \frac{N_{ij}(n)}{N_{i.}(n)}, \quad \hat{f}_{ij}(k; n) = \frac{N_{ij}(k; n)}{N_{ij}(n)}, \quad \hat{q}_{ij}(k; n) = \frac{N_{ij}(k; n)}{N_{i.}}.$$

$$\text{où } N_{ij}(n) = \sum_{m=1}^{N(n)-1} \mathbb{1}_{\{J_m=i; J_{m+1}=j\}}, \quad N_{i.}(n) = \sum_{m=1}^{N(n)-1} \mathbb{1}_{\{J_m=i\}},$$

$$N_{ij}(k; n) = \sum_{m=1}^{N(n)-1} \mathbb{1}_{\{J_m=i; J_{m+1}=j; X_{m+1}=k\}}$$

- Le cas $q_{ij}(k) = p_{ij}f_{i.}$; les estimateurs empiriques sont :

$$\hat{p}_{ij}(n) = \frac{N_{ij}(n)}{N_{i.}(n)}, \quad \hat{f}_{i.}(k; n) = \frac{N_{i.}(k; n)}{N_{ij}(n)}, \quad \hat{q}_{ij}(k; n) = \frac{N_{i.}(k; n)}{N_{i.}}.$$

$$\text{où } N_{i.}(k; n) = \sum_{m=1}^{N(n)-1} \mathbb{1}_{\{J_m=i; X_{m+1}=k\}}$$

- le cas $q_{ij}(k) = p_{ij}f_{.j}$; les estimateurs empiriques sont :

$$\hat{p}_{ij}(n) = \frac{N_{ij}(n)}{N_{i.}(n)}, \quad \hat{f}_{.j}(k; n) = \frac{N_{.j}(k; n)}{N_{ij}(n)}, \quad \hat{q}_{ij}(k; n) = \frac{N_{.j}(k; n)}{N_{i.}}.$$

$$\text{où } N_{.j}(k; n) = \sum_{m=1}^{N(n)-1} \mathbb{1}_{\{J_{m+1}=j; X_{m+1}=k\}}$$

- le cas $q_{ij}(k) = p_{ij}f(k)$; les estimateurs empiriques sont :

$$\hat{p}_{ij}(n) = \frac{N_{ij}(n)}{N_{i.}(n)}, \quad \hat{f}(k; n) = \frac{N(k; n)}{N_{ij}(n)}, \quad \hat{q}_{ij}(k; n) = \frac{N(k; n)}{N_{i.}}.$$

$$\text{où } N(k; n) = \sum_{m=1}^{N(n)-1} \mathbb{1}_{\{X_{m+1}=k\}}$$

Paramètre : l'estimation est réalisée par la fonction **estimSM()** et plusieurs paramètres doivent être donnés.

- **file** : chemin qui contient les séquences à partir desquelles on va effectuer une estimation.
- **seq** : liste des séquences à partir desquelles on veut effectuer une estimation.

- **E** : vecteur d'espace d'état de longueur S.
- **TypeSojournTime** : type de temps de séjour ; égale à "NP" pour l'estimation non paramétrique.
- **cens.beg** : type de censure au début d'échantillon ; 1 (si le premier temps de séjour est censuré) ou 0 (sinon). Toutes les séquences doivent être censurées de la même manière.
- **cens.end** : type de censure à la fin des trajectoires ; 1 (si le dernier temps de séjour est censurée) ou 0 (sinon).

```
## data
seqNP3-no = read.fasta("seqNP3-no.txt")
E = c("a","c","g","t")
## Estimation des séquences simulées
estSeqNP3= estimSM(seq = seqNP3-no, E = E, TypeSojournTime = "fj",
distr = "NP", cens.end = 0, cens.beg = 0)
```

Ici, nous estimons des séquences simulées sans censure.

Valeur : La fonction **estimSM()** renvoie une liste contenant :

- **init** : vecteur de taille S des probabilités initiales estimées de la chaîne semi Markovienne.

```
estSeqNP3$init
[1] 0.0000000 0.6666667 0.3333333 .0000000
```

- **Ptrans** : matrice de taille $S \times S$ des probabilités de transition estimées de la chaîne de Markov $J = (J_m)_m$

```
estSeqNP3$Ptrans
      [,1]      [,2]      [,3]      [,4]
[1,] 0.0000000 0.2057578 0.4928027 0.3014395
[2,] 0.2089796 0.0000000 0.2889796 0.5020408
[3,] 0.3134948 0.4934256 0.0000000 0.1930796
[4,] 0.3782051 0.2139423 0.4078526 0.0000000
```

- **laws** : tableau de taille $S \times S \times K_{\max}$ avec les valeurs estimées des distributions de temps de séjour

estSeqNP3\$laws[,1 :2]				
,,1				
	[,1]	[,2]	[,3]	[,4]
[1,]	0.000000000	0.4039248	0.4747405	0.2984
[2,]	0.1998307	0.00000000	0.4747405	0.2984
[3,]	0.1998307	0.4039248	0.000000000	0.2984
[4,]	0.1998307	0.4039248	0.4747405	0.0000
,,2				
	[,1]	[,2]	[,3]	[,4]
[1,]	0.000000000	0.4039248	0.4747405	0.2984
[2,]	0.1998307	0.00000000	0.4747405	0.2984
[3,]	0.1998307	0.4039248	0.000000000	0.2984
[4,]	0.1998307	0.4039248	0.4747405	0.0000

- **q** : tableau de taille $S \times S \times K_{\max}$ du noyau semi Markovien estimé :

```
estSeqNP3$q[,1:3]
,,1
      [,1]      [,2]      [,3]      [,4]
[1,] 0.000000000 0.08552075 0.2277731 0.08552075
[2,] 0.04163265 0.00000000 0.1420408 0.15510204
[3,] 0.06020761 0.19584775 0.000000000 0.05674740
[4,] 0.07852564 0.08814103 0.1947115 0.00000000
```

3.4 Estimation d'une chaîne de Markov d'ordre k-th

Estimation de la matrice de transition de loi initiale d'une chaîne de Markov d'ordre k à partir d'une ou plusieurs séquences.

Paramètres : Cette estimation est réalisée par la fonction **estimMk()**, les différents paramètres de la fonction sont :

- **file** : chemin contenant les séquences à partir desquelles on va effectuer une estimation.
- **seq** : liste des séquences à partir desquelles on veut effectuer une estimation.
- **E** : Vecteur d'espace d'état.
- **k** : Ordre de la chaîne de Markov.

Exemple 3

```
#
E <- c("a","c","g","t")
S = length(E)
init.distribution <- c(1/6,1/6,5/12,3/12)
k = 2
p = matrix(0.25, nrow = Sk, ncol = S)
## Simulation de 3 séquences avec la fonction simulMK
seq3 = simulMk(E = E, nbSeq = 3, lengthSeq = c(1000, 10000, 2000), Ptrans = p,
init = init.distribution, k = 2)
## Estimation des séquences simulées
res3 = estimMk(seq = seq3, E = E, k = 2)
## Résultats d'estimation
# Loi initiale
res3$init
# [1] 0.2469048 0.2573333 0.2483810 0.2473810
# matrice de transition
res3$Ptrans
```

	[,1]	[,2]	[,3]	[,4]
[1,]	0.2690616	0.2338710	0.2602639	0.2368035
[2,]	0.2507553	0.2673716	0.2651057	0.2167674
[3,]	0.2517758	0.2533544	0.2588792	0.2367798
[4,]	0.2522376	0.2432872	0.2481692	0.2563059
[5,]	0.2501949	0.2595479	0.2595479	0.2307093
[6,]	0.2492775	0.2492775	0.2586705	0.2427746
[7,]	0.2337662	0.2792208	0.2438672	0.2445887
[8,]	0.2381306	0.2833828	0.2292285	0.2492582
[9,]	0.2462745	0.2627451	0.2384314	0.2525490
[10,]	0.2259760	0.2530030	0.2424925	0.2785285
[11,]	0.2469512	0.2423780	0.2599085	0.2507622
[12,]	0.2318393	0.2673879	0.2403400	0.2604328
[13,]	0.2866192	0.2668250	0.2185273	0.2280285
[14,]	0.2237711	0.2553191	0.2611886	0.2597212
[15,]	0.2465863	0.2465863	0.2441767	0.2626506
[16,]	0.2511346	0.2541604	0.2420575	0.2526475

Exemple 4

```

E = c(1,2,3).
S = length(E).
init.distr = rep(1/S, 3).
p <- matrix(c(0.3,0.2,0.5,0.1,0.6,0.3,0.2,0.4,0.4), nrow = 3, byrow = TRUE).
## Simulation avec la fonction simulMK
seq1 = simulMk(E = E, nbSeq = 1, lengthSeq = 100, Ptrans = p, init =
init.distr, k = 1).
## Estimation
res1 = estimMk(seq = seq1, E = E, k = 1).
## Résultats de simulation
# Loi initiale
res1$init
      [,1]      [,2]      [,3]

[1,] 0.2500000 0.1875000 0.5625000

[2,] 0.0500000 0.5500000 0.4000000

[3,] 0.2093023 0.3488372 0.4418605

```

Valeur : **EstimMK** renvoie la matrice de probabilité de transition de taille $S^k \times S$ (avec $S = \text{longueur}(E)$) et la loi initiale de taille S estimée à partir des séquences avec un modèle de Markov d'ordre K .

La matrice de transition est toujours donnée dans l'ordre alphabétique et numérique, même si le vecteur de l'espace d'états n'est pas indiqué dans cet ordre.

3.5 Simulation d'une chaîne de Markov d'ordre k-th

Simulation d'une chaîne de Markov d'ordre k à partir d'une matrice de transition et d'une distribution initiale.

Paramètres : Cette estimation est réalisée par la fonction **simulMk()**, les différents paramètres de la fonction sont :

- **E** : Vecteur d'espace d'état.
- **nbSeq** : Nombre de séquence simulées.
- **lengthSeq** : vecteur contenant les longueurs de chaque séquence simulée (les longueurs des séquences peut être différentes).
- **TypeSojournTime** : caractère : doit être l'un des "*f_{ij}*", "*f_{i.}*", "*f_{.j}*", "*f*".
- **init** : vecteur de distribution initiale de longueur S avec S = longueur(E).
- **Ptrans** : matrice de probabilités de transition de taille $S^k \times S$, avec S = length(S).
- **k** : Ordre de la chaîne de Markov.
- **file** : chemin qui contient les séquences à partir desquelles, on désire effectuer une estimation.

Exemple 5

```
# E = c("a","c","g","t").
S = length(E).
init.distribution = c(1/6,1/6,5/12,3/12).
k = 2
p = matrix(0.25, nrow = Sk, ncol = S).
### On simule 3 séquences de tailles 1000, 10000 et 2000 respectivement
simulMK(E = E, nbSeq = 3, lengthSeq = c(1000, 10000, 2000), Ptrans = p,
init = init.distribution, k = 2).
```

Exemple 6

```
# Premier modèle avec
E <- c(1,2,3).
S <- length(E).
init.distr <- rep(1/S, 3).
p <- matrix(c(0.3,0.2,0.5,0.1,0.6,0.3,0.2,0.4,0.4), nrow = 3, byrow = TRUE).
# On simule une séquence de taille 100
simulMk(E = E, nbSeq = 1, lengthSeq = 100, Ptrans = p, init = init.distr, k = 1).
```

Valeur : **SimulMK** renvoie une liste de séquences de taille lengthSeq simulées avec une chaîne de Markov de paramètre init et Ptrans et un espace d'état E.

Si le paramètre File.out n'est pas égale à NULL, un fichier au format fasta contenant les séquences sera créé.

3.6 Application sur des données réelles

3.6.1 Code génétique du virus HPV

Données de séquences génétiques (DSG) : Le code génétique détermine la construction et les fonctions des organismes. Ce code est contenu dans les molécules d'ADN présentes dans les cellules humaines, animales et végétales, ainsi que dans les micro-organismes comme les bactéries et les virus. L'ADN comprend quatre composants, ou blocs de construction, appelés C (pour cytosine), G (pour guanine), A (adénine) et T (pour thymine). Ces nucléotides s'alignent dans un ordre particulier pour chaque organisme, composant ainsi le code génétique. Ce code peut être lu par la cellule pour produire de l'ARN- qui constitue également un type d'information génétique, encodé par les nucléotides A (adénine), C (cytosine), G (guanine) et U (uracile) et qui, à son tour, produit des protéines responsables de l'ensemble de la structure et du fonctionnement d'un organisme vivant.

L'ordre de présentation des nucléotides dans l'ADN et l'ARN- c'est

à dire, la séquence est déterminant car les séquences génétiques contiennent l'information relative à la structure et aux propriétés spécifiques d'un organisme. Dans le cas des virus, ces caractéristiques incluent la pathogénicité, la transmissibilité et la sensibilité aux médicaments antiviraux.

Les laboratoires sont capables de déterminer la séquence génétique d'un organisme particulier au moyen de technologies de séquençage. Les données générées par ce processus sont appelées données de séquences génétiques (DSG) et sont représentées par la liste des nucléotides dans leur ordre d'apparition (par exemple, une séquence d'ARN peut se présenter sous la forme AGAAAUGAAAUGG-CUCCUGUCAA).

Mieux dépister le cancer du col avec l'ADN. Il existe plusieurs tests génétiques de diagnostic du cancer du col de l'utérus. Ces tests pourraient identifier ce type de cancer, qui est provoqué par un virus, le papillomavirus humain (HPV). On compte chaque année un peu plus de 3000 nouveaux cas en France et plus de 1100 décès.

Un dépistage précoce permet de traiter les lésions précancéreuses et d'éviter qu'elle ne se transforment à terme en cancer. À l'heure actuelle, dans le cas où lors du dépistage le frottis présente des cellules anormales. Soit un deuxième frottis six mois plus tard, il est procédé à des tests supplémentaires, soit un test de détection du virus HPV. Ce test permet de savoir si le virus est intégré ou non dans les cellules des patientes, ce qui est un signe indiscutable du développement de la tumeur.

Le programme suivant utilise le code génétique du virus HPV pour estimer loi initiale, sa broba de transition et son noyau semi Markoven.

• **Code R. Estimer le modèle :**

```
=====
k=file.choose().
j=read.fasta().
E=c('a','c','g','t').
estimSM(file=k, seq=NULL, E=E,TypeSojourn='fij', distr ='NP',
```

```
cens=0, cens.beg=0).
```

```
=====
```

• **L'exécution :**

```
$init
```

```
0.2591716  0.2230716  0.3178972  0.1998597
```

```
$Ptrans
```

```
      [,1]      [,2]      [,3]      [,4]
```

```
[1,] 0.0000000 0.3345196 0.4234875 0.2419929
```

```
[2,] 0.3038462 0.0000000 0.4115385 0.2846154
```

```
[3,] 0.3782991 0.3079179 0.0000000 0.3137830
```

```
[4,] 0.2943548 0.2459677 0.4596774 0.0000000
```

```
$laws
```

```
      [,1]      [,2]      [,3]      [,4]
```

```
[1,] 0.0000000 0.6914894 0.6302521 0.6617647
```

```
[2,] 0.8354430 0.0000000 0.7196262 0.7162162
```

```
[3,] 0.6434109 0.6952381 0.0000000 0.6728972
```

```
[4,] 0.7945205 0.8360656 0.8070175 0.0000000
```

```
,,1
```

	[,1]	[,2]	[,3]	[,4]
[1,]	0.0000000	0.2340426	0.2521008	0.2941176
[2,]	0.1265823	0.0000000	0.2056075	0.1891892
[3,]	0.2713178	0.2190476	0.0000000	0.2616822
[4,]	0.1369863	0.1311475	0.1666667	0.0000000
,,2				
	[,1]	[,2]	[,3]	[,4]
[1,]	0.0000000	0.07446809	0.2521008	0.2941176
[2,]	0.03797468	0.0000000	0.2056075	0.1891892
[3,]	0.03875969	0.05714286	0.0000000	0.2616822
[4,]	0.04109589	0.03278689	0.1666667	0.0000000
,,3				
	[,1]	[,2]	[,3]	[,4]
[1,]	0.0000000	0.00000000	0	0
[2,]	0.0000000	0.0000000	0	0
[3,]	0.04651163	0.01904762	0	0
[4,]	0.0000000	0.0000000	0	0
,,4				

	[,1]	[,2]	[,3]	[,4]
[1,]	0.00000000	0.00000000	0.00000000	0.01470588
[2,]	0.00000000	0.00000000	0.009345794	0.04054054
[3,]	0.00000000	0.00952381	0.00000000	0.01869159
[4,]	0.02739726	0.00000000	0.00000000	0.00000000
\$q				
,,1				
	[,1]	[,2]	[,3]	[,4]
[1,]	0.00000000	0.2313167	0.2669039	0.1601423
[2,]	0.2538462	0.00000000	0.2961538	0.2038462
[3,]	0.2434018	0.2140762	0.00000000	0.2111437
[4,]	0.2338710	0.2056452	0.3709677	0.00000000
,,2				
	[,1]	[,2]	[,3]	[,4]
[1,]	0.00000000	0.07829181	0.10676157	0.07117438
[2,]	0.03846154	0.00000000	0.08461538	0.05384615
[3,]	0.10263930	0.06744868	0.00000000	0.08211144
[4,]	0.04032258	0.03225806	0.07661290	0.00000000
,,3				

	[,1]	[,2]	[,3]	[,4]
[1,]	0.0000000	0.024911032	0.04982206	0.007117438
[2,]	0.01153846	0.0000000	0.02692308	0.015384615
[3,]	0.01466276	0.017595308	0.0000000	0.014662757
[4,]	0.01209677	0.008064516	0.01209677	0.0000000
,,4				
	[,1]	[,2]	[,3]	[,4]
[1,]	0.0000000	0.0000000	0.0000000	0.007117438
[2,]	0.0000000	0.0000000	0.003846154	0.011538462
[3,]	0.01759531	0.002932551	0.0000000	0.005865103
[4,]	0.0000000	0.0000000	0.0000000	0.0000000
,,5				
	[,1]	[,2]	[,3]	[,4]
[1,]	0.0000000	0.0000000	0	0
[2,]	0.0000000	0.0000000	0	0
[3,]	0.0000000	0.005865103	0	0
[4,]	0.008064516	0.0000000	0	0

Remarque :

On peut alors comparer la loi initiale et la proba de transition d'une personne suspecte d'avoir cette maladie avec cette loi initiale et cette proba de transition pour affirmer l'existence ou non du tumeur du col.

3.6.2 Données sur l'insuffisance rénale

Le rein joue le rôle de centrale d'épuration de l'organisme. Les reins élaborent l'urine et éliminent les déchets. Ils participent à la transformation de la vitamine D qui joue un rôle dans l'absorption de calcium nécessaire à la bonne santé osseuse.

Les reins produisent l'hormone EPO (érythropoïétine) impliquée dans la production des globules rouges et une enzyme (la rénine) qui régule la tension artérielle. Ils normalisent le volume des liquides et la concentration du potassium, du sodium et du phosphate dans le sang.

L'insuffisance rénale chronique désigne la diminution progressive, prolongée et irréversible des fonctions du rein. Cette maladie silencieuse s'installe petit à petit, mais de manière définitive en détruisant les néphrons (unités fonctionnelles du rein).

Le plus souvent aucun symptôme particulier n'est ressenti, on calcule alors la clairance de la créatine.

• **Evaluation de la clairance de la créatinine par le quotient de COCKROFT-GOLT**

$$\frac{(140 - \text{age}) \times \text{poids}}{\text{créatininémie}} \times \begin{cases} 1.23 & \text{pour les hommes} \\ 0.23 & \text{pour les femmes} \end{cases}$$

Tableau 1.1. Formule de COCKROFT

Pour les différents stades de cette maladie on a :

stade	TFG	que faire ?
1 atteinte	> 90	Diagnostic. traiter les facteurs de risque (hypertension artérielle, lipides, tabagisme...)
2 légère	60-89	Evaluation de la progression
3 modérée	30-59	traitement des complications (anémie, hyperparathyroïdie)
4 sévère	15-29	Néphrologue, préparation à la substitution de la fonction rénale
5 terminale	< 15	Néphrologue, analyse / transplantation

Tableau 1.2. Les différents stades de l'insuffisance rénale.

Le tableau ci-dessous représente les données de 20 patients qui ont été hospitalisés dans l'hôpital de Mohamed Boudiaf (EL BAYADH).

où

P : désigne patients.

A : Age.

p : pois

P	A	p	stade1	stade2	stade3	stade4	stade5
M	80	78	91 (2 ans)	71(10ans)	36 (4 ans)	27 (1 ans)	14 (2 ans)
F	20	54			38 (12ans)	20 (1 ans)	7 (2 ans)
M	69	70			31 (8 ans)	25 (1 ans)	10 (4 ans)
M	79			44 (5 ans)	29 (1 ans)	13 (1 ans)	
F	51	82		65 (12ans)	45 (7 ans)	28 (2 ans)	9 (1 ans)
M	27	48				25 (1 ans)	6 (3 ans)
M	60	73			32 (7 ans)	26 (2 ans)	12 (3 ans)
F	54	119		60 (7 ans)	49 (4 ans)	19 (1 ans)	5 (4 ans)
F	43	95		58 (10ans)	1 (3 ans)	20 (3 ans)	11 (3 ans)
M	60	77		75 (15ans)	33 (5 ans)	27 (1 ans)	10 (1 ans)
F	79	85			57 (9 ans)	26 (2 ans)	14 (2 ans)
F	41	95	95 (1 ans)	77 (5 ans)	49 (3 ans)	25 (1 ans)	8 (1 ans)
M	57	83			33 (6 ans)	27 (1 ans)	12 (3 ans)
M	79	82			32 (9 ans)	18 (1 ans)	9 (4 ans)
M	89	58	94 (12ans)	82 (10ans)	49 (5 ans)	21 (1 ans)	14 (2 ans)
F	65	35			35 (15 ans)	22 (2 ans)	7 (2 ans)
F	53	102		69 (10 ans)	35 (3 ans)	25 (1 ans)	11 (3 ans)
M	71	86		71 (10 ans)	42 (7 ans)	38 (1 ans)	11 (1 ans)
M	78	94	97 (7 ans)	81 (4 ans)	50 (3 ans)	23 (1 ans)	13 (1 ans)
F	75	75		62 (9 ans)	41 (5 ans)	18 (2 ans)	11 (2 ans)

Tableau 1.2. Données sur l'insuffisance rénale.

Le programme suivant utilise les données de l'insuffisance rénale pour estimer la loi initiale, la proba de transition et le noyau semi Markovien.

• **Code R. Estimation du modèle :**

```
E=c('a','c','g','t').
malades=list(c('1','1',rep('2',10),rep('3',4),'4',rep('5',2)),
c(rep('3',12),'4','5'),c(rep('3',8),'4','5','5'),c(rep('3',5),
'4',rep('5',4)),c(rep('2',12),rep('3',7),rep('4',2),'5'),c('4',
rep('5',3),c(rep('3',7),rep('4',2),'5','5','5'),c(rep('2',7),
rep('3',4),'4',rep('5',4)),c(rep('2',10),rep('3',3),rep('4',3),
rep('5',3),c(rep('2',15),rep('3',5),'4','5'),c(rep('3',9),rep('4',
```

```
,2),rep('5',2),c('1',rep('2',5),rep('3',3),'4','5'),c(rep('3',6)
,'4',rep('5',3),c(rep('3',6),'4',rep('5',4)),c(rep('3',9),'4',rep('5'
,2)),c(rep('1',12),rep('2',10),rep('3',5),'4',rep('5',2))
,c(rep('3',15),rep('4',2),rep('5',3)),c(rep('2',10),rep('3'
,3),'4','5')),c(rep('2',10),rep('3',7),'4','5'),c(rep('1',7)
,rep('2',4),rep('3',3),'4','5')),c(rep('2',9),rep('3',5),rep('4',2),'5'))).
estimSM(file=malades, seq=NULL, E=E, TypeSojourn='fij', distr ='NP',
cens=0, cens.beg=0).
```

```
=====
```

● **Extrait de l'exécution :**

```
$init
```

```
0.18181818  0.31818182  0.40909091  0.09090909  0.00000000
```

```
$Ptrans
```

```
      [,1] [,2] [,3] [,4] [,5]
```

```
[1,]  0    1    0    0    0
```

```
[2,]  0    0    1    0    0
```

```
[3,]  0    0    0    1    0
```

```
[4,]  0    0    0    0    1
```

```
[5,]  0    0    0    0    0
```

```
$laws
```

```
.,1
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0	0	0	0	0.0000000
[2,]	0	0	0	0	0.0000000
[3,]	0	0	0	0	0.0000000
[4,]	0	0	0	0	0.7142857
[5,]	0	0	0	0	0.0000000
,,7					
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0	0.3333333	0	0	0.0000000
[2,]	0	0.0000000	0	0	0.0000000
[3,]	0	0.0000000	0	0	0.0000000
[4,]	0	0.0000000	0	0	0.1111111
[5,]	0	0.0000000	0	0	0.0000000

Remarque : On a par exemple dans `estim$Q` la probabilité qu'une personne passe de l'état 4 à l'état 5 dans une durée d'un an est de 0.71.

Bibliographie

- [1] Anselone. *"Ergodic theory for discrete semi-Markov chains"*. Duke Mathematical Journal. (1960).
- [2] Barbu V.S. C Bérard, D. Cellier, M.Sautereuil, N. Vergne. "SMM. *An R package for estimation and simulation of discrete-time semi-Markov models*", to appear in the R journal. (2018).
- [3] Barbu VS, Bérard C, Cellier D, Sautereuil M, Vergne N. "*Parametric estimation of semi-Markov chains*". Simulation and Estimation of Multi-State Discrete Time Semi-Markov and Markov Models. (2017).
- [4] Barbu VS, Karagrigoriou A, Makrides A. "*Semi Markov Modeling for Multi State Systems*". Methodology and computing in applied Probability. (2016).
- [5] Barbu V, Limnios N. "*Semi-Markov chains and Hidden Semi Markov Models toward Applications*". vol.191. Spring, New York. (2008a).
- [6] Barbu. V.S et N. Limnios. "*Semi-Markov Chains and Hidden Semi-Markov Models Toward Applications*". Springer, New York, (2008).
- [7] Barbu VS, Limnios N. "*Empirical estimation for discrete time semi Markov processes with applications in reliability*". Journal of nonparametric Statistics. (2006)
- [8] Barbu. V.S et M. Boussemart et N. Limnios. "*Discrete-time semi-Markov model for reliability and survival analysis*". Communication in Statistics-theory and Methods.

- [9] Barbu, Boussemart and Nikolas Limnios. "*Discrete Time Semi-Markov Model for Reliability and Survival Analysis*". Communication in Statistics- Theory and Methods. (2004).
- [10] Chryssaphinou O, Janssen J, Manca R. "*Downward migration credit risk problem : a non-homogeneous backward semi-Markov reliability approach*". (2016).
- [11] D'amico.G, Janssen J, Manca R (2006). "*Downward migration credit risk problem : a non-homogenous backward semi Markov reliability approach*". Journal of the Operational Research Society.
- [12] D'amico.G, Manca R, Corini C, Petroni F, Prattico F. "*Tornadoes and related damage costs statistical modeling with a semi-Markov approach*". (2016).
- [13] Feller.W On semi-Markov processes. "*Proceedings of the National Academy of Sciences of the USA*".
- [14] Gerontidis. "*Semi-Markov replacement chains. Advance in applied probability*", (1994).
- [15] Heutte N, Huber-carol C. "*Semi-Markov Models for Quality of life Data with Censoring*". In M Mesbah, ML Lee, B Cole (eds.), Statistical Methods for Quality of Life Studies. (2002).
- [16] Howard.R Dynamic "*Probabilistic Systems*", volume II. Wiley, New York, (1971).
- [17] Janssen.J et N. Limnios, éditeurs. "*Semi-Markov Models and Applications*". Kluwer, Dordrecht, (1999).
- [18] Janssen.J et R. Manca. "*Applied Semi-Markov Processes*". Springer, New York, (2006).
- [19] Krol A, Saint-Pierre P. "*Semi Markov : An R Package for Parametric Estimation in Multi-State Semi-Markov Models*". Journal of Statistical Software. (2015).
- [20] Lévy P (1954). "*Processus semi-Markoviens*". In Proc. of International Congress of Mathematics, Amsterdam.
- [21] Limnios, N. and Oprisan, G. "*Semi-Markov Processes and Reliability*". (Boston : Birkhäuser). (2001).

-
- [22] Pyke.R. "*Markov renewal processes with finitely many states*". Annals of Mathematical Statistics. (1259, 1961).
 - [23] Pyke.R. "*Markov renewal processes : Deffinitions and preliminary properties*". Annals of Mathematical Statistics.
 - [24] Smith.W.L. "*Regenerative stochastic processes*". Dans Proceedings of the Royal Society.