UNIVERSITÉ DU QUÉBEC À MONTRÉAL

MODÈLES DE MARKOV CACHÉS

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN MATHÉMATIQUES

(STATISTIQUE)

PAR

JEAN-BAPTISTE VOUMA LEKOUNDJI

SEPTEMBRE 2014

REMERCIEMENTS

J'adresse mes remerciements aux personnes qui m'ont apporté du soutien tout au long de mes études et qui ont contribué à la réalisation de ce mémoire.

D'abord, je tiens à exprimer toute ma reconnaissance à mon directeur de recherche François Watier. Je le remercie pour tout l'encadrement, l'orientation, l'aide et les conseils à mon égard.

Aussi, j'adresse mes sincères remerciements à la faculté des Sciences de l'UQÀM et à tous les professeurs de ma maîtrise en particulier qui m'ont enseigné et m'ont permis d'acquérir divers outils statistiques. Je remercie également Gisèle Legault pour le soutien sur LaTeX lors de la rédaction de ce mémoire.

Je remercie mes très chers parents, mes frères et soeurs, qui ont toujours été là pour moi et sans qui je ne serai pas là aujourd'hui.

Finalement, je profite de cette occasion pour remercier ma partenaire de vie, Nora, pour tout ce qu'elle a fait pour moi lors de ma maîtrise. Faire de la recherche n'est pas toujours une tâche facile. Cependant tu as toujours été là, même pendant les jours les plus sombres. Merci Nora pour tout ce que tu as fait.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

RÉSUMÉ

Les modèles de Markov cachés (MMC) connaissent aujourd'hui un grand succès dans divers domaines d'application. Ils ont été initialement introduits dans la reconnaissance vocale par Baker (1975) et Rabiner (1989), et plus tard dans des domaines tels que l'analyse de séquences biologiques par R. Durbin et Mitchison (1998), l'ingénierie financière par Weigend et Shi (1997) et bien d'autres.

Ils sont utilisés pour modéliser des séquences d'observations qualitatives ou quantitatives. La plupart des méthodes d'utilisation et de développement des MMC ont été développées dans le cadre de la reconnaissance vocale. Par la suite ces mêmes techniques ont été appliquées et adaptées à d'autres domaines.

Notre objectif dans ce mémoire est de présenter une vue d'ensemble de la théorie des MMC à temps discret. Nous exposons les trois problèmes classiques et développons différents algorithmes susceptibles de les résoudre en effectuant de l'inférence sur les états du processus.

Les différents algorithmes dont nous traitons sont : l'algorithme *Forward-Backward* développé par Rabiner et Juang (1986) pour le problème d'évaluation de l'état le plus probable de générer une observation particulière, ou "symbole", à un certain instant défini (évaluation), l'algorithme de Viterbi (1967) pour le problème de calcul de la trajectoire d'états la plus probable de générer une séquence d'observations (décodage) et finalement l'algorithme de Baum-Welch traité par Baum et Eagon (1967) pour la construction d'un modèle adapté aux séquences d'états ou d'observations à modéliser (apprentissage).

Nous illustrons ensuite ces algorithmes en les appliquants à des exemples plus démonstratifs .

MOTS-CLÉS : États, séquences, symboles observables, processus de Markov à temps discret, MMC, algorithme *Forward-Backward*, algorithme de Viterbi, algorithme de Baulm-Welch, inférence.

INTRODUCTION

Les processus stochastiques, notamment markoviens dans notre cas, sont des outils importants en théorie de probabilités. Ils permettent de modéliser plusieurs types de phénomènes dans des domaines tels que la génétique des populations ou l'évolution des cours de marchés boursiers en finance mathématique par exemple.

L'utilisation de tels processus suggère que les états sont les seules observations de la chaîne. Il serait judicieux de se demander quels types de modèles utilisés dans des situations où ces états ne sont pas directement observables, mais produisent des observations particulières ("symboles"). Pour ce fait, on s'intéresse aux MMC et aux problèmes qu'ils permettent de résoudre.

Tel qu'énoncé plus haut, dans les processus markoviens, les observations sont les états du processus. Pour les MMC, c'est bien plus complexe. En effet, dans un MMC on ne peut observer directement les états du processus mais des symboles générés par les états selon une certaine loi de probabilité. Ainsi, à partir d'une séquence de symboles, il n'est pas évident de connaître la trajectoire (séquence d'états) par laquelle est passée le processus. D'où le nom de modèles de Markov «cachés».

Le premier chapitre de ce mémoire est consacré au rappel sur la théorie des processus markoviens à temps discrets avec quelques exemples.

Le deuxième chapitre présente de manière formelle les MMC et les trois principaux problèmes qu'ils permettent de résoudre : l'évaluation du modèle pour expliquer une séquence de symboles observées, trouver le chemin qui optimise le mieux une

séquence de symboles observés et un modèle donné, et enfin la construction d'un modèle adapté aux séquences d'observations à modéliser.
Dans ce chapitre, on expose le développement mathématique derrière les algorithmes *Forward-Backward*, de Viterbi et de Baum-Welch permettant de résoudre les différents problèmes cités plus haut. Nous illustrons également par quelques exemples.

Enfin, le troisième chapitre est la mise en application de tout ce qui est établi dans le chapitre 2. Nous utiliserons les algorithmes et applications développés sur des données réelles obtenues par simulations. Nous illustrons le bon fonctionnement des algorithmes et donnons autant que possible des conclusions à partir des résultats obtenus.

CHAPITRE I

THÉORIE ET INFÉRENCE SUR LES CHAÎNES DE MARKOV À TEMPS DISCRET

Les chaînes de Markov sont des suites de variables aléatoires caractérisées par une aspect particulier de dépendance, qui leur attribue des propriétés singulières et un rôle important en modélisation. Elles ont été introduites par Andreï Andreïevitch Markov (1856-1922) vers 1906. Ce chapitre introduit les principales notions développées par Taylor et Karlin (1998) sur les chaînes de Markov à temps discret et met l'accent sur les points essentiels que nous utilisons tout au long de ce mémoire.

1.1 Préalables des chaînes de Markov à temps discret

Une chaîne de Markov est un réseau spécifique de variables aléatoires décrit par un système dynamique d'états de transitions. Elle respecte les propriétés suivantes tirées de Howard (1971).

Définition 1.1.1. *Soient $\mathcal{S}$ un ensemble fini ou dénombrable et $X = \{X_n\}_{n\in\mathbb{N}}$ un processus stochastique à valeurs dans $\mathcal{S}$.*
On dit que X est une chaîne de Markov si pour tout $n \in \mathbb{N}$, et pour tout $x_0, x_1, \ldots, x_{n+1} \in \mathcal{S}$ telle que $\mathbb{P}(X_0 = x_0, X_1 = x_1, \ldots, X_n = x_n) > 0$,

$$\mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n, X_{n-1} = x_{n-1}, \ldots, X_0 = x_0) = \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n) \tag{1.1}$$

$\mathcal{S}$ est appelé espace d'états.

Remarque 1.1.1. La condition de Markov peut aussi s'écrire pour tout $m, n \in \mathbb{N}$ et $\{x_i,\ i \in \mathbb{N}\}$ à valeurs dans $\mathcal{S}$, sous la forme :

$$\mathbb{P}(X_{m+n} = x_{m+n} | X_m = x_m, X_{m-1} = x_{m-1}, \ldots . X_0 = x_0) = \mathbb{P}(X_{m+n} = x_{m+n} | X_m = x_m) \tag{1.2}$$

Ainsi, on peut dire d'un processus X qu'il est markovien si son état actuel fourni toute l'information nécessaire pour connaître son évolution future. Sa distribution dans le futur étant donné le présent et le passé ne dépend que du présent. On parle alors d'absence de mémoire.

Définition 1.1.2. *Une chaîne de Markov est dite homogène lorsque la probabilité de transition* (2.1) *ne dépend pas de n, c'est à dire :*

$$\mathbb{P}(X_{n+1} = j | X_n = i) = \mathbb{P}(X_1 = j | X_0 = i), \text{ pour tout entier } n. \tag{1.3}$$

L'homogénéité d'une chaîne de Markov précise donc que la probabilité de passer de l'état i à l'état j reste la même à travers le temps. Ainsi, elle permet de condenser dans une seule matrice les probabilités de transitions entre deux états quelconques. La définition **2.1.2** permet alors de caractériser une chaîne de Markov homogène à l'aide d'une matrice de transition et d'un vecteur d'états initiaux.

Définition 1.1.3. *Soit $X = \{X_n\}_{n \in \mathbb{N}}$ une chaîne de Markov d'espace d'états $\mathcal{S}$. Soient $(i, j) \in \mathcal{S}^2$ deux états. La probabilité de transition de i à j est notée*

$$p_{ij} = \mathbb{P}(X_{n+1} = j | X_n = i) = \mathbb{P}(X_1 = j | X_0 = i), \text{ pour tout entier } n. \tag{1.4}$$

On appelle matrice de transition de la chaîne X, la famille $\mathbf{P} = \{p_{ij}\}_{i,j \in \mathcal{S}}$, telle que $0 \leq p_{ij} \leq 1$ et $\sum_{j \in \mathcal{S}} p_{ij} = 1$. Elle est dite stochastique.

Remarque 1.1.2. L'expression $\sum_{j\in\mathcal{S}} p_{ij}$ représente la somme de tous les éléments de la ligne j de la matrice de transition $\mathbf{P}$. Elle vaut 1 pour tout $i \in \mathcal{S}$. En effet, on remarque que :

$$\sum_{j\in\mathcal{S}} p_{ij} = \sum_{j\in\mathcal{S}} \mathbb{P}(X_1 = j|X_0 = i) = \mathbb{P}(\bigcup_{j\in\mathcal{S}} \{X_1 = j\}|X_0 = i) = 1,$$

car ce sont des évènements disjoints.

Exemple 1.1.1. *Supposons l'espace d'états* $\mathcal{S} = \{1, 2, 3\}$ *avec des probabilités de transitions* $p_{11} = p_{12} = p_{13} = 1/3$, $p_{21} = 0, p_{22} = p_{23} = 1/2$, $p_{31} = p_{32} = 0$, *et* $p_{33} = 1$. *La matrice de transition est :*

$$\mathbf{P} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \end{bmatrix}$$

La figure 1.1 *est une représentation graphique de la chaîne de Markov définie. Les états sont représentés par des cercles numérotés et les probabilités de transitions positives par des flèches.*

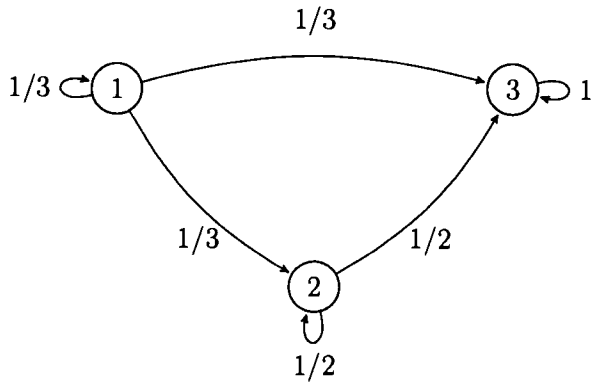

Figure 1.1: Diagramme chaîne de Markov Exemple **1.1.1**

La loi d'une une chaîne de Markov homogène X est déterminée par son espace d'états $\mathcal{S}$, sa matrice de transition $\mathbf{P}$ et son vecteur de distribution initiale $\mu = (\mu_i)_{i\in\mathcal{S}}$ où $\mu_i = \mathbb{P}(X_0 = i)$ pour tout $i \in \mathcal{S}$. C'est la probabilité que la chaîne démarre dans l'état i.

Dansce qui suit, nous considérons que les chaînes de Markov sont homogènes à espace d'états dans $\mathcal{S}$, de matrice de transition $\mathbf{P}$ et de loi initiale μ.

Définition 1.1.4. *La matrice de transition à k pas de la chaîne X, notée* $\mathbf{P}^{(k)} = \{p_{ij}^{(k)}\}$ *pour tout* $k, n \in \mathbb{N}$ *et* $i, j \in \mathcal{S}$ *est la matrice d'éléments :*

$$p_{ij}^{(k)} = \mathbb{P}(X_{n+k} = j | X_n = i) = \mathbb{P}(X_k = j | X_0 = i) \tag{1.5}$$

La notion de *"k pas"* vient du changement d'états après un certain temps k. Plus clairement, la matrice de transition à k pas nous donne la probabilité qu'après un temps k on soit dans l'état j sachant que l'on etait initialement dans l'état i.

Propriété 1.1.1. *Quelques propriétés de la matrice de transition à k pas.*

1. $0 \leq p_{ij}^{(k)} \leq 1$;
2. $\sum_{j\in\mathcal{S}} p_{ij}^{(k)} = 1$ *pour tout* $i \in \mathcal{S}$;

Elle est dite matrice stochastique.

Démonstration. La première propriété est immédiate car $p_{ij}^{(k)}$ est une probabilité. La preuve de la deuxième propriété se fait de la même manière que dans la remarque **2.1.1.** :

$$\begin{aligned}
\sum_{j\in\mathcal{S}} p_{ij}^{(k)} &= \sum_{j\in\mathcal{S}} \mathbb{P}(X_{n+k} = j | X_n = i) \\
&= \sum_{j\in\mathcal{S}} \frac{\mathbb{P}(X_{n+k} = j, X_n = i)}{\mathbb{P}(X_n = i))} \\
&= \frac{\mathbb{P}(X_n = i)}{\mathbb{P}(X_n = i)} \\
&= 1
\end{aligned}$$

car les évènements $\{X_{n+k} = j\}_{j \in \mathcal{S}}$ sont une partition disjointe de $\mathcal{S}$ et donc :

$$\mathbb{P}(X_n = i) = \sum_{j \in \mathcal{S}} \mathbb{P}(X_{n+k} = j, X_n = i).$$

□

La matrice $\mathbf{P}^{(k)}$ nous donne la probabilité d'atteindre l'état j au temps $n + k$ sachant que la chaîne était à l'état i au temps n.

Exemple 1.1.2. *La matrice* $\mathbf{P} = \begin{bmatrix} a & 1-a \\ 1-b & b \end{bmatrix}$ *est une matrice stochastique seulement si* $0 \leq a, b \leq 1$.

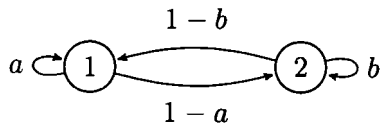

Figure 1.2: Diagramme chaîne de Markov Exemple **1.1.2**

Théorème 1.1.1. *Pour tout* $n \in \mathbb{N}$, *on a :*

$$\mathbf{P}^{(n)} = \mathbf{P}^n. \tag{1.6}$$

Démonstration. Montrons le théorème par induction.

On peut déjà remarquer que la matrice de transition à un pas n'est autre que la matrice de transition elle même, soit $\mathbf{P}^{(1)} = \mathbf{P}$. La condition est donc vraie pour $n = 1$.

Supposons que la condition est vraie pour $n - 1 \geq 1$ et montrons que c'est aussi le cas pour n :

$$\begin{aligned} \mathbf{P}^{(n)} &= \mathbf{P}^{(n-1)}\mathbf{P}^{(1)} = \mathbf{P}^{n-1}\mathbf{P} \\ &= \mathbf{P}^n. \end{aligned}$$

Le cas $n = 0$ est trivial. Il est donc nécessaire de pouvoir calculer la puissance d'une matrice car elle nous permet de déterminer directement la matrice de transition.

□

Théorème 1.1.2. *Equation de Chapman-Kolmogorov*
Pour tout état $i, j \in \mathcal{S}$, pour tout $n, m \in \mathbb{N}$ et $k \in [0, n]$, on a l'égalité

$$p_{ij}^{(n+m)} = \sum_{k \in \mathcal{S}} p_{ik}^{(m)} \, p_{kj}^{(n)} \tag{1.7}$$

En notation matricielle, on a $\mathbf{P}^{(m+n)} = \mathbf{P}^{(m)} \, \mathbf{P}^{(n)}$.

Démonstration. Soient k l'état de la chaîne au temps m. On a

$$\begin{aligned}
p_{ij}^{n+m} &= \mathbb{P}(X_{n+m} = j \,|\, X_0 = i) \\
&= \sum_{k \in \mathcal{S}} \mathbb{P}(X_{n+m} = j, X_m = k | X_0 = i) \\
&= \sum_{k \in \mathcal{S}} \frac{\mathbb{P}(X_{n+m} = j, X_m = k, X_0 = i)}{\mathbb{P}(X_0 = i)} \\
&= \sum_{k \in \mathcal{S}} \frac{\mathbb{P}(X_{n+m} = j | X_m = k, X_0 = i) \mathbb{P}(X_m = k | X_0 = i) \mathbb{P}(X_0 = i))}{\mathbb{P}(X_0 = i)} \\
&= \sum_{k \in \mathcal{S}} \mathbb{P}(X_{n+m} = j | X_m = k, X_0 = i) \mathbb{P}(X_m = k | X_0 = i) \\
&= \sum_{k \in \mathcal{S}} \mathbb{P}(X_m = k | X_0 = i) \mathbb{P}(X_{n+m} = j | X_m = k) \\
&= \sum_{k \in \mathcal{S}} p_{ik}^{(m)} p_{kj}^{(n)}.
\end{aligned}$$

□

Remarque 1.1.3. Les deux démonstrations précédentes sont effectuées par la définition d'une probabilité conditionnelle, pour deux évènements A et B d'un espace de probabilité Taylor et Karlin (1998) :

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}. \tag{1.8}$$

et pour trois évènements A, B et C :

$$\mathbb{P}(A \cap B|C) = \mathbb{P}(B|C)\mathbb{P}(A|B \cap C). \tag{1.9}$$

Une interprétation de l'équation de $Chapman - Kolmogorov$ est la suivante : La probabilité de passer de l'état i à j en $n+m$ pas, revient à calculer les probabilités d'aller de l'état i à l en m pas et ensuite de l'état l à j en n pas, où l est un état intermédiaire quelconque.
Étudier une chaîne de Markov peut se réduire à l'étude des propriétés algébriques de sa matrice de transition.

Exemple 1.1.3. *Considérons deux urnes, une de couleur noire contenant 2 boules noires et 3 boules blanches, et l'autre de couleur blanche contenant 4 boules noires et une boule blanche. On effectue un tirage avec remise. Après avoir tiré une boule, on note sa couleur et on la replace dans son urne. La boule suivante est tirée de l'urne dont la couleur correspond à celle de la dernière boule sortie.*
L'espace d'états correspond à la couleur des urnes où sont effectués les tirages. On a donc $\mathcal{S} = \{1 = Blanche, 2 = Noire\}$.
La matrice de transition des états est :

$$\mathbf{P} = \begin{bmatrix} 1/5 & 4/5 \\ 3/5 & 2/5 \end{bmatrix}$$

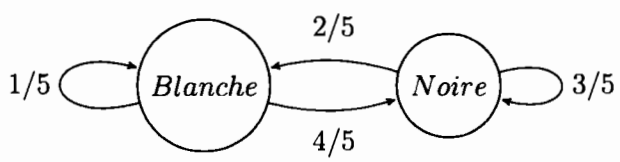

Figure 1.3: Diagramme chaîne de Markov Exemple **1.1.3**

La première colonne correspond aux probabilités de rester ou de quitter l'urne blanche au prochain tirage et la deuxième colonne de rester ou de quitter l'urne noire au prochain tirage. La première ligne correspond aux probabilités de tirer une boule dans l'urne blanche, et la deuxième ligne correspond à tirer une boule dans l'urne noire.

Notre but est de déterminer la probabilité que la troisième boule soit tirée de l'urne noire sachant que la première boule provient de l'urne blanche. On veut

$$\mathbb{P}(X_3 = Noire | X_1 = Blanche) = \mathbb{P}(X_3 = 2 | X_1 = 1) = p_{12}^{(2)} ,$$

on doit commencer par trouver la matrice de transition à deux pas :

$$\mathbf{P}^{(2)} = \mathbf{P}^2 = \begin{bmatrix} 13/25 & 12/25 \\ 9/25 & 16/25 \end{bmatrix}$$

La probabilité voulue est : $p_{12}^{(2)} = 12/25$.

1.2 Classification des états

Il existe différents types d'états. Leur classification permet de mieux étudier les propriétés asymptotiques des chaînes de Markov. Maintenant, classifions les états possibles selon diverses caractéristiques.

1.2.1 Caractères récurrents et transitoires

Les définitions qui suivent sont tirées des ouvrages de Taylor et Karlin (1998) et de Lessard (2013).

Définition 1.2.1. *Soit $i \in \mathcal{S}$, où $\mathcal{S}$ est l'espace des états.*

1. *Un état i est dit récurrent si, $\mathbb{P}(\exists\, n \geq 1,\ X_n = i | X_0 = i) = 1$. C'est à dire que la probabilité d'un éventuel retour à l'état i vaut 1, sachant que la chaîne a commencé à l'état i.*

   Sinon, on dit que l'état est transitoire ;

2. *Un état i est récurrent nul si, $\rho_i = E[R_i | X_0 = i] = \infty$ où $R_i = min\{r : X_r = i\}$. Sinon, l'état i est récurrent positif.*

Définition 1.2.2. *Soient $i, j \in \mathcal{S}$ et $n \geq 1$ un entier.*
La probabilité du premier temps de passage à l'état j, au n-ième pas, sachant que le processus démarre à l'état i est définie par :

$$\begin{aligned} f_{ij}^{(n)} &= P(X_n = j, X_k \neq j,\ k = 1, 2, \ldots, n-1 | X_0 = i) \\ &= P(R_i = n | X_0 = i) \qquad\qquad , n = 1, 2, \ldots \end{aligned} \tag{1.10}$$

Par convention $f_{ij}^{(0)} = 0$.

Proposition 1.2.1. *La probabilité f_{ij} d'un possible passage de la chaîne à l'état j sachant que'elle démarre à l'état i vaut*

$$f_{ij} = \sum_{n=1}^{\infty} f_{ij}^{(n)}. \tag{1.11}$$

Démonstration.

$$\begin{aligned} f_{ij} &= \mathbb{P}(\text{un possible passage à l'état } j | X_0 = i) \\ &= \mathbb{P}(\bigcup_{n=1}^{\infty} \{\text{le premier passage à l'état } j \text{ se fait au temps } n\} | X_0 = i) \\ &= \sum_{n=1}^{\infty} \mathbb{P}(\text{le premier passage à l'état } j \text{ se fait au temps } n | X_0 = i) \\ &= \sum_{n=1}^{\infty} f_{ij}^{(n)}. \end{aligned}$$

□

Remarque 1.2.1. On peut donc affirmer qu'un état i est récurrent si $f_{ii} = 1$ et transitoire si $f_{ii} < 1$.

Proposition 1.2.2. *Pour* $n \in \mathbb{N}$, $p_{ij}^{(n)} = \sum_{k=0}^{n} p_{jj}^{(n-k)} f_{ij}^{(k)}$.

En effet si le processus passe de l'état i à l'état j en n pas, on s'intéresse au k-ième instant où il atteint pour la première fois l'état j.

Démonstration. Soient $i, j \in \mathcal{S}$ et l'évènement $Z_k = \{X_k = j\} \cap \{X_r \neq j, r = 1, 2, \ldots, k-1\}$ pour tout entier $k \leq n$. On peut alors écrire :

$$\begin{aligned} p_{ij}^{(n)} &= \mathbb{P}(X_n = j | X_0 = i) \\ &= \sum_{k=0}^{n} \mathbb{P}(X_n = j,\ Z_k | X_0 = i). \end{aligned}$$

De l'égalité (1.9) on peut écrire :

$$\begin{aligned} p_{ij}^{(n)} &= \sum_{k=0}^{n} \mathbb{P}(Z_k |\ X_0 = i) \mathbb{P}(X_n = j | Z_k,\ X_0 = i) \\ &= \sum_{k=0}^{n} \mathbb{P}(Z_k |\ X_0 = i) \mathbb{P}(X_n = j | X_k = j) \ \ (car\ le\ processus\ est\ markovien) \\ &= \sum_{k=0}^{n} f_{ij}^{(n)} p_{jj}^{(n-k)}. \end{aligned}$$

□

La formule de la proposition 1.2.2 permet de calculer de manière récursive les probabilités $f_{ij}^{(n)}$ à partir des probabilités de transitions $p_{ij}^{(n)}$.

Maintenant, l'objectif serait d'établir un critère de dépendance entre les probabilités de transitions à n pas et celles des premiers temps de passage. Lequel nous permettra de montrer certaines de leurs propriétés . Pour ce fait, on définie les fonctions génératrices suivantes pour $i, j \in \mathcal{S}, n \in \mathbb{N}$ et $|s| < 1$ Lévêque (2014).

$$P_{ij}(s) = \sum_{n=0}^{\infty} s^n p_{ij}^{(n)} \tag{1.12}$$

$$F_{ij}(s) = \sum_{n=0}^{\infty} s^n f_{ij}^{(n)}. \tag{1.13}$$

Proposition 1.2.3. *Pour tous $i, j \in \mathcal{S}$, on a :*

(i) $P_{ii}(s) = 1 + F_{ii}(s)P_{ii}(s)$;

(ii) $P_{ij}(s) = F_{ij}(s)P_{jj}(s)$, *si* $i \neq j$.

Démonstration. Preuve du point (i).

$$\begin{aligned}
P_{ii}(s) &= \sum_{n=0}^{\infty} s^n p_{ii}^{(n)} = p_{ii}^{(0)} + \sum_{n=1}^{\infty} s^n p_{ii}^{(n)} = 1 + \sum_{n=1}^{\infty} s^n p_{ii}^{(n)} \qquad (car\ p_{ii}^{(0)} = 1) \\
&= 1 + \sum_{n=1}^{\infty} s^n \sum_{k=0}^{n} p_{ii}^{(n-k)} f_{ii}^{(k)} = 1 + \sum_{n=0}^{\infty} \sum_{k=0}^{n} s^n p_{ii}^{(n-k)} f_{ii}^{(k)} \quad (\text{proposition } \mathbf{2.2.2} \text{ et } f_{ii}^{(0)} = 0) \\
&= 1 + \sum_{n=0}^{\infty} \sum_{k=0}^{n} s^{n-k} p_{ii}^{(n-k)}\, s^k f_{ii}^{(k)} = 1 + \sum_{k=0}^{\infty} \sum_{n=k}^{\infty} s^{n-k} p_{ii}^{(n-k)}\, s^k f_{ii}^{(k)} \\
&= 1 + \sum_{k=0}^{\infty} s^k f_{ii}^{(k)} \sum_{n-k=0}^{\infty} s^{n-k} p_{ii}^{(n-k)} \\
&= 1 + F_{ii}(s)P_{ii}(s).
\end{aligned}$$

Preuve du point (ii).

$$\begin{aligned}
P_{ij}(s) &= \sum_{n=0}^{\infty} s^n p_{ij}^{(n)} = \sum_{n=0}^{\infty} s^n \sum_{k=0}^{n} p_{jj}^{(n-k)} f_{ij}^{(k)} \quad (\text{proposition } \mathbf{2.2.2}) \\
&= \sum_{n=0}^{\infty} \sum_{k=0}^{n} s^{n-k} p_{jj}^{(n-k)}\, s^k f_{ij}^{(k)} \\
&= \sum_{k=0}^{\infty} \sum_{n=k}^{\infty} s^{n-k} p_{jj}^{(n-k)}\, s^k f_{ij}^{(k)} \\
&= \sum_{k=0}^{\infty} s^k f_{ij}^{(k)} \sum_{n-k=0}^{\infty} s^{n-k} p_{jj}^{(n-k)} \\
&= F_{ij}(s)P_{jj}(s).
\end{aligned}$$

□

Corollaire 1.2.1. *Caractères récurrents et transitoires*

- *Un état $i \in \mathcal{S}$ est récurrent si et seulement si $\sum_{n=1}^{\infty} p_{ii}^{(n)} = \infty$;*
- *Un état i est transitoire si et seulement si $\sum_{n=1}^{\infty} p_{ii}^{(n)} < \infty$.*

Démonstration. En prenant la limite des expressions (1.12) et (1.13) quand s se rapproche de 1, on obtient :

$$\lim_{s\longrightarrow 1} F_{ii}(s) = \lim_{s\longrightarrow 1}\sum_{n=0}^{\infty} s^n f_{ii}^{(n)} = \sum_{n=0}^{\infty} f_{ii}^{(n)} = f_{ii} = 1 \ (récurrent)$$
$$\lim_{s\longrightarrow 1^-} P_{ii}(s) = \lim_{s\longrightarrow 1^-}\sum_{n=0}^{\infty} s^n p_{ii}^{(n)} = \sum_{n=0}^{\infty} p_{ii}^{(n)}.$$

D'après l'assertion (i) de la proposition 1.2.3, on peut écrire pour $i, j \in \mathcal{S}$ et $|s| < 1$:

$$P_{ii}(s) = 1 + F_{ii}(s)P_{ii}(s)$$
$$P_{ii}(s)(1 - F_{ii}(s)) = 1$$
$$P_{ii}(s) = \frac{1}{1 - F_{ii}(s)}.$$

Par conséquent $\lim_{s\longrightarrow 1^-} P_{ii}(s) = \infty$. En invoquant le lemme d'Abel (Théorème **A.2.1** en annexe), on en déduit que :

$$\sum_{n=0}^{\infty} p_{ii}^{(n)} = \lim_{s\longrightarrow 1^-} P_{ii}(s) \Longleftrightarrow f_{ii} = 1$$
$$\sum_{n=0}^{\infty} p_{ii}^{(n)} = \infty \Longleftrightarrow f_{ii} = 1$$

On vient donc de prouver le premier point du corollaire. Le deuxième point n'est qu'une conséquence du premier. □

1.2.2 Partition des états

Définition 1.2.3. *Soient i et j deux états de $\mathcal{S}$. L'état j est accessible depuis l'état i, noté $i \longrightarrow j$, si*

$$\exists\, n \in \mathbb{N},\ p_{ij}^{(n)} = \mathbb{P}(X_n = j | X_0 = i) > 0. \tag{1.14}$$

On dit que les états i et j communiquent si ils sont tous deux accessibles l'un de l'autre. On note $i \longleftrightarrow j$.

En d'autres termes, on dit que l'état j est accessible depuis l'état i si la probabilité d'atteindre j en n transitions depuis i est strictement positive .

Définition 1.2.4. *La période d'un état $i \in \mathcal{X}$ notée $d(i)$ est l'entier définie par :*

$$d(i) = PGCD\{n \geq 1 : p_{ii}^{(n)} > 0\} \tag{1.15}$$

On dit que i est périodique si $d(i) > 1$, sinon il est apériodique .

Proposition 1.2.4. *La relation $\longleftrightarrow$ est une relation d'équivalence sur $\mathcal{S}$. Elle est*

- *réflexive : i communique avec i ;*
- *symétrique : $\forall i, j \in \mathcal{S}, i \longleftrightarrow j$ si $j \longleftrightarrow i$;*
- *transitive : $\forall i, j, k \in \mathcal{S}$, si $i \longleftrightarrow j$ et $j \longleftrightarrow k$ alors $i \longleftrightarrow k$.*

Démonstration. Pour prouver cette équivalence, nous devons montrer que la relation est réflexive, symétrique et transitive.

- *réflexivité* : $p_{ii}^{(0)} = P(X_0 = i | X_0 = i) = 1 > 0$ alors $i \longleftrightarrow i$;
- *symétrie* : par définition si $i \longleftrightarrow j$ alors $\exists m, n \in \mathbb{N}$ tels que $p_{ij}^{(m)} > 0$ et $p_{ji}^{(n)} > 0$. Inversement, on a $j \longleftrightarrow i$;
- *transitivité* :

  Si $i \longleftrightarrow j$ alors $p_{ij}^{(n_1)} > 0$ avec $n_1 \in \mathbb{N}$,

  Si $j \longleftrightarrow k$ alors $p_{jk}^{(n_2)} > 0$ avec $n_2 \in \mathbb{N}$.

  De l'équation de Chapman-Kolmogorov, on a pour $k \in \mathcal{X}$

$$p_{ik}^{(n_1+n_2)} = \sum_{l \in \mathcal{X}} p_{il}^{(n_1)} p_{lk}^{(n_1)} \geq p_{ij}^{(n_1)} p_{jk}^{(n_2)} > 0.$$

  Ainsi $i \longleftrightarrow k$.

□

L'espace d'états $\mathcal{S}$ de la chaîne de Markov peut être partitionné en differentes classes d'équivalence. La chaîne est dite irréductible lorsque chacun de ses états est accessible depuis tous les autres états. Nous verrons par la suite que les états dans classe irréductible ont les mêmes caractères récurrent et transitoire.

Définition 1.2.5. *Soit $C \subseteq \mathcal{X}$ une classe d'états.*

1. *C est dite fermée si $\forall i \in C$, $j \notin C$ et $n \geq 1$, $p_{ij}^{(n)} = 0$;*
2. *C est dite irréductible si $\forall i, j \in C$, $i \longleftrightarrow j$.*

Une classe d'état est donc fermée si aucun état hors d'elle n'est accessible depuis ses états intérieurs. De plus, une chaîne de Markov est irréductible si elle n'est formée que d'une unique classe fermée.

Théorème 1.2.1. *Décomposition de l'espace d'états Irwin (2006)*
Par la relation d'équivalence $\longleftrightarrow$, il existe une unique partition de l'espace d'états $\mathcal{X}$ telle que

$$\mathcal{S} = T \cup (\cup_{k=1}^{\infty} C_k), \tag{1.16}$$

où T est une classe uniquement constitué d'états transitoires et $\{C_k\}_{k\geq 1}$ est une suite de classes irréductibles fermées d'états récurrents.

Démonstration. Soit $\{C_k\}_{k\geq 1}$ une suite de classe d'états récurrents pour la relation $\longleftrightarrow$.
Montrons que les C_k sont fermées :
On procède par l'absurde. Supposons qu'il existe $i \in C_k$ et $j \notin C_k$ tel que $p_{ij} > 0$.
On a j ne communique pas avec i donc :

$$\mathbb{P}(X_n \neq i, \forall n \geq 1 | X_0 = i) \geq \mathbb{P}(X_1 = j | X_0 = i) = p_{ij} \geq 0$$

Or i est un état récurrent, ce qui est contradictoire car si i est récurrent alors $\exists n \geq 1$ tel que :

$$\mathbb{P}(X_n = i | X_0 = i) = 1$$

D'où :

$$\mathbb{P}(X_n \neq i, \forall n \geq 1 | X_0 = i) = 0$$

□

Ainsi, on vient de prouver que toutes les classes irréductibles d'états récurrents sont fermées. En outre, puisque la relation $\longleftrightarrow$ est une relation d'équivalence, on a directement la partition contenant la classe d'états transitoires T et la suite des classes irréductibles fermées d'états récurrents.

Remarque 1.2.2. Si l'espace d'états $\mathcal{S}$ est fini alors il existe au moins un état récurrent et tous les états récurrents sont non-nuls.

Théorème 1.2.2. Soient i et j deux états dans $\mathcal{S}$. Si $i \longleftrightarrow j$ alors

(*i*) ils sont tous les deux transitoires ou tous les deux récurrents ;

(*ii*) Dans le cas où les deux sont récurrents, ils sont récurrents nuls ou récurrents positifs ;

(*iii*) $d(i) = d(j)$, ils ont la même période.

Démonstration. Voir en Annexe **A**. □

Exemple 1.2.1. *Considérons l'espace d'états* $\mathcal{X} = \{1, 2, 3, 4, 5\}$. *Soient la chaîne de Markov de diagramme et matrice de transition*

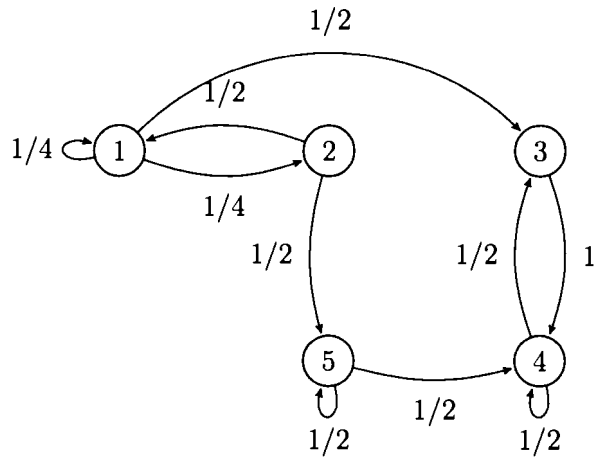

Figure 1.4: Diagramme chaîne de Markov Exemple **1.2.1**

$$\mathbf{P} = \begin{bmatrix} 1/4 & 1/4 & 1/2 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3 \end{bmatrix}$$

La chaîne de markov a trois classes d'états $\{1,2\}$, $\{3,4\}$ *et* $\{5\}$.
Dans la classe $\{1,2\}$, $p_{13} = p_{25} = \dfrac{1}{2} \neq 0$. *Alors la classe n'est pas fermée, donc transitoire et périodique car*

$$d(1) = d(2) = pgcd\{n \geq 1 : p_{11}^{(n)} > 0 \text{ et } p_{22}^{(n)} > 0\} = pgcd\{2,4,6,8,\ldots\} = 2.$$

Dans la classe $\{3,4\}$, *on remarque que* $\forall n \geq 1$ *et* $\forall j \notin \{3,4\}$, $p_{3j}^{(n)} = p_{4j}^{(n)} = 0$. *Elle est alors fermée sur un nombre d'états fini, donc récurrente positive et apériodique car*

$$d(1) = d(2) = 1.$$

Dans la classe $\{5\}$, $p_{54} = \frac{1}{2} \neq 0$. *Elle n'est pas fermée, donc transitoire et apériodique car*

$$d(5) = pgcd\{n \geq 1 : p_{55}^{(n)} > 0\} = pgcd\{1, 2, 3, \ldots\} = 1.$$

1.3 Distribution stationnaire et théorème ergodique

L'objectif de cette section est de trouver les conditions simples permettant d'approximer la loi d'une chaîne de Markov $\{X_n\}_{n\geq 1}$ sur une longue période, plus clairement sous quelles conditions pourrons nous trouver la limite $\lim_{n\to\infty} X_n$ afin de pouvoir identifier facilement la chaîne.

1.3.1 Distribution stationnaire

Définition 1.3.1. *Un vecteur* $\pi = \{\pi_i\}_{i\in\mathcal{S}}$ *sur* $\mathcal{S}$ *est stationnaire pour la chaîne de Markov* X *si*

(i) $\pi_i \geq 0, \forall i \in \mathcal{S}$ *et* $\sum_{i\in\mathcal{S}} \pi_i = 1$;

(ii) $\pi_j = \sum_{i\in\mathcal{S}} \pi_i P_{ij}$, $\forall j \in \mathcal{S}$ *ou en notation matricielle* $\pi = \pi\mathbf{P}$.

Une chaîne de Markov est donc stationnaire sous $\mathbb{P}$ si pour tout $k, n \in \mathbb{N}$, la distribution du vecteur aléatoire $(X_1, X_2, ..., X_n)$ est identique à celle du vecteur $(X_k, X_{k+1}, ..., X_{n+k})$.

Remarque 1.3.1. *La stationnarité de cette distribution se voit en itérant l'égalité de l'assertion (ii), soit* $\pi\mathbf{P}^2 = (\pi\mathbf{P})\mathbf{P} = \pi\mathbf{P} = \pi$. *De la même manière, on a* $\pi\mathbf{P}^n = \pi$,
$n \in \mathbb{N}$.

1.3.2 Théorème ergodique

Une chaîne de Markov irréductible et apériodique est dite ergodique lorsque tous ses états sont récurrents positifs et apériodique, et non-ergodique lorsque tous ses états sont transitoires ou récurrents nuls.
Le théorème qui suit joue un rôle important dans l'étude des chaînes de Markov pour des très longues périodes(transitions).

Théorème 1.3.1. *Théorème ergodique*
Soit une chaîne de Markov irréductible et apériodique $X = \{X_n\}_{n\geq 1}$. *Alors*

1. $\lim_{n\longrightarrow\infty} p_{ij}^{(n)} = \frac{f_{ij}}{\rho_i}$;
2. *Sous les mêmes conditions,*
   $\lim_{n\longrightarrow\infty} \frac{1}{n} \sum_{k=0}^{n-1} p_{ij}^{(n)} = \frac{f_{ij}}{\rho_i}$, *pour tous les entiers* i, j *dans* $\mathcal{S}$.

Démonstration. Voir Lessard (2013).

□

L'étude des points précédents suppose que les matrices de transition sont connues à l'avance. Cependant, on se pose la question de savoir comment estimer à partir d'une de ses trajectoires la matrice de transition d'une chaîne de Markov.
Aussi, on se demande comment statuer si la trajectoire que l'on observe provient bel et bien d'une chaîne de Markov. Nous en discutons dans le point qui suit.

1.4 Inférence sur les chaînes de Markov à temps discret

Cette partie traite de l'inférence basée sur les probabilités pour des chaînes de Markov ergodique finies. On discute des méthodes d'estimation des paramètres d'une chaîne de Markov basées sur les travaux de Billingsley (1960) et de Anderson et Goodman (1957).

1.4.1 Matrice de dénombrement des transitions d'une chaîne de Markov

Soit l'espace des états $\mathcal{S} = \{1, 2, \ldots, m\}$ et $\{X_k,\ k = 1, 2, \ldots, n+1\}$ un échantillon provenant d'une chaîne de Markov ayant pour matrice de transition $\mathbf{P} = \{p_{ij}\}$ et de distribution stationnaire $\pi = \{\pi_i\}_{i \in \mathcal{S}}$, pour tous les entiers i et j dans $\mathcal{S}$.
Si $a = \{a_1, a_2, \ldots, a_{n+1}\}$ est une séquence de $n+1$ états dans $\mathcal{S}$, alors la probabilité que cette séquence corresponde à l'échantillon s'écrit :

$$\mathbb{P}(X_i = a_i,\ i \in \mathcal{S}) = \mathbb{P}(X_1 = a_1)\mathbb{P}(X_2 = a_2 | X_1 = a_1) \ldots \mathbb{P}(X_{n+1} = a_{n+1} | X_n = a_n) \tag{1.17}$$

$$\mathbb{P}(X_i = a_i,\ i \in \mathcal{S}) = \mu_{a_1} p_{a_1 a_2} p_{a_2 a_3} \cdots p_{a_n a_{n+1}} \tag{1.18}$$

La première égalité découle de la définition d'un processus markovien vue dans le chapitre 2.

On définit maintenant la variable r_{ij} qui donne le nombre de fois que la chaîne fait une transition de l'état i vers l'état j, donc lorsqu'on obtient successivement $X_k = i$ et $X_{k+1} = j$, pour $1 \leq k \leq n$:

$$r_{ij} = \sum_{k=1}^{n} \mathbb{1}_{\{X_k = i,\, X_{k+1} = j\}} \tag{1.19}$$

La matrice $R = \{r_{ij}\}$ sera appelée matrice de dénombrement des transitions de la séquence a. On a alors :

$$\mathbb{P}(X_1 = a_1,\ X_2 = a_2, \ldots, X_{n+1} = a_{n+1}) = \mu_{a_1} \prod_{i,j} p_{ij}^{r_{ij}} \tag{1.20}$$

De plus posons $r_{i.} = \sum_{j=1}^{m} r_{ij}$ et $r_{.j} = \sum_{i=1}^{m} r_{ij}$ qui correspondent respectivement aux fréquences des états $\{a_1, a_2, \ldots, a_n\}$ et $\{a_2, a_3, \ldots, a_{n+1}\}$. Whittle (1955) a pu

établir que :

$$r_{i.} - r_{.i} = \delta_{i,a_1} - \delta_{i,a_{n+1}} \text{ . pour tout entier } i \text{ dans } \mathcal{S} \tag{1.21}$$

$$\sum_{i.j} r_{ij} = \sum_{i=1}^{m} r_{ij} = \sum_{j=1}^{m} r_{ij} = n \tag{1.22}$$

où δ_{ij} est le symbole de Kronecker.

L'égalité (1.21) s'interprète à partir de l'égalité (1.19). En effet, de (1.19) on constate qu'à l'exception de l'état initial a_1 et de l'état final a_{n+1}, chaque transition dans un état i doit être suivie d'une sortie de cet état i. On remarque d'abord qu'avec probabilité μ_{a_1} on est intialement dans l'état a_1, ensuite avec $p_{a_1 a_2}$ on transite de a_1 vers a_2 mais avec $p_{a_2 a_3}$, on sort de l'état a_2 pour transiter vers a_3 et ainsi de suite jusqu'à la transition finale de a_n vers a_{n+1}.
Le théorème suivant est une conséquence des conditions décrites ci-dessus, il a été prouvé par Whittle (1955). Voir aussi les travaux deBillingsley (1960) pour plus de détails.

Théorème 1.4.1. *Lemme de Whittle.*
Soit $R = r_{ij}$ une matrice $m \times m$ à composantes entières positives telles que $\sum_{i,j} r_{ij} = n$ et $r_{i.} - r_{.i} = \delta_{i,u} - \delta_{i,v}$, $i \in \mathcal{S}$ pour une certaine paire (u, v).
Si $N_{uv}^{(n)}(R)$ est le nombre de séquences $(a_1, a_2, \ldots, a_n)$ avec matrice de dénombrement R satisfaisant $a_1 = u$ et $a_{n+1} = v$, alors

$$N_{uv}^{(n)}(R) = R_{vu}^{*} \frac{\prod_{i=1}^{m} r_{i.}!}{\prod_{i.j} r_{ij}!} \tag{1.23}$$

où R_{vu}^{*} est le (v, u)-ième cofacteur de la matrice $R^{*} = \{r_{ij}^{*}\}$ de composantes

$$r_{ij}^{*} = \begin{cases} \delta_{ij} - \dfrac{r_{ij}}{r_{i.}} & si\ r_{i.} > 0 \\ \delta_{ij} & si\ r_{i.} = 0 \end{cases} \tag{1.24}$$

Démonstration. La preuve se fait par induction. Voir Whittle (1955). □

Des égalités (1.20) et (1.23), on peut constater que la probabilité qu'une séquence $\{x_1, x_2, \ldots, x_{n+1}\}$ ait une matrice de dénombrement R avec état initial $x_1 = u$ et état final $x_{n+1} = v$ est :

$$\mu_u R^*_{vu} \frac{\prod_{i=1}^m r_{i.}!}{\prod_{i,j} r_{ij}!} \prod_{i,j} p_{ij}^{r_{ij}}. \tag{1.25}$$

C'est la formule de *Whittle*, voir Billingsley (1960), Anderson et Goodman (1957). Elle permet d'estimer par le maximum de vraisemblance les paramètres pour une chaîne de Markov finie.

1.4.2 Estimation par le maximum de vraisemblance (EMV) de la matrice de transition

La probabilité de transition p_{ij} peut être estimée en maximisant la formule de *Whittle*(1.25) tout en gardant les propriétés des probabilités de transition, soient $p_{ij} \geq 0$ et $\sum_{j=1}^m p_{ij} = 1$ pour tout entier i dans $\mathcal{S}$. Anderson et Goodman (1957) Avant de revenir sur la formule (1.25), introduisons d'abord la loi multinomiale $\mathcal{M}(n, p_1, p_2, \ldots, p_m)$, $n \in \mathbb{N}$ et $m \geq 1$ deux entiers. La proposition qui suit, définit une loi multinomiale et donne l'EMV pour un vecteur aléatoire de loi multinomiale. Son espace paramétrique s'écrit :

$$\{(p_1, p_2, \ldots, p_m) \in [0,1]^m : \sum_{i=1}^m p_i = 1\}. \tag{1.26}$$

Proposition 1.4.1. *Soit $X = (X_1, X_2, \ldots, X_m)$ un vecteur aléatoire qui suit une loi multinomiale $\mathcal{M}(n, p_1, p_2, \ldots, p_m)$, $n \in \mathbb{N}$ d'espace paramétrique* (1.26)*, alors, sa densité de probabilité est :*

$$\mathbb{P}(X_1 = x_1, X_2 = x_2, \ldots, X_m = x_m) = \frac{n!}{\prod_{j=1}^m x_j!} \prod_{j=1}^m p_j^{x_j}, \tag{1.27}$$

où $\sum_{j=1}^{m} x_j = n$.

L'EMV du vecteur de probabilités $p = (p_j,\ j = 1, 2, \ldots, m)$ *associé à* X *est donné par le vecteur :*

$$\hat{p} = \left(\frac{x_j}{n},\ j = 1, 2, \ldots, m\right). \tag{1.28}$$

Démonstration. La loi multinomiale est une généralisation de la loi binomiale dans laquelle chaque expérience possède m issues possibles avec la probabilité (1.27). La preuve suivante est pour l'EMV.

On veut maximiser la vraisemblance (1.27), posons :

$$L(p_1, p_2, \ldots, p_m) = \frac{n!}{\prod_{j=1}^{m} x_j!} \prod_{j=1}^{m} p_j^{x_j}.$$

On calcule d'abord le logarithme de la vraisemblance :

$$l(p_1, p_2, \ldots, p_m) = log(L(p_1, p_2, \ldots, p_m)) = log\left(\frac{n!}{\prod_{j=1}^{m} x_j!}\right) + \sum_{j=1}^{m} x_j\, log(p_j).$$

Afin de maximiser cette expression, nous devons prendre en considération la contrainte $\sum_{j=1}^{m} p_j = 1$. Celle-ci nous permet de construire la fonction de Lagrange (Voir **Annexe A** pour plus de détails) suivante :

$$Lg(p_1, p_2, \ldots, p_m, \lambda) = f(p_1, p_2, \ldots, p_n) - \lambda\, h(p_1, p_2, \ldots, p_n)$$

où $f(p_1, p_2, \ldots, p_n) = log\left(\frac{n!}{\prod_{j=1}^{m} x_j!}\right) + \sum_{j=1}^{m} x_j\, log(p_j)$,

$h(p_1, p_2, \ldots, p_n) = \sum_{j=1}^{m} p_j - 1$ et λ est le coefficient de Lagrange.

Ensuite on détermine les points zéros des dérivées partielles par rapport à λ et $p_i\,(i = 1, 2, \ldots, m)$:

$$\frac{\partial}{\partial p_i} Lg(p_1, p_2, \ldots, p_m, \lambda) = \frac{x_i}{p_i} - \lambda \qquad \frac{\partial}{\partial \lambda} Lg(p_1, p_2, \ldots, p_m, \lambda) = \sum_{j=1}^{m} p_j - 1$$

$$\frac{x_i}{\hat{p}_i} - \hat{\lambda} = 0 \qquad \sum_{j=1}^{m} \hat{p}_j - 1 = 0$$

$$\hat{p}_i = \frac{x_i}{\hat{\lambda}} \text{ ou } x_i = \hat{p}_i \hat{\lambda} \qquad \sum_{j=1}^{m} \hat{p}_j = 1$$

Or on sait que $\sum_{i=1}^{m} x_i = n$ et $\sum_{i=1}^{m} \hat{p}_i = 1$ par conséquent,

$$\sum_{i=1}^{m} x_i = \hat{\lambda} \sum_{i=1}^{m} \hat{p}_i$$
$$\hat{\lambda} = n.$$

On obtient finalement l'EVM de p_i :

$$\hat{p}_i = \frac{x_i}{n}, \ i = 1, 2, \ldots, m.$$

□

Revenons à la formule de *Whittle*. On laisse de côté les deux premiers facteurs μ_u et R^*_{vu} car ils ne s'expriment pas en fonction des probabilités p_{ij}. Ils n'ont donc pas d'influence dans leurs estimations par maximum de vraisemblance. Le terme qui nous intéresse est alors :

$$\frac{\prod_{i=1}^{m} r_{i.}!}{\prod_{i,j} r_{ij}!} \prod_{i,j} p_{ij}^{r_{ij}} = \prod_{i=1}^{m} \left[\frac{r_{i.}!}{\prod_{j=1}^{m} r_{ij}!} \prod_{j=1}^{m} p_{ij}^{r_{ij}} \right]. \tag{1.29}$$

En observant le terme (1.29) on remarque une certaine analogie avec la probabilité fonctionnelle d'une loi multinomiale pour des observations indépendantes. En effet, on sait que $\sum_{j=1}^{m} r_{ij} = r_{i.}$, que $0 \leq p_{ij} \leq 1$ et que $\sum_{j=1}^{m} p_{ij} = 1$ pour $i \in \mathcal{S}$. Ce sont les mêmes conditions énoncés dans la proposition **1.4.1**, on peut poser $k = m$, $n = r_{i.}$ et $x_j = r_{ij}$. Par conséquent le terme (1.29) représente la probabilité d'obtenir m fréquences $(r_{i1}, r_{i2}, \ldots, r_{im})$ dans m échantillon indépendant de taille $r_{i.}$ $(i = 1, 2, \ldots, m)$, pour une famille de lois multinomiales de paramètres respectifs $(p_{i1}, p_{i2}, \ldots, p_{im})$.

Par la proposition **1.4.1**, l'EMV de la matrice des probabilités de transition

$P^* = \{p_{ij}\}_{i,j\in\mathcal{S}}$ correspondant à ce type d'échantillon est :

$$\hat{P}^* = \left\{\frac{r_{ij}}{r_{i.}}\right\}_{i,j\in\mathcal{S}} \tag{1.30}$$

$$= \left\{\frac{r_{ij}}{\sum_{j=1}^{m} r_{ij}}\right\}_{i,j\in\mathcal{S}} \tag{1.31}$$

Cet estimateur possède des propriétés asymptotiques importantes décrites par Billingsley (1960) et Anderson et Goodman (1957), lesquelles sont tirées du comportement des suites $\{r_{i.}\}$ et $\{r_{ij}\}$, pour tous les entiers i et j dans $\mathcal{S}$. Ces propriétés permettent, entre autres, d'établir un test d'ajustement de Khi-deux.

Théorème 1.4.2. *Pour tout $(i,j) \in \mathcal{S}^2$, on a lorsque $n \longrightarrow \infty$:*

$$\frac{r_{i.}}{n} \xrightarrow{p.s.} \pi_i \ ; \tag{1.32}$$

$$\frac{r_{ij}}{n} \xrightarrow{p.s.} \pi_i p_{ij} \ ; \tag{1.33}$$

$$\frac{1}{\sqrt{n}}(r_{ij} - r_{i.}p_{ij}) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \pi_i p_{ij}(1-p_{ij})) \ . \tag{1.34}$$

Démonstration. Il se démontre à partir du théorème ergodique, voir Dacunha-Castelle et Duflo (1993). □

De ce théorème, on déduit les propriétés de convergence de l'EMV $\hat{p}^*_{ij}$.

Propriété 1.4.1. *Pour tout (i,j) dans $\mathcal{S}^2$, on a lorsque $n \longrightarrow \infty$:*

$$\hat{p}^*_{ij} \xrightarrow{p.s.} p_{ij} \tag{1.35}$$

$$\sqrt{n}(\hat{p}^*_{ij} - p_{ij}) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \Gamma) \tag{1.36}$$

où $\Gamma = diag\left(\frac{p_{ij}(1-p_{ij})}{\pi_i}\right)$ est une matrice diagonale, $(i,j) \in \mathcal{S}^2$.

Démonstration. Il se démontre à partir du théorème précédent, voir Dacunha-Castelle et Duflo (1993). □

1.4.3 Test de Khi-deux pour validation d'une chaîne de Markov

Le théorème que nous allons voir apporte l'information nécessaire pour la validation d'une chaîne de Markov, Billingsley (1960) et Dacunha-Castelle et Duflo (1993).

Le résultat qui suit est démontré dans Dacunha-Castelle et Duflo (1993).

Théorème 1.4.3. *Soit $\{X_n\}_{n\geq 1}$ une chaîne de Markov de matrice de transition $\mathbf{P} = \{p_{ij}\}$ sur un espace $\mathcal{S}$ à s éléments qui forme une seule classe de récurrence, et soit k le nombre de couples (i,j) dans $\mathcal{S}$ pour lesquelles $p_{ij} > 0$. On a que :*

$$Z = \sum_{i,j} \frac{(r_{ij} - p_{ij} r_{i.})^2}{p_{ij} r_{i.}} = \sum_{i,j} \frac{(\hat{p}_{ij} - p_{ij})^2}{p_{ij}} \xrightarrow{\mathcal{L}} \begin{cases} \chi^2_{m(m-1)} & \text{si tous les } p_{ij} > 0 \\ \chi^2_{(k-s)} & \text{s'il existe au moins un } p_{ij} = 0. \end{cases} \tag{1.37}$$

Grâce à la convergence en loi de la statistique (1.37), il est possible d'établir un test d'ajustement de type χ^2 sur les probabilités de transition estimées par maximisation de vraisemblance afin de valider si la suite de variables aléatoires à l'étude est bel et bien une chaine de Markov de matrice de transition $\mathbf{P}$.

Remarque 1.4.1. *L'interprétation du nombre de degrés de liberté est la suivante : il y a k paramètres non-nuls, avec $k \geq s$ vu qu'au moins un coefficient sur chaque ligne de la matrice stochastique $\mathbf{P}$ est non-nul.*

Ce chapitre résume les points essentiels dont on a besoin pour entrer dans le vif du sujet de ce mémoire.

Dans le chapitre suivant nous introduisons un cas particulier des processus markoviens. Il s'agit des modèles de Markov cachés (MMC).

CHAPITRE II

MODÈLES DE MARKOV CACHÉS À TEMPS DISCRET

2.1 Présentation et caractéristiques

2.1.1 Notions de base

Certains phénomènes peuvent se décrire adéquatement par des chaînes de Markov. Cela suppose alors que les états et les probabilités de transition sont bien connus. Toutefois il est fréquent d'observer une fonction de ces états, et plus généralement une variable aléatoire associée aux états. On parle alors de Modèles de Markov Cachés (MMC). On peut les décrire comme des fonctions probabilistes d'une chaîne de Markov.

Brièvement, un MMC à temps discret peut se définir comme une modélisation doublement stochastique : un processus dit «caché» parfaitement modélisé par une chaîne de Markov discrète et un processus observable dont la distribution dépend des états du processus caché.

Il existe diverses sortes de MMC afin de répondre à plusieurs types de problèmes. Dans ce mémoire, on s'intéresse particulièrement aux MMC de premier ordre.

Les éléments exposés dans ce chapitre sont, de façon générale, tirés des articles et des ouvrages suivants :

Rabiner (1989), Rabiner et Juang (1993), Weigend et Shi (1997), Mamon et Elliott

(2007) et Bhar et Hamori (2004).

2.1.2 Caractéristiques des MMC

Nous introduisons les caractéristiques des MMC. Pour faire cela, considérons les éléments notés N, M, A, B, μ et $\underset{\sim}{\lambda}$, et définie comme suit :

1. $\underset{\sim}{\lambda}$ est un ensemble paramètrique ;
2. N est le nombre d'états cachés dans le modèle. On note l'ensemble d'états cachés par $S = \{S_1, S_2, \ldots, S_N\}$ et l'état au temps t par y_t ;
3. M est le nombre de symboles distincts observables par états. On note ces symboles par o_k où $k = 1, 2, \ldots, M$, et l'observation au temps t par O_t ;
4. $A = \{a_{ij}\}$ est la matrice de transition des états cachés où

$$a_{ij} = \mathbb{P}(y_{t+1} = S_j | y_t = S_i, \underset{\sim}{\lambda}), \text{ avec } i = 1, 2, \ldots, N \text{ et } j = 1, 2, \ldots, N ; \quad (2.1)$$

5. $B = \{b_{S_i}(o_k)\}$ est la matrice de probabilité des observations k dans l'état S_i où

$$b_{S_i}(o_k) = \mathbb{P}(O_t = o_k | y_t = S_i, \underset{\sim}{\lambda}), \text{ avec } i = 1, 2, \ldots, N \text{ et } k = 1, 2, \ldots, M ; \quad (2.2)$$

   La matrice B contient les probabilités d'observer au temps t le symbole k sachant qu'au même instant le modèle est dans l'état caché S_i ;

6. $\mu = \{\mu_i\}$ la distribution de l'état initial du modèle où

$$\mu_i = \mathbb{P}(y_1 = S_i | \underset{\sim}{\lambda}), \ i = 1, 2, \ldots, N ; \quad (2.3)$$

   Ce vecteur contient la probabilité qu'au moment initial ($t = 1$), le modèle se trouve dans l'état caché S_i.

Un MMC est un quintuplet $\underset{\sim}{\lambda}$ qui se définie par

$$\underset{\sim}{\lambda} = (N, M, \mu, A, B), \quad (2.4)$$

Soit T le nombre d'observations. On définit respectivement par $O(T) = O_1 O_2 \dots O_T$ et $y(T) = y_1\, y_2\, \dots\, y_T$, des séquences d'observations ainsi que d'états cachés pouvant être obtenues à partir du modèle $\underset{\sim}{\lambda}$.

Remarque 2.1.1. La notation des MMC est très souvent réduite au triplet $\underset{\sim}{\lambda} = (\mu, A, B)$ car A est une matrice $N \times N$, et B une matrice $N \times M$.

Exemple 2.1.1. *Considérons le modèle λ décrit par la* ***figure 2.1****. Les symboles observables possibles sont $o_1 = I$, $o_2 = II$ et $o_3 = III$. Nous pouvons voir que $M = N = 3$. Supposons que la loi de l'état initial est $\mu = \{0,\ 1/2,\ 1/2\}$ et qu'on a les valeurs suivantes :*

$$b_{S_1}(I) = 0, \quad b_{S_1}(II) = 1/2, \quad b_{S_1}(III) = 1/2;$$
$$b_{S_2}(I) = 1/2, \quad b_{S_2}(II) = 1/2, \quad b_{S_2}(III) = 0;$$
$$b_{S_3}(I) = 1/2, \quad b_{S_3}(II) = 0, \quad b_{S_2}(III) = 1/2.$$

On a :

$$\mathbf{A} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 2/3 & 0 & 1/3 \\ 2/3 & 1/3 & 0 \end{bmatrix} \quad \textit{et } \mathbf{B} = \begin{bmatrix} 0 & 1/2 & 1/2 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1/2 \end{bmatrix}.$$

Générons une séquence d'observations $O(3) = O_1\, O_2\, O_3$ à partir de ce modèle.
On sait que $\mu_1 = 0$, donc le modèle ne peut pas démarrer dans l'état S_1.
Choisissons alors aléatoirement entre les états S_2 et S_3, et disons que le modèle s'initialise en S_3. Après, nous devons produire un symbole observable en choisissant au hasard entre I et II (car $b_{S_3}(II) = 0$, il est donc impossible d'observer II à partir de l'état caché S_3), disons que l'on obtient I.

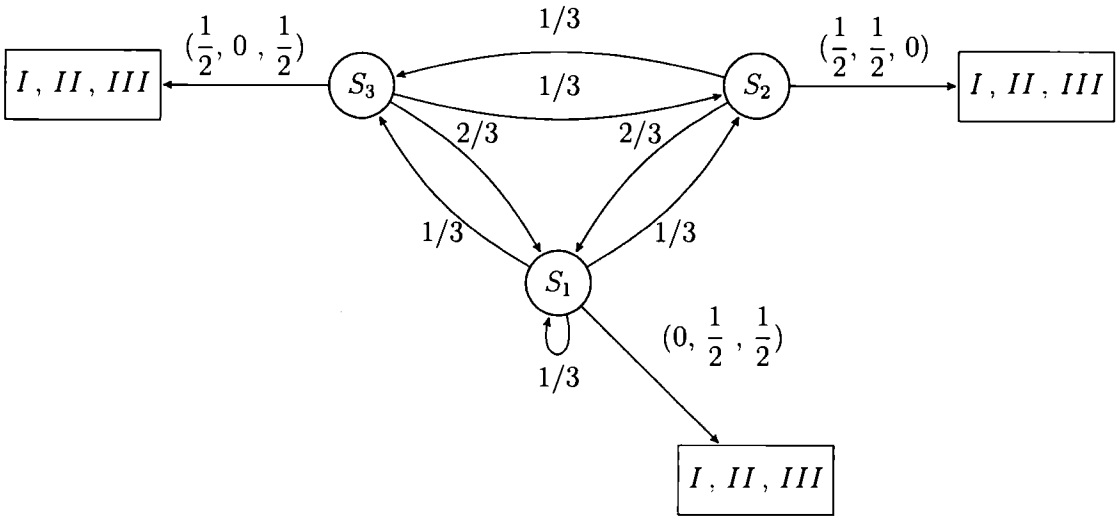

Figure 2.1: Diagramme MMC Exemple **2.1.1**

Le couple état-observation obtenu est (S_3, I). *En procédant de façon identique, nous pourrions obtenir la séquence suivante de couples :*

$$\{(S_3, I), (S_1, II), (S_1, II)\}.$$

La séquence d'observation est $O(3) = (I, II, II)$.
Notons qu'une autre séquence aurait pu être observée, mais que toutefois dans notre exemple, elle débuterait par l'observation I.

Il existe un algorithme dans l'article de Rabiner (1989) utilisé pour générer effacement par simulation une séquence d'observations à partir d'un MMC.

2.1.3 Production d'une séquence d'observations par simulation

Pour des valeurs données de N, M, A, B et μ, *le MMC peut être utilisé pour générer une séquence d'observations* $O(T) = O_1 O_2 \ldots O_T$ *de la manière*

suivante :

Algorithme de Génération d'une séquence d'état

1. *Pour* $i = 1 : N$, *choisir aléatoirement un état initial* $y_1 = S_i$ *selon la loi* μ ;
2. *définir* $t = 1$;
3. *Pour* $k = 1 : M$, *choisir* $O_t = o_k$ *selon la distribution des observations dans l'état* S_i, *c'est à dire selon* $b_{S_i}(k)$;
4. *Pour* $t = 1 : T$ *et* $j = 1 : N$, *choisir l'état* $y_{t+1} = S_j$ *selon les probabilités de transitions de l'état* S_i, *c'est à dire selon* a_{ij} ;
5. *définir* $t = t + 1$; *si* $t < T$ *alors retour à l'étape* ***3.*** *Sinon fin de la procédure.*

L'exemple **2.1.1** illustre comment il est possible de générer une séquence d'observations à partir de la distribution de l'état initial, les probabilités de transition et des observations du modèle. En situation réelle, nous n'observons que les sorties $O(3) = (I, II, II)$ et devons alors faire de l'inférence sur les états sous-jacents pour que le modèle soit efficace.

Afin de pouvoir exploiter le modèle, trois problèmes fondamentaux doivent être résolus, à savoir :

1. l'évaluation de la probabilité d'observer une séquence de symboles donnée à partir d'un modèle $\underset{\sim}{\lambda}$;
2. le décodage de la séquence d'états optimale susceptible d'avoir générée une séquence d'observations arbitraire ;
3. la réestimation des paramètres du modèles (modélisation) afin de maximiser la probabilité d'observer une séquence d'observations.

Dans la section 2.2 suivante, on décrit en profondeur ces trois problèmes et on

donne des méthodes de résolution.

2.2 Les propriétés fondamentales des MMC

2.2.1 Problème d'évaluation efficace de l'état le plus probable

Le premier but de ce principe est de déterminer une manière efficace pour évaluer la probabilité $\mathbb{P}(O(t)|\underset{\sim}{\lambda})$ d'observer la séquence $O(t) = O_1 O_2 \ldots O_t$ étant donné le MMC de paramètres $\underset{\sim}{\lambda}$.

Il est primordial de voir que cette probabilité peut s'exprimer sous la forme suivante :

$$\mathbb{P}(O(t)|\,\underset{\sim}{\lambda}) = \sum \mathbb{P}(O(t),\, y(t)|\underset{\sim}{\lambda}). \tag{2.5}$$

La somme est faite pour toutes les séquences possibles d'états $y(t)$.

Ainsi, nous pouvons évaluer cette probabilité de manière directe en procédant comme suit :

$$\begin{aligned} \mathbb{P}(O(t)|\underset{\sim}{\lambda}) &= \sum \mathbb{P}(O(t),\, y(t)|\underset{\sim}{\lambda}), & (2.6)\\ &= \sum \mathbb{P}(O(t)|y(t), \underset{\sim}{\lambda})\ \mathbb{P}(y(t)|\underset{\sim}{\lambda}), & (2.7) \end{aligned}$$

avec la séquence d'états $y(t) = y_1\, y_2 \ldots y_t$ où y_1 est l'état initial du modèle.

Evaluons ensuite les probabilités $\mathbb{P}(O(t)|y(t), \underset{\sim}{\lambda})$ et $\mathbb{P}(y(t)|\underset{\sim}{\lambda})$ (2.7). On a :

$$\begin{aligned} \mathbb{P}(O(t)|y(t), \underset{\sim}{\lambda}) &= \prod_{j=1}^{t} \mathbb{P}(O_j|y_j, \underset{\sim}{\lambda}) & (2.8)\\ &= \prod_{j=1}^{t} b_{y_j}(O_j) & (2.9)\\ &= b_{y_1}(O_1)\ b_{y_2}(O_2) \ldots b_{y_t}(O_t). & (2.10) \end{aligned}$$

La probabilité d'avoir une séquence d'états $y(t)$ peut s'exprimer de la manière

suivante par le théorème **2.2.1** :

$$\begin{aligned}\mathbb{P}(y(t)|\underset{\sim}{\lambda}) &= \mathbb{P}(y_1|\underset{\sim}{\lambda})\ \mathbb{P}(y_2|y_1,\underset{\sim}{\lambda})\ \mathbb{P}(y_3|y_1,y_2,\underset{\sim}{\lambda}) \qquad (2.11)\\ &\quad \ldots\ \mathbb{P}(y_t|y_1,\ldots,y_{t-1},\underset{\sim}{\lambda})\\ &= \mu_{y_1}\ a_{y_1y_2}\ a_{y_2y_3}\ \ldots\ a_{y_{t-1}y_t}. \qquad (2.12)\end{aligned}$$

Nous pouvons alors réécrire les probabilités de l'égalité (2.7) à partir des expressions (2.10) et (2.12), soit :

$$\begin{aligned}\mathbb{P}(O(t),\, y(t)|\underset{\sim}{\lambda}) &= \mathbb{P}(O(t)|y(t),\underset{\sim}{\lambda})\ \mathbb{P}(y(t)|\underset{\sim}{\lambda})\\ &= \mu_{y_1}b_{y_1}(O_1)\ a_{y_1y_2}b_{y_2}(O_2)\ \ldots\ a_{y_t\ _1y_t}b_{y_t}(O_t). \qquad (2.13)\end{aligned}$$

Ainsi en sommant la probabilité conjointe (2.13) sur toutes les séquences d'états possible $y(t)$, nous obtenons la probabilité d'observer la séquence $O(t)$ étant donné le modèle (égalité (2.5)).

$$\mathbb{P}(O(t)|\underset{\sim}{\lambda}) = \sum_{y_1,y_2,\ldots,y_t} \mu_{y_1}b_{y_1}(O_1)\ a_{y_1y_2}b_{y_2}(O_2)\ \ldots\ a_{y_{t-1}y_t}b_{y_t}(O_t). \qquad (2.14)$$

Cette égalité peut être décrite de façon algorithmique à partir des probabilités définies dans les caractéristiques des MMC :

(i) Au temps $t = 1$, le MMC démarre initialement à l'état y_1 avec probabilité μ_{y_1} et produit une observation O_1 avec probabilité $b_{y_1}(O_1)$;

(ii) À l'instant suivant $t = 2$, le MMC transite à l'état y_2 de l'état précédent y_1 avec probabilité $a_{y_1y_2}$ et produit une observation O_2 avec probabilité $b_{y_2}(O_2)$;

(iii) La procédure se répète jusqu'à un certain temps t ;

(iv) La dernière transition de l'état y_{t-1} à l'état y_t se fait avec probabilité $a_{y_{t-1}y_t}$ et la production de l'observation finale O_t avec probabilité $b_{y_t}(O_t)$.

Il s'agit là d'un calcul direct de la probabilité $\mathbb{P}(O(t)|\underset{\sim}{\lambda})$. Mais cette façon de procéder nous mène à des calculs d'ordre exponentiel ($T \times N^T$), qui sont très coûteux en temps de calcul.
Par la suite, nous décrivons un procédé récursif plus astucieux et plus rapide qui permet réduire le temps de calcul de la probabilité (2.5).

Il s'agit de l'algorithme *Forward-Backward* ou *progressif-rétrogressif* en français Rabiner (1989), Miller (2011a). Cette méthode suppose que la distribution initiale du modèle, les probabilités de transition des états, les probabilités des observations dans chaque état sont connues ($\underset{\sim}{\lambda}$ connu).
Avant de poursuivre sur cette méthode, nous introduisons d'abord un théorème sur les probabilités conditionnelles très utile pour la suite. Dantzer (2007)

Théorème 2.2.1. *Si $n \geq 2$ et $F_1, F_2, \ldots, F_n$ sont des évènements dans l'ensemble $\Im$ tels que $\mathbb{P}\left(\bigcap_{k=1}^{n-1} F_k\right) \neq 0$, alors :*

$$\mathbb{P}\left(\bigcap_{k=1}^{n} F_k\right) = \mathbb{P}(F_1)\ \mathbb{P}(F_2|F_1)\ \mathbb{P}(F_3|F_1 \cap F_2)\ \ldots\ \mathbb{P}\left(F_n|\bigcap_{k=1}^{n-1} F_k\right). \tag{2.15}$$

Pour une séquence ordonnée en fonction du temps, dans le cas des modèles Markoviens on a :

$$\mathbb{P}\left(\bigcap_{k=1}^{n} F_k\right) = \mathbb{P}(F_1)\ \mathbb{P}(F_2|F_1)\ \mathbb{P}(F_3|F_2)\ \ldots\ \mathbb{P}\left(F_n|F_{n-1}\right). \tag{2.16}$$

Démonstration. Remarquons que $\bigcap_{k=1}^{p} F_k \subset \bigcap_{k=1}^{n-1} F_k$ pour tout entier p et $n \geq 2$ tels que $1 \leq p \leq n-1$ et donc que $\mathbb{P}\left(\bigcap_{k=1}^{p} F_k\right) \geq \mathbb{P}\left(\bigcap_{k=1}^{n-1} F_k\right) > 0$. Par conséquent les probabilités conditionnelles introduites sont bien définies. Avec la définition des

probabilités conditionnelles, on peut vérifier :

$$\begin{aligned}\mathbb{P}(F_1)\prod_{p=1}^{n-1}\mathbb{P}(F_{p+1}|\bigcap_{k=1}^{p}F_k) &= \mathbb{P}(F_1)\ \mathbb{P}(F_2|F_1)\ \mathbb{P}(F_3|F_1\cap F_2)\ \ldots\ \mathbb{P}\left(F_n|\bigcap_{k=1}^{n-1}F_k\right)\\ &= \mathbb{P}(F_1)\ \frac{\mathbb{P}(F_1\cap F_2)}{\mathbb{P}(F_1)}\ \frac{\mathbb{P}(F_1\cap F_2\cap F_3)}{\mathbb{P}(F_1\cap F_2)}\ \cdots\ \frac{\mathbb{P}\left(\bigcap_{k=1}^{n}F_k\right)}{\mathbb{P}\left(\bigcap_{k=1}^{n-1}F_k\right)}\\ &= \mathbb{P}\left(\bigcap_{k=1}^{n}F_k\right)\end{aligned}$$

La preuve pour les modèles de Markov est directe, ces modèles étant sans mémoire, on peut donc écrire pour des séquences ordonnées $F_1, F_2, \ldots, F_n$:

$$\mathbb{P}\left(\bigcap_{k=1}^{n}F_k\right) = \mathbb{P}(F_1)\ \mathbb{P}(F_2|F_1)\ \mathbb{P}(F_3|F_2)\ \ldots\ \mathbb{P}(F_n|F_{n-1})\,. \tag{2.17}$$

□

Méthode *Forward*

Considérons une séquence donnée d'observations $O(T) = O_1\, O_2\, \ldots\, O_T$.
Le but de cette méthode est d'effectuer un calcul progressif (*Forward*) qui permet par la suite d'obtenir la probabilité des t premières observations $O(t)$, se terminant dans l'état S_i, $i = 1, 2, \ldots, N$.
Soit la quantité $\alpha_t(i)$ définie par :

$$\alpha_t(i) = \mathbb{P}(O(t),\, y_t = S_i|\underline{\lambda}),\ t = 1, 2, \ldots, T \text{ et } i = 1, 2, \ldots, N. \tag{2.18}$$

A l'instant initial, on a la valeur de $\alpha_1(i)$:

$$\begin{aligned}\alpha_1(i) &= \mathbb{P}(O_1,\, y_1 = S_i|\underline{\lambda}) &(2.19)\\ &= \mathbb{P}(y_1 = S_i|\underline{\lambda})\,\mathbb{P}(O_1|y_1 = S_i, \underline{\lambda}) &(2.20)\\ &= \mu_i\ b_{S_i}(O_1). &(2.21)\end{aligned}$$

On remarque que le MMC s'initialise à l'état S_i avec probabilité μ_i et produit une observation O_1 avec probabilité $b_{S_i}(O_1)$ (2.21).

Ensuite, analysons plus en détails la valeur de $\alpha_{t+1}(j)$ afin d'en ressortir une équation récursive :

$$\begin{aligned}
\alpha_{t+1}(j) &= \mathbb{P}(O(t+1),\, y_{t+1} = S_j|\underset{\sim}{\lambda}) && (2.22)\\
&= \sum_{i=1}^{N} \mathbb{P}(O(t) \wedge O_{t+1},\, y_t = S_i,\, y_{t+1} = S_j|\underset{\sim}{\lambda}) && (2.23)\\
&= \sum_{i=1}^{N} \mathbb{P}(O(t),\, y_t = S_i|\underset{\sim}{\lambda})\ \mathbb{P}(O_{t+1},\, y_{t+1} = S_j|O(t),\, y_t = S_i,\, \underset{\sim}{\lambda}) && (2.24)\\
&= \sum_{i=1}^{N} \alpha_t(i)\ \mathbb{P}(O_{t+1},\, y_{t+1} = S_j|y_t = S_i,\, \underset{\sim}{\lambda}) && (2.25)\\
&= \sum_{i=1}^{N} \alpha_t(i)\ \mathbb{P}(O_{t+1}|y_{t+1} = S_j,\, \underset{\sim}{\lambda})\, \mathbb{P}(y_{t+1} = S_j|y_t = S_i,\, \underset{\sim}{\lambda}) && (2.26)\\
&= \sum_{i=1}^{N} \alpha_t(i)\ b_{S_j}(O_{t+1})\ a_{ij}. && (2.27)
\end{aligned}$$

Illustrons les valeurs de $\alpha_t(i)$ par un exemple.

Exemple 2.2.1. *Reprenons les données de l'exemple* ***2.1.1****. La séquence d'observations était* $O(3) = (I,\, II,\, II)$*. Calculons* $\alpha_t(i)$*, avec* $t = 1,\, 2,\, 3$ *et* $i = 1,\, 2,\, 3$*.*
À l'instant initial $t = 1$ *pour chaque état* $S_1,\, S_2,$ *et* S_3 *:*

$$\alpha_1(1) = \mathbb{P}(O_1 = I, y_1 = S_1|\underset{\sim}{\lambda}) = \mu_1 * b_{S_1}(I) = 0 * 0 = 0;$$

$$\alpha_1(2) = \mathbb{P}(O_1 = I, y_1 = S_2|\underset{\sim}{\lambda}) = \mu_2 * b_{S_2}(I) = 1/2\ *\ 1/2 = 1/4;$$

$$\alpha_1(3) = \mathbb{P}(O_1 = I, y_1 = S_3|\underset{\sim}{\lambda}) = \mu_3 * b_{S_3}(I) = 1/2\ *\ 1/2 = 1/4.$$

Aux instants $t = 2$ *et* $t = 3$ *pour chaque état* $S_1,\, S_2,$ *et* S_3 *:*

$$\begin{aligned}
\alpha_2(1) &= \sum_{i=1}^{3} \alpha_1(i) * b_{S_1}(O_2) * a_{i1}\\
&= \alpha_1(1) * b_{S_1}(II) * a_{11} + \alpha_1(2) * b_{S_1}(II) * a_{21} + \alpha_1(3) * b_{S_1}(II) * a_{31}\\
&= 0 * 1/2 * 1/3\ +\ 1/4 * 1/2 * 2/3\ +\ 1/4 * 1/2 * 2/3\\
\alpha_2(1) &= 1/6.
\end{aligned}$$

De la même manière on obtient :

$$\alpha_2(2) = 1/24;\ \alpha_2(3) = 0;\ \alpha_3(1) = 1/24;\ \alpha_3(2) = 1/36;\ \alpha_3(3) = 0.$$

Par cette méthode progressive, en sommant la quantité $\alpha_t(i)$ à chaque état S_i, on obtient la probabilité de n'observer que la séquence $O(t)$ compte tenu du MMC de paramètre λ.

$$\sum_{i=1}^{N} \alpha_t(i) = \mathbb{P}(O(t)|\underset{\sim}{\lambda}). \tag{2.28}$$

De toute cette analyse, on comprend donc que l'algorithme *Forward* donne deux informations, à savoir $\alpha_t(i)$ et $P(O(T)|\underset{\sim}{\lambda})$. Résumons les étapes de cette méthode :

Algorithme *Forward*

1. ***Pour*** $i = 1 : N$ (croissant), ***faire***
   $\alpha_1(i) = \mu_i \; b_{S_i}(O_1)$;
2. ***Pour*** $t = 1 : T-1$, ***faire***
   Pour $j = 1 : N$, ***faire***
   $\alpha_{t+1}(i) = \left[\sum_{i=1}^{N} \alpha_t(i) \; a_{ij}\right] \; b_{S_j}(O_{t+1})$;
   Fin Pour
   Fin Pour
   Fin Pour
3. *Finalisation*
   $\mathbb{P}(O(T)|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$.

Méthode *Backward*

Soit la quantité rétrogressive (*Backward*) $\beta_t(i)$ définie par :

$$\beta_t(i) = \mathbb{P}(O_{t+1:T}|\, y_t = S_i, \underset{\sim}{\lambda}), \quad t = T-1, T-2, \ldots, 1, \tag{2.29}$$

où $O_{t+1:T} = O_{t+1} \, O_{t+2} \ldots O_T$.

$\beta_t(i)$ est la probabilité d'observer la séquence partielle ultérieure $O_{t+1:T}$, sachant que le MMC de paramètre $\underset{\sim}{\lambda}$ était dans l'état S_i à l'instant t.

Contrairement à la méthode précédente, la valeur de $\beta_t(i)$ à l'échéance est arbitrairement choisie :

$$\beta_T(i) = 1. \tag{2.30}$$

Rappelons que les seules observations données sont $O_1, O_2, \ldots, O_T$ et que la probabilité $\beta_t(i)$ n'est définie que pour des temps discrets strictement inférieurs à notre échéance T. Cela peut nous aider à justifier ce choix.
De même que pour la variable progressive, la transformation de la probabilité conditionnelle $\beta_t(i)$ que nous traitons ci-dessous permet de retrouver une forme récursive rétroactive :

$$\begin{aligned}
\beta_t(i) &= \mathbb{P}(O_{t+1:T} | \, y_t = S_i, \, \underset{\sim}{\lambda}) \\
&= \sum_{j=1}^{N} \mathbb{P}(O_{t+1:T}, \, y_{t+1} = S_j | \, y_t = S_i, \, \underset{\sim}{\lambda}) \\
&= \sum_{j=1}^{N} \mathbb{P}(O_{t+1} \wedge O_{t+2:T}, \, y_{t+1} = S_j | \, y_t = S_i, \, \underset{\sim}{\lambda}) \\
&= \sum_{j=1}^{N} \mathbb{P}(y_{t+1} = S_j | y_t = S_i, \, \underset{\sim}{\lambda}) \, \mathbb{P}(O_{t+2:T} | y_{t+1} = S_j, \, y_t = S_i, \, \underset{\sim}{\lambda}) \\
&\qquad\qquad \mathbb{P}(O_{t+1} | y_{t+1} = S_j, \, O_{t+2:T}, \, y_t = S_i, \, \underset{\sim}{\lambda}); \\
&= \sum_{j=1}^{N} \mathbb{P}(y_{t+1} = S_j | \, y_t = S_i, \, \underset{\sim}{\lambda}) \, \mathbb{P}(O_{t+2:T} | y_{t+1} = S_j, \, \underset{\sim}{\lambda}) \\
&\qquad\qquad \mathbb{P}(O_{t+1} | y_{t+1} = S_j, \, \underset{\sim}{\lambda}); \\
&= \sum_{j=1}^{N} a_{ij} \; \beta_{t+1}(j) \; b_{S_j}(O_{t+1}).
\end{aligned} \tag{2.31}$$

On illustre les valeurs de $\beta_t(i)$ par un exemple.

Exemple 2.2.2. *On considère les données de l'exemple* ***4.1.1****. On a la séquence d'observation* $O(3) = (I, \, II, \, II)$*. On évalue les valeurs de* $\beta_t(i)$ *pour* $t = 1, 2, 3$ *et* $i = 1, 2, 3$.

À l'échéance $t = 3$ pour chaque états S_1, S_2, et S_3 :

$$\beta_3(1) = 1$$

$$\beta_3(2) = 1$$

$$\beta_3(3) = 1$$

Aux instants $t = 2$ et $t = 3$ pour chaque états S_1, S_2, et S_3, on obtient par récursion :

$$\begin{aligned}
\beta_2(1) &= \sum_{j=1}^{3} \beta_3(j) * b_{S_j}(O_3) * a_{1j} \\
&= \beta_3(1) * b_{S_1}(II) * a_{11} + \beta_3(2) * b_{S_2}(II) * a_{12} + \beta_3(3) * b_{S_3}(II) * a_{13} \\
&= 1 * 1/2 * 1/3 + 1 * 1/2 * 1/3 + 1 * 0 * 1/3 \\
\beta_2(1) &= 1/3
\end{aligned}$$

De la même manière on obtient :

$$\beta_2(2) = 1/3;\ \beta_2(3) = 1/2;\ \beta_1(1) = 1/9;\ \beta_1(2) = 1/9;\ \beta_1(3) = 1/6.$$

On peut dès lors affirmer que l'algorithme *Backward* ne produit qu'une seule information, soit $\beta_t(i)$:

Algorithme *Backward*

1. ***Pour*** $i = 1 : N$, ***faire***

   $\beta_T(i) = 1$;

2. ***Pour*** $t = T - 1 : (-1) : 1$ (décroissant), ***faire***

   Pour $j = 1 : N$, ***faire***

   $\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j)\ a_{ij}\ b_{S_j}(O_{t+1})$;

   Fin Pour

   Fin Pour

   Fin Pour.

Remarque 2.2.1. L'algorithme *Backward* est construit similairement à l'algorithme *Forward*. Cependant, il ne permet pas de calculer directement la probabilité $\mathbb{P}(O(T)|\underset{\sim}{\lambda}))$. La différence se voit au niveau des observations dès l'instant initial. En d'autres mots, dans le cas de l'algorithme *Forward* la première observation de la séquence correspond à O_1, la deuxième à O_2, ainsi de suite jusqu'à la dernière correspondant à O_T ; alors que pour l'algorithme *Backward* la première observation correspond à O_T, la deuxième à O_{T-1}, ainsi de suite jusqu'à la dernière qui correspond à O_1.

À partir des quantités $\alpha_t(i)$ et $\beta_t(i)$ (des deux algorithmes réunis), on peut calculer la probabilité $\mathbb{P}(O(T)|\,\underset{\sim}{\lambda})$ d'observer la séquence $O(T)$ à chaque instant t et aussi évaluer plus facilement la probabilité $\gamma_t(i) = \mathbb{P}(y_t = S_i|O(T), \underset{\sim}{\lambda})$ d'être dans l'état S_i à un certain temps t étant donné la séquence d'observations $O(T)$ et les paramètres $\underset{\sim}{\lambda}$. On a :

$$\begin{aligned}
\mathbb{P}(O(T)|\,\underset{\sim}{\lambda}) &= \sum_{i=1}^{N} \mathbb{P}(O(T),\, y_t = S_i|\,\underset{\sim}{\lambda}) = \sum_{i=1}^{N} \mathbb{P}(O(t), O_{t+1:T},\, y_t = S_i|\,\underset{\sim}{\lambda});\\
&= \sum_{i=1}^{N} \mathbb{P}(O(t),\, y_t = S_i|\,\underset{\sim}{\lambda})\ \mathbb{P}(O_{t+1:T}|\,O(t),\, y_t = S_i,\, \underset{\sim}{\lambda});\\
\mathbb{P}(O(T)|\,\underset{\sim}{\lambda}) &= \sum_{i=1}^{N} \alpha_t(i)\ \beta_t(i). \qquad (2.32)
\end{aligned}$$

$$\gamma_t(i) = \frac{\mathbb{P}(y_t = S_i, O(T)|\underset{\sim}{\lambda})}{\mathbb{P}(O(T)|\underset{\sim}{\lambda})} = \frac{\alpha_t(i)\ \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\ \beta_t(i)}. \qquad (2.33)$$

Remarque 2.2.2. Les observations sont indépendantes à chaque instant t. Dans le cas ci-dessus par exemple, la séquence partielle O_{t+1}, O_{t+2}, ..., O_T ne dépend pas des observations précédentes O_1, O_2, ..., O_t.

L'évaluation de $\mathbb{P}(O(T)|\lambda)$ grâce à la procédure *forward-backward* passe de l'ordre de calcul $T \times N^T$ à $T \times N^2$ (Pour $N = 3$ et $T = 100$ par exemple, la procédure réduit l'ordre du nombre de calcul d'environ 10^{50} à 900). Pour plus de détails voir

Rabiner (1989).

Algorithme *Forward-Backward*

1. ***Appliquer*** *initialement les algorithmes Forward et Backward* ;
2. ***Pour*** $t = 1 : T$, ***faire***

   Pour $i = 1 : N$, ***faire***

   $$\gamma_t(i) = \frac{\alpha_t(i)\ \beta_t(i)}{\sum_{k=1}^{N} \alpha_t(k)\ \beta_t(k)} ;$$

   Fin Pour

   Fin Pour.

Dans cette partie, nous venons d'expliquer une méthode efficace pour évaluer la probabilité d'observer une séquence $O(T)$ à partir d'un MMC de paramètres $\underset{\sim}{\lambda}$. Cependant, avec cette méthode, il nous est impossible de déterminer une séquence d'états la plus probable pour générer une séquence d'observations connues.
Dans la section suivante, nous discutons d'une nouvelle méthode permettant d'évaluer ce genre de problème.

2.2.2 Décodage de la séquence optimale d'états cachés $y(T)$ ayant produit la séquence d'observations $O(T)$

Nous cherchons à découvrir dans ce cas la partie cachée du modèle, trouver la séquence d'états la plus probable d'avoir généré une séquence précise d'observation à partir du modèle donné.
Plusieurs critères d'optimalité peuvent être considérés. Nous verrons que le choix de ces critères a un impact impotant sur l'estimation de la séquence des états à trouver.
Plus mathématiquement, notre but est de calculer la quantité, Rabiner (1989) et

Miller (2011b), $\underset{y(T)}{argmax}\ \mathbb{P}(y(T) \mid O(T),\ \underset{\sim}{\lambda})$.
On peut remarquer que :

$$\underset{y(T)}{argmax}\ \mathbb{P}(y(T) \mid O(T),\ \underset{\sim}{\lambda}) = \underset{y(T)}{argmax}\ \frac{\mathbb{P}(O(T),\ y(T) \mid \underset{\sim}{\lambda})}{\mathbb{P}(O(T) \mid \underset{\sim}{\lambda})} \tag{2.34}$$

$$= \underset{y(T)}{argmax}\ \mathbb{P}(y(T), O(T) \mid \underset{\sim}{\lambda}). \tag{2.35}$$

Une approche de résolution de ce problème provient de la programmation dynamique, qui est un mode opératoire algorithmique pour des problèmes d'optimisation dans un ensemble fini de solutions, mais de grande cardinalité.
En supposant une trajectoire décomposée en plusieurs étapes, l'idée générale est de prendre une décision optimale à chaque étape pour chaque état possible. C'est le *principe d'optimalité* de Bellman (1954).
Nous énonçons particulièrement ce principe car il joue un rôle important dans l'algorithme qui nous permet de résoudre le problème présenté.

Définition 2.2.1. *Notion de sous trajectoire*
Étant donné la trajectoire ou ensemble de décision, $U(T) = u_1\ u_2\ \ldots\ u_T$ *qui génère la séquence d'états* $y(T) = y_1\ y_2\ \ldots\ y_T$, *la séquence de décision* $U(t) = u_1\ u_2\ \ldots\ u_t$, $t = 1, 2, \ldots, T-1$, *qui génère la séquence d'états* $y(t) = y_1\ y_2\ \ldots\ y_t$ *est appelée une sous trajectoire de* $U(T)$.

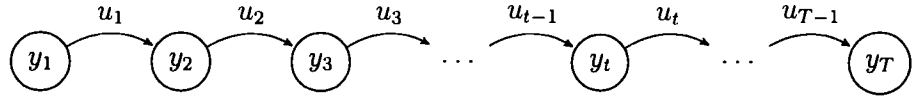

Figure 2.2: Trajectoire et sous trajectoire

Proposition 2.2.1. *Principe d'optimalité de Bellman (1954)*
Toute sous trajectoire d'une trajectoire optimale est elle-même optimale.
De cette façon, si l'ensemble de décision $U(T) = u_1\ u_2\ \ldots\ u_T$ générant la séquence d'états $y(T) = y_1\ y_2\ \ldots\ y_T$ est optimale pour aller de l'état initial y_1 à l'état

échéant y_T alors la sous trajectoire $U(t) = u_1\ u_2\ \ldots\ u_t$ est optimale pour aller initialement de l'état y_1 à l'état y_t (Figure 2.2).

Ce principe explique que la solution d'un problème global peut être obtenue en le décomposant en sous-problèmes plus simples.

Maintenant, on revient au problème de maximisation de la probabilité en (2.36). On va développer un algorithme reposant sur le *principe d'optimalité de Bellman* appelé *algorithme de Viterbi.* Viterbi (1967), Omura (1969)
L'*algorithme de Viterbi* suppose que la séquence d'observations $O(T) = O_1, O_2, \ldots, O_T$ et les paramètres du MMC sont connus. On a pour but d'évaluer l'expression (2.35). Avant cela, introduisons la proposition suivante :

Proposition 2.2.2. *Si f et g sont deux fonctions telles que $f(a) > 0$ pour tout a, et $g(a,b) \geq 0$ pour tout a, b alors :*

$$\max_{a,b}\ f(a)\, g(a,b) = \max_{a}\ \left[f(a)\ \max_{b}\ g(a,b)\right]. \tag{2.36}$$

Démonstration. La fonction $f(.)$ ne dépend pas de b. On peut donc écrire :

$$\max_{b}\ f(a)\, g(a,b) = f(a)\,[\max_{b}\ g(a,b)],$$
$$\max_{a}\left[\max_{b}\ f(a)\, g(a,b)\right] = \max_{a}\left[f(a)\,\max_{b}\ g(a,b)\right],$$
$$\max_{a,b}\ f(a)\, g(a,b) = \max_{a}\left[f(a)\ \max_{b}\ g(a,b)\right].$$

□

Soit la quantité $\delta_t(j)$, $j = 1, 2, \ldots, N$, Rabiner (1989), Miller (2011b), définie par

$$\delta_t(j) = \max_{y(t-1)}\ \mathbb{P}(y(t-1),\, y_t = S_j,\, O(t) \mid \underset{\sim}{\lambda}),\ avec t = 1, 2, \ldots, T. \tag{2.37}$$

$\delta_t(j)$ est la probabilité maximale, étant donné le MMC $\underset{\sim}{\lambda}$ de parcourir la séquence d'états $y(t)$ qui s'achève en S_j au temps t et d'y observer la séquence $O(t)$.

Supposons qu'on soit dans les états S_j et S_i aux instants respectifs t et $t-1$. Développons analytiquement les valeurs de $\delta_t(j)$, $t = 2, 3, \ldots, T$ afin d'en ressortir une formule récursive :

$$\begin{aligned}
\delta_t(j) &= \max_{y(t-1)} \mathbb{P}(y(t-1),\, y_t = S_j,\, O(t) \,|\, \underset{\sim}{\lambda}) \\
&= \max_{y(t-1)} \mathbb{P}(y_t = S_j | y_{t-1} = S_i, \underset{\sim}{\lambda})\, \mathbb{P}(O_t | y_t = S_j, \underset{\sim}{\lambda}) \\
&\qquad\qquad \mathbb{P}(y(t-2),\, y_{t-1} = S_i,\, O(t-1) \,|\, \underset{\sim}{\lambda}) \\
&= \max_{y_{t-1}} \left[\mathbb{P}(y_t = S_j | y_{t-1} = S_i, \underset{\sim}{\lambda})\, \mathbb{P}(O_t | y_t = S_j, \underset{\sim}{\lambda}) \right. \\
&\qquad\qquad \left. \max_{y(t-2)} \mathbb{P}(y(t-2),\, y_{t-1} = S_i\, O(t-1) \,|\, \underset{\sim}{\lambda}) \right] \\
&= \max_{S_i} \left[a_{ij}\, b_{S_j}(O_t)\, \delta_{t-1}(i) \right] \\
&= \max_{S_i} \left[a_{ij}\, \delta_{t-1}(i) \right] b_{S_j}(O_t). \qquad (2.38)
\end{aligned}$$

L'égalité (2.36) permet le passage de la deuxième à la troisième égalité. La transformation des probabilités est évidente à partir de la formule (2.16) pour des processus markoviens.

Déterminer l'argument du maximum $\underset{S_i}{argmax}\, \delta_t(j)$ revient donc à obtenir $\underset{S_i}{argmax}\, a_{ij}\, \delta_{t-1}(i)$. Cela nous permet de voir que la maximisation à l'instant initial $t = 1$ est alors triviale.

L'algorithme s'initialise par la valeur :

$$\begin{aligned}
\delta_1(i) &= \max_{S_i} \mathbb{P}(y_1 = S_i,\, O_1 | \underset{\sim}{\lambda}) \\
&= \max_{S_i} \left[\mathbb{P}(y_1 = S_i | \underset{\sim}{\lambda})\ \mathbb{P}(O_1 | y_1 = S_i,\, \underset{\sim}{\lambda}) \right] \\
&= \max_{S_i} \left[\mu_i\, b_{S_i}(O_1) \right] \\
&= \mu_i\, b_{S_i}(O_1). \qquad (2.39)
\end{aligned}$$

Nous pouvons considérer que $\underset{S_i}{argmax}\, \delta_1(i) = 0$ car aucune séquence d'observations et d'état, n'est observable à l'instant 0.

Si nous observons la probabilité maximale à l'échéance T, soit $\delta_T(y_T)$, on a :

$$\delta_T(y_T) = \max_{y(T-1)} \mathbb{P}(y(T), O(T)|\underset{\sim}{\lambda}).$$

La maximisation de cette probabilité par rapport à l'état final y_T donne une approche pour obtenir la valeur recherchée en (2.36),

$$\begin{aligned} \max_{y_T} \delta_T(y_T) &= \max_{y_T}[\max_{y(T-1)} \mathbb{P}(y(T), O(T)|\underset{\sim}{\lambda})] \\ &= \max_{y(T)} \mathbb{P}(y(T), O(T)|\underset{\sim}{\lambda}). \end{aligned} \tag{2.40}$$

Ainsi, en prenant l'argument du maximum des $\delta_t(j)$ pour tous les $t = 1, 2, \ldots, T$ et $j = 1, 2, \ldots, N$, on obtient la séquence optimale d'états . Dénotons par $\psi_t(S_j)$ cet argument à l'instant t.

On peut dès lors résumer la procédure complète de l'algorithme de Viterbi :

Algorithme de Viterbi

1. ***Pour*** $i = 1 : N$, ***faire***

   $\delta_1(i) = \mu_i b_{S_i}(O_1)$ *et* $\psi_1(i) = 0$;

2. ***Pour*** $t = 2 : T$, ***faire***

   Pour $j = 1 : N$, ***faire***

   $\delta_t(j) = \max_{S_i} [a_{ij}\, \delta_{t-1}(i)]\; b_{S_j}(O_t)$;

   $\psi_t(S_j) = \underset{S_i}{argmax}\; [a_{ij}\, \delta_{t-1}(i)]$;

   Fin Pour

   Fin Pour

3. $\mathbb{P}^* = \max_{S_i} \delta_T(i)$;

   $S^*_{i,T} = \underset{S_i}{argmax}\; \delta_T(i)$;

   Fin Pour

4. *Construction de la séquence d'états.*

   Pour $t = T - 1 : (-1) : 1$, ***faire***

   $S^*_{i,t} = \psi_{t+1}(S^*_{i,t+1})$;

   Fin Pour.

La probabilité $\mathbb{P}^*$ dans l'étape 3 correspond à celle de l'égalité (2.40). Cette étape permet d'obtenir à la fois, la probabilité maximale et la séquence d'états associée. Dans les sections 2.2.1 et 2.2.2, nous avons développé des procédures permettant d'obtenir des séquences d'observations et d'états les plus probables à observer à partir des paramètres $\underset{\sim}{\lambda}$ du MMC.
La prochaine section traite du troisième et dernier problème sur les MMC. Nous voulons cette fois extraire de l'information sur le MMC à partir des observations données.

2.2.3 Réestimation des paramètres du MMC afin de maximiser la vraisemblance de la séquence d'observations $O(T)$

Le but de ce problème est d'optimiser les paramètres du modèle $\underset{\sim}{\lambda}$ pour mieux expliquer comment une séquence donnée d'observations survient. Autrement dit, le but est de trouver le $\underset{\sim}{\lambda}$ qui maximise la probabilité $\mathbb{P}(O(T)|\,\underset{\sim}{\lambda})$.
De façon immédiate et mathématique, maximiser ou minimiser la probabilité $\mathbb{P}(O(T)|\,\underset{\sim}{\lambda})$ par rapport à $\underset{\sim}{\lambda}$ consiste à résoudre l'équation suivante :

$$\frac{\partial}{\partial \underset{\sim}{\lambda}}\mathbb{P}(O(T)|\,\underset{\sim}{\lambda}) = 0\,. \tag{2.41}$$

La résolution de ce type d'équation est difficile à obtenir, voir impossible en pratique.
Essentiellement, pour résoudre ce type problème nous devons faire appel à l'*algorithme de Baum-Welch* ou l'*algorithme EM* (*Expectation-Maximization*),voir Baum et Eagon (1967).
L'objectif de ce dernier étant d'optimiser la vraisemblance d'un modèle probabiliste, markovien dans notre cas, Rabiner (1989) et Moore (2005).

L'*algorithme de Baum-Welch* permet d'obtenir ainsi :

$$\underset{\sim}{\lambda}^* = \underset{\underset{\sim}{\lambda}}{argmax}\ \ \mathbb{P}(O(T)|\ \underset{\sim}{\lambda}). \tag{2.42}$$

Il réestime les différents paramètres du modèle, lesquels sont le vecteur de distribution initiale μ, la matrice de transition des états A et la matrice de distribution des observations selon les états observables B.

Pour calculer ces nouveaux paramètres, l'algorithme de Baum-Welch se sert de deux nouvelles matrices de de probabilités que nous dénotons par $\Theta = \{\theta_t(i,j)\}$ et $\Upsilon = \{\upsilon_t(i)\}$ où les coefficients sont définies par :

$$\theta_t(i,j) = \mathbb{P}(y_t = S_i,\ y_{t+1} = S_j|O(T),\ \underset{\sim}{\lambda}) \tag{2.43}$$

$$\gamma_t(i) = \mathbb{P}(y_t = S_i|O(T),\ \underset{\sim}{\lambda}). \tag{2.44}$$

Le coefficient $\theta_t(i,j)$ représente la probabilité selon le modèle de paramètres $\underset{\sim}{\lambda}$ et la séquence d'observations $O(T)$ de passer de l'état S_i au moment t à l'état S_j au moment $t+1$. Le coefficient $\gamma_t(i)$ quant à lui, est la probabilité selon le modèle de paramètres $\underset{\sim}{\lambda}$ et la séquence d'observations $O(T)$ d'être dans l'état S_i à l'instant t.

À l'aide des probabilités décrites par l'algorithme *Forward-Backward*, on peut très vite obtenir les valeurs des coefficients $\theta_t(i,j)$ et $\upsilon_t(i)$, $i = 1, 2, \ldots, N$ et $j = 1, 2, \ldots, N$.

Pour commencer on peut écrire :

$$\begin{aligned}\sum_{j=1}^{N} \theta_t(i,j) &= \sum_{j=1}^{N} \mathbb{P}(y_t = S_i,\ y_{t+1} = S_j|O(T),\ \underset{\sim}{\lambda}),\\ &= \mathbb{P}(y_t = S_i|O(T),\ \underset{\sim}{\lambda}),\\ \sum_{j=1}^{N} \theta_t(i,j) &= \gamma_t(i).\end{aligned} \tag{2.45}$$

On sait de (2.33) que :

$$\begin{aligned}\gamma_t(i) &= \frac{\mathbb{P}(y_t = S_i,\, O(T)|\, \underset{\sim}{\lambda})}{\mathbb{P}(O(T)|\, \underset{\sim}{\lambda})}, \\ &= \frac{\alpha_t(i)\ \beta_t(i)}{\mathbb{P}(O(T)|\, \underset{\sim}{\lambda})}.\end{aligned}$$

Ensuite, on peut réécrire le coefficient $\theta_t(i,j)$ comme une fonction des probabilités vues dans l'algorithme *Forward-Backward* :

$$\begin{aligned}\theta_t(i.j) &= \mathbb{P}(y_t = S_i,\, y_{t+1} = S_j | O(T),\, \underset{\sim}{\lambda}), \\ &= \frac{\mathbb{P}(y_t = S_i,\, y_{t+1} = S_j,\, O(T)|\, \underset{\sim}{\lambda})}{\mathbb{P}(O(T)|\underset{\sim}{\lambda})}, \\ &= \frac{\mathbb{P}(y_t = S_i,\, y_{t+1} = S_j,\, O(t), O_{t+1}, O_{t+2:T}|\, \underset{\sim}{\lambda})}{\mathbb{P}(O(T)|\underset{\sim}{\lambda})}.\end{aligned}$$

Par la définition des probabilités conditionnelles, on peut transformer le numérateur comme suit :

$$\begin{aligned}\mathbb{P}(y_t = S_i,\, y_{t+1} = S_j,\, O(T)|\, \underset{\sim}{\lambda}) &= \mathbb{P}(O_{t+1}|\, y_{t+1} = S_j,\, \underset{\sim}{\lambda})\, \mathbb{P}(y_t = S_i,\, y_{t+1} = S_j,\, O(t),\, O_{t+2:T}|\, \underset{\sim}{\lambda}), \\ &= b_{S_j}(O_{t+1})\, \mathbb{P}(O_{t+2:T}|\, y_{t+1} = S_j,\, \underset{\sim}{\lambda})\, \mathbb{P}(y_t = S_i,\, y_{t+1} = S_j,\, O(t)|\underset{\sim}{\lambda}), \\ &= b_{S_j}(O_{t+1})\, \beta_{t+1}(j)\, \mathbb{P}(y_{t+1} = S_j|\, y_t = S_i,\, \underset{\sim}{\lambda})\ \mathbb{P}(y_t = S_i,\, O(t)|\, \underset{\sim}{\lambda}), \\ &= b_{S_j}(O_{t+1})\ \beta_{t+1}(j)\ a_{ij}\ \alpha_t(i). \qquad (2.46)\end{aligned}$$

Remarque 2.2.3.

1. La $(t+1)$-ième observation dépend uniquement du $(t+1)$-ième état et des paramètres du modèles ;
2. La séquence partielle d'observations $O_{t+2:T}$ dépend uniquement de l'état caché à l'instant précédent, soit y_{t+1} et des paramètres du modèle ;
3. L'état caché y_{t+1} est indépendant des observations $O_1, O_2, \ldots, O_t$. Il ne dépend que de l'information contenue dans l'état actuel y_t et des paramètres du modèle.

On peut également transformer le dénominateur en se basant sur l'expression (2.46) par :

$$\begin{aligned}\mathbb{P}(O(T)|\underset{\sim}{\lambda}) &= \sum_{m=1}^{N}\sum_{n=1}^{N}\mathbb{P}(y_t = S_m,\, y_{t+1} = S_n,\, O(T)|\underset{\sim}{\lambda}),\\ &= \sum_{m=1}^{N}\sum_{n=1}^{N} b_{S_n}(O_{t+1})\ \beta_{t+1}(n)\ a_{mn}\ \alpha_t(m). \qquad (2.47)\end{aligned}$$

À partir des expressions (2.46) et (2.47), on réecrit les valeurs des coefficient $\theta_t(i,j)$ et $\gamma_t(i)$ par :

$$\theta_t(i,j) = \frac{b_{S_j}(O_{t+1})\ \beta_{t+1}(j)\ a_{ij}\ \alpha_t(i)}{\sum_{m=1}^{N}\sum_{n=1}^{N} b_{S_n}(O_{t+1})\,\beta_{t+1}(n)\,a_{mn}\,\alpha_t(m)}. \qquad (2.48)$$

$$\gamma_t(i) = \frac{\beta_t(i)\ \alpha_t(i)}{\sum_{m=1}^{N}\sum_{n=1}^{N} b_{S_n}(O_{t+1})\,\beta_{t+1}(n)\,a_{mn}\,\alpha_t(m)}. \qquad (2.49)$$

On peut interpréter la fréquence relative des transitions à travers l'état S_i en sommant le coefficient $\gamma_t(i)$ sur les instants $t = 1, 2, \ldots, T-1$: $\sum_{t=1}^{T-1}\gamma_t(i)$. De la même manière, on peut interpréter la fréquence relative des transitions de l'état S_i à l'état S_j par la sommation : $\sum_{t=1}^{T-1}\theta_t(i,j)$.

Ainsi, en allant dans le même sens d'interprétation, la réestimation des trois paramètres $A = \{a_{ij}\}$, $B = \{b_{S_i}(k)\}$ et $\mu = \{\mu_i\}$ du MMC donnant les nouveaux paramètres que l'on notera $\hat{\underset{\sim}{\lambda}} = (\hat{\mu}, \hat{A}, \hat{B})$ est telle que pour $i = 1, 2, \ldots, N$ et $j = 1, 2, \ldots, N$:

$$\hat{\mu}_i = \gamma_1(i), \qquad (2.50)$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1}\theta_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}, \qquad (2.51)$$

$$\hat{b}_{S_i}(k) = \frac{\sum_{t=1\cap O_t=k}^{T-1}\gamma_t(i)}{\sum_{t=1}^{T-1}\gamma_t(i)}. \qquad (2.52)$$

Une simple interprétation des trois nouveaux paramètres ci-dessus :

(i) L'estimation $\hat{\mu}_i$ (2.50) représente le fréquence espérée des transitions à travers l'état S_i au temps initial $t = 1$.

(ii) Le coefficient $\hat{a}_{ij}$ (2.51) est la proportion de la fréquence espérée des transitions de l'état S_i vers l'état S_j.

(iii) Le coefficient $\hat{b}_{S_i}(k)$ (2.52) représente la proportion du nombre espéré de transition à travers l'état S_i et d'observer le symbole o_k.

Voici un résumé des étapes de cette méthode :

Algorithme de Baum-Welch

1. ***Appliquer*** *les algorithmes Forward-Backward sur le MMC initial de paramètres arbitraire que l'on note* $\underset{\sim}{\lambda}(0)$, $z = 0$;
2. ***Faire*** $z = z + 1$;
3. ***Pour*** $t = 1 : T$, ***faire***
   Pour $i = 1 : N$, ***faire***
   Pour $j = 1 : N$, ***faire*** (*Calculer sous* $\underset{\sim}{\lambda}(z)$)
   $\theta_t(i, j)$;
   Fin Pour
   $\gamma_t(i)$;
   Fin Pour
   Fin Pour
4. ***Calculer*** *les fréquences espérées*
   $\sum_{t=1}^{T-1} \theta_t(i, j)$ *et* $\sum_{t=1}^{T-1} \gamma_t(i)$;
5. ***Réestimer*** *les paramètres du modèle*
   $\hat{\mu} = \{\hat{\mu}_i\}$, $\hat{A} = \{\hat{a}_{ij}\}$ *et* $\hat{B} = \{\hat{b}_{S_i}(k)\}$. *On pose* $\underset{\sim}{\lambda}(z+1) = (\hat{\mu}, \hat{A}, \hat{B})$;
6. ***Retour à l'étape*** 2, *tant qu'il y a augmentation de la probabilité* $\mathbb{P}(O(T)|\underset{\sim}{\lambda}(z))$ *ou tant qu'il y a encore des itérations à faire.*

Après réestimation des paramètres du modèle, l'algorithme de Baum-Welch commence par réévaluer la vraisemblance avec les nouveaux paramètres du modèle, soit $\hat{\underset{\sim}{\lambda}}$. Ensuite, l'algortithme recalcule les opérations de réestimation (2.50, 2.51, 2.52) avec les paramètres $\hat{\underset{\sim}{\lambda}}$ tant que la vraisemblance $\mathbb{P}(O(T)|\,\underset{\sim}{\lambda})$ n'est pas maximale, autrement dit, tant que la vraisemblance n'est pas très proche de 1.

Notons qu'il y a plusieurs réestimations des paramètres, mais que le modèle final ou adéquat est sélectionné selon le type de données à l'étude (étape 6 de l'algorithme). On peut également constater que l'algorithme de Baum-Welch ne réestime pas le nombre d'états cachés N, ce dernier doit donc être donné.

Pour conclure, à l'issu de ce chapitre, on est capable de définir un MMC et d'évaluer les probabilités des séquences d'observations avant-arrière (*Forward-Backward*) en temps discrets $\alpha_t(i)$ et $\beta_t(i)$. De même, on a pris connaissance de l'utilité de l'algorithme de Viterbi et de la schématisation de l'algorithme de Baum-Welch.

Ces méthodes sont mises en pratique au chapitre suivant. On met en application les algorithmes et les méthodes de calculs vues afin d'illustrer leur fonctionnement et d'observer des convergences possibles des paramètres estimés ou réestimés vers les modèles de référence.

CHAPITRE III

APPLICATION DES THÉORÈMES ET ALGORITHMES

Dans la section **3.1** de ce chapitre, nous revenons sur l'inférence dans les chaînes de Markov, vue au chapitre I. La matrice de transition à l'étude dans cette section est aussi utilisée à la section **3.2**.

À l'exception de cette matrice, il n'existe aucun lien entre les deux sections. Le choix des données ne s'inscrit pas dans un contexte particulier, il est totalement arbitraire.

3.1 Exemple sur l'inférence dans les chaînes de Markov

On considère la matrice de transition de probabilité (7×7) suivante :

$$\mathbf{P}^0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0.10 & 0.15 & 0.30 & 0.10 & 0.10 & 0.15 & 0.10 \\ 0.10 & 0.10 & 0.20 & 0.20 & 0.20 & 0.05 & 0.15 \\ 0.20 & 0.10 & 0.15 & 0.10 & 0.25 & 0.10 & 0.10 \\ 0.10 & 0.15 & 0.25 & 0.25 & 0.10 & 0.10 & 0.05 \\ 0.15 & 0.10 & 0.20 & 0.10 & 0.15 & 0.10 & 0.20 \\ 0.30 & 0.20 & 0.17 & 0.03 & 0.10 & 0.10 & 0.10 \\ 0.21 & 0.12 & 0.17 & 0.20 & 0.10 & 0.07 & 0.10 \end{pmatrix} \end{array} \tag{3.1}$$

Le but de cet exemple est de montrer que l'estimation par maximum de vraisem-

blance (EMV) $\hat{\mathbf{P}}^{\star} = \{\hat{p}^{\star}_{ij}\}_{i,j\in\mathcal{S}}$ de la matrice de transition d'une chaîne de Markov converge vers la matrice initiale de transition de probabilité $\mathbf{P^0} = \{p^0_{ij}\}$, $i, j \in \mathcal{S}$. On effectue une simulation à partir d'un grand nombre de transition entre les états ($n = 65100$).

Après avoir incrémenter $n = 65100$ fois à partir de l'état initial $X_0 = 1$, la fonction **SimulMarkov** en annexe B.1 nous permet d'obtenir l'estimateur par maximum de vraisemblance suivant :

$$\hat{\mathbf{P}}^{\star} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0.0965 & 0.1494 & 0.2966 & 0.1057 & 0.1006 & 0.1511 & 0.1001 \\ 0.1011 & 0.1016 & 0.2036 & 0.2007 & 0.1935 & 0.0463 & 0.1532 \\ 0.1983 & 0.0944 & 0.1460 & 0.0992 & 0.2587 & 0.1039 & 0.0995 \\ 0.0973 & 0.1445 & 0.2528 & 0.2477 & 0.1107 & 0.0990 & 0.0480 \\ 0.1505 & 0.1014 & 0.2007 & 0.1027 & 0.1486 & 0.0970 & 0.1991 \\ 0.2966 & 0.2030 & 0.1689 & 0.0300 & 0.0984 & 0.1000 & 0.1031 \\ 0.1985 & 0.1265 & 0.1732 & 0.2005 & 0.1001 & 0.0723 & 0.1289 \end{pmatrix} \end{array} \tag{3.2}$$

On peut voir qu'il y a effectivement une forte convergence de la matrice estimée $\hat{\mathbf{P}}^{\star}$ vers la matrice originelle $\mathbf{P}^0$. Nous avons également caculé la statistique de test d'ajustement pour la validation de la chaîne de Markov, on a obtenu $Z = 48.6289$.

Au niveau 5% la statistique de test $Z = 48.6289 < \chi^2_{5\%,42} = 58.1240$, on ne peut donc pas rejeter l'hypothèse nulle selon laquelle l'EMV de la matrice de transition $\hat{\mathbf{P}}^{\star}$ est égale à la matrice de transition initiale du modèle $\mathbf{P}^0$. La **Figure 3.1** illustre bien la convergence en loi de la matrice estimée. La convergence n'est pas parfaite mais assez bien représentative. Le degré de liberté de la Khi-deux ici vaut 42 car tous les p^0_{ij} de la matrice de transition initiale sont strictement positifs. Étant donné que l'espace d'états est de taille 7, alors le degré de liberté de la Khi-deux vaut $m(m-1) = 7 * 6 = 42$ (voir **Théorème 1.4.3**).

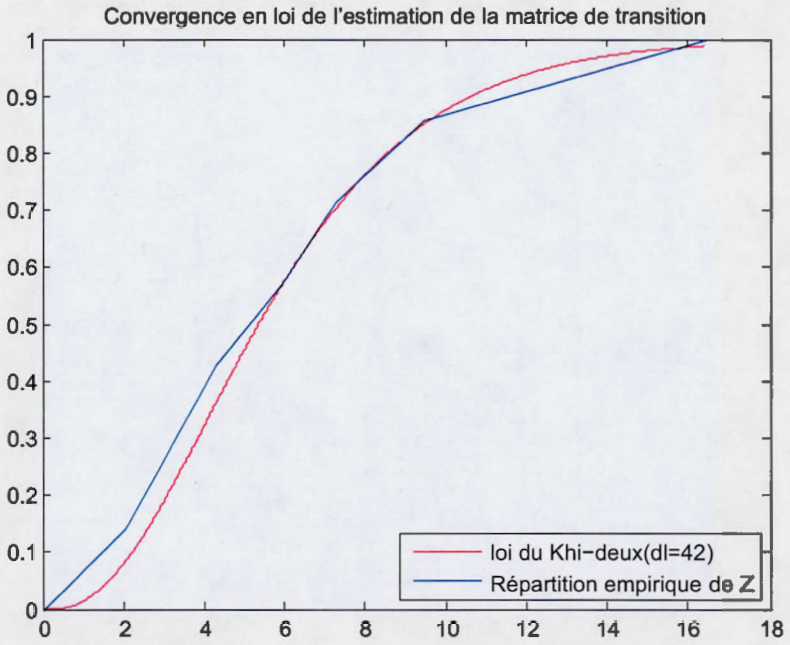

Figure 3.1: Convergence en loi de l'estimation de la matrice de transition

Ainsi s'achève l'exemple sur l'inférence dans les châines de Markov. À présent, nous allons voir des exemples sur les algorithmes.

3.2 Application des algorithmes

Maintenant notre objectif ici est de mettre en application les différents algorithmes vus au chapitre II via le langage MATLAB.

On considère la matrice de transition des états $A = \mathbf{P^0}$, oú $\mathbf{P^0}$ est la matrice donnée dans l'exemple précédent. On suppose que les états i ($i = 1, 2, \ldots, 7$) peuvent générer des symboles (des lettres) observables U, V, W, X et Y ($U = 1$, $V = 2$, $W = 3$, $X = 4$, $Y = 5$). On suppose aussi que l'information que nous avons indiqué une corrélation entre les lettres observables et les états i.

Enfin, on suppose la relation probabiliste entre les lettres et les états suivants :

$$\mathbf{B} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccc} U & V & W & X & Y \end{array} \\ \begin{pmatrix} 0.2 & 0.3 & 0.1 & 0.15 & 0.25 \\ 0.15 & 0.25 & 0.35 & 0.05 & 0.2 \\ 0.09 & 0.24 & 0.12 & 0.21 & 0.34 \\ 0.28 & 0.15 & 0.21 & 0.17 & 0.19 \\ 0.17 & 0.08 & 0.22 & 0.23 & 0.3 \\ 0.22 & 0.18 & 0.15 & 0.25 & 0.2 \\ 0.12 & 0.22 & 0.36 & 0.15 & 0.15 \end{pmatrix} \end{array} \tag{3.3}$$

Pour ce système, l'espace d'états est $\mathcal{S} = \{1, 2, 3.4, 5, 6.7\}$. La transition d'un état à un autre est un processus markovien d'ordre 1 car l'état suivant dépend uniquement de l'état courant et les probabilités (3.1) sont fixes. Toutefois les états du processus sont cachés car il n'est pas possible d'observer directement les états. Malgré que les états soient cachés, on peut observer les symboles U, V, W, X et Y. Grâce à (3.3), les lettres observables nous donnent de l'information probabiliste sur les états. Le système décrit ici est donc un MMC.

On considère la distribution initiale du modèle arbitrairement choisie :

$$\mu = [0.15,\ 0.20,\ 0.10,\ 0.15,\ 0.09,\ 0.20,\ 0.11]. \tag{3.4}$$

On considère pour la suite que le triplet $\underset{\sim}{\lambda} = \{\mu, A, B\}$ est connu. Afin de mieux comprendre et illustrer le comportement des MMC, on applique les trois algorithmes vus dans le chapitre II.

3.2.1 Application de l'algorithme *Forward-Backward*

On génère aléatoirement une séquence de $T = 10$ symboles via la fonction ***hmmgenerate(Taille voulue, A,B)*** de MATLAB. Nous avons obtenu :

$$O(10) = U\,Y\,Y\,W\,X\,U\,W\,Y\,X\,Y\ . \tag{3.5}$$

On évalue les probabilités *Forward* et *Backward* séparément par les algorithmes *Forward* et *Backward* basé sur la séquence (3.5). Ces dernières nous permettrons d'appliquer l'agorithme *Forward-Backward*. On obtient en exécutant les codes MATLAB en annexe B.1 et B.2, les probabilités *Forward*, *Backward* des **Tableaux 3.1** et **3.2**.

	$\alpha_t(1)$	$\alpha_t(2)$	$\alpha_t(3)$	$\alpha_t(4)$	$\alpha_t(5)$	$\alpha_t(6)$	$\alpha_t(7)$
t=1	0.0300	0.0300	0.0090	0.0420	0.0153	0.0440	0.0132
t=2	0.0076	0.0053	0.0135	0.0049	0.0070	0.0036	0.0030
t=3	0.0018	0.0011	0.0031	0.0011	0.0022	0.0009	0.0008
t=4	0.0002	0.0005	0.0003	0.0003	0.0004	0.0002	0.0005
t=5	0.0001	0.0000	0.0001	0.0001	0.0001	0.0000	0.0000
t=6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
t=7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
t=8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
t=9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
t=10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Tableau 3.1: Résultats algorithme *Forward*

	$\beta_t(1)$	$\beta_t(2)$	$\beta_t(3)$	$\beta_t(4)$	$\beta_t(5)$	$\beta_t(6)$	$\beta_t(7)$
t=1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
t=2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
t=3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
t=4	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
t=5	0.0003	0.0004	0.0004	0.0004	0.0003	0.0004	0.0004
t=6	0.0021	0.0023	0.0021	0.0021	0.0022	0.0021	0.0021
t=7	0.0108	0.0105	0.0107	0.0105	0.0103	0.0105	0.0101
t=8	0.0437	0.0432	0.0442	0.0425	0.0430	0.0390	0.0410
t=9	0.2510	0.2435	0.2500	0.2450	0.2395	0.2435	0.2358
t=10	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Tableau 3.2: Résultats algorithme *Backward*

	$\gamma_t(1)$	$\gamma_t(2)$	$\gamma_t(3)$	$\gamma_t(4)$	$\gamma_t(5)$	$\gamma_t(6)$	$\gamma_t(7)$
t=1	0.1682	0.1620	0.0499	0.2297	0.0814	0.2394	0.0694
t=2	0.1716	0.1178	0.3069	0.1089	0.1527	0.0792	0.0628
t=3	0.1597	0.1060	0.2743	0.0981	0.2097	0.0801	0.0721
t=4	0.0815	0.2151	0.1222	0.1304	0.1766	0.0666	0.2076
t=5	0.1290	0.0357	0.2475	0.1623	0.1872	0.1219	0.1163
t=6	0.1976	0.1205	0.1046	0.2088	0.1566	0.1291	0.0828
t=7	0.0776	0.2326	0.1321	0.1476	0.1427	0.0751	0.1924
t=8	0.1641	0.1014	0.2881	0.1235	0.1849	0.0645	0.0735
t=9	0.1402	0.0349	0.2493	0.1276	0.2069	0.1427	0.0985
t=10	0.1787	0.1059	0.2786	0.0954	0.1847	0.0834	0.0733

Tableau 3.3: Résultats algorithme *Forward-Backward*

L'algorithme *Forward-Backward* obtenu en combinant les résultats des **Tableaux 3.1** et **3.2** (voir annexe B.3), nous donne l'état le plus probable à chaque position dans la séquence de symboles. On peut par exemple dire que l'état 4 est le plus probable de générer le symbole initial U de la séquence $O(10)$ car il a la plus grande probabilité au temps 1 ($\gamma_1(4) = 0.2297$).
Cet algortihme est bien meilleur que de simples suppositions aléatoires pour donner de l'information sur les états et les symboles. Les résultats obtenus sont affichés dans le **Tableau 3.3**.

3.2.2 Application de l'algorithme de Viterbi

Dans cette partie, on veut déterminer la séquence d'états la plus probable de générer la séquence de symbole observale en (3.5). Pour ce fait, on exécute le code MATLAB en annexe B.5 de l'algorithme de Viterbi qui donne la probabilité maximale et la séquence d'états correspondante.
Nous avons obtenu les résultats suivant :

Séquence d'états $y(10)$ la plus probable	Probabilité maximale
4,3,5,7,4,4,4,3,5,3	8.0056e-13

Tableau 3.4: Résultats algorithme de Viterbi

Pour générer la séquence de symbole $O(10) = U\,Y\,Y\,W\,X\,U\,W\,Y\,X\,Y$, il est plus probable de partir initialement de l'état $y_1 = 4$ ensuite d'aller à l'état $y_2 = 3$, ainsi de suite, et de finir au temps $T = 10$ à l'état $y_{10} = 3$. La probabilité maximale peut sembler très petite, cela s'explique par la multitude de séquences d'états de taille 10 possible ($7^{10} = 282475249$).

3.2.3 Application de l'algorithme de Baum-Welch

L'application de l'algorithme de Baum-Welch est un peu plus complexe. Avant de poursuivre avec les données utilisées précédemment, nous appliquons d'abord cet algorithme sur des modèles de base à 2 et 3 états.

1. On considère le modèle initial à 2 états de matrices de transition et d'émission d'observations ($N = 2, M = 3$) :

$$\mathbf{A_2} = \begin{pmatrix} 0.84 & 0.16 \\ 0.22 & 0.78 \end{pmatrix}, \ \mathbf{B_2} = \begin{pmatrix} 0.17 & 0.49 & 0.34 \\ 0.5 & 0.09 & 0.41 \end{pmatrix}.$$

Notre but est de réestimer les paramètres de ce modèle à partir de l'algorithme *EM*. Pour ce fait, à l'aide de la fonction ***hmmgenerate***, on génère une séquence de symboles grâce au modèle initial. Ensuite on réestime les paramètres du MMC grâce à la fonction ***hmmtrain*** qui détermine les probabilités de transition d'un MMC à partir d'une séquence de symboles.
Pour mieux présenter les résultats de l'algorithme EM, nous avons réestimé les paramètres pour 100 et 10,000 observations simulées. On a obtenu les résultats suivants :

- pour 100 symboles simulés

$$\hat{\mathbf{A}}_\mathbf{2} = \begin{pmatrix} 0.8998 & 0.1002 \\ 0.1302 & 0.8698 \end{pmatrix}, \ \hat{\mathbf{B}}_\mathbf{2} = \begin{pmatrix} 0.1923 & 0.5601 & 0.2476 \\ 0.5868 & 0.0424 & 0.3708 \end{pmatrix}.$$

- pour 10,000 symboles simulés

$$\hat{\mathbf{A}}_\mathbf{2} = \begin{pmatrix} 0.8502 & 0.1498 \\ 0.2309 & 0.7691 \end{pmatrix}, \ \hat{\mathbf{B}}_\mathbf{2} = \begin{pmatrix} 0.1734 & 0.4819 & 0.3447 \\ 0.5115 & 0.0716 & 0.4169 \end{pmatrix}.$$

2. On applique similairement au point précédent l'algorithme EM sur un modèle

initale à 3 états de paramètres (N=3, M=3) :

$$\mathbf{A_3} = \begin{pmatrix} 0.12 & 0.54 & 0.34 \\ 0.76 & 0.18 & 0.06 \\ 0.41 & 0.33 & 0.26 \end{pmatrix}, \mathbf{B_3} = \begin{pmatrix} 0.19 & 0.45 & 0.36 \\ 0.29 & 0.20 & 0.51 \\ 0.91 & 0.06 & 0.03 \end{pmatrix}.$$

On a obtenu les résultats suivants :

- pour 100 symboles simulés

$$\hat{\mathbf{A}}_3 = \begin{pmatrix} 0.0000 & 0.0129 & 0.9871 \\ 0.5511 & 0.4489 & 0.0000 \\ 0.2249 & 0.5201 & 0.2549 \end{pmatrix}, \hat{\mathbf{B}}_3 = \begin{pmatrix} 0.0000 & 0.6483 & 0.3517 \\ 0.5314 & 0.0000 & 0.4686 \\ 0.7581 & 0.1395 & 0.1024 \end{pmatrix}.$$

- pour 10,000 symboles simulés

$$\hat{\mathbf{A}}_3 = \begin{pmatrix} 0.2407 & 0.4613 & 0.2980 \\ 0.7416 & 0.2174 & 0.0411 \\ 0.3022 & 0.2852 & 0.4125 \end{pmatrix}, \hat{\mathbf{B}}_3 = \begin{pmatrix} 0.0742 & 0.4905 & 0.4353 \\ 0.4045 & 0.1259 & 0.4695 \\ 0.8958 & 0.0708 & 0.0334 \end{pmatrix}.$$

Grâce à cet exemple, on peut tout d'abord remarquer que les résultats obtenus sont meilleurs pour 10,000 simulations, donc les réestimations convergent mieux lorsqu'il y a plus d'observations. Ensuite, en comparant les modèles à 2 et 3 états, on remarque une convergence plus précise de celui à 2 états ($\hat{\mathbf{A}}_2$ versus $\hat{\mathbf{A}}_3$ et $\hat{\mathbf{B}}_2$ versus $\hat{\mathbf{B}}_3$). Nous constatons que pour des modèles à plus de 3 états, les paramètres réestimés ne convergent que pour de grandes séquences d'observations. Pour mieux illustrer ceci, nous revenons au modèle à 7 états vu dans les points précédents.

En effet, nous avons essayé par l'éxécution du code en annexe B.6, de réestimer les paramètres du modèles à partir de la séquence de symboles en (3.5) mais il semblerait, comme dit plus haut, que pour des modèles à plus de 3 états la

convergence ne soit réalisable que pour de longue séquences de symboles. C'est assez logique, avec un modèle à 7 états et une séquence de 10 observations, estimer les probabilités de transition des états ou d'émission des observations n'est pas évident (il y a peu d'information).

Les paramètres que nous avons essayé d'estimer avec la séquence $O(10)$ ne convergeait pas vers les vraies valeurs. Par conséquent, pour pouvoir illustrer le bon fonctionnement de cet algorithme, nous avons simulé $T = 100,000$ symboles observables et effectuer 100 itérations pour l'estimation du modèle, A et B ont été pris pour matrices du modèle initial. On a obtenu les résultats suivants :

$$\hat{\mathbf{A}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0.0843 & 0.1424 & 0.2923 & 0.0902 & 0.1092 & 0.1715 & 0.1102 \\ 0.0862 & 0.0674 & 0.2231 & 0.2150 & 0.2405 & 0.0562 & 0.1115 \\ 0.1595 & 0.1057 & 0.1298 & 0.0952 & 0.2844 & 0.1103 & 0.1151 \\ 0.1122 & 0.1397 & 0.2527 & 0.2713 & 0.0847 & 0.0982 & 0.0411 \\ 0.1763 & 0.1431 & 0.1755 & 0.0950 & 0.1082 & 0.0846 & 0.2173 \\ 0.3394 & 0.2219 & 0.1670 & 0.0235 & 0.0732 & 0.0840 & 0.0911 \\ 0.2083 & 0.1010 & 0.1920 & 0.2119 & 0.1074 & 0.0735 & 0.1058 \end{pmatrix} \end{array} \tag{3.6}$$

$$\hat{\mathbf{B}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccc} U & V & W & X & Y \end{array} \\ \begin{pmatrix} 0.1875 & 0.3557 & 0.1020 & 0.1304 & 0.2243 \\ 0.1186 & 0.3016 & 0.3526 & 0.0401 & 0.1871 \\ 0.0740 & 0.2535 & 0.1129 & 0.2143 & 0.3452 \\ 0.3274 & 0.1078 & 0.1780 & 0.1799 & 0.2069 \\ 0.1491 & 0.0566 & 0.2234 & 0.2342 & 0.3366 \\ 0.2479 & 0.1152 & 0.1605 & 0.3008 & 0.1756 \\ 0.1333 & 0.2043 & 0.3991 & 0.1215 & 0.1419 \end{pmatrix} \end{array} \tag{3.7}$$

La réestimation du vecteur de loi initiale du modèle s'obtient, tel que vu, en prenant les valeurs $\gamma_1(i)$, $i \in \mathcal{S}$. On a également compilé à partir de la même séquence de symboles de taille $T = 100,000$, l'algorithme *Forward-Backward* (celui qui nous donne les $\gamma_t(i)$). On a obtenu :

$$\hat{\mu} = [\gamma_1(1)\ \gamma_1(2)\ \gamma_1(3)\ \gamma_1(4)\ \gamma_1(5)\ \gamma_1(6)\ \gamma_1(7)], \tag{3.8}$$

$$= [0.1388\ \ 0.1880\ \ 0.3360\ \ 0.0686\ \ 0.0402\ \ 0.1242\ \ 0.1041]. \tag{3.9}$$

La convergence des paramètres réestimés vers le modèle initial n'est pas parfaite, elle reste tout de même considérable. Les valeurs des nouveaux paramètres sont assez proche des valeurs recherchées.

Il est sûr qu'en simulant une plus grande quantité de symboles, ces paramètres convergeraient de façon plus évidente. On peut donc dire que la réestimation des paramètres du MMC par l'algorithme EM nécessite une grande quantité d'information surtout pour les modèles à plus de 3 états.

CONCLUSION

Au cours de ce mémoire, nous avons présenté la notion des MMC comme une extension possible des chaînes de Markov, notion capable d'exprimer des systèmes de dépendance plus complexes. On a expliqué que cette dépendance vient du fait que chaque état de la chaîne est lié à une loi de probabilité, laquelle permet d'émettre les symboles.

La compréhension des MMC est simplifiée par la manière dont les paramètres du modèle sont construits.

Les algorithmes permettant de faire de l'inférence sur les MMC ont été codés sur MATLAB à partir des formules récursives obtenues dans le développement du Chapitre 2. Ce qui renforce la flexibilité de la paramétrisation des MMC car l'indiciation mathématique est légère et la majorité des formules développées sont souvent réduites à de simples sommes. De cette manière le codage en MATLAB est beaucoup simplifié.

Nous avons utilisé l'algorithme *Forward-Backward* et l'algorithme de Viterbi afin d'évaluer et donner de l'information sur la différence entre les états du modèle à partir d'une séquence de symboles observés. Grâce aux résultats obtenus dans le chapitre 3, on a pu voir effectivement qu'à travers le temps, certains états sont plus probables que d'autres pour générer des symboles.

Finalement, nous avons également pu comprendre le fonctionnement de l'algorithme de Baum-Welch. Ce dernier, qui a été utilisé pour la réestimation et l'ajustement des paramètres du modèle à partir d'une séquence de symboles observée,

a permis de constater qu'il dépend fortement du nombre de données (symboles et états).

On a pu en effet, illustrer à partir d'une longue séquence de symboles (100,000 symboles et 7 états), qu'il y a une convergence des paramètres réestimés par cet algorithme.

ANNEXE A

MATÉRIAUX PRÉLIMINAIRES

Les définitions et les théorèmes présentés et montrés dans cet annexe sont issues des articles et livres suivants :

Dacunha-Castelle et Duflo (1993), Dantzer (2007), Hairer et Wanner (1996) et Lessard (2013).

A.1 Notions de probabilités

Commençons tout d'abord par donner les notions de mesure et espace de probabilité.

Définition A.1.1. *Soit* $(\Omega, \Im)$ *un espace mesurable. La fonction* $\mathbb{P} : \Im \to [0, 1]$ *est une mesure de probabilité si :*

(i) $\mathbb{P}(\Omega) = 1$;
(ii) $\forall F \in \Im, 0 \leqslant \mathbb{P}(F) \leqslant 1$, *où* F *est un évènement ;*
(iii) $\forall F_1, F_2, F_3, \ldots \in \Im$ *tels que* $F_i \cap F_j = \emptyset$ *si* $i \neq j$, $\mathbb{P}(\cup_{i\geq 1} F_i) = \sum_{i\geq 1} \mathbb{P}(F_i)$.

Définition A.1.2. *Un espace de probabilité est un triplet* $(\Omega, \Im, \mathbb{P})$ *où* $(\Omega, \Im)$ *est un espace mesurable et* $\mathbb{P}$ *une mesure de probabilité.*

Ensuite, nous introduisons le concept de variable aléatoire.

Définition A.1.3. *Soit* $(\Omega, \Im, \mathbb{P})$ *un espace de probabilité et* $(A, \mathcal{A})$ *un espace mesurable. On appelle variable aléatoire de* Ω *vers* A*, toute fonction mesurable* X *de* Ω *vers* A*.*

Maintenant, on introduit la notion de processus stochastique.

Définition A.1.4. *Soit* $(\Omega, \Im, \mathbb{P})$ *un espace de probabilité. Un processus stochastique est une famille* $\{X_t, t \in T\}$ *de variables aléatoires indexées par un ensemble d'indice* T*, définies dans le même espace de probabilités et à valeurs dans un même ensemble dénombrable* $\mathcal{S}$*.*

Exemple A.1.1. *On parle de processus stochastique en temps discret pour* $T = \mathbb{N}$ *et de processus stochastique en temps continue pour* $T = \mathbb{R}$*.*

On s'intéresse particulièrement au processus stochastique en temps discret.
Pour finir, on introduit aussi les notions de loi marginale et de loi conjointe.

Définition A.1.5. *Loi conjointe et marginale*

- *Soient* X *et* Y *deux variables aléatoires discrètes définies dans un ensemble dénombrale* $\mathcal{S}$ *donné. La loi de probabilité conjointe* $p(x, y)$ *est définie pour chaque paire de nombres* (x, y) *par :*

$$p(x, y) = \mathbb{P}(X = x, Y = y) \tag{A.1}$$

- *Les lois de probabilités marginales de* X *et de* Y *respectivement notées* $p_X(x)$ *et* $p_Y(y)$ *sont données par :*

$$p_X(x) = \mathbb{P}(X = x) = \sum_y p(x, y) \tag{A.2}$$

$$p_Y(y) = \mathbb{P}(Y = y) = \sum_x p(x, y) \tag{A.3}$$

Les sommes (1.2) *et* (1.3) *se font pour chaque valeurs possibles de* x *et de* y*.*

Remarque A.1.1. La définition **1.1.5** est valable si les conditions ci-dessous sont satisfaite :

- $p(x, y) > 0$;
- $\sum_x \sum_y p(x, y) = 1$.

A.2 Autres notions mathématiques

Théorème A.2.1. *Théorème d'Abel en analyse Hairer et Wanner (1996)*
Soit $f(x) = \sum_{n=0}^{\infty} a_n x^n$ *une série entière qui converge pour* $|x| < 1$. *Si la série* $\sum_{n=0}^{\infty} a_n$ *converge, alors*

$$\lim_{x \longrightarrow 1^-} f(x) = \sum_{n=0}^{\infty} a_n \,. \tag{A.4}$$

En d'autres mots, si la série converge en $x = 1$, *alors sa valeur en* $x = 1$ *est égale à sa limite* $\lim_{x \longrightarrow 1^-} f(x)$.

Démonstration. On utilisera la sommation par parties pour deux séquences $u_1, u_2, \ldots, u_N$ et $v_1, v_2, \ldots, v_N$ suivante :

$$\sum_{n=1}^{N} u_n(v_n - v_{n-1}) = (u_N v_N - u_1 v_0) - \sum_{n=0}^{N-1} v_n(u_{n+1} - u_n) \tag{A.5}$$

Selon l'énoncé du théorème, on suppose que $\sum_{n=0}^{\infty} a_n x^n$ converge pour $|x| < 1$ et pour $x = 1$.

Pour prouver $(A.4)$, on travaillera avec les sommes finies $\sum_{n=0}^{N} a_n x^n$ et $\sum_{n=0}^{N} a_n$.

Soit $s_n = a_0 + a_1 + \ldots + a_n$, pour $n \geq 0$. Il est facile de voir que $s_n - s_{n-1} = a_n$,

pour $n \geq 1$. On a alors :

$$\begin{aligned}
\sum_{n=0}^{N} a_n x^n &= a_0 + \sum_{n=1}^{N} (s_n - s_{n-1}) x^n, \\
&= a_0 + \sum_{n=1}^{N} (s_n - s_{n-1}) u_n && (où\ u_n = x^n), \\
&= a_0 + u_N s_N - u_1 s_0 - \sum_{n=1}^{N-1} s_n (u_{n+1} - u_n), && (par (A.5)) \\
&= a_0 + x^N s_N - x a_0 - \sum_{n=1}^{N-1} s_n (x^{n+1} - x^n), && (car s_0 = a_0) \\
&= a_0 (1 - x) + x^N s_N + \sum_{n=1}^{N-1} s_n x^n (1 - x), \\
&= x^N s_N + \sum_{n=0}^{N-1} s_n x^n (1 - x). && \text{(A.6)}
\end{aligned}$$

Par hypothèse le terme à gauche de l'égalité $(A.6)$ converge quand $N \longrightarrow \infty$. On a aussi que le terme $x^N s_N \longrightarrow 0$ quand $N \longrightarrow \infty$ car $\lim_{N \longrightarrow \infty} x^N = 0$ pour $-1 < x < 1$ et s_N est borné (s_N converge car la série $\sum_{n=0}^{\infty} a_n$ converge).

Soit $s = \sum_{k=0}^{\infty} a_k$.

Ainsi quand $N \longrightarrow \infty$ et $|x| < 1$, l'égalité $(A.4)$ devient :

$$\begin{aligned}
\sum_{n=0}^{\infty} a_n x^n &= \sum_{n=0}^{\infty} s_n x^n (1 - x), \\
\sum_{n=0}^{\infty} a_n x^n - s &= \sum_{n=0}^{\infty} s_n x^n (1 - x) - s, \\
\sum_{n=0}^{\infty} a_n x^n - s &= \sum_{n=0}^{\infty} s_n x^n (1 - x) - s \left((1 - x) \sum_{n=0}^{\infty} x^n \right), \\
\sum_{n=0}^{\infty} a_n x^n - s &= (1 - x) \sum_{n=0}^{\infty} (s_n - s) x^n.
\end{aligned}$$

Remarquons par les séries de Taylor que : $(1 - x) \sum_{n=0}^{\infty} x^n = 1$.

Le but revient à montrer que $\lim_{x \longrightarrow 1^-} (1 - x) \sum_{n=0}^{\infty} (s_n - s) x^n = 0$.

Par hypothèse s_n converge vers s quand $n \longrightarrow \infty$. On peut choisir une valeur $\varepsilon > 0$ telle que pour des grandes valeurs de n, $n > M$ on ait $|s_n - s| \leq \varepsilon$. On peut

partitionner le terme de droite :

$$\sum_{n=0}^{\infty} a_n x^n - s = (1-x)\sum_{n=0}^{M-1}(s_n - s)x^n + (1-x)\sum_{n=M}^{\infty}(s_n - s)x^n.$$

On applique l'inégalité triangulaire sur les sommes :

$$\begin{aligned}
|\sum_{n=0}^{\infty} a_n x^n - s| &\leq |1-x|\sum_{n=0}^{M-1}|s_n - s||x|^n + |1-x|\sum_{n=M}^{\infty}|s_n - s||x|^n, \\
&\leq |1-x|\sum_{n=0}^{M-1}|s_n - s||x|^n + |1-x|\sum_{n=M}^{\infty}\varepsilon|x|^n, \\
&= |1-x|\sum_{n=0}^{M-1}|s_n - s||x|^n + |1-x|\varepsilon\frac{|x|^M}{1-|x|}, \\
&< |1-x|\sum_{n=0}^{M-1}|s_n - s||x|^n + |1-x|\varepsilon\frac{1}{1-|x|}.
\end{aligned}$$

On sait que $|x| < 1$. On sait aussi que pour $0 < x < 1$, $|1-x| = 1-x$ et que $1-|x| = 1-x$. On peut alors obtenir la borne supérieure suivante :

$$|\sum_{n=0}^{\infty} a_n x^n - s| < |1-x|\sum_{n=0}^{M-1}|s_n - s| + \varepsilon. \tag{A.7}$$

Quand $x \longrightarrow 1^-$, le terme $|1-x|$ est proche de 0. Vu que la somme $\sum_{n=0}^{M-1}|s_n - s|$ ne dépend pas de x, elle ne change pas. C'est aussi le cas pour ε. Par conséquent, quand $x \longrightarrow 1^-$ on peut faire en sorte que $|1-x|\sum_{n=0}^{M-1}|s_n - s| \leq \varepsilon$. Alors, quand $x \longrightarrow 1^-$ on a :

$$|\sum_{n=0}^{\infty} a_n x^n - s| \leq \varepsilon + \varepsilon = 2\varepsilon.$$

Puisque ε est un nombre positif arbitraire, le terme $|\sum_{n=0}^{\infty} a_n x^n - s|$ doit partir de 0 quand $x \longrightarrow 1^-$. □

Nous définissons maintenant une fonction pour la méthode d'optimisation du Lagrangien

Définition A.2.1. *Méthode d'optimisation du Lagrangien*

Le Lagrangien du problème d'optimisation de f sous la contrainte $h(x) = 0$ est la fonction $Lg(x, \lambda)$ où $(x, \lambda) = (x_1, x_2, \ldots, x_n, \lambda) \in \mathbb{R}^{n+1}$ définie par :

$$Lg(x, \lambda) = f(x) - \lambda h(x). \tag{A.8}$$

Définition A.2.2. *Un point $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ est dit point stationnaire pour le problème d'optimisation de f (**Définition A.2.1**) s'il existe un paramètre $\lambda \in \mathbb{R}$ tel que :*

$$\frac{\partial}{\partial x_i} Lg(x, \lambda) = 0,\ i = 1.2, \ldots, n \qquad et \qquad \frac{\partial}{\partial \lambda} Lg(x, \lambda) = 0.$$

Le paramètre λ est appelé multiplicateur de Lagrange du point stationnaire x.

Le point (x, λ) est un point stationnaire pour la fonction Lg. Ainsi pour déterminer les extrémums de la fonction f sous la contrainte $h(x) = 0$, la première étape consiste à chercher ce point stationnaire.

Définition A.2.3. *Convergence en Loi ($\xrightarrow{\mathcal{L}}$)*

Soient $F_1, F_2, \ldots, F_n$ une suite de fonctions de répartition associée aux variables aléatoires réelles $X_1, X_2, \ldots, X_n$, et F la fonction de répartition associée à la variable aléatoire X.

La suite $\{X_n\}_{n \in \mathbb{N}}$ converge en loi vers X si :

$$\lim_{n \longrightarrow \infty} F_n(x) = F(x),\ \forall x \in \mathbb{R}.$$

On note cette convergence par $X_n \xrightarrow{\mathcal{L}} X$.

Définition A.2.4. *Convergence presque sûrement ($\xrightarrow{p.s.}$)*

Une suite $\{X_n,\ n \in \mathbb{N}\}$ converge presque sûrement vers X si la convergence est vraie avec probabilité 1,

$$\mathbb{P}(\omega \in \Omega | \lim_{n \longrightarrow \infty} X_n(\omega) = X(\omega)).$$

On note cette convergence par $X_n \xrightarrow{p.s.} X$.

A.3 Démonstration théorème chapitre 1

A.3.1 Preuve du **Théorème 1.2.2**

Démonstration. Lessard (2013)

Preuve de l'assertion (i).

Par hytpothèse $i, j \in \mathcal{S}$ tels que $i \longleftrightarrow j$, alors il existe $m, n \geq 0$ tels que $eta = p_{ij}^{(m)} p_{ji}^{(n)} > 0$, par définition de $i \longrightarrow j$ et $j \longrightarrow i$.

D'après les équations de Chapman-Kolmogorov, on a pour tout entier $k \geq 0$:

$$p_{ii}^{(m+k+n)} \geq p_{ij}^{(m)} p_{jj}^{(k)} p_{ji}^{(n)} = \eta p_{jj}^{(k)}.$$

En sommant sur tous les k, on obtient :

$$\sum_{k \geq 0} p_{jj}^{(k)} < \infty si \sum_{k \geq 0} p_{ii}^{(k)} < \infty.$$

D'après le **Corollaire 1.2.1**, on a j est transitoire si i l'est aussi. Et ineversement par symétrie, i est transitoire si j l'est aussi. Par conséquent i récurrent si et seulement si j est récurrent.

Preuve de l'assertion (ii).

Par l'assertion (i) on sait que si i est récurrent alors j l'est aussi et inversement. Si $p_{ii}^{(m+k+n)} \overset{k \uparrow \infty}{\longrightarrow} 0$, c'est à dire que i est récurrent nul, alors $p_{jj}^{(k)} \overset{k \uparrow \infty}{\longrightarrow} 0$ c'est à dire que j est récurrent nul, et inversement par symétrie. Par conséquent i est récurrent positif si et seulement si i est récurrent positif.

Preuve de l'assertion (iii).

Par les inégalités en (i), si $p_{jj}^{(k)} > 0$ alors $p_{ii}^{(m+k+n)} > 0$ c'est à dire que $m + k + n$ est un multiple de $d(i)$, la période de l'état i. Mais alors on a aussi

$$p_{jj}^{(2k)} \geq p_{jj}^{(k)} p_{jj}^{(k)} > 0 \ ,$$

d'où $m + 2k + n$ est un multiple de $d(i)$.

Donc $k = (m + 2k + n) - (m + k + n)$ est un multiple de $d(i)$. Par définition de la période de j, on a alors $d(j) \geq d(i)$. Inversement , on a $d(j) \leq d(i)$ par symétrie. On conclut donc que $d(j) = d(i)$. □

ANNEXE B

CODES MATLAB

Certaines parties de ce code sont inspirées du projet de recherche d'Alneberg (2011).

B.1 Estimation et convergence en loi de la matrice de transition

```
1  function X = SimulMarkov(n,P,X0)
2  %fonction donnant l'estimation d'une matrice de transition
3  % P est la matrice de transition de la chaîne de Markov
4  % n est le nombre d'observation à simuler
5  % X est le vecteur contenant n observations simulées d'états de
6  % la chaîne de Markov de matrice de transition P
7
8  fix=rng; %on utilise 'rng' pour fixer le vecteur aléatoire qui ...
       sera généré
9  s = length(P); % la taille de l'espace d'états de la chaîne
10 Pe=zeros([s,s]); % On initialise la matrice de transition à ...
       estimer Pe
11 rij=zeros([s,s]);
12 riP=zeros([s,s]);
13 Z=zeros(1,s);
14
```

```
15 X = zeros(n,1); % On initialise le vecteur d'observations simulées
16
17 X(1) = X0; % On fixe l'état initial de la chaîne à X0
18
19 Q = cumsum(P,2); % Q est la matrice de probabilités de transition
20 % cumulées (elle est obtenue de P)
21 for i=1:n-1
22     r = rand; %r = variable aléatoire de distribution uniforme ...
          [0,1]
23
24     for j=1:s
25         if (r < Q(X(i),j)) %Si r plus petit que la proba. cumulée
26             X(i+1) = j;    %on fixe le prochain état à simuler
27             break; % sortir de la boucle contenant le if
28         end
29     end
30
31 end
32 rng(fix);
33
34 for i=1:n-1
35     rij(X(i),X(i+1))= rij(X(i),X(i+1))+1; %On incrément les ...
          transitions
36 end
37
38 ri=sum(rij,2); %Somme sur les colonnes de {r_ij}, ce sont les r_i.
39
40 for i=1:s
41     Pe(i,:)=rij(i,:)/ri(i); % Calcul de l'EMV
42     riP(i,:)=ri(i)*P(i,:);
43 end
44
45
```

```
for j=1:s
    %V=((rij-riP).^2)./riP;
    Z(j)=sum(((rij(:,j)-riP(:,j)).^2)./riP(:,j));

end

Z=sort(Z);
disp(Z)

%affichage du graphique pour comparaison du khi-deux et de D_n


x=(0:0.01:max(Z)); %axe des abscisses
%f=zeros(1,length(x)); % on initialise le vecteur de khi deux
%degrelib=length(nonzeros(P))-length(nonzeros(P(:,1))); % ...
    degré de freedom
vdegrel=6*ones(1,length(x)); %vecteur de degré de freedom
f=chi2cdf(x,vdegrel); % distribution de Khi-deux


plot(x, f, 'b',[0 Z],[0 (1/s:1/s:1)],'r')
title('Convergence en loi de l''estimation de la matrice de ...
    transition',...
     'fontsize', 10)
legend('Répartition empirique de Z','loi du Khi-deux(dl=6)',4)


disp(P) % affichage de la matrice de transition P
disp(Pe) % affichage de la matrice estimée de transition Pe
end
```

B.2 Code pour l'Algorithme *Forward*

```
function [forward]=algoforward(O,A,B,mu)
%la fonction algoforward(O,A,B,mu) évalue les probabilités ...
    forward pour les
%séquence d'observation 'O' à partir du MMC de distribution ...
    initiale 'mu',
%de matrice de transition 'A' et de matrice de prob. ...
    d'observations B.

n=length(A(1,:));
T=length(O);
forward=zeros(T, n);

% Calcul les probabilités forward .

forward(1,:)=mu.*(B(:,O(1)))';
for t=2:T
    for j =1:n
        forward (t,j)=(forward(t-1,:)*(A(:,j)))*B(j,O(t));
    end
end
end
```

B.3 Code pour l'Algorithme *Backward*

```
function [backward]=algobackward(O,A,B)
%la fonction algobackward(O,A,B) évalue les probabilités ...
    forward pour les
```

```
%séquence d'observation 'O' à partir du MMC de matrice de ...
    transition 'A' et de %matrice de prob. d'observations B.

%initialisation
n=length(A(1,:));
T=length(O);
backward=ones(T, n);

% Calcul les probabilités backward .
    for t=(T-1):(-1):1
    x=B(:,O(t+1)).*backward(t+1,:)';
    backward(t,:)=A*x;
    end
end
```

B.4 Code pour l'Algorithme *Forward-Backward*

```

function [gamma]=algoforwback(O,A,B,mu)

%calul des probabilités forward et backward par leur fonction
forward=algoforward(O,A,B,mu);
backward=algobackward(O,A,B);

%calcul des différents vecteurs de probabilité
gamma=forward.*backward;

for t=1:length(O)
    gamma(t,:)=gamma(t,:)/(forward(t,:)*((backward(t,:))'));
end

```

```
15 end
```

B.5 Code pour l'Algorithme de Viterbi

```
1  function [proba,sequence]=algoviterbi(O,A,B,mu)
2  %La fonction algoviterbi trouve la séquence d'états la plus ...
       probable et %calcule la plus grande probabilité associée à ...
       la séquence 'O', pour le MMC %de matrice de transition A, ...
       de matrice de prob. des observations B et de %distribution ...
       initiale 'mu'
3
4
5  %initialisation
6  T=length(O);
7  N=length(A(1,:));
8
9  % construction des vecteurs Δ(i,t) et psi(i,t) où 'i' ...
       correspond à un
10 %état dans l'espace d'états et 't' correspond à un temps.
11
12 Δ=zeros(N,T);
13 psi=zeros(N,T);
14
15 %Initialisation de Δ
16 Δ(:,1)=B(:,O(1)).*mu';
17
18 %calcul des séquences Δ et psi
19 for t=2:T
20     for j=1:N
21         [H,M]=max(Δ(:,t-1)).*A(:,j));
22         Δ(j,t)=H*B(j,O(t));
```

```
23          psi(j,t)=M;
24      end
25  end
26  [H,M]=Max(A(:,T));
27  proba=H;
28  sequence=zeros(1,T);
29  sequence(T)=M;
30  for t=(T-1):(-1):1
31      sequence(t)=psi(sequence(t+1),t+1);
32  end
33  end
```

B.6 Code pour l'Algorithme de Baum-Welch. MathWorks (2014)

```
1
2  %la fonction hmmgenerate(Seq,Transition,Emission) est une
3  %fonction MATLAB qui peut génèrerer aléatoirement une
4  %séquence d'états ou de symboles observable de taille
5  %'Seq' à partir du modèle initiale de matrice de transition
6  %'Transition' et de matrice d'observations 'Emission'.
7
8  %la fonction hmmtrain(Obs,Transition,Emission) est une
9  %fonction MATLAB qui permet de réestimer les paramètres
10 %d'un MMC à partir de la séquence d'observations 'Obs',
11 %du modèle initiale de matrice de transition 'Transition'
12 % et de matrice d'observations 'Emission'.
13
14 A2=[0.84 0.16;
15      0.22 0.78];
16 B2=[0.17 0.49 0.34;
17      0.5 0.09 0.41];
```

```
mu2=[0.65 0.35];

%On simule 100 et 10000 observations.
h2=rng;
Obs100=hmmgenerate(100,A2,B2);
Obs10000=hmmgenerate(10000,A2,B2);
rng(h2);

On ré-estime les paramètres du modèles à 2 états.
[A2estim100,B2estim100]=hmmtrain(Obs100,A2,B2); %100 obs.
[A2estim10000,B2estim10000]=hmmtrain(Obs10000,A2,B2); %10000 obs.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Ré-estimation des paramètres du modèles.
%à 3 états et 3 observations.
A3=[0.12 0.54 0.34;
    0.76 0.18 0.06;
    0.41 0.33 0.26];

B3=[0.19 0.45 0.36;
    0.29 0.20 0.51;
    0.91 0.06 0.03];

%On simule 100 et 10000 observations.
h3=rng;
Obs3_100=hmmgenerate(100,A3,B3);
Obs3_10000=hmmgenerate(10000,A3,B3);
rng(h3);

%On ré-estime les paramètres du modèle à 3 états.
[A3estim100,B3estim100]=hmmtrain(Obs3_100,A3,B3); %100 obs.
```

```
50 [A3estim10000,B3estim10000]=hmmtrain(Obs3_10000,A3,B3); %10000 ...
       obs.
51
52 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
53
54 On simule 100000 obs.
55 h=rng;
56 Obs=hmmgenerate(100000,A,B);
57 rng(h);
58
59 %On ré-estime les paramètres du modèle à 7 états.
60 [Aestim,Bestim]=hmmtrain(Obs,A,B);
```

BIBLIOGRAPHIE

J. ALNEBERG : Movement of a prawn a hidden markov model approach, 2011. URL `http://uu.diva-portal.org/smash/get/diva2:429623/FULLTEXT01.pdf`.

T. W. ANDERSON et L. A. GOODMAN : *Statistical Inference about Markov Chains.* Institute of Mathematical Statistics, 1957.

J. K. BAKER : The dragon system-an overview. *IEEE Transactions on Acoustics Speech and Signal Processing*, 23:24–29, 1975.

L. E. BAUM et J. A. EAGON : An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73, Number 3:360–363, 1967.

R. BELLMAN : The theory of dynamic programming. *The RAND corporation*, 1954.

R. BHAR et S. HAMORI : *Hidden Markov Models : Applications to Financial Economics.* Springer, 2004.

P. BILLINGSLEY : Statistical methods in markov chains. *Institute of Mathematical Statistics - University of Chicago*, p. 14–16, 1960.

D. DACUNHA-CASTELLE et M. DUFLO : *Probabilités et Statistiques, Tome 2. Problèmes à temps mobile.* 1993.

J.-F. DANTZER : *Mathématiques pour l'agrégation interne. Analyse et probabilités.* 2007.

E. HAIRER et G. WANNER : *Analysis by Its History.* Springer-Verlag, New York, 1996.

R. A. HOWARD : *Dynamic Probabilistic Systems, Volume I : Markov Models.* 1971.

M. E. IRWIN : Markov chains, 2006. URL `http://www.markirwin.net/stat110/Lecture/MarkovChains.pdf`.

S. LESSARD : Cours de processus stochastiques, 2013. URL `http://www.dms.umontreal.ca/~lessards/ProcessusStochastiquesLessard2014.pdf`.

O. LÉVÊQUE : Markov chains, recurrence, transience, 2014. URL `http://lthiwww.epfl.ch/~leveque/Random_Walks/lecture_notes2.pdf`.

R. S. MAMON et R. J. ELLIOTT : *Hidden Markov Models in Finance.* Springer Science+Business Media, LLC, 2007.

MATHWORKS : hmmtrain, hidden markov model parameter estimates from emissions, 2014. URL `http://www.mathworks.fr/fr/help/stats/hmmtrain.html`.

J. W. MILLER : Forward backward algorithm for hmms, 2011a. URL `http://www.youtube.com/watch?v=7zDARfKVm7s`.

J. W. MILLER : Viterbi algorithm, 2011b. URL `http://www.youtube.com/watch?v=RwwfUICZLsA`.

A. W. MOORE : Hidden markov model, school of computer science, carnegie mellon university, 2005. URL `http://www.cs.cmu.edu/~awm/10701/slides/hmm14a.pdf`.

J. OMURA : On the viterbi decoding algorithm. *IEEE Transactions on Information Theory*, 15:177–179, 1969.

A. K. R. DURBIN, Sean R. Eddy et G. MITCHISON : *Biological Sequence Analysis : probabilistic models of proteins and nucleic acids.* Cambridge University Press, 1998.

L. R. RABINER : A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, p. 257–286, 1989.

L. R. RABINER et B.-H. JUANG : An introduction to hidden markov models. *IEEE ASSP Magazine*, p. 4–16, 1986.

L. R. RABINER et B.-H. JUANG : *Fundamentals of Speech Recognition.* Prentice Hall Press, 1993.

H. M. TAYLOR et S. KARLIN : *An Introduction to Stochastic Modeling.* Academic Press, 1998.

A. J. VITERBI : Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13: 260 – 269, 1967.

A. S. WEIGEND et S. SHI : Taking time seriously : Hidden markov experts applied to financial engineering. *Proceedings of the IEEE/IAFE*, p. 244–252, 1997.

P. WHITTLE : Some distribution and moment formula for the markov chain. *Journal of the Royal Statistical Society, Serie B. 17*, p. 235–242, 1955.