N° d'ordre : 25/2008-M/MT

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENNE

FACULTÉ DES MATHÉMATIQUES

MÈMOIRE Présenté pour l'obtention du diplôme de MAGISTER

En Mathématiques

Spécialité :Probabilité Statistique

Par

ZIREM Djamila

THÈME

APRENTISSAGE DANS LES CHAÎNES DE MARKOV CACHÈES

Soutenu, le 12 /05 /2008, devant le jury composé de :

Mr.	**A. AISSANI**	Professeur	U.S.T.H.B.	Président.
Mr.	**M. DJEDOUR**	Professeur	U.S.T.H.B.	Directeur de thèse.
Mr.	**DJ. CHAABANE**	Maître de Conférences	U.S.T.H.B.	Examinateur.
Mr.	**T. LARDJANE**	Chargée de Cours	U.S.T.H.B.	Examinateur.

Remerciements

Je tiens en premier lieu à exprimer mes plus vifs remerciements à Monsieur **M. *Djedour***, *mon* ***Directeur*** *de thèse pour l'intéressant sujet qu'il m'a proposé. Je lui suis également reconnaissant pour la confiance qu'il ma accordée. Il m'est impossible de lui exprimer toute ma gratitude en seulement quelques lignes.*

Je souhaite également adresser ma plus vive gratitude aux membres de mon jury.

Je tiens à remercie **A. AISSANI** qui me fait l'honneur de bien vouloir présider le jury de ma soutenance.

Je remercie également **M. T.LARDJANE** et **M. D.CHAARBANE,** pour les conseils

Pratiques et les remarques judicieuses qu'ils m'ont apportés et d'avoir accepté de faire partie du jury.

J'adresse mes remerciements à $\mathbf{M}^{elle}$ **Radja**, chargé de cours du département de l'électronique pour sa gentillesse et sont aide.

Je tiens aussi à remercier toutes mes amies, en particulier Karima et Kheira. Surtout un grand merci à Halim, Nordine qui m'ont aidé dans la programmation.

Enfin j'aimerais associer à ce moment particulier de ma vie mes parents, mes sœur et mes frères. Sans oublier mon mari pour son encouragement, et sa patience. Sans oublier ma fille Ines et mon fils Anas. Que mes parents et mon mari trouvent ici l'expression de toute ma gratitude.

Table des matières

Introduction

Le présent mémoire concèrne l'étude de l'apprentissage dans les Chaînes de Marov cachées, qui est caractérisé par deux processus, un est caché, il correspond à la séquence des états que suit le phénomène étudié et l'autre est observable, il correspond à la séquence de l'événement secondaire.

Le problème général est celui de l'estimation de la réalisation inobservable de la chaîne. La modélisation générale permet l'application des techniques bayesiennes et les résultats obtenus sont, en général, satisfaisants. En situations réelles, les paramètres définissant le modèle sont inconnus dans de nombreux cas. On est ainsi confronté, au problème de l'estimation de tous les paramètres du modèle à partir des données bruités.

Un certain nombre de techniques comme la méthode d'estimation Expectation Maximisation (EM), l'algorithme de Viterbi a été proposé par les auteurs.

Ici dans ce mémoire en s'intéresse essentiellement à une méthode d'estimation par Baum-Welch qui est un cas particulier de EM.

Pour illustrer notre travail nous présentant un exemple d'apprentissage par les CMC sur un problème générales auquel nous allons essayez d'apporter une solution. Notons que une fois un modèle est construit, dans le cas générales, il est facile de l'adapter au domaine dans lequel se pose le même problème. La question qui se pose dans ce cas est la suivante :

Étant donné une observation de l'événement secondaire comment retrouver l'état du phénomène qui a permit la réalisation de cet événement?

On constate sur le progrès réalisés ces dernières années dans tous les domaines sont dues en grand partie à l'utilisation d'approches statistique.

Parmi, celles-ci la modélisation baseés sur les Chaînes de Markov Cachées, qui sont des fonctions aléatoires de chaînes de Markov, qui ont été introduit par Baum, Petrie et Eagon

à la fin des années soixante[15] Il fallait attendre une quinzaine d'année pour trouver leur première application. Pendant la dernière décennie, les Chaînes de Markov cachées, sont devenus un outil très utilisé dans la modélisation des séquences de variables aléatoires. Ces modèle ont permis de représenter des séries temporelles dépendantes de facteurs non observables, mais restent encore mal connus dans certains domaines comme, le traitement d'images dynamiques et l'estimation des paramètres des système dynamiques non stationnaires [12]. Néanmoins, les Chaînes de Markov cachées connaissent actuellement un net regain d'intérêt dans leurs aspects théoriques que dans leurs aspects appliquées. Actuellement, ces modèles sont largement appliqués dans plusieurs domaines tels que : intelligence artificielle, reconnaissance de la parole et des forme [49], reconnaissance de l'écriture [6] traitement su signal, biologie moléculaire[39], [89] ; l'économie (le marché financier), [36] ; ou le diagnostique médical, robotique,... etc.

Une Chaîne de Markov cachée est un processus doublement stochastique dont une composante est une chaîne de Markov non observable (qui est la chaîne d'états) et l'autre composante est une séquence d'observations (selon une certaine loi probabiliste) produite par la chaîne des états. Ainsi, la Chaîne de Markov cachée est un modèle statistique qui peut être estimé à partir de la séquence observée.

Notre choix de l'application des Chaînes de Markov cachées porte sur la loi de Bernoulli de paramètre p. Plus précisément, nous considérons une série de lancers de pièces de monnaies "1", "2"et "3". Le résultat observé est une suite de piles et faces tandis qu'un autre résultat non observé est une suite de "1", "2" et de "3" qui donne l'ordre dans lequel sont lancées les deux pièces de monnaies. Dans notre cas, le modèle de Chaînes de Markov cachées est caractérisé par un modèle Markovien à espace d'états fini et par un ensembles de distributions de sortie. Les paramètres de transition dans la chaîne de Markov modélisent le passage d'un état à un autre dans le système tandis que les paramètres de sortie (les paramètres d'émission) modélisent l'émission d'une observation par un état du système.

Le modèle de Markov que nous étudions est limité par les caractéristiques suivantes :

(i) les observations sont supposées être conditionnements indépendantes par rapport aux états et (ii) les facteurs cachés obéissent à une loi markovienne d'ordre un. Notre travail est

organisé en quatre chapitres et une conclusion, suivis d'une bibliographie. Certains rappels de définition et les organigrammes des algorithmes utilisés sont donnés en annexe.

Le chapitre 1, comporte des notions générales sur l'apprentissage, et une synthèse sur les différents types de celui-ci, souvent l'homme est confrontée au phénomène d'apprentissage qui fourni un grand nombre d'outils aux industriels et aux entrepreneurs depuis bientôt un demi-siècle. Mais Qu'est-ce qu'un apprentissage ?

Apprendre est une transformation de l'espace d'entrée vers l'espace de sortie, l'entrée (observation) constitue les variable indépendante: les attributs, les symptômes et les sortie sont les objectifs qui représentes les classes ou variable indépendantes.

La théorie de l'apprentissage aborde les problème de régression, d'estimation de densité, mais elle se définit d'abord autour des problème de classification[23] de données, qui consiste à déterminer les paramètres du système adaptables, probabiliste ou déterministe, à partir d'ensemble d'exemples, dans le but de généraliser, c'est- à-dire classer correctement de nouvelle données.

Ces problèmes sont généralement abordés comme un problème de " minimisation " la perspective visée est le plus souvent " minimax ". L'objectif n'est pas d'estimer correctement les paramètres qui sous-tendent la production des données, mais plus d'ailleurs la minimisation de l'erreur de prédictions (mal classer une donnée, mal prédire la valeur d'observation).

Il n'existe pas de définition générale, universellement, ce concept touche à trop de notions distinctes, qui dépendant du point de vue que l'on adopte.

Dans le chapitre 2, nous rappelons quelques notions sur les processus aléatoires, les chaînes de Markov, les matrices de probabilités de transitions, le graphe de transitions ainsi que le traitement des propriétés des processus de Markov tel que l'érgodicité et les conditions de stationnarités. Les propriétés de Markov ainsi traitées sont utiles dans notre étude de la chaîne de Markov cachée. Par ailleurs, nous introduisons les Chaînes de Malakov cachées en : (i) donnant leurs définition, (ii) citant un historique de l'évolution du traitement des chaîne de Markov cachés ainsi que les différents domaines d'application actuels, (iii) schématisant des exemples qui illustrent les constituants d'une CMC, (iv) spécifiant une CMC, par la définition de ses paramètres et leurs notations et enfin (v) nous donnons une classification dans la structure des chaînes de Markov cachées.

Le chapitre 3, est une étude probabiliste de la résolution des problèmes d'évaluation et de recherche d'optimum pour une CMC. Dans cette étude nous présentons les différentes procédures utilisées dans l'étude des CMC tel que (i) la procédure forward et la procédure backward pour le calcul de la vraisemblance du modèle, (ii) l'algorithme de Viterbi pour la détermination du chemin optimal, ainsi que (iii) l'alternative de la méthode forward-backwar appelée l'algorithme Baum-Welch ou EM afin de calculer l'ensemble de paramètres optimal, c'est à dire trouver les valeurs de probabilités de transitions et d'émissions qui maximisent la vraisemblance du modèle.

Dans le chapitre 4, nous considérons notre modèle de Chaînes de Markov cachées à trois états (trois pièces de monnaies) où chaque état émet une observation selon une loi de Bernoulli de paramètre p, Nous mettons ensuite en application des algorithme du chapitre 3 sur une séquence donnée, à savoir l'apprentissage de Baum-Welch où nous étudions le comportement des paramètres estimés en fonction des valeurs initiales des modèles.

Nous terminons ce mémoire par une conclusion générale dans laquelle nous donnons une synthèse de l'application du chapitre 4, les problèmes pratiques de l'implémentation des programmes des algorithmes du chapitre 3 et les possibilités d'extensions de ce travail.

Chapitre 1

Généralités sur l'apprentissage

1.1 Introduction

Le mot « Apprentissage » désigne changement sur un système qui améliorent son fonctionnement dans une direction donnée. Donc, selon les types de changements et les différentes directions, on peut trouver plusieurs définitions proposées.

Définition 1.1.1 *Selon H.A. Simon[83] l'apprentissage désigne les changements, dans le système, qui sont adaptatifs dans le sens où ils permettent au système d'exécuter la même tâche du même échantillon, et ceci d'une manière plus efficace, la prochaine fois.*

On peut remarquer que cette définition n'est pas générale. Ainsi, il est possible de trouver des phénomènes d'apprentissage mais le critère d'amélioration n'y est pas facile à appliquer. Si on prend le contre exemple :" l'huile d'olives s'améliore avec le temps ", il est clair que cette amélioration ne peut jamais qualifier un phénomène d'apprentissage repproche aussi à la définition de Simon le fait que l'homme, notre référence en matière d'intelligence artificielle, n'apprend pas uniquement pour refaire la même tache plusieurs fois.

Définition 1.1.2 *Minsky a donner une définition plus générale : « l'apprentissage est la construction de changements utiles dans nos cerveaux».*

Mais l'apprentissage a des relations plus étroites avec une autre activité plus complexe parmi lesqueles on peut citer les phénomènes de la découverte, de la compréhension et de la résolution de problèmes. En effet, l'apprentissage est un cas particulier de résolution de problèmes, à partir d'un environnement de données, il s'agit pour le système d'en sélectionner certaines, de les structurer et de les intégrer à l'ensemble déjà mémorisé, afin de pouvoir les utiliser ultérieurement.

Ils s'intéressent aux processus par lesquels l'homme développe ses capacités au fur et à mesure s'il exerce une tache donnée. Les psychologues, à leur tour, définissent l'apprentissage comme « l'acquisition d'habilité ».

Ils s'intéressent aux processus par lesquels l'homme développe ses capacités au fur et à mesure s'il exerce une tache donnée.

Un autre point de vue, adopté par les chercheurs qui identifient l'apprentissage à « l'acquisition de connaissance explicite. ».

Cette acquisition doit être sélective, i.e elle doit simplifier et préciser, plus rapide et à moindre coût. Cette sélection n'est pas toujours évidente ainsi, l'événement le plus précis peut servir d'exemple général pour une catégorie d'événements [54].Il est aussi évident sue le système d'acquisition de connaissance ne va pas copier la mémoire de la machine il doit par contre la représenter sous forme adaptée au système expert. C'est pourquoi R.S.Michalski a caractérisèe l'apprentissage par :

« L'apprentissage est la construction ou la modification de la représentation de ce qui a été expérimenté ».

Cette représentation est qualifié par :

Sa validité : elle représente le degré d'exactitude avec lequel la représentation cadre avec la réalité.

Son efficacité : ce facteur caractérise l'utilité de la représentation et son effet sur le système d'exécution.

Son niveau d'abstraction : qui identifie le degré ou la puissance d'explication de la représentation.

La meilleure définition ne pourrait être donnée qu'au moment où l'on peut affirmer que l'on a bien compris ce processus.

1.2 Historique

L'apprentissage est une discipline relativement jeune. Sa naissance s'est manifestée par la première conférence internationale (The first Machin learning work-shop) qui a eu lieu en 1980 à carnegie Mellon. Ceci n'exclue cependant pas son existence depuis presque une cinquantaine d'années, vers les années 50 tout au début de l'intelligence artificielle. D'une façon plus formelle, les travaux dans ce domaine se distinguent par :

- La qualité de la connaissance initiale préconçue dans le système.
- La façon de présenter et modifier cette connaissance.

1.3 Nécessité de l'apprentissage

Le processus d'apprentissage consiste en une modification systématic en cas de répétition d'une situation stimulante, ou encore en dépendance d'un contact antérieur avec une situation donnée.

C'est un attribut déterminant de l'intelligence. C'est pourquoi, il fait l'objet d'étude dans plusieurs domaines s'intéressent à la compréhension de l'intelligence. Parmi ces domaines, on peut citer : les sciences cognitives, l'intelligence artificielle, la reconnaissance des forme, la psychologie,....

Plusieurs raisons pratiques et théoriques on motivé la recherche dans le domaine de l'apprentissage.

1.3.1 Au niveau pratique

Les systèmes existants (sans apprentissage), malgré leur succès, présentent des limites [23]. Ainsi par exemple dans une application opérationnelle de l'intelligence artificielle on recense les limites suivantes.

- La connaissance d'un système doit être entièrement programmée, et par conséquent, plusieurs hommes année sont nécessaires pour la construction d'un système dans le domaine riche en matière de connaissance.
- Toute erreur contenue dans la base de connaissance, et sans une autre récriture de cette base, ne peut être évitée ultérieurement.

- Le système ne peut pas améliorer sa connaissance en utilisant son expérience. Ceci présente un inconvénient pour de tels systèmes qui ont une durée de vie relativement longue.

Donc, il ne peut ni formuler de nouvelles abstractions, ni trouver des analogies aux solutions anciennes, ni découvrir des nouvelles. Ces système sont alors dits purement déductifs du moment où il ne peuvent que tirer des conclusions incorporées dans leurs bases de connaissance et éliminer les parties inutiles en fonction de ses besoins. Un besoin est encore senti dans le domaine de la vision. Ainsi, pour que la machine puisse voir un objet, il faut lui introduire plusieurs concepteurs géométriques, des descriptions physiques et des informations relationnelles de ces objets. Ce qui alourdie la tache au concepteur. Dans le domaine de la compréhension de la parole, pour pouvoir, par exemple, dialoguer en langue naturel avec l'homme, le système doit avoir toutes les propriétés syntaxiques du langage et des milliers de concepts et de structures représentant sa sémantique.

Il est donc évident qu'un système d'apprentissage facilite ces taches en automatisant la construction de tels systèmes.

1.3.2 Au niveau théorique

La compréhension de la nature du processus de l'apprentissage chez l'homme représente un outil voir un but très important. Ainsi, pour avoir des modèles bien solides de l'apprentissage, il faut bien comprendre le processus utilisé par l'homme. On pourra peut-être non pas simuler ce processus mais le sophistiquer. En, effet, l'utilisation de la machine va nous éviter la lenteur du processus chez l'homme ; tout en nous permettent d'en faire des copies, et accélérer le transfert de la connaissance.

Ainsi, la compréhension de l'apprentissage a un effet important sur les systèmes D'éducation. On pourra améliorer les méthodes D'enseignement et les systèmes d'enseignements interactifs assister par ordinateur. Dans de tels systèmes l'instructeur et l'élève doivent tous les deux apprendre.

L'apprentissage représente aussi un grand intérêt pour les machines à temps partagées. Dans de telles machines, où des programmes différents utilisent la mémoire d'une façon compliquée, une modification ou une correction n'est pas une tache facile. Ainsi,

L'apprentissage, par lequel l'homme modifie et corrige une mémoire extrêmement complexe, sera un outil important de déboguage pour ces machines.

Pour toute ces raisons, on voit ses dernières années l'importance de l'apprentissage croître (Y. Kodratoff) affirme que : « résoudre le problème de l'apprentissage et donner au système des connaissances lui permettant de découvrir de nouvelle connaissances sera certainement l'axe le plus important dans les prochaines années ». Et pour (Y. Kodratoff) : « l'initiative personnelle des ordinateurs ne peut que croître, et on arrivera à des machine ayant des personnalités propres ».

1.4 Objectif des systèmes d'apprentissage

Les objectifs fixés par l'ensemble des travaux d'apprentissage sont :

Analyse théorique et développement d'algorithmes généraux d'apprentissage :

Dans le but de simplifier le processus d'apprentissage, des algorithmes généraux sont développés. Ces algorithmes peuvent ne pas être similaires au processus utilisé par l'homme. Néanmoins, leurs structures de connaissances produites sont similaires.

Le développement de modèles artificiels des processus d'apprentissage

L'objectif visé par de telles recherches est le développement de théories et de modèles expérimentaux de l'apprentissage humain. Ces recherches présentent un grand intérêt pour améliorées les systèmes d'éducation existant et pour développer des techniques d'apprentissage.

Construction des systèmes d'apprentissage pour des domaines spécifiques

Ici, les recherches sont spécialisées. Des systèmes d'apprentissage particuliers et liés à des domaines spécifiques sont développés. Ayant un contact direct avec le monde réel, ces systèmes servent de guide pour les deux autres types de recherche. En, effet, une solution pour un problème donné peut être généralisée pour toute une classe de problèmes.

Ces trois directions de recherches dépendant l'une de l'autre, les travaux dans un domaine utilisent les résultats des autre approches.

Pour situer notre travail, on dira qu'il représente une méthode d'apprentissage, qui est un problème d'optimisation c'est à dire chercher la meilleur hypothése par les chaînes de Markov cachées (CMC).

Chapitre 2

Chaîne de Markov cachée

2.1 Introduction

Les Chaînes de Markov Cachées ont été proposées pour la première fois (en 1966) par Baum et Petrie dans le cas des observations à valeurs discrètes. Une quinzaine d'années plus tard, plusieurs applications ont été proposées dans différentes domaines, la première application étant la reconnaissance de la parole [49]. Nous notons que dans le traitement des Chaînes de Markov cachées, nous utilisons certaines propriétés des chaînes de Markov telles que l'homogénéité et la stationnarité.

En premier, il nous a paru indispensable de passer en revue l'historique des Chaînes de Markov cachées depuis leurs introduction jusqu'à leurs dernières applications. Ensuite, nous définissons brièvement les Chaînes de Markov et nous continuons à étudier leurs propriétés telles que le théorème ergodique et les conditions de stationnarité que nous utilisons ultérieurement dans les applications des CMC.

Par ailleurs, nous définissons les Chaînes de Markov cachées, et nous donnons leur spécification et leur structure générale. Nous exposons dans la même partie des exemples illustrateurs des Chaînes de Markov cachées pour mieux comprendre la distinction entre l'espace des états et celui des observations. Nous terminons ce chapitre par le classement des principaux types de CMC.

Dans ce qui suit, nous présentons les diverses étapes de l'évolution des modèles de Chaînes de Markov cachés et leurs différentes applications.

2.2 Historique et applications

En 1913, une première application des Chaînes de Markov a été développée pour le texte de " eugen onyegin [57]" À la fin des années 40 et le début des années 50, on continuait à appliquer les Chaînes de Markov pour traiter le langage, [82]. Dans les années 50, plusieurs travaux ont été entrepris concernant la programmation dynamique, leurs applications et le calcul du maximum de vraisemblance pour les données manquantes [38].Patric Billingsley a exposé, dans son article de 1960, l'aspect mathématique de l'inférence statistique appliquée aux Chaînes de Markov cachées et en 1961 Emanuel Parzen a discuté le problème de l'estimation de la fonction de densité de probabilités.

En 1966, Baum et Petrie introduisent les Chaînes de Markov cachées en les appelant fonctions probabilistes des Chaîne de Markov finies. À partir de cette année et jusqu'aux années 70, des recherches très actives ont abouties au développement d'algorithmes efficaces pour l'estimation des paramètres du modèle de CMC, ainsi que le décodage de la suite d'états cachés. Nous citons les travaux de Baum, Eagon Sell et Petrie,[18], [66] où ont été développés des algorithmes itératifs, basés sur le maximum de vraisemblance pour l'estimation des états cachés et l'estimation des paramètres du modèle.

En 1970, l'utilisation des CMC devient plus claire grâce à leur description par l'exemple des urnes de la part de Neuwrith qui a employé pour la première fois l'expression "Chaînes de Markov cachées (CMC)" au lieu de "(fonction probabilistes de chaînes de Markov). Des algorithmes efficaces de décodage utilisés en théorie de l'information ont servi à l'estimation de la suite d'états cachés. Parmi ces algorithmes, l'algorithme de Viterbi [87] et[35] a été largement exploité au décodage de la parole, possède une complexité de calcul linéaire avec la longueur de la suite d'observations à décoder et permet d'estimer des états du modèle correspondantes au meilleur chemin. D'autre part, l'algorithme EM est aussi largement exploité dans ce domaine [28],[41], [40].

À partir de 1975, de nombreuses applications ont été développées. Ces applications ne se bornent pas aux modèle Markoviens cachés de base, mais plusieurs notions,provenant des extensions théoriques de ces modèles et de leurs variantes, ont été introduites dans ce but d'améliorer les (DHMC : Discret Hidden Markov Chain, CHMC : Continuous Hidden Markov Chain, etc). Parmi les domaines d'application des CMC, nous citons la recon-

naissance automatique de la parole, le traitement du signal, la modélisation du langage, la robotique, les finances, la biologie et la médecine.

Reconnaissance automatique de la parole (RAP) :

Dès leur introduction en 1975, indépendamment par un groupe d 'IBM Et par Baker[11], les CMC ont pris une importance considérable au point où la quasi-totalité des systèmes de la RAP utilises cette modélisation. Toutefois, on continuait à utiliser des résultats des processus acoustiques pour la reconnaissance de la parole [9]. Les Chaînes de Markov cachées supposent que le phénomène modélisé est un processus aléatoire inobservable qui se manifeste par des émissions elles même aléatoires. Ces deux phénomènes donnent à l'approche markovienne une flexibilité pour la modélisation d'un phénomène complexe qui est la production de la parole. Depuis les années 80, des études approfondies pour évaluer et tester cette nouvelle technique ont été entreprises dans les grands laboratoires et dans plusieurs secteurs tels que : (i) les mots isolés[53], [74], (ii) la parole continue et discrète , et (iii) les vocabulaires[91].

Traitement du signal :

A u départ les signaux étaient traités par l'analyse spectrale, mais depuis l'introduction des CMC dans le domaine, les application n'ont cessées de s'élargir,[50], utilisant les CMCs pour le traitement des signaux et tout spécialement l'algorithme EM pour extraire des signaux markoviens dans l'espace d'états fini à temps discret. La même estimation a été appliquée par Fredkin et Rice (1992) pour l'identification des paramètres d'CMC à partir d'un canal

D'enregistrement, et ce pour une application dans le domaine de la biologie.

Modélisation des langages:

En 1996, Huang et Jack ont étudié la reconnaissance des mots isolés dans un langage continu et, en 1987, Katz a donné une description d'un modèle de langage de type "m-gram".

La théorie du décodage:

Omura (1969) a appliqué l'algorithme de Viterbi comme un décodeur optimum des régulateurs de contrôle par les technique de programmation dynamique. Par ailleurs, Bahl et All. (1975) ont utilisé une approche de décodage des codes linéaires par minimisation du ratio de l'erreur des symboles.

La robotique:

Dans l'étude des mouvements d'obstacles d'un robot mobile, Quilan [72] a développé une approche qui impose une stratégie de contrôle pour que le robot puisse bouger dans un environnement dynamique avec la vitesse prédéfinie et sous certaines contraintes. Les CMC sont utilisés pour prédire les sauts d'obstacles qui sont considérés comme des processus stochastiques. L'étude a permis de conclure que la modèle donne une description des mouvements aléatoires et que les CMC se veulent très cohérents avec les caractéristiques de ces mouvements et de leur nature stochastique.

La finance:

Cover (1984) s'est intéressé à la maximisation du revenu d'un capital investi et a utilisé les techniques de programmation dynamique pour atteindre son objectif. Ces dernières années, l'intérêt de l'application des CMC porte de plus en plus sur le domaine des finances. A titre d'exemple, Zucchini (1991) a considéré le modèle de commercialisation des actions sur un ou plusieurs sites et a utilisé les CMC pour identifier la présence ou l'absence des anticipations des actions sur des journées successives.

La biologie:

La première application des chaînes de Markov cachées dans le domaine de la biologie a été réalisée par Churchil en 1989. Pendant la dernière décennie, les CMCs sont devenus un outil largement utilisé pour la modélisation des séquences biologiques. Nous citons à titre d'exemples:

- La modélisation des familles des séquences biologiques[13] ;

- L'alignement multiple des séquences ;

- L'identification de l'activité électrique des structures moléculaires appelée canaux ioniques (ion channels) ;

- La détection des régions homogènes dans le génome.

Par ailleurs, ce domaine d'activité et de recherche a été soutenu par des chercheurs en informatique et des statisticiens mathématiciens pour la résolution des problèmes d'ordre biologique, [80] et Nous pouvons aussi trouver une riche information en matières sur le web, toutes les adresses correspondantes ayant été collationnées par Attwood et Parry- Smith (1999).

La médecine:

Dans son article sur l'analyse des données concernant les rechutes et les remises d'une maladie, Albert (1994) a montré que les modèles markoviens conviennent bien à l'état de santé des malades et fournissent une estimation efficace des caractéristiques importantes du déroulement de maladie. Par ailleurs, en présence d'une série de données concernant le nombre de naissances par césarienne, Zucchini et MacDonald (1997) ont montré que les CMC sont les meilleurs parmi toutes les méthodes autorégressives appliquées et donne une flexibilité considérable pour les séries temporelles ayant une tendance et une dépendance dans le temps.

Les CMC ne cessent d'être appliqués dans plusieurs autres domaines que le cadre de notre travail ne nous permet pas d'étaler. Nous citons l'étude des comportement des animaux et la climatologie , [58]

2.3 Le modèle de Markov

2.3.1 Processus aléatoire

Considérons les deux espaces probabilisables $(\Omega,\ A)$ et $(\mathbb{R}, \beta_{\mathbb{R}})$ et l'application $X(.)$ de l'espace d'épreuves Ω dans la droite des nombres réels $\mathbb{R}$:

$$\begin{aligned} X(.) \quad &: \quad \Omega \to \mathbb{R}, \\ \forall\omega \quad &\in \quad \Omega \rightsquigarrow X(\omega) \in \mathbb{R}. \end{aligned}$$

Définition 2.3.1 *l'application $X(.)$ est une variable aléatoire réelle si, l'image réciproque, $X^{-1}(B)$, par X de chaque borélien B, de la $\sigma - a\lg èbre$ de borele $\beta_{\mathbb{R}}$, est un événement aléatoire de la $\sigma - a\lg èbre$ A.*

$$\forall B \in \beta_{\mathbb{R}} \Longrightarrow X^{-1}(B) = \{\omega \in \Omega : X(\omega) \in B\} \in A \tag{2.3.1}$$

Remarque 2.3.1 *Du fait que la $\sigma - a\lg èbre$ de borele $\beta_{\mathbb{R}}$ est engendré par tous les intervalles semi-ouverts de la forme $]-\infty,\ x]$, alors une variable aléatoire peut être définie par la définition équivalente suivante :*

Définition 2.3.2 *l'application* $X(.)$ *de* Ω *dans* $\mathbb{R}$, *est une variable aléatoire réelle si pour tout* $x \in \mathbb{R}$, *le sous ensemble :*

$$\forall B \in \beta_{\mathbb{R}} \Longrightarrow A_x = X^{-1}(]-\infty, x]) = \{\omega \in \Omega : X(\omega) \leq x\} \tag{2.3.2}$$

est un événement aléatoire $A_x \in A$.

Une séquence de variables aléatoires indépendantes et identiquement distribuées est appelée processus stochastique, mais n'est pas intéressant en tant que modèle stochastique parce que les variables se comportent plus ou moins de la même manière. Pour introduire une plus grande variabilité, nous pouvons considérer le cas où ces variables dépendent du passé par des équations de récurrence.

Les chaînes de Markov à temps discret possèdent cette caractéristique. Dans une chaîne de Markov d'ordre 1, la dépendance (par rapport au temps) des états prend en considération le seul état précédent. Cependant, cette limitation de mémoire, suffit pour produire une large diversité dans le comportement des variables, chose pour laquelle les chaîne de Markov trouvent plusieurs applications dans divers domaines.

Une suite $(X_n)_{n\geq 0}$ de variables aléatoires dans l'ensemble E est appelée processus stochastique à temps discret dans l'espace des états E. Nous nous intéressons au cas ou l'ensemble E est fini $E = \{i_1,\ i_2, ...i_t, ...,\ i_n\}$.

Si

$$X_n\ (\omega)\ = i_t$$

le processus est dit être à l'état i_t au temps n .

2.3.2 Chaînes de Markov homogènes

Soit $(X_n)_{n\geq 0}$ une suite de variables aléatoires définie sur $(\Omega, @, P)$ à valeurs dans $(E, \mathcal{F})$ un processus stochastique à temps discret et à espace d'états E fini ou dénomrable.Alors $(X_n)_{n\geq 0}$ est une chaîne de Markov si pour tout entier pour tous les états $i_0, i_1, ...i_{n-1}, i, j$

$$P\left(X_{n+1} = j\ /\ X_n = i, , X_{n-1} = i_{n-1}, .., X_0 = i_0\right) = P\left(X_{n+1} = j\ /X_n = i\right) \tag{2.3.3}$$

Cette définition signifie que, pour le futur l'histoire du processus jusqu'à l'instant n est entièrement résumée par son état à l'instant n. La chaîne de Markov est homogène si la partie droite de l'équation est indépendante de n [Reinhard (1996)], cette propriété est dite de Markov.

La matrice $P = \{p_{ij}\}_{ij \in E}$ telle que

$$p_{ij} = P\left(X_{n+1} = j \ / X_n = i\right)$$

est la matrice de probabilité de transition de la chaîne de Markov homogène, et vérifie pour tous les états i et j :

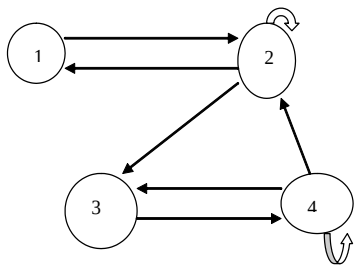
$$\begin{aligned} 0 &\leq p_{ij} \leq 1 \\ &\quad et \\ \sum_{j \in E} p_{ij} &= 1 \end{aligned} \tag{2.3.4}$$

La matrice P est appelée matrice sthocastique.

2.3.3 Graphe de transition

La matrice de transition dans une chaîne de Markov est représentée par un graphe de transition G ayant pour noeuds les états de E.Un arc (i, j) de ce graphe est orienté de l'état i à l'état j si et seulement si $p_{ij} \geq 0$. Les arcs et les noeuds correspondants à une matrice de transition d'une chaîne de Markov sont représentés dans la figure ci-dessous.

$$\begin{pmatrix} 0 & p_{12} & 0 & 0 \\ p_{21} & p_{22} & p_{23} & 0 \\ 0 & 0 & 0 & p_{34} \\ 0 & p_{42} & p_{43} & p_{44} \end{pmatrix}$$



Les transitions dans la chaîne de Markov

2.3.4 Etats communicants et récurrence

Un état j est dit accessible à partir de l'état i s'il existe un entier $m \geq 0$ tel que la probabilité d'atteindre l'état j à partir de l'état i après m transitions n'est pas nulle, c'est à dire $p_{ij}^m > 0$; en particulier un état i est toujours accessible à partir de lui même, du fait que $p_{ii}^0 = 1$. Deux états i et j sont dit communicants si i est accessible à partir de j et j est accessible à partir de i et on note $i \longleftrightarrow j$. On montre que la communication est une relation d'équivalence, et elle génère une partition de l'espace E des états en classe d'équivalence disjointes appelées classes de communication. S'il existe une seule classe de communication, alors la chaîne de Markov, sa matrice de transition et son graphe de transition sont dits irréductibles.

Dans une chaîne de Markov irréductible, l'unique partition de E est composée de d classes C_0, C_1, ..., C_{d-1} tel que pour tout k et $i \in C_k$

$$\sum_{j \in C_{k+1}} p_{ij} = 1$$

Si d est le nombre maximal de partitions, alors par convention on peut écrire $C_d = C_0$. $d \geq 1$ est appelé la période la chaîne de chaîne de Markov (et aussi de la matrice de transition

et du graphe de transition). Les classes C_0, C_1, ..., C_{d-1} sont dites classes cycliques. En effet, la chaîne peut se déplacer d'une classe à une autre d'une manière cyclique. Par définition la période d_i d'un état $i \in E$ est le plus grand commun diviseur du nombre de transitions n tel que:

$$\begin{aligned} n &\geq 1 \\ p_{ii}^n &> 0 \end{aligned}$$

Si $d = 1$, la chaîne est dite apériodique. Dans le cas où $d \neq 1$, la chaîne est apériodique et un état $i \in E$ peut être revisité par la chaîne après un nombre n de transitions. Le temps T_i mis pour revenir à cet état i est appelé temps de retour. Il est défini comme suit:

$$T_i = \inf \{n \geq 1; X_n = i\}$$

Nous remarquons que $T_i \geq 1$: en particulier si $X_n = i$, ceci n'implique pas que $T_i = 0$ et de plus si $X_n \neq i$, $\forall n \geq 1$ alors $T_i = \infty$.

Un état $i \in E$ est dit récurrent si $P_i(T_i = \infty) = 1$; sinon il est transient. Un état récurrent $i \in E$ est dit récurrent positif si $E_i[T_i] < \infty$.

2.3.5 Distribution stationnaire

Une distribution de probabilité π qui satisfait $\pi = \pi P$ où pour tous les états j et i de l'ensemble E, le vecteur π qui satisfait:

$$\pi(j) = \sum_{i \in E} \pi(i) p_{ij} \tag{2.3.5}$$

est la distribution stationnaire de la chaîne de Markov homogène. Par conséquent, une chaîne qui commence par une distribution stationnaire est stationnaire [24].

Théorème 2.3.1 ***Théorème 2.3.2 Théorème 2.3.3 Théorème 2.3.4*** *Une chaîne de Markov homogène irréductible à espace d'états fini est récurrente positive.*

Théorème 2.3.5 *Une chaîne de Markov irréductible homogène est récurrente positive si et seulement si il existe une distribution stationnaire. De plus, la distribution stationnaire π, quand elle existe , est unique et positive.*

Théorème 2.3.6 *(théorème érgodique)*

Soit $\{X_n\}\, n \geq 0$ *une chaîne de Markov irréductible récurrente positive avec une distribution stationnaire* π, *et soit* $f : E \rightarrow R$ *telle que*

$$\sum_{i \in E} |f(i)|\, \pi(i) < \infty$$

alors pour toute distribution initiale μ $: P_\mu$ *presque sûrement*

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=1}^{N} f(X_k) = \sum_{i \in E} f(i)\, \pi(i)$$

Ces résultats et leurs démonstrations peuvent être retrouvés dans Brémaud (1991).

On adopte les notations suivantes:

• la loi initiale :

$$P(X_1 = \omega_i\ ) = \pi_i \qquad , 1 \leq i \leq N \tag{2.3.6}$$

• la matrice de transition :

$$P(X_t = \omega_j\ / X_{t-1} = \omega_i) = a_{ij} \qquad , 1 \leq i \leq N \tag{2.3.7}$$

La loi de $(X_1, ..., X_t)$ est donnée par :

$$P\left(X_1 = \omega_{i_1}\ , ..., X_{t-1} = \omega_{i_{t-1}}, X_t = \omega_{i_t}\right) = \pi_{i_1} \cdot a_{i_1 i_2} ..... a_{i_{t-1} i_t} \tag{2.3.8}$$

La loi de probabilité commandant l'évolution du système est ainsi entièrement déterminée par la probabilité initiale et les matrices de transitions successives. Nous supposerons dans la suite que les chaînes considérées sont stationnaires, i.e. que la matrice de transition est indépendante du site considéré.

2.3.6 Spéfication des Chaîne de Markov caché

Les CMC sont caractérisés par les cinq paramètres suivants (nous reprenons les notations de (Levinson et Rabiner [53], [73]) :

• N, le nombre des états du modèle dont l'espace d'états est S est :

$$S = \{S_1,\ ...,\ S_N\}$$

on note l'état au temps t par S_t, $S_t \in S$;

• M, le nombre d'observations (symboles) distinctes, ces symboles correspondent à la sortie physique du système. Dans notre cas, l'observation du système est discrète. et on le note $X_t \in X$ tel que :

$$X = \{X_2,\ ...,\ X_M\}$$

où M est le cardinal de X .

• A, Une matrice des probabilités de transition entre les états de la chaîne:

$$A = \{a_{ij}\}, \qquad 1 \leq\ i, j \leq N$$

où

$$\begin{cases} a_{ij} = P\left(S_{t+1} = j\ / S_t = i\right) & , 1 \leq i, j \leq N \\ \qquad\qquad et & \\ \sum_{j=1}^{N} a_{ij} = 1 & 1 \leq i \leq N \end{cases} \tag{2.3.9}$$

• B, Matrice des distribution des probabilités des observations dans chaque état j :

$$B = \{b_j\left(X_t\right)\} \qquad 1 \leq j \leq N$$

avec

$$b_j\left(X_t = k\right) = p\left(X_t = k\ /\ S_t = j\right) \qquad 1 \leq j \leq N, 1 \leq k \leq M \tag{2.3.10}$$

$$et \tag{2.3.11}$$

$$\sum_{k=1}^{M} b_j\left(X_t = k\right) = 1 \quad , \quad 1 \leq j \leq N \tag{2.3.12}$$

• $\Pi = (\pi_i)$ où π_i représente la distribution des probabilités des états de départ (initiales) du modèle :

$$\Pi = \{\pi_i\}$$

où

$$\pi_i = p\left(S_1 = i\right)\ , \qquad 1 \leq i \leq N \tag{2.3.13}$$

$$et \tag{2.3.14}$$

$$\sum_{i=1}^{N} \pi_i = 1 \tag{2.3.15}$$

On peut conclure que la spécification complète d'une CMC requiert la spécification des paramètres :

• Deux paramètres (N et M pour un CMC discret) définissant les cardinaux des vecteurs d'observations, respectivement ;

• les distributions de probabilités A, B et Π.

On note par $\lambda = (N, M, A, B, \Pi)$ les paramètres définissant la Chaîne de Markov cachée, qui appartient à l'ensemble

$$\Gamma = \left\{ \begin{array}{c} \lambda \in R^d / \ \forall 1 \leq i, j \leq N, \ \forall 1 \leq k \leq M, \ 0 \leq \ \pi_i \leq 1, \ \sum_{i=1}^{N} \pi_i = 1, \\ 0 \leq \ a_{ij} \leq 1, \ \sum_{i=1}^{N} a_{ij} = 1, \ 0 \leq \ b_j(k) \leq 1, \ \sum_{i=1}^{M} b_j(k) = 1 \end{array} \right\}$$

Dans ce qui suit nous désignons la Chaîne de Markov caché par λ.

Nous supposons que la probabilité de transition des états $p_{ij} \geq 0$ pour tous les états i, $j = 1, 2, ..., N$.

Ainsi, la chaîne est irréductible

$$\forall \ i, j \in S, \ \exists \ m \ tel \ que \ p_{ij}^{m} \geq 0$$

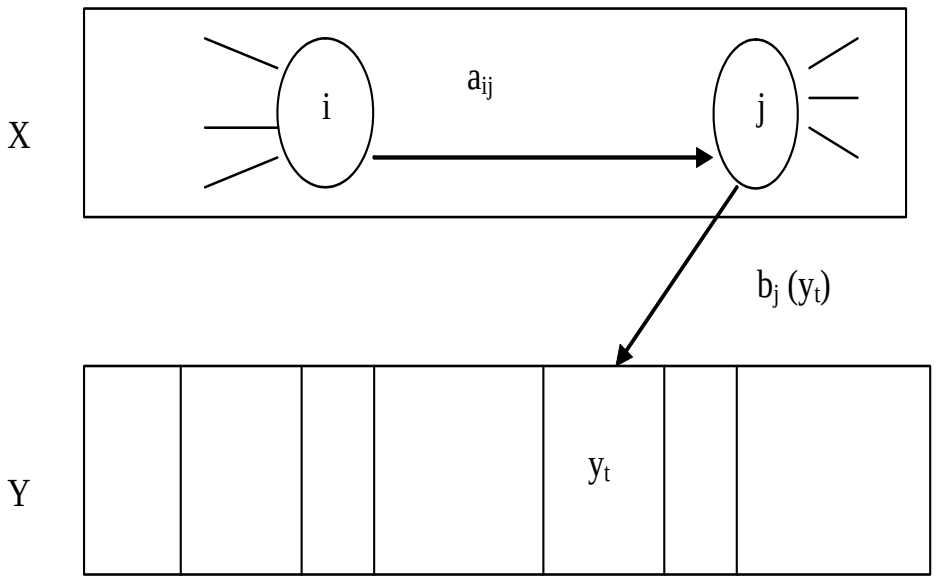
Elle est aussi apériodique. De plus l'espace S des états est fini ; alors la chaîne $\{S_t\}$ est érgodique. Par le théorème ergodique, la distribution initiale π est choisie comme l'unique distribution stationnaire positive.

$$\pi_i = P(S_t = i) > 0 \qquad 1 \leq i \leq N, \forall \ t = 1, \ ..., \ L$$

Ainsi l'ensemble de paramètres $\lambda = (N, M, A, B, \pi)$ est réduit à l'ensemble $\lambda = (A, B) = (a_{ij}, \ b_j(k))$.

2.3.7 Structure générale d'une CMC

Une CMC est caractérisée par deux processus : le premier non directement observable, le second externe et observable ; ce dernier fournit les informations pour
l'identification du premier. Les réalisations du premier sont représentées par la chaîne $S = S_1, \ S_2, \ ..., S_t, \ ..., \ S_L \ (1 \leq t \leq L)$ dont les éléments sont dans l'ensemble des états $E = \{1, \ 2, ..., N\}$; les réalisations du second sont la suite $X = X_1, \ X_2, \ ..., \ X_t, \ ..., \ X_L$ où chaque X_t est un élément de l'espace d'observation X [65]; donc $X_t \in X$.



Les éléments d'un modèle HMM

la figure ci-dessus montre les probabilités a_{ij} mises en jeu lors d'une transition de l'état i à l'état j ; une telle transition provoque l'émission à partir de l'état j d'un élément X_t de X avec une probabilité $b_j(X_t)$.

2.4 Exemples des CMC

2.4.1 Exemple 1 : le modèle d'urnes

Considèrons un modèle constitué de deux urnes (1 *et* 2) et de trois gobelets (G_0, G_2, G_3). Chacune des urnes contient son propre mélange de balles colorées blanches et noires et chacun des gobelets contient son propre mélange de pierres marquées état$_1$ et état$_2$.

Notre vecteur de paramètres est

$$\lambda = \left((a_{01}, a_{02}), \quad A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad B = \begin{pmatrix} b_1(B) & b_1(N) \\ b_2(B) & b_2(N) \end{pmatrix} \right) \tag{2.4.1}$$

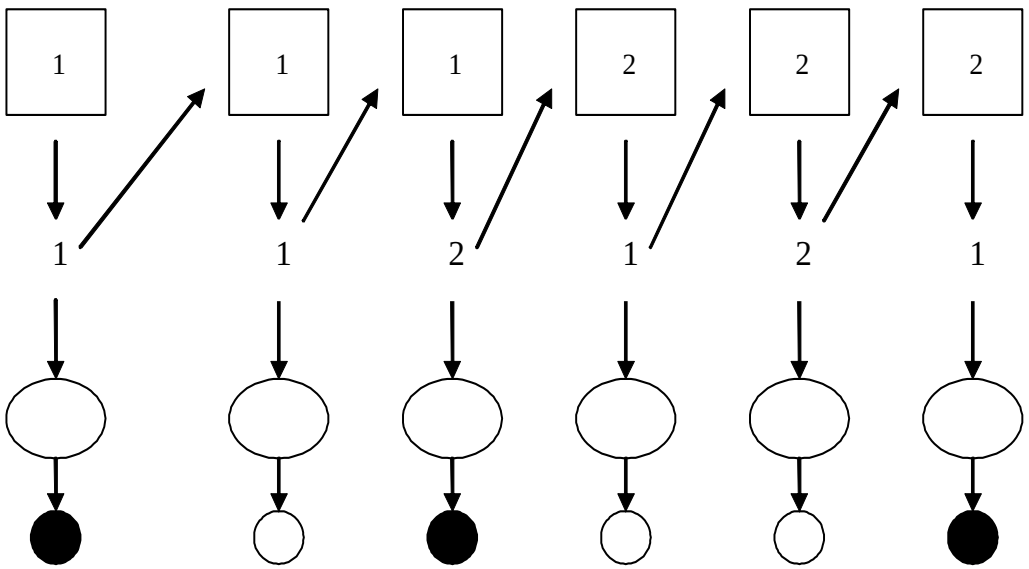
tel que :

a_{ij} : sont les fractions de pierres marquées état$_j$ dans le gobelet i

$b_i(N)$, $b_i(B)$ sont les fractions des balles noires et blanches (respectivement) dans l'urne i.

Nous générons une séquence d'observations de longueur L comme suit : nous tirons aléatoirement une pierre du gobelet G_0, sa marque est état_1, ensuite nous tirons alèatoirement une balle de l'urne 1, sa couleur est x_1 et tirons aléatoirement une pierre du gobelet G_1, la marque de la pierre du gobelet 1 est, par exemple, état_2 . Ce résultat nous permet de tirer une balle de l'urne 2 et une balle du gobelet G_2. Nous continuons le même procédé pour obtenir à la fois l'observation suivante (balle de l'urne) et l'état suivant (gobelet) à partir de l'état actuel jusqu'à la génération de L observations figure ci-dessous.

Ce modèle est appelé modèle de Ferguson et le voile de Ferguson est dû aux urnes qui cachent la succession des pierres.



Le Modèle de F erguson pour deux états et deux symboles

Nous notons la séquence d'observations par $X = (X_1,\ X_2,\ ...,\ X_L)$ et la séquence d'états cachés par $S = (S_1,\ S_2,\ ...,\ S_L)$. Le modèle en considération est un modèle d'ordre 1 car chaque état courant (gobelet) est lié à l'état prédécesseur. Nous remarquons que nous sommes en présence d'un modèle à deux état et deux symboles (couleurs).

Dans le cas général de N états et M couleurs ; l'ensemble des états (urnes) est $S = \{S_1,\ S_2, ...,\ S_N\}$. Chaque urne a son propre mélange de balles colorées (symboles). Chaque balle peut être colorée avec k couleurs possibles $(1 \leq k \leq M)$. Soit $b_i(k)$, la fraction (la probabilité) du symbole d'observation dans l'urne (état) $S_i, 1 \leq i \leq N$, où

$$\sum_{k=1}^{M} b_j(k) = 1 \qquad\qquad 1 \leq i \leq N$$

Soit $N+1$ gobelets : G_0, $G_1, ..., G_N$, chaque gobelet a son propre mélange de pierres portant des marques. La marque sur une pierre est considérée comme $état_1$, $état_2$, ... ou $état_N$, soit π_1, π_2, ..., π_N les fractions des pierres marquées $état_i$, $i = 1, 2, ..., N$, dans G_0 et soient a_{1i}, a_{2i}, ..., a_{Ni} les fractions des pierres marquées $état_i$ respectivement dans G_0, $G_1, ..., G_N$; avec :

$$\sum_{i=1}^{N} \pi_{0i} = 1 \qquad\qquad 1 \leq i \leq N$$

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad\qquad 1 \leq i \leq N$$

Générons une suite d'observations de couleurs de balles, soit $X = X_1, X_2, ..., X_L$ cette suite d'observations (figure ci-dessous). Tirons alèatoirement une pierre du gobelet G_0, sa marque est appelée état$_i$, $1 \leq i \leq N$ Ensuite tirons alèatoirement une balle de l'urne i, sa couleur est $x_1 = k,\ 1 \leq K \leq M$. Tirons encore alèatoirement une pierre du gobelet G_i, sa marque est $état_j,\ 1 \leq j \leq M$. Ainsi nous continuons le tirage en utilisant l'état courant pour obtenir à la fois l'observation courante (balle de l'urne) et l'état suivant (gobelet) jusqu'à un total L d'observations. À chaque tirage dans une urne, la balle est remise dans la même urne. Le voile de Ferguson cache cet unique échantillonnage des gobelets. L'observateur obtient seulement une information probabiliste concernant les pierres.

En générant la suite d'observations des couleurs X , une suite de pierres $S = S_1, S_2, ..., S_L$ est aussi générée mais elle est cachée. L'ensemble des paramètres du modèle stochastique est le vecteur $\lambda = (\Pi,\ A,\ B)$ tel que :

- le vecteur de probabilité $\Pi = (\pi_{01}, \pi_{02}, ..., \pi_{0N})$ est la distribution initiale des états;

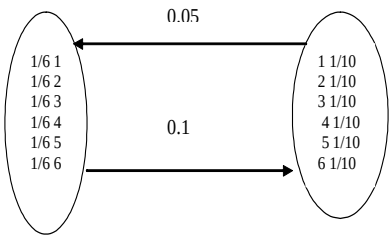
- la matrice stochastique est $A = (a_{ij})$, où la $i^{émé}$ rangée est associée au gobelet G_i, est la matrice de transition des états (urnes);

- pour chaque $état_i$, le vecteur $b_i = (b_i(1))\ b_i(2) ... b_i(M))'$ est appelé vecteur des probabilités de sortie pour l'état$_i$; ces probabilité tracent la suite d'états cachés S à partir de la suite d'observations X. La matrice $B = (b_1,\ b_2,\ ...,\ b_N)$ est appelée matrice des probabilité des observations.

2.4.2 Exemple 2 : le casino

Dans un casino malhonnête, on utilise une paire de dés, l'un équitable et l'autre biaisé.

Le dé biaisé a une probabilité de 0.5 d'obtenir un 6 et une probabilité de 0.1 d'obtenir l'un des autre numéros; on admet que le casino échange les dés : d'un équitable à un biaisé, avec une probabilité de 0.05 et d'un biaisé à un équitable avec une probabilité de 0.1. Le change entre les dés est un processus Markovien; le résultat de chaque état du processus a une probabilité différente et tout le processus est un exemple du modèle de Markov caché représenté graphiquement par :



Le modèle du casino

De la séquence observée, nous ne pouvons pas savoir à quel moment nous avons utilisé le dé équitable et à quel moment nous avons utilisé le dé biaisé. Le processus caché est aussi markovien à 2 états et d'ordre 1 pour la même raison que celle du premier exemple, l'observation étant de 6 alphabets (sorties). Ainsi, dans une chaîne de Markov cachée on

ne peut pas toujours connaître l'appartenance des observations ainsi que leurs probabilités d'où la nécessité d'estimer les probabilités du modèle.

2.5 Principaux types des CMCs

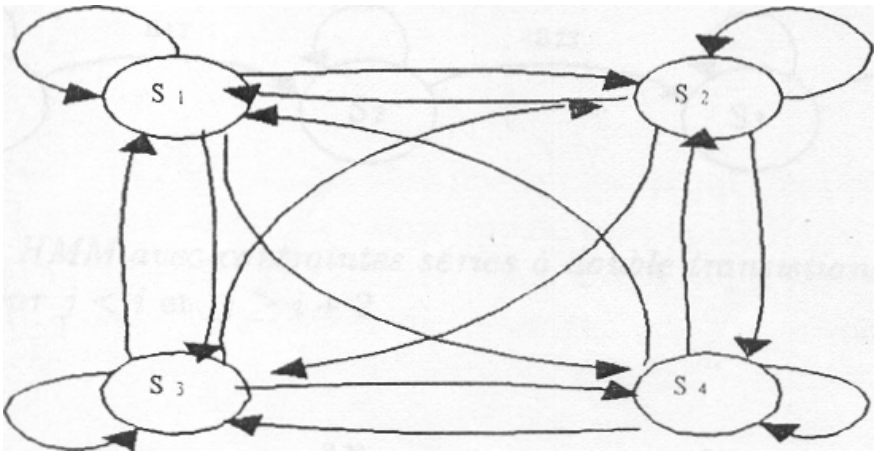
Nous exposons les différents types de CMC rencontrés dans le cas pratique et qui servent à donner une diversité dans le calcule et une richesse quand au traitement de ces modèles dans tous les domaines. Les CMC sont classés en différents types ou catégories sur la base :

1. Des contraintes sur les probabilités de transitions des états, (la matrice de transition A).
2. Des contraintes sur les probabilités d'émission des observations (le type de densité de probabilité d'observations $b_j\ (k)$).
3. Des contraintes sur la durée de séjour dans un état.
4. Des contraintes sur l'ordre de la chaîne de Markov.

Les contraintes sur les transitions

On peut obtenir deux types de modèles selon les contraintes imposées sur les éléments de la matrice A:

• **Modèle ergodique** C'est un modèle sans contrainte sur la matrice A, où toutes les transitions d'un état vers un autre sont possibles, c'est-à-dire que tous les états peuvent être atteints de n'importe où en un seul pas comme le montre la figure suivante . On peut remarquer que la connaissance de π est primordiale pour le choix de l'état de départ.



le modèle rgodique à 4 états

Le modèle gauche droite [11]:

C'est un modèle contenant des contraintes sur des transitions, c'est-à-dire il peut y avoir interdiction de certaines transitions, on cite par exemple l'interdiction du retour en arrière des transitions et on écrit mathématiquement :

$$a_{ij} = 0 \qquad si \quad j \leq i$$

Ils ont les propriétés suivantes [53]:

- La première observation est produite quand la chaîne est dans un état distingué appelé état début désigné généralement par S_1;

- La dernière observation est générée pendant que la chaîne est dans l'état fin ou absorbant désigné par S_N ;

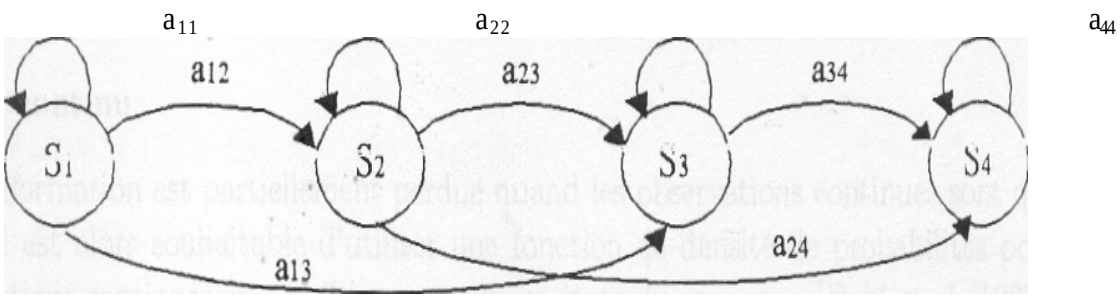
- Une fois que la chaîne de Markov quitte un état, cet état ne peut pas être visité ultérieurement.

D'Après les propriétés cités ci dessus, il existe deux sous-types de ce modèles qui sont le modèle parallèle et le modèle séquentiel comme le montre la (2.5) (2.5) . La première

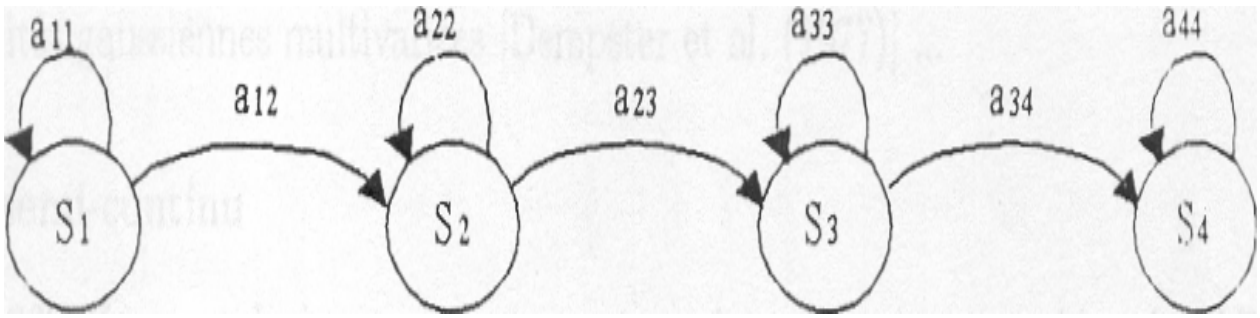
observation est produite pendant que la chaîne de Markov est dans un état initial S_1, avec

$$\begin{aligned} \pi_i &= 1, \quad i = 1; \\ \pi_i &= 0, \ 2 \leq \ i \leq N \ ; \\ a_{ij} &= 0, \quad j \leq i. \end{aligned}$$

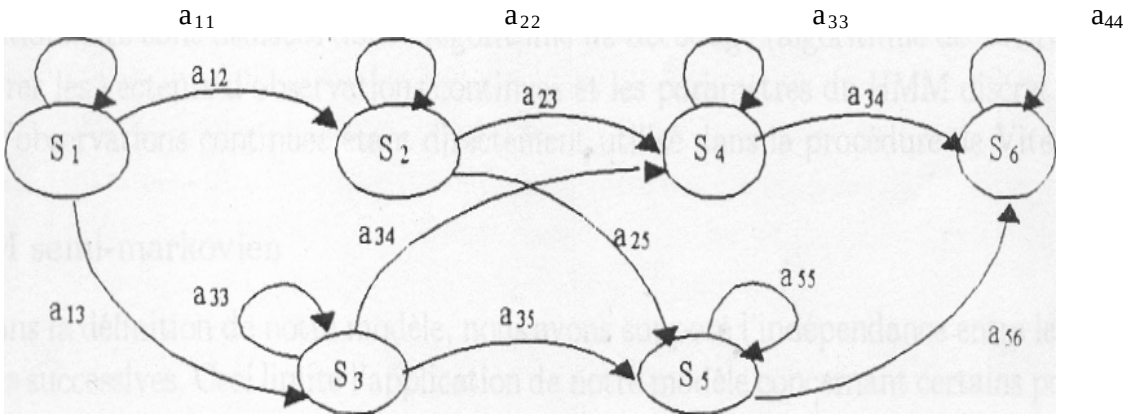
En général, nous choisissons $a_{ij} = 0$, si $j \geq i + \delta$ *avec* $\delta = 1, 2, ...$



HMM avec contraintes séries à une transition $a_{ij} = 0$, pour $j < i$ et $j \geq i + 3$



HMM avec contraintes séries à double transitions $a_{ij} = 0$, pour $j < i$ et $j \geq i + 2$



HMM avec contraintes paralléles

Contrairement au deux modèle précédents, qui transitent séquentiellement à travers les états, le modèle parallèle permet des trajectoires multiples à travers les états où chaque trajectoire peut sauter un ou plusieurs états. Ces modèles sont spécifique par le fait que dés que la chaîne de Markov quitte un état, cet état ne sera plus visité plus tard ; c'est à dire $a_{ij} = 0, \quad j \leq i$.

On peut remarquer que le modèle gauche droite couvre souvent la plupart des applications et de ce fait, il est largement suffisant, par exemple, en parole, le phénomène est continu et il n'y as pas lieu de revenir en arrière, on peut observer la même chose pour les phrases écrites où les caractères se suivent dans les mots tout en se répétant.

Contraintes sur les le type de densité de probabilité d'observations

Selon le type de densité de probabilité d'observations, discrète, continu, semi continu,... nous pouvons construire plusieurs types de modèles de CMC : soit une CMC discrete soit une CMC continue, CMC semi continue, CMC semi markovienne...

CMC discret "discret Hidden Markov Chain" (DHMC) Les observation en général sont continues puisqu'elle proviennent de phénomènes physiques continus. Dans le cas d'une CMC discrète, les observations continues sont quantifiées à l'aide d'un dictionnaire [58], [12].

CMC continue "Continious Hidden Markov Chain" (CHMC) Bien qu'il soit possible de quantifier les observation continues, il y a une sérieuse dégradation d'information associée à cette quantification [42]. Il sera, alors avantageux de choisir une fonction de densité de probabilité d'observations continues, conditionnée par les états du processus.

Cependant, pour estimer, d'une façon consistante, les paramètres de cette fonction, certaine restrictions doivent être précisées sur la forme de son modèle.

Il existe plusieurs formes pour lesquelles des procédures de ré estimation ont été formulées, à titre d'exemple:

- Fonction de densité de probabilité possédant une forme "strict log-concave", elliptique symétrique,à titre d'exemple:

* Fonction de densité de probabilité Gaussienne Scalaire [17].

* Fonction de densité de probabilité Gaussienne Multi variables .

- Mélange fini de densité de probabilité Gaussienne Multi variables [42].

- Fonctions Gaussiennes Autorègressives. C'est un cas particulièrement approprié aux systèmes dynamiques et notamment aux signaux évoluant dans le temps. Il existe trois types de ce modèles:

* CMC Autorègressif de mélange d'une seule fonction Gaussiennes par état .

* CMC Autoregressif de mélange de M fonctions Gaussiennes par état.

* CMC de mélange fini de densité de probabilité Gaussienne Autorégressive avec décodage par la technique de la quantification vectorielle.

Le tableau ci-dessous montre une comparaison, entre plusieurs critères, entre une CMC discrète et une CMC continue

	A. CMC continu	B. CMC discrete
Nombre de Paramètres à estimer	Un nombre élevé de paramètres	Mois de Paramètre que A
Précision de La classification	Précis	Mois précis que A
Hypothèse sur la Nature des observations	Importantes	Moins importantes que A
Implémentation	Difficile et lent	Plus facile est plus rapide que A
Nombre de Corpus d'apprentissage	Moyen	Plus élevé que A

CMC Semi Continue (Semi Continus Hidden Markov Chain (SCHMC)) En 1988, Huang et Jack ont proposé une approche qui consiste à combiner la CMC discrete et la CMC continue ; le modèle ainsi obtenu est appelé CMC semi continue. Son principe consiste à remplacer les probabilités des symboles des observations discrètes par une combinaison des probabilités des symboles discrets et des densités de probabilités continues dérivées à partir du dictionnaire de la quantification vectorielle. Les fonctions de densités de probabilités continues du dictionnaire sont utilisées dans l'algorithme de décodage (algorithme de Viterbi) pour pondérer les vecteurs d'observations continues et les paramètres de la CMC discrète ; le vecteur d'observations continues étant directement utilisé dans la procédure de Viterbi. C'est à dire dans le cas d'une CMC discrète, la perte de l'information liée à la quantification vectorielle peut être réduite si, durant le décodage (algorithme de viterbi), le vecteur de mesure (observation continue) est utilisé par la CMC discrète au lieu du symbole d'observation discret correspondant.

Corrélation du temps explicite " Semi Markov Hidden Chain (SMHMc) Jusqu'à présent, nous avons supposé l'indépendance entre les observations successives dans une suite d'observations. Cette hypothèse est assez limitée pour être appliquée en définissant une nouvelle probabilité d'émission d'observation qui prend en compte

la corrélation entre les vecteurs représentants les observations continue successives a été proposée . Par conséquent, les formules de ré estimation de Baum Welch et l'algorithme de Viterbi ont été modifiés.

La durée de séjour dans un état

L'un des inconvénients des CMC de base est le manque d'informations concernant la variabilité de la durée de séjour dans un état en favorisant les durées courtes.

Le problème de la variabilité de séjour dans un état est d'importance majeur dans certains processus physiques .Exemple une CMC permet de segmenter un mot prononcé en états (en utilisant l'algorithme de viterbi). Nous pouvons, alors, utiliser les durées mesurées pour qu'un état ω_i reste pendant une durée d avant la transition vers un autre état j avec une probabilité a_{ij}. La densité de la durée d associée à l'état i, $p_i(d)$ est,

$$p(X/\lambda, S_1 = i) = (a_{ij})^{d-1}(1 - a_{ii}) = p_i(d)$$

Cette équation possède une propriété géométrique puisque :

$$p_i(d+1) = a_{ii}p_i(d)$$

nous constatons, alors que la durée la plus probable est celle qui est plus courte.

L'ordre de la chaîne de Markov

Une limitation des modèles de Markov de base est de supposer que le processus markovien est d'ordre un ce qui n'est pas le cas dans de nombreuses applications.

Le travail de Kerdem (1976) a montré l'avantage et l'efficacité de la CMC du deuxième ordre dans le traitement des séries temporelles météorologiques.

2.6 Conclusion

Dans ce chapitre, nous avons présenté la Chaîne de Markov cachée et nous avons mis l'accent sur la distribution entre l'ensemble des états et l'ensemble des observations ainsi que les

distributions des lois de probabilités les générant. Par ailleurs, nous avons montré comment les Chaînes de Markov cachées ont été introduit comme modèle probabiliste et la première des applications. Les domaines de ces dernières ne cessent de s'élargir et les études sur les CMC sont en croissance à savoir les études sur les modèle de Markov non stationnaire d'ordre supérieur à 1. Cependant dans l'utilisation et l'application de notre modèle CMC, nous avons validé les trois hypothèses suivantes :

1. Propriété de Markov

Les probabilité de transitions des états sont les suivantes :

$$\begin{aligned} a_{ij} &= P\left(S_{t+1} = j \ / \ S_t = i_0,\ S_{t-1} = i_1,\ ...,\ S_{t-k} = i_k\right) \\ &= P\left(S_{t+1} = j \ / \ S_t = i\right) \end{aligned}$$

La supposition de Markov est vérifiée ; c'est à dire, l'état courant dépend seulement de l'état précédent et notre modèle devient alors une CMC du premier ordre.

2. Homogénéité

Les probabilité de transitions des états sont indépendantes du temps de réalisation des transitions ; i.e., pour tout t_1 et t_2.

$$P\left(S_{t_1+1} = j \ / \ S_{t_1} = i\right) = P\left(S_{t_2+1} = j \ / \ S_{t_2} = i\right)$$

3. Indépendance des observations de sortie conditionnellement aux états.

L'observation actuelle, conditionnée par l'état actuel, est statistiquement indépendante de l'observation précédente conditionnée par l'état précédent. En considèrent la séquence d'observations

$$X = X_1,\ X_2,\ ..., X_L$$

et pour une CMC λ, cette indépendance est formulée comme suit :

$$P\left(X \ / \ S_1,\ S_2,\ ..., S_L,\ \lambda\right) = \Pi_{t=1}^{L} P\left(X_t \ / \ S_t,\ \lambda\right)$$

Sur la base de ces hypothèses, nous considérons dans la suite de notre travail les trois problèmes suivants :

♠ Problème1:L'évaluation de probabilité d'observation

Soit la suite d'observations $X = X_1,\ X_2, ...,\ X_L$ et un modèle $\lambda = (A,\ B,\ \pi)$ il est primordial de savoir comment évaluer efficacement la probabilité de la suite d'observation, sachant le modèle $P\,(X/\lambda)$?

♠ Problème 2 : La recherche du chemin le plus probable, ou estimation de la partie cachée, ou encore décision

Soit la suite d'observations X et un modèle λ,comment peut-on choisir une suite d'états $S = (S_1, ...,\ S_L)$ qui soit optimale selon un certain critère convenable?

♠ Problème3 : Évaluation du modèle (l'apprentissage) :

Nous estimons les paramètres du modèle de manière optimale, ce qui revient à ajuster les paramètres du modèle λ pour maximiser $P\,(Y/\lambda)$.

Chapitre 3

Les méthodes d'estimation

3.1 Introduction

A partir d'une séquence d'états, une Chaîne de Markov cachée génère une séquence d'observation suivant une loi de probabilités conditionnée par les états. Une bonne Chaîne de Markov cachée est celle qui génère la suite observée avec la vraisemblance maximale quand les paramètres du modèle sont donnés. La vraisemblance des observations est calculée comme le produit de toutes les probabilités de transition des états et de toutes les probabilités d'émission des observations conditionnellement aux états et sachant le modèle. Toutefois, ce calculs est très coûteux en temps de calcul [qui dans cette approche séquentielle, croît exponentiellement avec la longueur de la séquence] et en un espace de stockage [puisqu'il mémorise tous les états précédents]. Néanmoins, l'utilisation des techniques de programmation dynamique, connues sous le nom de procédure forward-backward, permet d'effectuer le même calcul dans un temps proportionnel au nombre d'états et à la longueur de la séquence d'observations.

L'estimation des paramètres du modèle peut se faire par la méthode du maximum de vraisemblance (ML) ou son alternative la méthode du maximum a posteriori (MAP) avec laquelle nous maximisons la probabilité a posteriori du modèle quand les séquences sont données ; le passage d'une méthode à l'autre est donné par la règle de Bayes :

$$P\left(S/X,\lambda\right)=\frac{P\left(S,X/\lambda\right)}{P\left(X/\lambda\right)} \tag{3.1.1}$$

Où $P(X/\lambda)$ est considérée comme constante de normalisation. L'estimation MAP est donc équivalente à maximiser $P(S/X,\lambda)$ sur toute la séquence et l'estimation ML est équivalente à maximiser $P(S,\ X/\lambda)$ sur toute la séquence. Partant d'un point arbitraire, nous pouvons calculer l'estimateur d'un modèle de CMC en utilisant les deux méthodes, ML et MAP, qui par ré estimation itérative maximise soit la vraisemblance du modèle, soit la probabilité a posteriori. Les étapes des algorithmes des deux méthodes sont les suivantes :

1. Un modèle initial est crée en attribuant des valeurs arbitraires à la distribution de probabilités de transition ;

2. Puis, en utilisant le modèle actuel, nous considérons tous les chemins possibles de chaque séquence pour avoir un nouvel estimateur des probabilités de transition et d'émission (dit paramètres du modèle);

3. Dans cette étape, un nouveau modèle est crée en remplaçant les paramètres par leurs estimateurs pour chaque état, chaque symbole et chaque séquence ;

4. Les étapes 2 et 3 sont répétées jusqu'à ce que les paramètres actuels ne changent pas de manière significative ou jusqu'à obtenir un nombre prédéterminé d'itérations.

Tant que la vraisemblance du modèle croît à chaque itération, il n'existe pas un bon modèle et, à chaque itération, nous créons un modèle meilleur, au moins localement, pour la séquence d'apprentissage.

3.2 Le calcul de la vraisemblance des observations

Etant donné un ensemble de séquences $\{S(j)\}$ et le modèle λ, la vraisemblance de l'ensemble des séquence $S = \{S(j)\}_{j\in N}$ est le produit des probabilités de chaque séquence $S(j)$ sachant le modèle :

$$P(S/\lambda) = \Pi_j P(S(j)/\lambda)$$

Où chaque terme de $P(S(j)/\lambda)$ est calculé en substituant $S(j)$ par $(S_1,\ S_2, ...,\ S_L)$ dans l'équation précédente ; ceci est appelé la vraisemblance du modèle dont on cherche la

valeur la plus élevée. Ainsi, l'objectif de la méthode du maximum de vraisemblance est de cherché à maximisé l'expression $\Pi_j \; P\left(S\left(j\right)/\lambda\right)$.

3.2.1 L'évaluation directe

Etant donné une suite d'observations $X = (X_1,\ X_2, ...,\ X_L)$ et un modèle de CMC $\lambda = (A,\ B,\ \Pi)$, comment peut-on calculer efficacement la probabilité que la suite d'observation X soit produite par λ, c'est-à-dire $P\left(X/\lambda\right)$. Autrement dit, comment évaluer le modèle afin de choisir parmi plusieurs celui qui génère le mieux cette suite d'observations. La méthode la plus simple est celle qui énumère toutes les séquences d'états possibles de longueur L (nombre d'observations dans une séquence) [2]. Pour chaque séquence d'états $S = S_1, ..., S_t, ..., S_L$, $S_t = i$ et $i = 1, 2, ..., N$, la vraisemblance de la séquence d'observations X est :

La probabilité $P\left(X/\lambda\right)$ d'une suite d'observation X, sachant qu'un modèle $\lambda = (A,\ B,\ \Pi)$ est donné, est la somme sur tous les chemins d'états possibles S , des probabilités conjointes de X et de S par rapport à ce modèle :

$$P\left(X/\lambda\right) = \sum_{S} P\left(X,\ S/\lambda\right) \tag{3.2.1}$$

La règle de bayes permet de calculer $P\left(S, X/\lambda\right)$:

$$P\left(X,\ S/\lambda\right) = P(X/S,\ \lambda)P(S/\lambda) \tag{3.2.2}$$

Si $S = S_1, ..., S_t, ..., S_L,\ S_t = i$ et $1 \leq i \leq N$ est une séquence d'état fixée, alors la probabilité d'observer la séquence X pour la séquence d'état S est :

$$P\left(X/S, \lambda\right) = P(X_1...X_L\ /\ S_1...S_L,\ \lambda) \tag{3.2.3}$$

On considère les observation comme étant indépendantes d'où:

$$\begin{aligned} P\left(X/S, \lambda\right) &= \Pi_{t=1}^{L} P(X_t/S_1...S_L,\ \lambda) \\ &= \Pi_{t=1}^{L} P(X_t/S_t,\ \lambda) \\ &= b_{S_1}\left(X_1\right) b_{S_2}\left(X_2\right) ... b_{S_L}\left(X_L\right) \end{aligned} \tag{3.2.4}$$

En outre on a :

$$P\left(S/\lambda\right) = \pi_{S_1} a_{S_1 S_2} a_{S_2 S_3} ... a_{S_{L-1} S_L} \tag{3.2.5}$$

D'où :

$$P(X/\lambda) = \sum_{S_1=1}^{N} ... \sum_{S_L=1}^{N} \pi_{S_i} b_{S_1}(X_1)\, a_{S_1S_2} b_{S_2}(X_2)\, a_{S_2S_3} ... b_{S_{L-1}}(X_{L-1})\, a_{S_{L-1}S_L} b_{S_L}(X_L) \tag{3.2.6}$$

On interprète le calcul ci-dessus comme suit :

• Initialement à $t = 1$ nous sommes dans l'état S_1 avec la probabilité π_{S_1} et produit le symbole (observation) X_1 (dans cette état) avec une probabilité $b_{S_1}(X_1)$;

• Après on change le temps de t à $t+1$ (i.e $t = 2$), en faisant une transition de l'état S_t à S_{t+1} (respectivement S_1 à S_2) avec une probabilité de transition $a_{S_{t+1}S_t}$ (respectivement $a_{S_1S_2}$) qui produit le symbole (observation) X_{t+1} (respectivement X_2) avec une probabilité $b_{S_{t+1}}(X_{t+1})$ (respectivement $b_{S_2}(X_2)$);

• Ainsi de suite jusqu'à ce qu'on arrive au temps L (dernière transition) de l'état S_{L-1} à l'état S_L avec une probabilité $a_{S_{L-1}S_L}$ qui produit le symbole X_L qui génère une probabilité transition $b_{S_L}(X_L)$.

3.2.2 La procédure forward-backward

D'après la formule de calcul direct de $P(X/\lambda)$, nous remarquons qu'il existe N états possibles et chacun de ces états peut être atteint à partir des N états précédents sur toute la longueur L de la séquence et donc nous avons N^L séquences d'états possibles [73] nous rappelons que N^L est le nombre de chemins possibles de longueur L. En effet, la formule (3-7) nécessite $(N^L - 1)$ additions et $.(2L-1)N^L$ multiplications. Soit environ $2LN^L$ opérations (complexité $O\left(2LN^L\right)$, ce qui est un ordre exponentiel considérablement lourd et coûteux en temps de calcul, et non faisable même pour des petite valeur de N et L. Par exemple pour N=5 et L=100 on obtient environ 10^{72} opérations. Il existe une autre variante de cette méthode d'évaluation de la vraisemblance appelée la procédure "forward-backward" qui permet un calcule moins complexe et demande moins d'espace mémoire pour le stockage des variables correspondantes [17].

Dans cette approche, on considère que la conséquence d'observations peut se faire en deux sous -séquences (étapes) :

1- Dans la première étape, il se produit l'émission de la suite d'observations $X_1, X_2, ..., X_t$ et la réalisation d'états $S_t = i$ au temps t;

2- Ensuite, l'émission de la suite d'observations $X_{t+1}, X_{t+2}, ..., X_L$ se produit à partir de l'états $S_t = i$ au temps t;

On définit ainsi deux probabilités pour évaluer la probabilité d'observations: la première connue sous le nom anglo-saxon de probabilité forward, se calculant par une récurrence progressive, et la seconde connue sous le nom de probabilité backward, se calculant par une récurrence régressive. La probabilité de l'observation :

$$P\left(X = x\ /\lambda\ \right) = \sum_{i} P\left(X_1 = x_1, ..., X_t = x_t, S_t = i\ /\lambda\right)$$

Elle peut être décomposée comme suit[17] :

$$\begin{aligned} P\left(X = x/\lambda\ \right) \ = \ & \sum_{1\ \leq\ i\ \leq\ N} P(X_1 = x_1, ..., X_t = x_t, S_t = i\ /\lambda) \\ & *P\left(X_{t+1} = x_{t+1}, ..., X_L = x_L\ /S_t = i, \lambda\right) \end{aligned}$$

Soit $\alpha_t\left(i\right)$ la probabilité forward correspondant au premier terme du produit :

$$\alpha_t\left(i\right) = P(X_1 = x_1, ..., X_t = x_t, S_t = i/\lambda) \tag{3.2.7}$$

Et $\beta_t\left(i\right)$ la probabilité backward correspondant au second terme :

$$\beta_t\left(i\right) = P\left(X_{t+1} = x_{t+1}, ..., X_L = x_L\ /S_t = i, \lambda\right) \tag{3.2.8}$$

On obtient ainsi:

$$P\left(X = x/\lambda\ \right) = \sum_{1\ \leq\ i\ \leq\ N} \alpha_t\left(i\right)\ \beta_t\left(i\right) \tag{3.2.9}$$

L'intérêt des probabilités $\alpha_t\left(i\right)$, $\beta_t\left(i\right)$ est qu'elles peuvent être calculées par des méthodes récursives :

Remarque 3.2.1 *Pour résoudre le premier problème énnoncé à la fin du chapitre 2, il suffit de calculer la variable forward;*

Le calcul de la variable backward permet de résoudre le troisième problème.

La procédure Forward:

$\alpha_t(i)$ désigne la probabilité conjointe de la la séquence partielle d'observations de l'instant initial 1 à l'instant t avec l'hypothèse que le processus est dans la classe i à l'instant t.

$$\alpha_t(i) = P(X_1 = x_1, ..., X_t = x_t, S_t = i/\lambda) \tag{3.2.10}$$

Pour les chaînes de Markov cachées, nous calculons la vraisemblance de la séquence $X_1, X_2, ..., , X_t$. Soit $P(X\ )$ cette probabilité:

$$P(X) = P(x_1\ )\,\Pi_{i=1}^{L} a_{X_{t-1}X_t}$$

A cause de la multitude des chemins pour une seule séquence, la probabilité de la séquence peut être écrite:

$$P(X\ ) = \sum_{S} P(X, S\ ) = \sum_{S_1=1}^{N} ... \sum_{S_1=L}^{N} P(X, S\ )$$

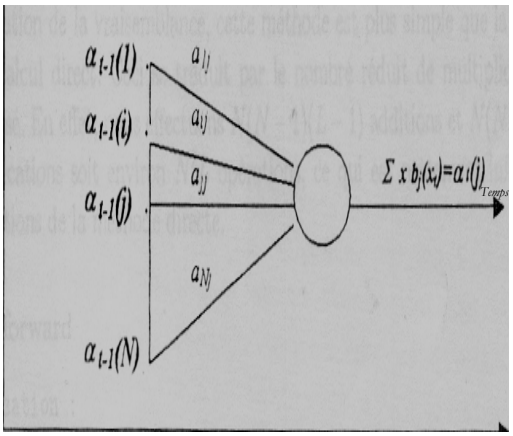
Ceci suppose l'énumération de tous les chemins possibles ; une méthode approximative consiste à considérer la somme sur tous les symboles de chaque état de la séquence; cette méthode est appelée la procédure forward. Soit la variable forward $\alpha_t(i)$ définie par [74] comme suit

Cette expression peut se calculer (pour chaque t) de manière récursive comme suit :

$$\alpha_t(i) = P(X_1 = x_1, ..., X_t = x_t, S_t = i/\lambda) \qquad 1 \leq t \leq L, 1 \leq i \leq N \tag{3.2.11}$$

Où $\alpha_t(i)$ désigne la probabilité de la la séquence partielle de d'observations jusqu'à l'état i et à l'instant t sachant le modèle λ.l'équation de récurrence est la suivante:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(x_{t+1}) \qquad pour \qquad 1 \leq t \leq L-1, 1 \leq j \leq N$$



Séquence partielle pour le calcul de la variable forward

Nous procédons récursivement comme suit:

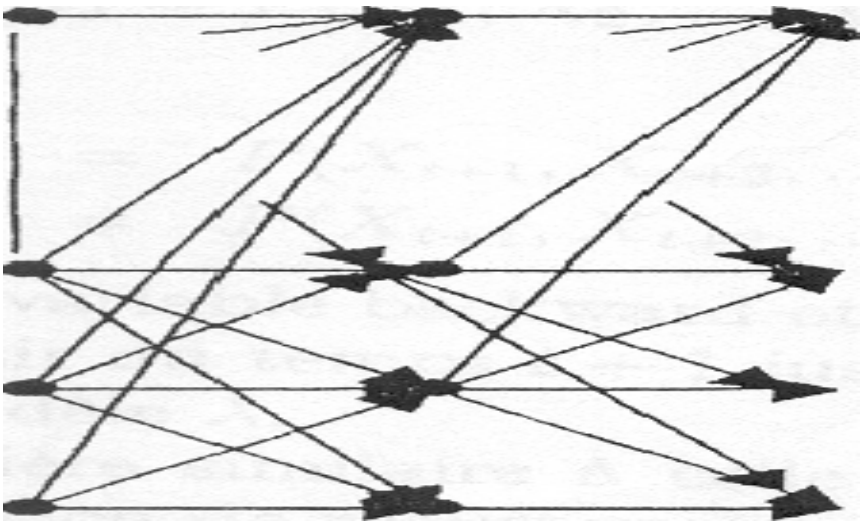
1. $\alpha_1(i) = \pi_i b_i(x_1) \qquad 1 \leq i \leq N$. Cette étape initialise la probabilité forward ; C'est la probabilité conjointe de l'état $S_1 = i$, $1 \leq i \leq N$ et l'observation initiale x_1.

2. $\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(x_{t+1})$. Cette étape correspond à l'induction illustrée dans la figure 3.1 qui montre que l'état $S_{t+1} = j$ peut être visité (atteint) au temps $t+1$ à partir de N états possibles $S_t = i$, $1 \leq i \leq N$ au temps t. $\alpha_t(i)$ et donc la probabilité jointe des observations $X_1, X_2, ..., X_t$ et l'état j est atteint au temps $t+1$ via l'état i au temps t. En sommant ce produit sur tous les N états possibles i, $1 \leq i \leq N$, au temps t , le résultat est la valeur de la probabilité de l'état j au temps $t+1$ avec toutes les séquences d'observations partielles inclues jusqu'au temps t .

Il est donc facile de marquer $\alpha_{t+1}(j)$ est obtenue en augmentant multiplicatives la quantité sommée avec la probabilité $b_j(x_{t+1})$.

3. Finalement, $P(X/\lambda) = \sum_{i=1}^{N} \alpha_L(i)$;cette équation donne le calcul désiré de $P(X/\lambda)$ comme étant la somme de la variable forward $\alpha_L(i)$, tel que $\alpha_L(i)$ est égale à $P(X_1, X_2, ..., X_L, S_L = i/\lambda)$. Toutes les transitions possibles entre les états peuvent être représentés sous forme du treillis ci -dessous où chaque noeud correspond à un état distinct au moment actuel t donné et chaque branche représente une transition à de nouveaux états

au moment suivant $t+1$. L a propriété la plus importante dans la structure de treillis est que à chaque séquence d'états S correspond un chemin unique et vice versa.



Implémentation du calcule de α_t ou β_t sous forme de Treillis

Pour l'évaluation de la vraisemblance, cette méthode est plus simple que la méthode précédente de calcul directe Ceci se traduit par le nombre réduit de multiplications et d'additions utilisé.

En effet, nous effectuons $N(N+1)(L-1)+N$ multiplications et $N(N-1)(L-1)$ additions soit environ N^2L opérations sont effectuées. Cette méthode permet de réduire la complexité du calcul de $P(x)$. Par exemple pour N=5 et L=100 on obtient 3000 opérations au lieu de 10^{72} opérations demandées par la méthode directe.

Algorithme 3.2.1 *Forward*

– ***Initialisation : t=1,***

$$\alpha_1(i) = \pi_i b_i(x_1) \qquad 1 \leq i \leq N \tag{3.2.12}$$

- ***Induction (Récurrence) : t=2, 3,..., L,***

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(x_{t+1}) \qquad pour \qquad 1 \leq t \leq L-1, 1 \leq j \leq N \tag{3.2.13}$$

-Terminaison (critère d'arrêt) :

$$P\left(X/\lambda\ \right) = \sum_{i=1}^{N}\ \alpha_L\left(i\right) \tag{3.2.14}$$

• **La procédure Backward**:

D'après l'équation (3.2.9) , nous savons que la probabilité de la séquence observée au temps t à l'état S_t est :

$$\begin{aligned} P\left(X = x, S_t = i\ /\lambda\ \right) &= P\left(X_1 = x_1, ..., X_t = x_t, S_t = i\ /\lambda\right) \\ &= P(X_1 = x_1, ..., X_t = x_t, S_t = i\ /\lambda) \\ &\quad *P\left(X_{t+1} = x_{t+1}, ..., X_L = x_L\ /X_1, ..., X_t, S_t = i, \lambda\right) \\ &= \alpha_t\left(i\right)\beta_t(i) \end{aligned}$$

Tel que

$$\alpha_t\left(i\right) = P(X_1, ..., X_t, S_t = i\ /\lambda) \quad 1 \leq t \leq L, 1 \leq i \leq N$$

Alors

$$\begin{aligned} \beta_t(i) &= P\left(X_{t+1}, X_{t+2}, ..., X_L\ /X_1, ..., X_t, S_t = i, \lambda\ \right) \\ &= P\left(X_{t+1}, X_{t+2}, ..., X_L\ /S_t = i, \lambda\right) \qquad 1 \leq t \leq L, 1 \leq i \leq N \end{aligned}$$

Où $\beta_t(i)$ est la variable backward et définie comme étant la probabilité de la séquence partielle à partir du temps $t+1$ jusqu'à la fin de la séquence $(t = L)$, étant donnés l'état $S_t = i$ et le modèle λ.

Le calcul est similaire à la démarche précédente de $\alpha_t\left(i\right)$. Le calcul de $\beta_t(i)$ se fait inductivement en utilisant l'équation de récurrence

$$\beta_t(i) = P\left(X_{t+1}, X_{t+2}, ..., X_L\ /S_t = i, \lambda\right) \qquad 1 \leq t \leq L, 1 \leq i \leq N \tag{3.2.15}$$

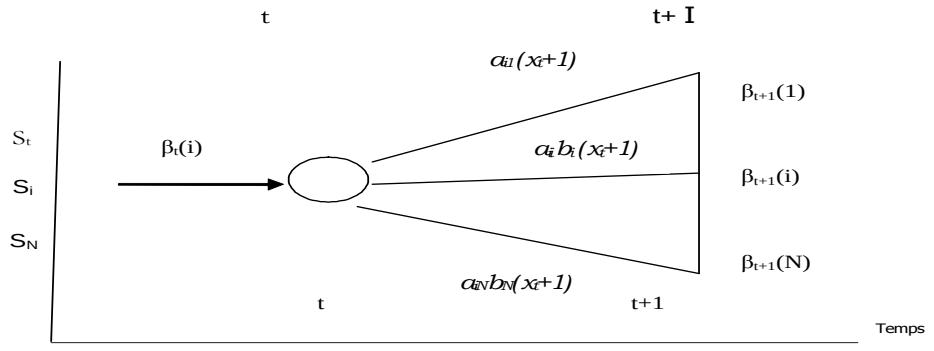
La procédure de calcul est la suivante:

1. $\beta_L(i) = 1 \quad pour \quad 1 \leq i \leq N$ cette étape définie arbitrairement $\beta_L(i) = 1$ pour tous les états i, $1 \leq i \leq N$;

2. à chaque instant $t = L-1, L-2, ..., 1$;nous avons

$$\beta_t(i) = \sum_{j=1} a_{ij} b_j(X_{t+1}) \beta_{t+1}(j)$$

Cette induction, explicitée dans la figure ci-dessous, explique que pour être à l'état i au temps t et en tenant compte du reste de la séquence d'observations, nous devons prendre en compte toutes les transitions vers chacun des N états possibles au temps $t+1$ et aussi tenir compte de la suite d'observations partielle restante à partir de l'observation X_{t+1}.



séquence partille pour le calcul de la variable backward

Algorithme 3.2.2 ***backward***

– **Initialisation : t = L**

$$\beta_L(i) = 1 \quad pour \quad 1 \leq i \leq N \tag{3.2.16}$$

Algorithme 3.2.3 - ***Induction :*** $t = L-1,\ L-2, ..., 1;$

$$\beta_t(i) = \sum_{j=1} a_{ij} b_j(y_{t+1})\,\beta_{t+1}(j) \qquad pour \qquad 1 \le t \le L-1, 1 \le i \le N$$

- ***Critère d'arrêt :***

$$P(X/\lambda) = \sum_{i=1}^{N} b_i(X_1)\,\beta_1(i)$$

D'une manière similaire à la procédure forward, cette procédure nécessite dans le calcul de $P(X/\lambda)$, $N(N+1)(L-1)+N$ multiplications et $N(N-1)(L-1)$ additions soit environ N^2L opérations sont effectuées. En utilisant la technique "forward-backward", la vraisemblance des observations peut être calculée, avec une complexité d'algorithme égale à $O(N^2L)$, des trois façons suivantes:

$$\begin{aligned}
1.\ P(X/\lambda) &= \sum_{1 \le i \le N} \alpha_t(i)\ \beta_t(i) \\
2.\ P(X/\lambda) &= \sum_{i=1}^{N} \alpha_L(i) \\
3.\ P(X/\lambda) &= \sum_{i=1ij}^{N} \pi_i \beta_0(i)
\end{aligned}$$

Remarque 3.2.2 ***1.*** *Le calcul respectif des probabilités forward et backward se heurte à des difficultés d'ordre numérique. En effet, les quantités figurant dans les expressions de* $\alpha_t(i)$ *et* $\beta_t(i)$ *sont très petites et cet algorithme donne des erreurs quand il est implémenté sur ordinateur dû au produit d'un grand nombres inférieurs à 1 (l'underflow) .*

2. *Pour l'implémentation sur ordinateur, tous ces algorithmes présentes le même problème que l'algorithme forward vu le produit de nombres inférieurs à 1 et qui tendent vers zéro quand le degré du produit augmente.*

3. *Les deux variables* $\alpha_t(i)$ *et* $\beta_t(i)$ *peuvent être utilisées pour calculer* $P(X/\lambda)$ *à chaque instant* $t, 1 \le t \le L$:

$$\begin{aligned}
P(X/\lambda) &= \sum_{i=1}^{N} \alpha_t(i) \cdot\ \beta_t(i) \qquad (3.2.17) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)\, a_{ij} b_j(X_{t+1})\,\beta_t(i)
\end{aligned}$$

4. *La probabilité $P(X)$ est résultat de l'algorithme forward ou backward; cette probabilité peut être calculée en posant $t = L$ dans l'équation (3.2.17) et nous retrouvons l'équation*

$$P(Y/y) = \sum_{i=1}^{N} \alpha_L(i)$$

5. *La probabilité $P(X)$ est habituellement calculée par la procédure forward.*

3.2.3 Algorithme de viterbi

Cette technique est basée sur l'algorithme de Viterbi qui sera explicité plus tard.

3.3 Calcul du chemin optimal (estimation de la suite cachée)

3.3.1 Introduction

Etant donné une suite d'observations $X_1....X_L$, et un modèle λ, comment peut-on choisir une suite d'états $S = S_1, S_2, ..., S_L$ qui soit optimale selon un critère convenable. La difficulté réside dans la définition de la suite optimale d'états, c'est-à-dire qu'il existe plusieurs critères d'optimalité à savoir si l'objectif est de maximiser la probabilité jointe $P(X,\ S\ /\ \lambda)$ ou la probabilité conditionnelle $P(S\ /X, \lambda)$. Selon le choix du critère nous proposons deux méthodes pour chercher la solution optimale :

1. estimation par le maximum de vraisemblance (algorithme de Viterbi),

2. estimation par le maximum a posteriori ou estimation Bayesienne,

Sous le critère de la probabilité de l'erreur minimale, il sera nécessaire de déterminer soit la vraisemblance conjointe

$$P(X, S/\lambda)$$

Ou la probabilité a posteriori

$$P(S/X, \lambda)$$

Qui sont liées par la relation suivante :

$$P\left(S/X,\lambda\right)=P(X,S/\lambda)P\left(X/\lambda\right)$$

Pour la vraisemblance, nous calculons

$$\vartheta_t\left(s_i\right)=P\left(X_1,X_2,..,X_t,\ S_t=i\ /\lambda\right)\qquad 1\leq i\leq N\ ,\ 1\leq t\leq L \tag{3.3.1}$$

qui est la probabilité de l'émission de la sous séquence $X_1,X_2,..,X_t$ et le système est à l'états i au temps t étant donné le modèle λ;

3.3.2 La procédure de Viterbi

La tache essentielle d'un modèle utilisant les CMC est de déterminée le chemin correspondant à l'observation, c.-à-d. de trouver dans le modèle la meilleur suite d'états qui maximise la quantité suivante :

$$P\left(S/X,\lambda\right)\Leftrightarrow P\left(S,X/\lambda\right)$$

Une technique formelle pour trouver le chemin optimal est basée sur les méthodes de programmation dynamique, c'est l'algorithme de Viterbi proposé par Viterbi [87].

C'est un Algorithme récursif qui permet de trouver à partir d'une suite d'observations provenant d'un canal sans mémoire, une solution optimale au problème d'estimation de la suite d'états d'un processus de Markov à temps discret qui produit cette suite d'observations.

Pour trouver le meilleur chemin $S=(S_1,S_2,..,S_L)$ pour une suite d'observations $X=(X_1,X_2,..,X_L)$, on définit la variable $\delta_t\left(i\right)$ qui est la probabilité du meilleur chemin amenant à l'état i a l'instant t, en étant guidé par les t premières observations :

$$\delta_t\left(i\right)=\max_{S_1,S_2,..,S_L}P\left(S_1,S_2,..,S_t=i,X_1,X_2,..,X_t\ /\lambda\right) \tag{3.3.2}$$

Par récurrence (induction), la probabilité du chemin le plus probable à l'instant t+1 étant donné que le système est à l'état $S_{t+1}=j$ est obtenue comme suit :

$$\delta_{t+1}\left(j\right)=[\max_{1\leq\ i\ \leq N}\delta_t\left(i\right)a_{ij}]b_j\left(x_{t+1}\right),\ 1\leq j\leq N \tag{3.3.3}$$

Pour retrouver la suite optimale d'états, nous devons garder une trace des arguments qui maximise l'équation (3.3.3) pour chaque t et j, lors du calcul. La suite d'états qui donne le meilleur chemin amenant à l'état i à t est représentée dans un tableau $\psi_t\left(j\right)$.

Principe de l'Algorithme de Viterbi est le suivant:

Nous avons déjà mentionné que l'algorithme de Viterbi est celui qui donne la meilleure séquence d'états $S = (S_1, S_2, .., S_L)$ pour une séquence d'observations donnée $X = (X_1, X_2, .., X_L)$. Afin d'atteindre cet objectif, il est nécessaire de maximiser la probabilité conjointe $P(X, S)$:

$$\arg\max_{S} P(X, S) \Rightarrow S_{optimal(Viterbi)} \tag{3.3.4}$$

Pour tout $1 \leq i, j, l \leq N$

$$\begin{aligned} P(X, S) &= P(S) P(X/S) \qquad (3.3.5)\\ &= P(S_1 = l) P(X_1/S_1 = l) \Pi_{t=2}^{L} P(S_t = j/S_{t-1} = i) \Pi_{t=2}^{L} P(X_t/S_t = j) \\ &= \pi_1 b_l(x_1) \Pi_{t=2}^{L} a_{ij} b_j(X_t) \end{aligned}$$

On a alors,

$$\max_{S} P(X, S) = (\pi_1 b_l(X_1)) \Pi_{t=2}^{L} \delta(S_t = j) \tag{3.3.6}$$

La probabilité $P(X, S)$ représente donc le score (coût) total pour le chemin S, où δ est le score (coût) d'un segment (une transition d'un état à un autre) de chemin S qu'on cherche à rendre meilleur.

Nous définissons $\psi_L(j)$ comme étant le chemin du meilleur score jusqu'à la fin de la séquence $t = L$ de l'état j,soit :

$$\psi_t(j) = \underset{\{S_1, S_2, .., S_L\} \subset S}{\arg\max} P(S_1, S_2, .., S_L = j/X_1, X_2, .., X_L, \lambda)$$

La règle de Bayes permet aussi d'écrire après simplification du terme constant[14] :

$$\psi_t(j) = \underset{\{S_1, S_2, .., S_L\} \subset S}{\arg\max} P(S_1, S_2, .., S_L = jX_1, X_2, .., X_L/\lambda)$$

Ainsi, nous avons défini précédemment $\delta_t(j)$ comme étant la probabilité jointe maximale d'occurrence de la suite d'observations jusqu'à l'instant t émise par le modèle λ en suivant un chemin qui arrive à l'état i.

$$\delta_t(i) = \max_{\{S_1,S_2,..,S_{t-1}\}} P(S_1, S_2, .., S_t = i, X_1, X_2, .., X_t \ /\lambda)$$

1. Initialement le processus est à l'instant $t = 1$ et le symbole du premier état est i,avec la probabilité

$$\delta(S_1 = i) = \pi_i b_i(x_1)$$

à cette étape aucun état n'est optimale, c'est à dire $\psi_1(i) = 0$

2. On utilise les formules de récurrence suivantes pour le calcul de :

La probabilité du chemin le plus probable

$$\delta_t(j) = b_j(x_t) \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \tag{3.3.7}$$

Et le chemin le plus probable

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \tag{3.3.8}$$

La probabilité du chemin le plus probable à une certaine étape $(t-1)$, $\delta_{t-1}(i)$ après transition a_{ij} calculé à son maximum et multipliée par la probabilité d'émission donne la probabilité du chemin le plus probable à l'étape suivante (t), soit $\delta_t(j)$ cette probabilité équation (3.3.7) . Un pointeur est mis en oeuvre pour localiser le meilleur symbole (état) équation(3.3.8) . Il est concrétisé par la fonction $\psi_t(i)$ qui permet de mémoriser l'indice i avec lequel la valeur de la fonction $|\delta_t(j)|$ est maximal. Enfin, ces équations de récurrence sont répétées jusqu'à la fin de la séquence

Algorithme 3.3.1 de Viterbi *[74], [73]*

1. Initialisation, t=1

Si S_1 *est connu a proiri, alors*

$$\begin{aligned} \delta_1(i) &= 0, \quad \forall i \quad \textit{(coût du sur viveur i)} \\ \psi_1(i) &= i \quad \textit{(cette variable stocke l'état optimal à l'instant t)} \end{aligned} \tag{3.3.9}$$

Autrement,

Si S_1 *est inconnu a priori, alors*

$$\delta_1(i) = (\pi_i b_i(x_1)) \quad 1 \leq i \leq N \tag{3.3.10}$$

$$\psi_1(i) = 0 \tag{3.3.11}$$

2. Induction (récurrence) : t=2,3,...,

$$\delta_t(j) = b_j(x_t) \max_{1 \leq i \leq N}[\delta_{t-1}(i)]a_{ij}] \qquad 1 \leq j \leq N \tag{3.3.12}$$

$$\Psi_t(j) = \arg \max_{1 \leq i \leq N}[\delta_{t-1}(i)\, a_{ij}] \qquad 1 \leq j \leq N, 2 \leq t \leq T \tag{3.3.13}$$

3. Terminaison:

$$P^* = \max_{1 \leq i \leq N}[\delta_L(i)] \tag{3.3.14}$$

$$S_L^* = \arg \max_{1 \leq i \leq N}[\delta_L(i)] \tag{3.3.15}$$

4. Chemin obtenu (recherche arrière) (Trace back) :

$$S_t^* = \Psi_{t+1}\left(S_{t+1}^*\right), \qquad L-1 \geq t \geq 1. \tag{3.3.16}$$

Viterbi appelle la quantité $\ln P^* = \max_{1 \leq i \leq N}[\delta_L(i)]$ le score. À chaque instant t et pour chaque état, il existe N prédécesseurs (N états de l'instant précédent $t-1$) ; l'algorithme nécessite à chaque instant la mémorisation de ces N sur viveurs (structure de Treillis) ainsi que leurs scores respectivement ; mais une fois les $\delta_t(i)$ sont calculés, il n'est pas nécessaire de stocker l'observation x_t, seule la variable Ψ_t stocke l'état optimal à chaque instant ce qui rend l'algorithme de Viterbi utilisable en temps réel.

Nous remarquons que dans son implémentation, l'algorithme de Viterbi est similaire à celui de l'algorithme forward à l'exception de l'étape trace back recherche arrière) : dans Viterbi, on teste le maximum sur toutes les probabilités des états à chaque instant.

Cependant, la multiplication d'un grand nombre de probabilités implique des problèmes d'underflow sur l'ordinateur. Une solution proposée pour l'algorithme de Viterbi est de calculer $\ln P(X/\lambda)$ au lieu de calculer $P(X/\lambda)$.

En effet, le logarithme qui est une fonction croissante, permet de préserver le sens de l'évolution de la probabilité du chemin le plus probable et transforme par ailleurs le produit dans l'équation (3.3.7) en somme ; ceci facilite l'implémentation sur machine du calcul des probabilités ainsi que du chemin le plus probable. Donc nous calculons δ_t et $S_{optimal(Viterbi)}$ comme suit :

$$\delta_t(i) = \max_{S_1,S_2,..,S_{t-1}} \ln P(S_1, S_2, .., S_t = i, X_1, X_2, .., X_t \ /\lambda) \qquad 2 \leq t \leq L$$

et la formule de récurrence devient :

$$\delta_t(j) = \max_{1 \leq i \leq N} \delta_{t-1}(i) + \ln a_{ij} + \ln \ b_j(x_t) \qquad 1 \leq \ j \leq N \tag{3.3.17}$$

L'objectif étant de maximiser $\ln P(X, S)$, on peut alors utiliser le même principe de Viterbi pour écrire :

$$\ln P(X, S) = \ln \pi_1 b_l(X_1) + \sum_{t=2}^{L} \delta(S_t = j) \qquad 1 \leq i, \ j, l \leq N$$

et

$$\delta(S_t = j) = \ln a_{ij} + \ln \ b_j(x_t) \tag{3.3.18}$$

on obtient donc l'algorithme' suivant :

Algorithme 3.3.2 *1. Initialisation, t=1*

$$\delta_1(i) = \ln \pi_i b_i(x_1) \qquad 1 \leq i \leq N \tag{3.3.19}$$

$\psi_1(i) = 0$

2. Induction (itération) : t=2,3,...,L

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) + \ln a_{ij}] \ + \ln b_j(x_t) \qquad 1 \leq \ j \leq N \tag{3.3.20}$$

$\Psi_t(j) = \arg\max_{1 \leq i \leq N} [\delta_{t-1}(i) + \ln a_{ij}] \qquad 1 \leq j \leq N, 2 \leq t \leq T$

3. Terminaison :

$$\ln P^* = \max_{1 \leq i \leq N} [\delta_L(i)] \tag{3.3.21}$$

$S_L^* = \arg\max_{1 \leq i \leq N}[\delta_L(i)]$

4. Chemin obtenu (recherche arrière) (Trace back) :

$$S_t^* = \Psi_{t+1}\left(S_{t+1}^*\right), \qquad L-1 \geq t \geq 1. \tag{3.3.22}$$

La fonction "$argmax$" permet de mémoriser l'indice i, entre 1 et N avec lequel on atteint le maximum des quantités $(\delta_{t-1}(i)\, a_{ij})$ le coût des opérations est en $O(N^2 L)$

3.3.3 La restauration par le maximum a posteriori

En général, l'utilisation de $P(S_t = i \ / \ X, \lambda)$ avec le décodage de viterbi est utile quand les différents chemins ont presque la même probabilité que le chemin le plus probable. Ceci justifie qu'il n'est pas toujours nécessaire de ne prendre en considération que le chemin le plus probable. Une des approches utilisées dans la littérature définit la séquence d'états qui maximise la probabilité $P(S_t = i \ / \ X, \lambda)$. Cette séquence d'états peut être plus appropriée quand on est plus intéressé par un état à un point particulier que par le chemin tout entier. En outre, la séquence d'états ainsi définie peut ne pas être similaire au chemin de Viterbi ; voir elle peut même être dans des cas particuliers un chemin non légitime si certaines transitions ne sont pas permises[71]. Nous choisissons l'état S_t le plus probable indépendamment des autres états, ce qui revient à choisir $P(S_t = i \ / \ X, \lambda)$ qui est la probabilité de l'état i au temps t sachant la suite d'observations et le modèle. Ce critère d'optimalité permet donc d'estimer le nombre maximum des états indépendants.

Nous essayons de calculer l'estimateur de S_t pour $1 \leq t \leq L$ étant donnée la réalisation de la suite d'observations (X) [2].

Pour tout $1 \leq t \leq L$ et $1 \leq i \leq N$, nous considérons la quantité :

$$\begin{aligned}
\vartheta_t^*(s_i) &= P(S_t = i \ / X_1, X_2, .., X_T, \ \lambda) \quad 1 \leq i \leq N, 1 \leq t \leq L && (3.3.23)\\
&= P(S_t = i \ / X_1, X_2, ..X_t, X_{t+1}, ..., X_L, \ \lambda) && (3.3.24)\\
&= P(S_t = i \ / X_1, X_2, ..X_t, \ \lambda) P(X_{t+1}, ..., X_L \ / S_t = i, \ \lambda) && (3.3.25)\\
&= \alpha_t(i)\beta_t(i), \ 1 \leq i \leq N, 1 \leq t \leq L && (3.3.26)
\end{aligned}$$

qui est la probabilité a posteriori que le système soit à l'état i au temps t connaissant la sous séquence d'observation $X_1, X_2, .., X_t$ et le modèle λ.

On définit la variable

$$\gamma_t(i) = \frac{\vartheta_t^*(i)}{P(X/\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)} \tag{3.3.27}$$

$P(X/\lambda)$ est appelé facteur de normalisation du fait que $\sum_{i=1}^{N}\alpha_t(i)\beta_t(i) = P(X/\lambda)$, et il peut être obtenu par l'application de la procédure forward ; d'où :

$$\sum_{i=1}^{N}\gamma_t(i) = 1 \tag{3.3.28}$$

En utilisant ainsi $\gamma_t(i)$ nous pouvons estimer l'état individuel S_t le plus probable au temps t

$$S_t = \arg \underset{1\leq i\leq N}{Max}\gamma_t(i) \tag{3.3.29}$$

Remarque 3.3.1 *Les variable α_t sont calculées et stockées de façon récursive. Elle sont utilisées ensuite pendant l'étape de régression "Backward" pour calculer $\vartheta_t^*(i) = \alpha_t(i)\beta_t(i)$, $1 \leq i \leq N$ et $t = L, L-1, ...1$.*

- Il est possible de résoudre le problème N° 1 vu précédemment par les formule suivante:

$$P(Y/\lambda) = \sum_{i=1}^{N}\alpha_T(i) = \sum_{i=1}^{N}\vartheta_t^*(i) \tag{3.3.30}$$

Bien que l'équation (3.3.29) maximise le nombre espéré des états individuels (ou indépendants) en sélectionnant l'état le plus vraisemblable à chaque instant, cependant on peut avoir quelques problèmes, relatifs à la suite d'états produite par cette équation [74] ; c'est le cas de la non existence (ou la non allocation) des transitions c'est à dire le HMM possède des transitions d'états nulles pour certains états i et j $(a_{ij} = 0)$, la suite d'états optimale estimée dans ce cas n'est pas valide. Ceci est due à la solution de l'équation (3.3.29) qui détermine l'état le plus vraisemblable à chaque instant sans prendre en compte la probabilité d'occurrence des suites d'états (l'état le plus probable à chaque instant sans considérer la structure de Treillis, le voisinage des états à chaque instant et la longueur de la séquence d'observations).

Algorithme 3.3.3 *Recurrence:*

t=1, 2,..., L

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(X\ /\lambda)} \qquad 1 \leq i \leq N$$

$$S_t = \arg \underset{1\leq i\leq N}{Max} \gamma_t(i) \qquad 1 \leq i \leq N$$

3.4 Estimation des Paramètres de la Chaîne de Markov cachée (λ)

3.4.1 Introduction

Il n'existe pas une méthode analytique directe pour l'estimation des paramètres du la Chaîne de Markov cachée (λ) qui maximise la vraisemblance des observations dans le cas où le processus $\{S_t\}$ n'est pas observé [73] cependant nous pouvons choisir λ de tel manière que $P(X/\lambda)$ ait un maximum local en utilisant une procédure itérative comme par exemple celle de Baum-welch appliqué au CMC ou son équivalence EM (Expectation Maximization ou modification).

L'approche repose sur la connaissance de la vraisemblance $P(X/\lambda)$ du modèle à données incomplets ; le calcul de cette dernière n'est pas facile à effectuer à cause de la perte de l'information contenu dans ce modèle. Hartley [38] a explicitement introduit l'algorithme EM comme une procédure de calcul qui approche l'estimateur de maximum de vraisemblance sachant un échantillon aléatoire de taille n d'une population à espace d'états discrets et a traité deux exemples du cas de l'information incomplète qui sont les données censurées et tronquées. Plus tard, les propriétés de l'algorithme EM ont été explicitées par Baum et Petrie [15] pour les chaînes de Markov stationnaires. Pendant la dernière décennie, nous avons assisté à l'étude des propriétés des estimateurs pour les Chaînes de Markov cachées généraux, c'est à dire non stationnaires à savoir les travaux de [79], [12], et [30].

Le principe du traitement des données incomplètes par le maximum de vraisemblance consiste à augmenter les données par la simulation du régime caché (S) et ensuite appliquer des algorithmes itératifs basés sur l'estimation par maximum de vraisemblance des paramètres du modèle. En travaillant sur des données complètes de modèle et la séquence

observée, on maximise la log vraisemblance des données complètes. Ces algorithmes diffèrent seulement dans la manière de simuler les séquences cachées.

3.4.2 Principe

Le principe dans les différentes méthodes d'estimation des paramètres du modèle de CMC, est d'approcher les éstimateurs du maximum de vraisemblance dans le cas de manque de données. La méthodes d'optimisation des paramètres appelés aussi apprentissage, consiste à ajuster les paramètres du modèle $\lambda(\pi,\ A,\ B)$ pour maximiser la vraisemblance des observations étant donnés le modèle λ ; c'est à dire maximiser $P(X/\lambda)$ [74].

Notre objectif est de définir une application F, tel que $\lambda^{(m+1)} = F\left(\lambda^{(m)}\right)$ améliore la vraisemblance d'observations $P(X/\lambda)$, ce qui permet de vérifier : que si $\lambda^{(m)}$ est notre modèle à l'itération m alors :

$$L\left(X/\lambda^{(m+1)}\right) \geq L\left(X/\lambda^{(m)}\right) \Leftrightarrow P\left(X/\lambda^{(m+1)}\right) \geq P\left(X/\lambda^{(m)}\right)$$

Ceci revient à écrire

$$\Pi_{t=1}^{L}\left(X_t\ /\lambda^{(m+1)}\right) \geq \Pi_{t=1}^{L}\left(X_t\ /\lambda^{(m)}\right) \tag{3.4.1}$$

Les paramètres à déterminer diffèrent selon que la distribution des observations soit continue ou discrète ; dans notre cas discret, le modèle est $\lambda = (\pi,\ A,\ B)$ dont les paramètres à estimer sont les composantes du vecteur de la probabilité initial π_i, les éléments de la matrice de la transition a_{ij} et la distribution de la probabilité d'émission $b_j(k)$.

$\lambda = (\pi,\ A,\ B)$ appartient à l'ensemble des paramètres Γ défini

$$\Gamma = \left\{ \begin{array}{c} \lambda \in IR^d\ /\ \forall 1 \leq i,j \leq N,\ \forall 1 \leq k \leq M, 0 \leq \pi_i \leq 1 \\ \sum_{i=1}^{N} \pi_i = 1,\ 0 \leq a_{ij} \leq 1, \sum_{i=1}^{N} a_{ij} = 1,\ 0 \leq b_j(k) \leq 1, \sum_{k=1}^{M} b_j(k = y_t) = 1 \end{array} \right\} \tag{3.4.2}$$

Nous avons supposé que toutes les transitions ont une probabilité positive (ie $a_{ij} \geq 0$, $\forall i,j = \overline{i,n}$, donc la chaîne de Markov $\{S_t\}$ est homogène irréductible et apériodique, et comme elle est à espace d'états fini alors d'après le théorème2.3.4 , elle est récurrente positive. Le théorème2.3.5 et le théorème érgodique2.3.6 nous permettent d'affirmer qu'il

existe une distribution π unique (stationnaire) qui est comme distribution initiale et elle est positive

On peut donc écrire :

$$\pi(i) = P(S_t = i) \geq 0,\ \forall 1 \leq i \leq N,\ \forall 1 \leq t \leq L$$

et l'ensemble des paramètres à estimer sera réduit à $\lambda = (A,\ B)$ tel que

$$\Gamma = \left\{ \begin{array}{c} \lambda \in IR^d \ / \ \forall 1 \leq i,j \leq N,\ \forall 1 \leq k \leq M, 0 \leq a_{ij} \leq 1 \\ \sum_{i=1}^{N} a_{ij} = 1,\ 0 \leq b_j(k) \leq 1, \sum_{k=1}^{M} b_j(k = y_t) = 1 \end{array} \right\} \tag{3.4.3}$$

Où d est le nombre de paramètres du modèle λ à estimer de telle manière que dans notre cas $d = N * N + N * M$ mais vu les contraintes sur les paramètres du modèle, le nombre d sera réduit à $d = N * (N-1) + N * (M-1)$.

L'idée est d'utiliser des procédures de ré estimation qui affine le modèle petit à petit en suivant les étapes suivantes :

$\star$ Choisir un ensemble initial de paramètres $\lambda^{(0)}\left(A^{(0)}, B^{(0)}\right)$;

$\star$ Calculer $\lambda^{(m+1)\left(A^{(m+1)},\ B^{(m+1)}\right)}$ à l'itération $(m+1)$ à partir de $\lambda^{(m)}\left(A^{(m)},\ B^{(m)}\right)$ à l'itération (m);

$\star$ Répéter ce processus jusqu'à un critère de fin (d'arrêt) qui est :

- partant de $\lambda^{(m)}, \lambda^{(m+1)}$ doit vérifier:

$$\Pi_{t=1}^{L} p\left(X_t \ /\lambda^{(m+1)}\right) \geq \Pi_{t=1}^{L} p\left(X_t \ /\lambda^{(m)}\right)$$

C'est à dire on cherche a définir une fonction F tel que:

$$\lambda^{(m+1)} = F\left(\lambda^{(m)}\right)$$

Généralement, pour l'estimation des paramètres on suppose l'existence de plusieurs séquences d'observations appelées corpus d'apprentissage. Soit X^1, X^2, ..., X^r ce corpus.

On suppose l'indépendance de ces séquences, la loi de la probabilité jointe de toutes ces séquences est donc équivalente au produit des probabilités des séquences prises individuellement

$$L\left(X^1, X^2, ..., X^r \ / \ \lambda\right) + \ln P\left(X^1, X^2, ..., X^r \ / \ \lambda\right) + \ln \Pi_{i=1}^{r} P\left(X^i \ / \ \lambda\right)$$

où λ est l'ensemble des paramètres du modèle, et $L\left(X^1, X^2, ..., X^r\right)$ est appelée la vraisemblance du modèle.

3.4.3 Les méthodes d'estimation des modèles de CMC

Lors de l'estimation des Chaînes de Markov cachées, deux cas peuvent se présenter :

(i) la séquence d'états est connue

Dans ce cas, les données sont complètes et l'application de la méthode du maximum de vraisemblance pour l'estimation des paramètres du modèle s'avère évidente.

(ii) la séquence d'états est inconnue, et notre modèle est en manque de données ; on fait ainsi recours à des méthodes de simulation de ces dernières et appliquer ensuite l'estimation par maximum de vraisemblance des paramètres du modèle.

(i) la séquence d'états est connue

lorsque la séquence d'états est connue, le calcul des valeurs des paramètres devient facile; il suffit d'appliquer les formules des estimateurs de mimum de vraisemblance [1], par le calcul du nombre de fois où chaque transition ou émission sont produites. Soient : A_{ij} : le nombre de fois que le système est à l'état j précédé par l'état i, et $B_j\left(x_t\right)$: le nombre de fois que l'alphabet k est émis par l'état j.

L'estimateur par maximum de vraisemblance des probabilité de transition et d'émissions donne respectivement :

$$a_{ij} = \frac{A_{ij}}{A_{i.}} = \frac{A_{ij}}{\sum_k A_{ik}} \tag{3.4.4}$$

et

$$b_j\left(x_t\right) = \frac{B_j\left(x_t\right)}{B_j\left(.\right)} = \frac{B_j\left(x_t\right)}{\sum_{y_t} B_j\left(y_t\right)} \tag{3.4.5}$$

- $A_{i.} = \sum_k A_{ik}$ est le nombre de fois que le système se trouve à l'état $S_t = i$, pour $i = 1, ..., N$, et $t = 1, ..., L-1$.

- $B_j\left(.\right) + \sum_{y_t} B_j\left(y_t\right)$ est le nombre de toutes les transitions à partir de l'état j pour $j = 1, ..., N$, et $t = 1, ..., L-1$.

Dans le cas de manque d'information, l'application des formules (3.4.4) et (3.4.5) est impossible. En effet, si un état i n'est jamais utilisé dans la séquence d'états, alors les équations d'estimation (3.4.4) et (3.4.5) sont indéfinies pour cet état. Car le nombre A_{ij} de transitions à partir de cet état vers n'importe quel état j est nul et par conséquent la somme

$\sum_k A_{ik}$ est aussi nulle; ce qui implique une indétermination dans le calcul de a_{ij}. Pour la même raison, nous obtenons une indétermination dans le calcul de $B_j(.)$.

Durbin et al en 1998 ont proposé d'ajuster à ces valeurs des pseudo nombres prédéterminés qui sont respectivement notés r_{ij} et $r_j(k)$, qui sont définis comme étant le biais à priori des valeurs de probabilité, et qui sont toujours positifs. Ces pseudo nombres sont tels que :

$$\begin{aligned} \text{Nouveau} \quad A_{ij} &= A_{ij} + r_{ij}. \\ &\text{et} \\ \text{Nouveau} \quad B_j(.) &= B_j(.) + r_j(k) \end{aligned}$$

Ces nombres r_{ij} et $r_j(k)$ sont de nature probabilistes et ils reflètent le biais a priori des valeurs de probabilités. De plus, ils sont interprétés comme les paramètres de la distribution a priori de Dirichlet pour chaque état.

L'estimation des Chaînes de Markov cachées étant basée essentiellement sur l'estimation par maximum de vraisemblance, nous discutons les équations de l'estimation par maximum de vraisemblance (MLE) et ses propriétés dans la prochaine section.

(ii) la séquence d'états est inconnue

Dans cette situation, l'application directe des estimateurs du maximum de vraisemblance n'est pas pratique, alors nous essayons d'approcher cet estimateur par les estimations itératives à partir d'un modèle initial choisi arbitrairement et d'appliquer les estimateurs du maximum de vraisemblance à chaque itération. Cette méthode est connue sous le nom de " l'algorithme EM " (Exeptation- Minimization) qui a été introduite par [28] pour l'estimation des modèles à données incomplètes. Nous notons que [15] ont été les premiers à prouver que l'information des paramètres de la Chaîne de Markov caché CMC par le maximum de vraisemblance augmente la vraisemblance des observations.

L'algorithme de Baum-welch appelé aussi algorithme "forward-backward " est un cas spécial de l'algorithme EM, il permet d'estimer les probabilités dans les Chaînes de Markov cachées où les données manquantes sont les éléments de la séquence d'états cachés.

Par ailleurs, il existe d'autres méthodes itératives qui sont soit des versions stochastiques de l'algorithme EM et qui approchent l'estimateur du maximum de vraisemblance tel que l'algorithme SEM et l'algorithme EM de Gibbs connus sous le nom de méthodes de Monte Carlo par les chaînes de Markov ; soit des méthodes basées sur l'estimation Bayesienne.

Nous notons que le cas des données incomplètes a été largement traité : Hartley [38] a donné une simplification du calcul des estimateurs de maximum de vraisemblance à partir des données incomplètes. Dempster [28] ont appliqué l'algorithme EM pour calculer les estimateurs du maximum de vraisemblance dans le cas de données incomplètes, et ont montré le caractère monotone de la fonction vraisemblance ainsi que la convergence de l'algorithme.

Dans le traitement des chaînes de Markov cachées, nous distinguons les méthodes basées sur l'estimation de maximum de vraisemblance (les algorithme EM, SEM, EM à la Gibbs) et la méthode basé sur l'estimation Bayesienne.

3.4.4 L'algorithme EM

L'algorithme EM, introduit pour la première fois par Dempster [28], est un algorithme itératif qui permet d'approcher l'Estimateur du maximum de vraisemblance du modèle à données incomplètes. Il consiste à remplacer la vraisemblance $P(X,\, S\, /\lambda)$ par son espérance conditionnelle sachant la séquence observé X et la valeur courante du paramètre $\lambda^{(m)}$ qui est à cette étape candidate au maximum ; cette espérance conditionnelle est appelée la fonction auxiliaire $Q\left(\lambda/\lambda^{(m)}\right)$. Elle s'écrit :

$$Q\left(\lambda/\lambda^{(m)}\right) = E\left(\ln P(X,\ S\ /\lambda^{(m)})\ /\ X,\lambda\right) = \sum_{s\in S} \ln P(X,\ S\ /\lambda) P(S\ /\ X,\lambda^{(m)}) \quad (3.4.6)$$

Et est maximisé en λ pour obtenir une nouvelle valeur du paramètre. Donc, partant d'un modèle initial $\lambda^{(0)}$ et connaissant les valeurs du paramètres actuels $\lambda^{(m)}$ à l'itération (m) , l'algorithme EM alterne deux étapes :

- La première étape qui est l'étape E (Estimation) consistante à calculer la valeur de la fonction $Q(\lambda/\lambda^m)$;

- La deuxième étape qui est l'étape M (Maximization) maximisant en λ la fonction auxiliaire $Q(\lambda/\lambda^m)$, c'est à dire choisir

$$\lambda^{m+1} = \arg\max_{\lambda} Q(\lambda/\lambda^m)$$

À cause du caractère incomplet du modèle $P(X/\lambda)$, les méthodes d'estimation sont difficiles à calculer ; nous essayons alors d'augmenter les donnés en donnant des valeurs aux états cachés et enfin calculer la vraisemblance du modèle complet $P(X,\, S\, /\lambda)$ dont le calcul devient facile après avoir éliminer tous les termes possibles des états cachés.

L'idée consiste à alterner deux étapes :

1. Attribuer des valeurs aux états cachés $S^{(m+1)} = \left(S_1^{(m+1)},\ ...,\ S_t^{(m+1)},\ ...,\ S_L^{(m+1)}\right)$ sachant l'estimation courante du paramètre du modèle λ^m et la séquence d'observations X .

2. Etant donné $S^{(m+1)}$, ré estimer λ^{m+1} à partir de λ^m sur la base de la log vraisemblance des données complètes $\ln P(X,\, S^{(m+1)}\, /\lambda^{(m)})$

Notons que la restauration de la séquence d'états cachés se fait en appliquant la loi conditionnelle

$$P(S/X,\lambda) = \frac{P(S,X/\lambda)}{P(X/\lambda)}$$

L'algorithme EM consiste à remplacer la vraisemblance $P(X,\ S\ /\ \lambda)$ par son espérance conditionnelle sachant la séquence d'observations et la valeur courante du paramètre $\lambda^{(m)}$. Cette espérance conditionnelle est la fonction auxiliaire Q .

Soient $X = X_1\ ,\ X_2,\ ..., X_L$ la séquence d'observations, $S = S_1\ ,\ S_2,\ ..., S_L$ la séquence d'états et $\lambda = (\pi,\ A,\ B)$ l'ensemble des paramètre du modèle.

L'estimateur du maximum de vraisemblance $\widehat{\lambda}$ s'obtient en maximisant la log vraisemblance des données complètes $\ln P(X,\ S\ /\ \lambda)$ pour tout λ. La solution originale de l'estimation du vrai modèle dans le sens du maximum de vraisemblance c'est à dire trouver $\widehat{\lambda}$ maximisant $P_{\widehat{\lambda}}(X)$, a apparu dans [15].

En effet, on cherche à calculer l'estimateur MLE de la fonction :

$$\ln P(X\ /\ \lambda) = \ln \sum_{S} P(X,\ S\ /\ \lambda)$$

Supposons un modèle actuel $\lambda^{(m)}$ et cherchons à estimer un nouveau modèle $\lambda^{(m+1)}$ qui soit meilleur, soit la vraisemblance du modèle

$$L\left(X \ / \ \lambda\right) = \ln P\left(X \ / \ \lambda\right)$$

du fait que

$$P\left(X \ / \ \lambda\right) = \frac{P\left(X,\ S \ / \ \lambda\right)}{P\left(S \ / \ X \ , \ \lambda\right)}$$

alors on peut écrire :

$$\ln P\left(X \ / \ \lambda\right) = \ln P\left(X,\ S \ / \ \lambda\right) - P\left(S \ / \ X \ , \ \lambda\right) \tag{3.4.7}$$

En multipliant cette équation par $P\left(S \ / \ X \ , \ \lambda\right)$ et en sommant sur l'ensemble des états cachés nous obtenons à chaque itération m :

$$\begin{aligned}\sum_S P\left(S \ / \ X \ , \ \lambda^{(m)}\right) \ln P\left(X \ / \ \lambda\right) &= \sum_S P\left(S \ / \ X \ , \ \lambda^{(m)}\right) \ln P\left(X,\ S \ / \ \lambda\right) \\ &\quad - \sum_S P\left(S \ / \ X \ , \ \lambda^{(m)}\right) \ln P\left(S \ / \ X \ , \ \lambda\right) \\ &= \ln P\left(X \ / \ \lambda\right)\end{aligned}$$

On appelle le premier terme la fonction auxiliaire $Q\left(\lambda/\lambda^{(m)}\right)$ tel que :

$$Q\left(\lambda/\lambda^{(m)}\right) = \sum_S P\left(S \ / \ X \ , \ \lambda^{(m)}\right) \ln P\left(X,\ S \ / \ \lambda\right) \tag{3.4.8}$$

le deuxième terme noté $H\left(\lambda/\lambda^{(m)}\right)$ et l'entropie et est égale à

$$H\left(\lambda/\lambda^{(m)}\right) = \sum_S P\left(S \ / \ X \ , \ \lambda^{(m)}\right) \ln P\left(S \ / \ X \ , \ \lambda\right) \tag{3.4.9}$$

on peut donc écrire

$$L\left(X \ / \ \lambda\right) = \ln P\left(X \ / \ \lambda\right) = Q\left(\lambda/\lambda^{(m)}\right) - H\left(\lambda/\lambda^{(m)}\right) \tag{3.4.10}$$

La fonction auxiliaire $Q\left(\lambda/\lambda^{(m)}\right)$ est d'une grande importance dans le calcul de l'estimateur obtenu par l'algorithme EM. Statistiquement cette fonction représente la moyenne de $\ln P\left(X,\ S \ / \ \lambda\right)$ sur la distribution de S connaissant la séquence d'observations et le modèle courant $\lambda^{(m)}$.

Q parait plus compliquée que P, mais il est plus facile de travailler avec Q. On démontre [54] que

$$Q\left(\widehat{\lambda}/\lambda^{(m)}\right) > Q\left(\lambda^{(m)}/\lambda^{(m)}\right) \Rightarrow P_{\widehat{\lambda}}\left(X\right) > P_{\lambda^{(m)}}\left(X\right)$$

Nous cherchons à trouver le modèle λ qui rend la log vraisemblance $\ln P\left(X \ / \ \lambda\right)$ plus grande que celle du modèle actuel ou :

$$\ln P\left(X \ / \ \lambda\right) > \ln P\left(X \ / \ \lambda^{(m)}\right)$$

nous calculons alors :

$$\begin{aligned} \ln P\left(X \ / \ \lambda\right) - \ln P\left(X \ / \ \lambda^{(m)}\right) &= Q\left(\lambda/\lambda^{(m)}\right) - Q\left(\lambda^{(m)}/\lambda^{(m)}\right) \\ &\quad + \sum_{S} P\left(S \ / \ X \ , \ \lambda^{(m)}\right) \frac{P\left(S \ / \ X \ , \ \lambda^{(m)}\right)}{P\left(S \ / \ X \ , \ \lambda\right)} \end{aligned}$$

Le deuxième terme est l'entropie relative [89] de $P\left(S \ / \ X \ , \ \lambda^{(m)}\right)$ à $P\left(S \ / \ X \ , \ \lambda\right)$ et elle est toujours positive d'où :

$$\ln P\left(X \ / \ \lambda\right) - \ln P\left(X \ / \ \lambda^{(m)}\right) > Q\left(\lambda/\lambda^{(m)}\right) - Q\left(\lambda^{(m)}/\lambda^{(m)}\right) \tag{3.4.11}$$

alors tant que $\lambda \neq \lambda^{(m)}$ l'inégalité reste stricte.

Pour rendre la différence maximale, nous devons maximiser $Q\left(\lambda/\lambda^{(m)}\right)$ et donc, nous devons choisir λ qui vérifie l'équation :

$$\lambda^{(m+1)} = \arg\max_{\lambda} Q\left(\lambda/\lambda^{(m)}\right) \tag{3.4.12}$$

Baum et al [17] ont montré que pour une large classe de probabilités fonctions du paramètres λ qu'on note par $P\left(\lambda\right)$, l'utilisation de la technique du maximum de vraisemblance dans l'estimation du modèle de Markov caché donne :

Théorème 3.4.1

Si $Q\left(\widehat{\lambda}/\lambda\right) \geq Q\left(\lambda/\lambda\right)$ *alors* $P\left(\widehat{\lambda}\right) \geq P\left(\lambda\right)$. *L'inégalité est stricte sauf si* $P\left(X/\widehat{\lambda}\right) = P\left(X/\lambda\right)$ *p.p sur le domaine de* X.

Ensuite, on montre que λ est un point critique de P si et seulement s'il est un point critique de Q fonction de $\widehat{\lambda}$:

$$\frac{\partial P_{\lambda}}{\partial \lambda_u/\lambda} = \frac{\partial Q\left(\widehat{\lambda}/\lambda^{(m)}\right)}{\partial \widehat{\lambda}_u/\widehat{\lambda} = \lambda}$$

pour chaque coordonnée de λ_u, $u = 1, ..., d$ de l'ensemble de λ

Enfin, Poritz [71] a monté que pour une large classe de modèles, Q fonction de λ a un seul point critique et ce point est l'unique maximum global.

L'idée est d'augmenter les paramètres du modèle à l'aide de la fonction Q commençant par un modèle initial $\lambda^{(0)}$ et de trouver les re-estimateurs de Baum- Welch $\widehat{\lambda}$ en maximisant la fonction Q .

L'algorithme EM peut être résumé comme suit :

- L'étape E : calculer la fonction $Q\left(\widehat{\lambda}/\lambda^{(m)}\right)$;
- L'étape M : maximiser $Q\left(\widehat{\lambda}/\lambda^{(m)}\right)$ sur l'ensemble des λ, autrement dit choisir

$$\lambda^{(m+1)} = \arg\max_{\lambda} Q\left(\lambda/\lambda^{(m)}\right)$$

Nous avons démontré d'après l'équation (3.4.11) que $\ln P\left(X/\lambda\right)$ croît à chaque itération quand $\lambda \neq \lambda^m$. Ce qui permet à la procédure d'atteindre un maximum local (et parfois global lorsque la taille de l'échantillon n augmente).

Prenons $\lambda^{(m)}$ comme étant le nouveau modèle initial et répéter le processus ; un seul des deux résultats se réalise :

- soit $P_{\lambda^{m+1}}\left(X\right) > P_{\lambda^m}\left(X\right)$
- soit λ^m est un point critique de $P_{\lambda}\left(X\right)$.

3.4.5 Propriétés de l'estimateur de l'algorithme EM

Nous avons vu que l'algorithme EM est une procédure itérative qui approche l'estimateur de maximum de vraisemblance pour les modèles à données incomplètes voire les Chaînes de Markov cachées où les données manquantes sont les états cachés. En effet, pour estimer les paramètre du modèle, on maximise l'espérance conditionnelle de la log vraisemblance du modèle complet qu'on obtient dans le calcul de l'étape E du même algorithme :

$\lambda^{(m+1)} \in \arg\max_{\lambda\in\Gamma} Q\left(\lambda/\lambda^{(m)}\right)$ $m = 0, 1, \ldots$ .Ainsi partant d'un modèle initial $\lambda^{(0)}$ et en appliquant les deux étapes de l'algorithme, on génère à chaque itération une suite de paramètres $\left(\lambda^{(m)}\right)_{m\geq 0} : \lambda^{(0)} \to \lambda^{(1)} \to \ldots\lambda^{(m)} \to \lambda^{(m+1)} \to \ldots$

Ainsi, la vraisemblance du modèle croît à chaque itération. Dans cette partie, nous étudions la convergence de cette vraisemblance.

Redner et Walker [75] ont établit le théorème suivant :

Théorème 3.4.2 *Soit $\left(\lambda^{(m)}\right)_{m\geq 0}$ une suite générée par l'algorithme EM dans Γ pour une valeur initiale $\lambda^{(0)} \in \Gamma$; la log vraisemblance des données incomplètes $L\left(X/\lambda^{(m)}\right)$ converge d'une manière monotone vers une limite L^* (finie ou non finie) , et si $\mathcal{L}$ est l'ensemble des points limites de $\left(\lambda^{(m)}\right)_{m\geq 0}$ dans Γ alors on vérifie les propriétés suivantes :*

1. *$\mathcal{L}$ est un ensemble fermé de Γ;*

2. *si $\left(\lambda^{(m)}\right)_{m\geq 0}$ est contenue dans un sous ensemble compact de Γ. , alors $\mathcal{L}$ est compact.*

3. *si $\left(\lambda^{(m)}\right)_{m\geq 0}$ est contenue dans un sous ensemble compact de Γ_0 et si pour une norme $||.||$ sur Γ, $\left|\left|\lambda^{(m+1)} - \lambda^{(m)}\right|\right| \to_{n\to\infty} 0$, alors $\mathcal{L}$ est connexe et compact .*

4. *si $L(X/\lambda)$ est continue dans Γ et $\mathcal{L} \neq \emptyset$ alors L^* est finie et $L\left(X/\widehat{\lambda}\right) = L^*$ pour tout point limite $\widehat{\lambda} \in \mathcal{L}$.*

5. *si $Q\left(\lambda/\lambda^{(m)}\right)$ et $H\left(\lambda/\lambda^{(m)}\right)$ sont continues en leurs deux variables λ et $\lambda^{(m)}$ dans Γ et sont différentiables en λ au point $\lambda = \lambda^{(m)} = \widehat{\lambda} \in \mathcal{L}$; alors $L(X/\lambda)$ est différentiables au point $\lambda = \widehat{\lambda}$ et les équations de vraisemblance $D_\lambda L(X/\lambda) = 0$ sont satisfaites par $\lambda = \widehat{\lambda}$.*

Notre modèle $\lambda(\pi_i,\ a_{ij},\ b_i(k))$ appartient à l'espace des paramètres :

$$\Gamma = \left\{ \begin{array}{c} \lambda \in IR^d \ / \ \forall 1 \leq i, j \leq N,\ \forall 1 \leq k \leq M, 0 \leq \pi_i \leq 1 \\ \sum_{i=1}^N \pi_i = 1,\ 0 \leq a_{ij} \leq 1, \sum_{i=1}^N a_{ij} = 1,\ 0 \leq b_j(k) \leq 1, \sum_{k=1}^M b_j(k = y_t) = 1 \end{array} \right\}$$

$\lambda^{(0)}$: l'ensemble des paramètres initiale doit avoir toutes ses composantes strictement positives. Ces paramètres sont ré estimés par les formules (3.4.21) , (3.4.22) ,(3.4.23) , et doivent vérifier à chaque itération les conditions de stricte positivité ; c'est à dire à l'itération m de l'algorithme EM $a_{ij}^{(m)} > 0,\ b_j(k) > 0$, $\pi_i > 0$; on admet de plus que $\sum_{i=1}^N \pi_i^{(m)} = 1$.

Sous ces conditions l'ensemble des points limites de la suite $\left(\lambda^{(m)}\right)_{m>0}$ générée par l'algorithme EM est un sous ensemble compact non vide de $\overline{\Gamma}$.

Théorème 3.4.3 *Soit* $\lambda^{(0)} \in \Gamma^0$ *et* $\left(\lambda^{(m)}\right)_{m>0}$ *la suite générée par l'algorithme EM pour une Chaîne de Markov caché. si* $\mathcal{L}$ *désigne l'ensemble des points limites de* $\left(\lambda^{(m)}\right)_{m>0}$ *alors :*

1. $L\left(X/\lambda^{(m)}\right)$ *converge quand* $m \to \infty$ *vers une limite finie* L^*.

2. pour tout point limite $\widehat{\lambda} \in \mathcal{L}$; $L\left(X/\widehat{\lambda}\right) = L^*$ *et* $D_\lambda L\left(X/\widehat{\lambda}\right) = 0$; ($\widehat{\lambda}$ *est un point stationnaire de* $L(X/\lambda)$).

3. $\mathcal{L}$ *est connexe est compact.*

Convergences vers la solution consistante des équations de maximum de vraisemblance :

L'existence d'une solution consistante des équations de vraisemblance pour les Chaînes de Markov cachées est établie sous les conditions suivantes :

- La vraie valeur du paramètre λ^* appartient à l'espace Γ_δ définie par

$$\Gamma_\delta = \{\lambda \in \Gamma \ / \ \forall i = 1, \ ..., \ d, \ \lambda_i \geq \delta, \ \delta > 0\}$$

- La matrice

$$\sigma_\lambda = -D^2 H_{\lambda^*}(\lambda) = -D^2 E_{\lambda^*}(\ln P(X_0 \ / \ X_{-1}, \ X_{-2}, \ ..., \ \lambda))$$

Est définie positive en λ^*.

D'après le théorème 3.4.1 tout ouvert contenant la classe d'équivalence de λ^* contient pour tout n assez grand celle de $\widehat{\lambda}_n$ et si de plus Γ'_δ est un compact de Γ_δ, alors pour n assez grand, $\widehat{\lambda}_n$ est dans l'intérieur de Γ'_δ et est l'unique solution des équations de vraisemblance dans Γ'_δ.

Nous venons de vérifier que l'ensemble des points limites de la suite $\left(\lambda^{(m)}\right)_{m>0}$ générée par l'algorithme EM est inclus dans un sous ensemble compact de $\overline{\Gamma}$ qui n'est pas automatiquement inclus dans Γ'_δ . pour pouvoir appliquer le théorème précédent, il faut donc s'assurer que la suite $\left(\lambda^{(m)}\right)_{m>0}$ ne sort pas de Γ'_δ.

Théorème 3.4.4 *Soit* Γ'_δ *un compact de* Γ_δ *contenant la classe d'équivalence* M_{λ^*} *du vrai paramètres dans son intérieur. Si pour* $\lambda^{(0)} \in \Gamma'_\delta$; $\left(\lambda^{(m)}\right)_{m>0}$ *désigne la suite générée par l'algorithme EM dans* Γ'_δ *pour une Chaîne de Markov caché, alors pour* n *assez*

grand, l'unique estimateur du maximum de vraisemblance $\widehat{\lambda}_n$ *est bien défini dans* Γ'_δ *et est consistant.*

$$\lambda^{(m)} \to \widehat{\lambda}_n$$

Dés que $\lambda^{(0)}$ *est suffisamment proche de* $\widehat{\lambda}_n$.

D'après Muri (1997), le théorème 3.4.2 montre l'existence de points limites établis par l'algorithme EM ; ensuite dans le théorème(3.4.3 , 3.4.4), il est montré la convergence de l'estimateur obtenue par l'algorithme EM $\left(\lambda^{(m)}\right)_{m>0}$ vers la solution consistante des équations de maximum de vraisemblance. Cependant, le dernier théorème n'est pas valable pour toute suite générée par l'algorithme EM puisqu'il est nécessaire de vérifier que la suite $\left(\lambda^{(m)}\right)_{m>0}$ ne sort pas de Γ'_δ.

3.4.6 L'apprentissage de Baum-Welch

Lorsque la séquence d'états est inconnue, nous ne pouvons pas estimer les paramètres directement par la méthode du maximum de vraisemblance ML. Dans ce cas, des procédures itératives sont utilisées et elles consistent à augmenter les données du modèle pour ensuite appliquer les formules de ré estimation sur les modèles supposés ainsi à données complètes. Parmi ces procédures itératives, la méthode la plus manipulée est l'algorithme Baum-welch ou l'algorithme forward-backward qui n'est qu'un cas particulier de l'algorithme EM.

[17] ont utilisé l'algorithme EM comme méthode itérative qui consiste à maximiser la fonction auxiliaire $Q(\lambda/\lambda^m)$ à l'étape E ont montré que cette maximisation augmente la log vraisemblance $\ln P(X/\lambda)$ à chaque itération de l'algorithme. Puis, l'algorithme Baum-Welch consiste à estimer les paramètres A_{ij}, et $B_j(k)$ en considérant les chemins les plus probables des séquences d'apprentissage et les valeurs courantes des probabilités a_{ij}, et $b_j(k)$, pour appliquer les estimateurs du maximum de vraisemblance précédemment cité pour calculer les nouvelles valeurs de a_{ij}, et $b_j(k)$.

Cette itération continue jusqu'à vérification d'un critère d'arrêt prédéterminé qui est :

- Soit une variation non significative de la vraisemblance du modèle au cours des itérations ; cette variation étant mesurée par

$$\varepsilon = \left| L\left(X/\lambda^{(m)}\right) - L\left(X/\lambda^{(m-1)}\right) \right|$$

de telle sorte que ε prend une valeur d'une grandeur relative à la valeur de $L(X/\lambda)$. Cette stabilité significative des valeurs des paramètres du modèle à savoir les paramètres ré-estimer $a_{ij}^{(m)}$, et $b_j^{(m)}(k)$.

- Soit le nombre d'itérations fixé au départ.

Nous montrons plus tard que la vraisemblance du modèle croit à chaque itération et par conséquent, le processus converge vers un maximum local. Cependant, il existe plusieurs maximaux locaux qui dépendent tous des valeurs initiales des paramètres.

L'algorithme Baum-welch calcul A_{ij}, et $B_j(k)$ comme étant le nombre espéré de fois que chaque transition ou émission est réalisée sachant la séquence d'apprentissage ; pour cela nous utilisons les procédures Forward et Backward dans le calcule des probabilités à posteriori.

$$a_{ij} = P(S_t = i,\ S_{t+1} = j \ /\ X,\ \lambda) = \frac{\alpha_t(i)\,\beta_{t+1}(j)\,b_j(x_{t+1})}{P(X\ /\ \lambda)} \tag{3.4.13}$$

À partir de cette équation, le nombre espéré de fois que la probabilité a_{ij} est utilisée, peut être obtenu en sommant sur tous les états d'une séquence et sur toutes les séquences d'apprentissage la quantité a_{ij}^s de la seule séquence s, on obtient

$$A_{ij} = \sum_{s=1}^{r} \frac{1}{P(X^s\ /\ \lambda)} \sum_{t=1}^{L} \alpha_t^s(i)\,\beta_{t+1}^s(j)\,b_j\left(x_{t+1}^s\right) \tag{3.4.14}$$

Tel que :

- $\alpha_t^s(i)$ est la variable Forward $\alpha_t(i)$ calculée pour la séquence s,

- $\beta_{t+1}^s(j)$ est la variable Backward $\beta_{t+1}(j)$ calculée pour la séquence s,

- $b_j\left(x_{t+1}^s\right)$ est la probabilité d'émission de l'observation x_{t+1} à partir de l'état j pour la séquence s.

- L est la taille de la séquence d'apprentissage,

- r es t le nombre de séquences dans le corpus d'apprentissage.

De même pour une séquence donnée, la probabilité d'émettre au temps t une observation $x_t = k$ est donné par la formule suivante :

$$b_j(k) = P\left(x_{t\{t\ /\ S_t\ =\ j\}} = k\ /\ X,\ \lambda\right) = \frac{\alpha_t(i)\,\beta_t(i)\,b_{j\{t\ /\ x_t\ =\ k\}}}{P(X\ )} \tag{3.4.15}$$

Ainsi le nombre de fois que la lettre k soit émise par l'états j est :

$$B_j(k) = \sum_{s=1}^{r} \frac{1}{P(X^s \ / \ \lambda)} \sum_{\{t \ / \ x_t^s = k\}} \alpha_t^s(i)\, \beta_t^s(i) \tag{3.4.16}$$

Nous remarquons que la somme du produit de deux variables Forward et Backward se fait seulement sur les positions (les états) où le symbole k est émis tout au long de la séquence.

En calculant ces estimations, les probabilités du nouveau modèle sont obtenues en appliquant les équations de maximum de vraisemblance :

$$a_{ij} = \frac{A_{ij}}{A_{i.}} \qquad \text{et} \quad b_j(x_t) = \frac{B_j(x_t)}{B_j(.)}$$

On continue l'itération jusqu'au moment où la vraisemblance ne change pas de manière significative. Dans la pratique, le critère d'arrêt est le nombre d'itérations fixé au préalable.

Ainsi, l'algorithme Baum-welch consiste en ces trois étapes suivantes :

1. Initialisation

prendre des paramètres arbitraire pour le premier modèle.

2. Récurrence

Calculer la variable Forward $\alpha_t(i)$ **pour chaque séquence** s,

Calculer la variable Backward $\beta_t(i)$ **pour chaque séquence** s,

Calculer la log vraisemblance du modèle pour chaque séquence s.

3. Arrêt

arrêter quand la logvraisemblance est inférieur à certain seuil prédéterminer ou quand le nombre fixe d'itérations est atteint .

3.4.7 Justification des estimateurs de Baum-welch

L' 'algorithme de Baum-welch cherche à maximiser la vraisemblance :

$$\ln P(X \ / \ \lambda) = \sum_{S} P(X, \ S \ / \ \lambda)$$

Dans le cas des données manquantes, la fonction Q est :

$$Q\left(\lambda/\lambda^{(m)}\right) = \sum_{S} P(S \ / \ X, \lambda) \ln P(X, S \ /\lambda^{(m)})$$

Pour un chemin donné, chaque paramètre du modèle apparaît un certain nombre de fois dans la probabilité $P(X)$.

Soient

- $A_{ij}(s)$: le nombre de transitions de l'état i à l'état j pour chaque séquence d'états s,

- $B_j(k,\ s)$: le nombre d'émission du symbole k via l'état j pour chaque séquence d'états s,

- $\pi_i(s)$: le nombre de fois que séquence S commence par l'états i,

Etant donné l'ensemble des chemins, la log vraisemblance de $P(X,\ S\ /\lambda)$ peut s'écrire :

$$\ln P(X,\ S\ /\lambda) = \ln\left(P(s_1)b_{s_1}(x_1)\left(\prod_{t=2}^{L} a_{ij}b_j(x_t)\right)\right)$$

que l'on peut écrire sous la forme :

$$\begin{aligned}\ln P(X,\ S\ /\lambda) \ &= \ \ln\left(\left(\prod_{i=1}^{N}\pi_i^{1\{s_1=i\}}\right)\left(b_j(k)^{1\{s_t=i,\ x_t=k\}}\right)\right)\\ &\left(\left(\prod_{t=2}^{L}\prod_{i=1}^{N-1}\prod_{j=2}^{N} a_{ij}^{1\{s_{t-1}=i,\ s_t=j\}}\right)\right)\end{aligned}$$

ce qui permet de développer :

$$\ln P(X,\ S\ /\lambda) = \ln\left(\begin{array}{c}\left(\prod_{i=1}^{N}\pi_i^{1\{s_1=i\}}\right)\left(\prod_{i=i}^{N}\prod_{k=1}^{M} b_i(k)^{\sum_{t=1}^{L}1\{s_t=i,\ x_t=k\}}\right)\\ \left(\prod_{i=1}^{N-1}\prod_{j=2}^{N} a_{ij}^{\sum_{t=2}^{L}1\{s_{t-1}=i,\ s_t=j\}}\right)\end{array}\right)$$

D'où pour chaque séquence la logvraisemblance du modèle est :

$$\ln P(X,\ S\ /\lambda) = \ln\left(\prod_{i=1}^{n-1}\prod_{j=2}^{n} a_{ij}^{A_{ij}(s)}\prod_{i=i}^{n}\prod_{k=1}^{m} b_i(k)^{B_j(k,\ s)}\prod_{j=1}^{n}\pi_i^{\pi_i(s)}\right)$$

où :

- $A_{ij}(s) = \sum_{t=2}^{L} 1\{s_{t-1} = i,\ s_t = j\}$ est le nombre de transitions de l'état i à l'état j pour la séquence d'états S,

- $B_j(k,\ s) = \sum_{t=1}^{L} 1\{s_t = i,\ x_t = k\}$ est le nombre de fois que l'alphabet k apparaît à partir de l'état j pour la séquence S.

Puisque nous considérons la chaîne stationnaire, nous allons ignorer le terme de la probabilité initiale dans le calcul de $Q\left(\lambda/\lambda^{(m)}\right)$, ce qui permet d'établir l'égalité suivante :

$$Q\left(\lambda/\lambda^{(m)}\right) = \sum_{S} P(S \ / \ X \ , \lambda^m) \left(\sum_{i=1}^{n-1} \sum_{j=2}^{n} A_{ij}\,(s) \ln a_{ij} \sum_{j=i}^{n} \sum_{k=1}^{m} B_j\,(k,\ s) \ln b_j\,(k) \right) \tag{3.4.17}$$

Nous remarquons que les valeurs estimées A_{ij} et $B_j\,(k)$ dans les équations (3.4.14) et (3.4.16) de l'algorithme Baum-Welch peuvent être vues comme les espérances de $A_{ij}\,(s)$ et $B_j\,(k,\ s)$ sur $P(S \ / \ X \ , \lambda^m)$; et on peut écrire :

$$\begin{aligned} A_{ij} &= \sum_{S} P(S \ / \ X \ , \lambda^m) A_{ij}\,(s) \\ B_j\,(k) &= \sum_{S} P(S \ / \ X \ , \lambda^m) B_j\,(k,\ s) \end{aligned}$$

Si on somme l'ensemble des séquences d'états l'équation (3.4.17) , nous pouvons écrire $Q\left(\lambda/\lambda^{(m)}\right)$ comme suit :

$$\begin{aligned} Q\left(\lambda/\lambda^{(m)}\right) &= \sum_{S} P(S \ / \ X \ , \lambda^m) \ln \left(\prod_{i=1}^{n-1} \prod_{j=2}^{n} a_{ij}^{A_{ij}(s)} \prod_{i=i}^{n} \prod_{k=1}^{m} b_i\,(k)^{B_j(k,\ s)} \right) \\ &= \sum_{i=1}^{n-1} \sum_{j=2}^{n} \sum_{S} P(S \ / \ X \ , \lambda^m) A_{ij}\,(s) \ln a_{ij} \\ &\quad + \sum_{i=i}^{n} \sum_{k=1}^{m} \sum_{S} P(S \ / \ X \ , \lambda^m) B_j\,(k,\ s) \ln b_i\,(k) \\ &= \sum_{i=1}^{n-1} \sum_{j=2}^{n} A_{ij}\,(s) \ln a_{ij} + \sum_{i=i}^{n} \sum_{k=1}^{m} B_j\,(k,\ s) \ln b_i\,(k) \end{aligned}$$

Nous allons montrer comment l'estimation par maximum de vraisemblance de a_{ij} et de $b_i\,(k)$ maximise $Q\left(\lambda/\lambda^{(m)}\right)$.

Soit le modèle $\widehat{\lambda}$ dont les paramètres constituent les estimateurs par maximum de vraisemblance :

$$\widehat{a}_{ij} = \frac{A_{ij}}{\sum_l A_{il}}$$

d'où :

$$A_{ij} = \widehat{a}_{ij} \sum_{l} A_{il}$$

et

$$\widehat{b}_j(x_t) = \frac{B_j(k)}{\sum_{k'} B_j(k')}$$

et on a

$$B_j(k) = \widehat{b}_j(x_t) \sum_{k'} B_j\left(k'\right)$$

Nous cherchons à calculer la différence : $Q\left(\widehat{\lambda}/\lambda^{(m)}\right) - Q\left(\lambda/\lambda^{(m)}\right)$ sachant que :

$$Q\left(\lambda/\lambda^{(m)}\right) = \sum_{i=1}^{n-1}\sum_{j=2}^{n} A_{ij} \ln \widehat{a}_{ij} + \sum_{j=i}^{n}\sum_{k=1}^{m} B_j(k) \ln \widehat{b}_j(k)$$

alors

$$Q\left(\widehat{\lambda}/\lambda^{(m)}\right) - Q\left(\lambda/\lambda^{(m)}\right) = \sum_{i=1}^{n-1}\sum_{j=2}^{n} A_{ij} \ln \frac{\widehat{a}_{ij}}{a_{ij}} + \sum_{j=i}^{n}\sum_{k=1}^{m} B_j(k) \ln \frac{\widehat{b}_j(k)}{b_j(k)}$$

Considérons l'équation terme par terme :

$$\sum_{i=1}^{n-1}\sum_{j=2}^{n} A_{ij} \ln \frac{\widehat{a}_{ij}}{a_{ij}} = \sum_{i=1}^{n-1}\left(\sum_{l} A_{il}\right)\sum_{j=2}^{n} \widehat{a}_{ij} \ln \frac{\widehat{a}_{ij}}{a_{ij}}$$

tel que $\sum_{j=2}^{n} \widehat{a}_{ij} \ln \frac{\widehat{a}_{ij}}{a_{ij}}$ est l'entropie de $\widehat{a}_{ij}$ relative à a_{ij} et elle est toujours strictement positive tant que $\widehat{a}_{ij} \neq a_{ij}$.

On procède de la même manière pour $b_j(k)$

$$\sum_{j=i}^{n}\sum_{k=1}^{m} B_j(k) \ln \frac{\widehat{b}_j(k)}{b_j(k)} = \sum_{j=i}^{n}\left(\sum_{k'} B_j\left(k'\right)\right)\sum_{k=1}^{m} \widehat{b}_j(k) \ln \frac{\widehat{b}_j(k)}{b_j(k)}$$

De même $\sum_{k=1}^{m} \widehat{b}_j(k) \ln \frac{\widehat{b}_j(k)}{b_j(k)}$ est l'entropie de $\widehat{b}_j(k)$ relative à $b_j(k)$ et donc toujours positive sauf si $\widehat{b}_j(k) \neq b_j(k)$, et on conclut que le maximum est atteint quand $\widehat{a}_{ij} \neq a_{ij}$ et $\widehat{b}_j(k) \neq b_j(k)$.

Ainsi, le maximum est atteint aux valeurs des paramètres obtenues par l'estimation du maximum de vraisemblance. Notons que Baum et Eagon [15] ont été les premiers à montrer qu'en appliquant une transformation de la probabilité de transition dans une chaîne de

Markov, cette transformation étant l'estimateur a posteriori de la probabilité de transition obtenu par maximum de vraisemblance : $T\left\{\frac{A_{ij}}{\sum A_{ij}}\right\}$, on augmente la vraisemblance de l'observation c'est à dire $P\left(T\left\{a_{ij}\right\}\right) \geq P\left(a_{ij}\right)$.

Dans l'algorithme de Baum-welch, l'étape E consiste à calculer les estimations de A_{ij} et $B_j\left(k\right)$ en utilisant les procédures forward-backward pour enfin évaluer la fonction Q, l'étape M consiste à injecter les valeurs de A_{ij} et $B_j\left(k\right)$ dans les formules de ré estimation de a_{ij} et $b_j\left(k\right)$ des équations (3.4.4) et (3.4.5) .

Dans la procédure de ré estimation du modèle on procède comme suit :

- Partir d'un modèle initial $\lambda^{(0)}\left(A^{(0)},\ B^{(0)}\right)$.

- Sachant l'estimation courante $\lambda^{(m)}\left(A^{(m)},\ B^{(m)}\right)$, attribuer une valeur aux éléments de la séquence d'états cachés $S^{(m+1)} = \left(S_1^{(m+1)},\ S_2^{(m+1)}, ...,\ S_n^{(m+1)}\right)$ à partir du modèle λ^m et de la séquence d'observation X.

- Sachant $S^{(m+1)}$ actualiser $\lambda^{(m)}$ en $\lambda^{(m+1)}$ sur la base de la log vraisemblance des données complètes $\ln\left(X, S^{\ (m+1)} / \ \lambda^m\right)$.

La restauration de la séquence cachée est fondée sur la loi conditionnelle sachant la séquence observée X et la valeur courante du paramètre λ, notée $P\left(S_t / X_1,\ X_2,\ ...,\ X_t, \lambda\ \right)$, et donnée par :

$$P\left(S_t / X_1,\ X_2,\ ...,\ X_t, \lambda\ \right) = \frac{P\left(S_t, X_{\ 1},\ X_2,\ ...,\ X_t\ /\ \lambda\ \right)}{P\left(X\ /\ \lambda\right)}$$

Nous allons décrire la procédure itérative basée sur le travail classique de Baum [17] pour le choix des paramètres du modèle. Commençons par définir les fréquences d'utilisation de l'ensemble d'apprentissage (paramètres du modèle).

Soit la quantité :

$$\zeta_t\left(i, j\ \right) = \frac{P\left(S_t = i, S_{t+1} = j\ /\ X,\ \lambda\ \right)}{P\left(X\ /\ \lambda\right)} \qquad t = 1,\ 2,\ ...,\ L-1$$

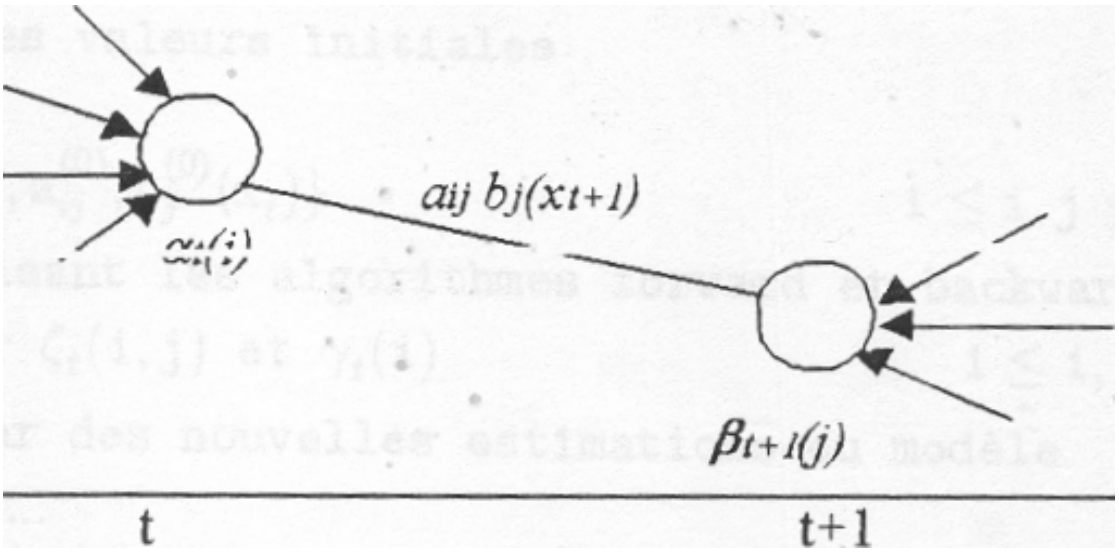
$\zeta_t\left(i, j\ \right)$ est la probabilité que le chemin à l'état i au temps t fait une transition à l'état j au temps $t+1$ étant donnés l'observation et le modèle, pondérée par la probabilité de la séquence d'observations connaissant le modèle.

On reconnaît que $P\left(X\ /\ \lambda\right)$ est un facteur de normalisation de $\zeta_t\left(i, j\ \right)$.

Nous pouvons écrire $\zeta_t(i,j\ )$ comme :

$$\zeta_t(i,j\ ) = \frac{\alpha_t(i)\, a_{ij} b_j(x_{t+1})\, \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)\, a_{ij} b_j(x_{t+1})\, \beta_{t+1}(j)} \tag{3.4.18}$$

$\alpha_t(i)$ considère les t premières observation, et au temps t le système est à l'état i, le terme $a_{ij}b_j(x_{t+1})$ comptabilise la probabilité d'une transition de l'état i à l'état j ainsi que la probabilité d'occurrence du symbole x_{t+1}, et le terme $\beta_{t+1}(j)$ prend en compte le reste de la séquence d'observations à partir de $t+1$; nous pouvons facilement montrer que $\forall j$, $\zeta_L(i,\ j) = \alpha_L(i)$. Le processus est expliqué par la figure ci-dessous [74].



Séquence partielle pour le calcul de l'evénement transition de l'état i au temps t à l'état j au temps $t+1$

On a définit précédemment $\gamma_t(i)$ comme étant la probabilité que le système est à l'état i au temps t, étant donnés la séquence d'observations et le modèle, alors on peut écrire l'expression suivante:

$$\gamma_t(i) = \sum_{j=1}^{N} \zeta_L(i,\ j)$$

On définit aussi

$$\gamma_i = \sum_{t=1}^{L-1} \gamma_t(i) = \sum_{j=1}^{N} \gamma_{ij}$$

et

$$\delta_{ij} = \sum_{t=1}^{L-1} \zeta_L(i,\ j) = \frac{\sum_{t=1}^{L-1} \alpha_t(i)\, a_{ij} b_j(x_{t+1})\, \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)\, a_{ij} b_j(x_{t+1})\, \beta_{t+1}(j)} \tag{3.4.19}$$

où

- $\gamma_i = \sum_{t=1}^{L-1} \gamma_t(i)$ est le nombre espéré de transitions à partir de l'état i.
- $\delta_{ij} = \sum_{t=1}^{L-1} \zeta_L(i,\ j)$ est le nombre espéré de transitions de l'état i à l'état j.

Cette méthode de maximum de vraisemblance est la plus utilisée dans les applications [2].

3.4.8 L'algorithme de Baum-welch

L'algorithme de Baum-welch est appliqué pour trouver l'estimateur du maximum de vraisemblance des paramètres; le résultat donne des paramètres d'estimation meilleurs que les paramètres de la CMC d'origine. En, effet, à partir d'un modèle a priori et des observations supposées émises par le modèle, on cherche les probabilités de transitions et d'émissions maximisant la vraisemblance d'observations à chaque itération . L'algorithme utilise des valeurs intermédiaires des probabilité calculées dans la procédure forward-backward pour ajuster les paramètres du modèle telle que la vraisemblance du modèle est maximisée (localement dans la plupart des cas). On peut résumer la procédure itérative de Baum-welch sous l'algorithme suivant :

Algorithme 3.4.1 ***1- Fixer des valeurs initiales:***

$$\begin{aligned} k &= 0 \\ \lambda &= \left\{\pi_i^0,\ a_{ij}^0, b_j^0(x_t)\right\} \end{aligned}$$

2-calculer

en utilisant l'algorithme forward-backward:

$$\zeta(i,j) \;\; et \;\; \gamma_t(j) \qquad\qquad 1 \leq i,j \leq N,\; 1 \leq t \leq L-1$$

3- effectuer des nouvelles estimations du modèle

$$\begin{aligned} m &= 1,\ 2,\ .... \\ \widehat{\lambda} &= \left\{ \widehat{\pi}_i^{(m)}, \widehat{a_{ij}}^{(m)}, \widehat{b_j}^{(m)}(X_t) \right\} \qquad 1 \leq i,j \leq N \end{aligned}$$

4- Recommencer en $\left(2^{(0)}\right)$ ***et*** $\left(3^{(0)}\right)$ ***jusqu'à un certain critère d'arrêt.***

Nous avons déjà mentionné que le test d'arrêt est en général le nombre d'itérations fixé au départ ; il peut aussi être une variation non significative des paramètres du modèle.

Cependant, le choix du modèle initial influe sur les résultats, toutes les valeurs nulles de A et B au départ restent zéro à la fin de l'apprentissage.

Par ailleurs, l'algorithme peut converger vers des valeurs de paramètres qui forment un point critique de $P\left(X/\widehat{\lambda}\right)$, où nous obtenons un maximum local ou un point d'inflexion ; d'où la nécessité d'un bon choix du modèle pour éviter ces point d'inflexions [2]

Une alternative de l'algorithme de Baum-welch fréquemment utilisé, est l'apprentissage de Viterbi. Dans cette approche, les chemins le plus probables des séquences d'apprentissage sont dérivés en utilisant l'algorithme de Viterbi cité précédemment, et elle sont utilisés dans le processus de ré estimation de Baum-welch. Ensuite, le processus est mis en itération quand les nouvelles valeurs des paramètres sont obtenues, et on continue le processus jusqu'au moment où les chemins ne change plus. En ce point les estimateurs des paramètres ne change plus aussi du fait qu'il sont complètement déterminés par le chemins.

Cette procédure ne maximise pas la vraie vraisemblance : $P(X^1,\ X^2,\ ...,\ X^r/\lambda)$ comme une fonction des paramètres du modèle λ, mais elle trouve les valeurs de λ qui maximisent la contribution de la vraisemblance

$$P\left(X^1,\ X^2,\ ...,\ X^r/\ \lambda,\ \widehat{S}\left(X^1\right),\ \widehat{S}\left(X^2\right),\ ...,\ \widehat{S}\left(X^r\right)\right)$$

À partir des chemins les plus probables des séquences.

Contrairement à l'algorithme de Viterbi l'algorithme de Baum-welch teste aussi les moins bons chemins (en les pondérant) alors que le premier ne considère que les meilleurs ; Ceci revient à dire que Baum-welch souffre beaucoup moins d'un mauvais modèle de départ que Viterbi, car il reste tout de même les chemins alternatifs. Néanmoins, ces deux algorithmes risque de rester coincés sur un optimum local. Souvent, et dans de nombreux problème, le domaine d'optimisation est très complexe et possède plusieurs maximaux locaux.

Pour avoir une estimation convenable du modèle, les ré estimation se fond sur un ensemble de plusieurs suites d'observations appelées corpus d'apprentissage. Il est à noter aussi que la taille du corpus influe sur les résultats.

3.4.9 Les formules de ré estimation de $\lambda\,(\Pi, A,\ B)$

Nous avons vu que l'étape M de l'algorithme EM consiste à la maximisation en λ de la fonction $Q\left(\lambda/\lambda^{(m)}\right)$ ce qui revient à maximiser

$$\begin{aligned} Q\left(\lambda/\lambda^{(m)}\right) &= \sum_{i=1}^{N} P\left(S_1 = i \ / \ X,\ \lambda^m\right) \ln \pi_i \\ &\quad + \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{L-1} P\left(S_t = i, S_{t+1} = j \ / \ X,\ \lambda^m\right) \ln a_{ij} \\ &\quad + \sum_{j=1}^{N}\sum_{t=1\{x_t=k\}}^{L} P\left(S_t = j \ / \ X,\ \lambda^m\right) \ln b_j\left(k\right) \end{aligned}$$

La maximisation se fait en π_i, a_{ij}, $b_j\left(x_t\right)$ et on obtient les estimateurs $\lambda^{(m+1)}$ à l'itération m par les formules suivantes :

$$a_{ij}^{(m+1)} = \frac{\sum_{t=1}^{L-1} P\left(\ S_t \ = \ i,\ S_{t+1} \ = \ j \ / \ X,\ \lambda^{(m)}\ \right)}{\sum_{t=1}^{L-1} P\left(\ S_t \ = \ j \ / \ X,\ \lambda^{(m)}\ \right)}$$

$$b_j^{(m+1)}\left(x_t\right) = \frac{\sum_{t=1}^{L} 1_{\{y_t \ = \ x_t\}} P\left(\ S_t \ = \ j \ / \ X,\ \lambda^{(m)}\ \right)}{\sum_{t=1}^{L} P\left(\ S_t \ = \ j \ / \ X,\ \lambda^{(m)}\ \right)}$$

et

$$\pi_i^{(m+1)} = \frac{\sum_{t=1}^{L} P\left(\ S_t \ = \ i \ / \ X,\ \lambda^{(m)}\ \right)}{n}$$

on définit $\gamma_t(i)$ qui signifie le nombre de fois possibles d'être dans l'état i à l'instant t alors :

- Le nombre espéré de fois (fréquence) d'être dans l'état i au temps $t = 1$ est

$$\widehat{\pi} = \gamma_1(i) \qquad 1 \leq i \leq N \tag{3.4.20}$$

$$= \frac{\alpha_1(i)\,\beta_1(i)}{\sum_{i=1}^{N} \alpha_L(i)} \qquad 1 \leq i \leq N \tag{3.4.21}$$

- Le nombre espéré de transition de i vers j pondéré par le nombre espéré de fois d'être dans l'état j donne $\widehat{a}_{ij}$ tel que :

$$\widehat{a}_{ij} = \frac{\sum_{t=1}^{L-1} \zeta_t(i,j)}{\sum_{t=1}^{L-1} \gamma_t(i)} \tag{3.4.22}$$

$$= \frac{\sum_{t=1}^{L-1} \alpha_t(i)\, a_{ij} b_j(x_{t+1})\, \beta_{t+1}(j)}{\sum_{t=1}^{L-1} \alpha_t(i)\,\beta_t(j)} \qquad 1 \leq i,\ j \leq N$$

- Le ratio du nombre espéré de fois d'être dans l'état j en observant le symbole k divisé par le nombre de fois d'être dans l'état j est :

$$\widehat{b}_j(x_{t+1}) = \frac{\sum_{t=1\{x_t = k\}}^{L} \gamma_t(j)}{\sum_{t=1}^{L} \gamma_t(i)} \tag{3.4.23}$$

$$= \frac{\sum_{t=1\{x_t = k\}}^{L} \alpha_t(i)\,\beta_t(j)}{\sum_{t=1}^{L} \alpha_t(i)\,\beta_t(j)} \qquad 1 \leq\ j \leq N$$

La dernière expression représente la fréquence d'occurrence du symbole k à l'état j par rapport à la fréquence d'occurrence de n'importe quel symbole à l'état j (au même état j) ; on remarque que pour le calcul de $b_j(k)$ la somme va jusqu'à $t = L$.

Les équations de ré estimations (3.4.21) , (3.4.22) et (3.4.23) sont les formules de l'algorithme de Baum-welch. Chaque application de ces formules garantie la croissance de la probabilité de l'observation $P(X)$ sauf si on est au point critique ; dans ce cas les valeurs des paramètres du modèle courant $\lambda = (\Pi,\ A,\ B)$ et les estimateurs obtenus à partir de ce modèle après application de l'algorithme de Baum-welch $\widehat{\lambda} = \left(\widehat{\Pi},\ \widehat{A},\ \widehat{B}\right)$ sont identiques.

Rabiner en 1989 d'après [17] a montré que :

1. Le modèle initial λ définit un point critique de la fonction de vraisemblance pour laquelle $\widehat{\lambda} = \lambda$.

2. Sinon, le modèle $\widehat{\lambda}$ est plus vraisemblable (plus probable), c'est à dire $\widehat{\lambda}$ vérifie $P\left(X/\widehat{\lambda}\right) \geq P\left(X/\lambda\right)$, on a trouvé donc un autre modèle $\widehat{\lambda}$ dans lequel la séquence d'observations a plus de probabilité d'occurrence.

3. Ainsi, si on utilise $\widehat{\lambda}$ à la place de λ d'une manière itérative et on ré-estime le modèle on peut augmenter la vraisemblance d'observations de la séquence X sachant le modèle jusqu'à un certain critère d'arrêt.

Enfin, le résultat de la procédure itérative de la ré estimation des paramètres du modèle est appelé l'estimateur du maximum de vraisemblance.

3.4.10 Estimation du modèle λ par la technique d'optimisation

Notons que les formule de ré estimation du modèle HMM vu en tant que processus stochastique doivent vérifier automatiquement et à chaque itération les condition suivantes :

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad 1 \leq \ i \leq N \tag{3.4.24}$$

$$\sum_{i=1}^{N} \pi_i = 1 \qquad 1 \leq \ i \leq N \tag{3.4.25}$$

et

$$\sum_{j=1}^{M} b_j\left(x_{t+1}\right) = 1 \qquad 1 \leq \ j \leq N \tag{3.4.26}$$

D'où on peut considéré le problème de ré estimation traité ci-dessus comme étant un problème d'optimisation de $P\left(X/\lambda\right)$ avec les conditions (3.4.24) , (3.4.25) et (3.4.26) et ce en utilisant la technique du multiplicateur de Lagrange (la méthode du gradient) [53].

Soit L le lagrangien du $P\left(X/\lambda\right)$ sous les contraintes (3.4.24) , (3.4.25) et (3.4.26) :

$$L\left(P\left(X/\lambda\right)\right) = P\left(X/\lambda\right) + \sum_{i=1}^{N}\lambda_i\left(\sum_{j=1}^{N}a_{ij}-1\right) + \sum_{i=1}^{N}\mu_i\left(\sum_{i=1}^{N}\pi_i-1\right) + \sum_{i=1}^{N}\beta_i\left(\sum_{j=1}^{N}b_j\left(x_t=k\right)-1\right)$$

Les variables λ_i, μ_i, β_i sont les multiplicateurs de Lagrange à déterminer.

Estimation de a_{ij}

Par rapport à a_{ij}, $L\left(P\left(X/\lambda\right)\right)$ s'écrit :

$$L\left(P\left(X/\lambda\right)\right) = P\left(X/\lambda\right) + \sum_{i=1}^{N}\lambda_i\left(\sum_{j=1}^{N}a_{ij}-1\right)$$

a_{ij}^{*} est tel que

$$\frac{\partial L\left(P\left(X/\lambda\right)\right)}{\partial a_{ij}} = 0$$

Ceci revient à écrire

$$\frac{\partial P\left(X/\lambda\right)}{\partial a_{ij}} + \lambda_i = 0 \qquad 1 \leq\ i,\ j \leq N \tag{3.4.27}$$

En multipliant (3.4.27) par a_{ij} et en sommant sur j, on obtient :

$$\sum_{j=1}^{N}a_{ij}\frac{\partial P\left(X/\lambda\right)}{\partial a_{ij}} = -\lambda_i\left(\sum_{j=1}^{N}a_{ij}\right) = -\lambda_i = \frac{\partial P\left(X/\lambda\right)}{\partial a_{ij}} \tag{3.4.28}$$

L'équation (3.4.28) montre que $P\left(X/\lambda\right)$ est maximisée quand

$$a_{ij}^{*} = \frac{a_{ij}\frac{\partial P(X/\lambda)}{\partial a_{ij}}}{\sum_{k=1}^{N}a_{ik}\frac{\partial P(X/\lambda)}{\partial a_{ik}}} \tag{3.4.29}$$

Il est évident que l'équation (3.4.29) est analytiquement insoluble. On rappelle que $P\left(X/\lambda\right)$ s'écrit :

$$P\left(X/\lambda\right) = \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t\left(i\right)a_{ij}b_j\left(x_{t+1}\right)\beta_{t+1}\left(j\right) \qquad 1 \leq\ t \leq L-1 \tag{3.4.30}$$

Donc

$$\frac{\partial P\left(X/\lambda\right)}{\partial a_{ij}} = \sum_{t=1}^{L-1}\alpha_t\left(i\right)b_j\left(x_{t+1}\right)\beta_{t+1}\left(j\right) \tag{3.4.31}$$

En considération les équations (3.4.29) , (3.4.30) et (3.4.31) , on trouve :

$$\begin{aligned} a_{ij} &= \frac{\sum_{t=1}^{L-1} \alpha_t(i)\, a_{ij} b_j(x_{t+1})\, \beta_{t+1}(j)}{\sum_{j=1}^{N}\sum_{t=1}^{L-1} \alpha_t(i)\, a_{ij} b_j(x_{t+1})\, \beta_{t+1}(j)} \\ &= \frac{\sum_{t=1}^{L-1} \alpha_t(i)\, a_{ij} b_j(x_{t+1})\, \beta_{t+1}(j)}{\sum_{t=1}^{L-1} \alpha_t(i)\, \beta_t(j)} \end{aligned} \tag{3.4.32}$$

Cette équation est équivalente à l'équation (3.4.22)

Estimation de π_i et $b_j(x_t = k)$

De la même manière π_i et $b_j(x_t = k)$ sont calculés à partir de :

$$\begin{aligned} \frac{\partial P(X/\lambda)}{\partial \pi_i} &= \sum_{j=1}^{N} a_{ij} b_i(x_1)\, b_j(x_2)\, \beta_2(j) \\ &= b_i(x_1)\, \beta_1(i) \end{aligned} \tag{3.4.33}$$

Et

$$\frac{\partial P(X/\lambda)}{\partial b_j(x_t = k)} = \sum_{t=1\{x_t = k\}}^{L} \sum_{i=1}^{N} \alpha_t(i)\, a_{ij} \beta_{t+1}(j) + \delta(x_1,\ k)\, \pi_j \beta_1(j) \tag{3.4.34}$$

Où δ est la fonction de kronecker ;

En substituant (3.4.33) , (3.4.34) dans leurs équations respectives, on aura :

$$\pi_i^* = \frac{\pi_i \frac{\partial P(X/\lambda)}{\partial \pi_i}}{\sum_{k=1}^{N} \pi_k \frac{\partial P(X/\lambda)}{\partial \pi_k}}$$

$$b_j^* = \frac{b_j(x_t = k)\, \frac{\partial P(X/\lambda)}{\partial a_{ij}}}{\sum_{k=1}^{M} b_j(x_t = k)\, \frac{\partial P(X/\lambda)}{\partial b_K(x_t = k)}}$$

Et on obtient les équations de ré estimation de π_i et $b_j(x_t = k)$ qui sont similaires aux équations (3.4.21) , (3.4.23)

Chapitre 4

Application

4.1 Introduction

La CMC a des domaines d'application très variés telle que la reconnaissance de la parole, la biologie l'ordonnancement des tâche, les technologies de l'information et de la communication ou la reconnaissance d'image, la climatologie et aussi à l'économie.

Dans cette partie, nous procédons à l'application des différents algorithmes étudiés dans le chapitre 3, ces derniers étant réalisés avec le langage de programmation Delphi sous environnement Windows.

Nous procédons à l'apprentissage des paramètres d'une CMC pour une séquence d'observation donnée en appliquant les formules de ré estimation de Baum-welch. Les valeurs estimées des paramètres du modèle sont ainsi différentes selon les valeurs initiales du modèle choisi au départ, les résultats des algorithmes sont détaillés.

4.2 Présentation

4.2.1 Position du problème

Le problème auquel nous allons essayez d'apporter une solution peut survenir dans n'importe quelle domaine. Nous donnons un exemple économique de la loi de Bernoulli. Soit l'économie d'un pays où l'existence des échanges avec les autres pays n'est pas très importante, dans la

mesure où seule l'économie en question est prise en considération (économie isolée). Cette économie possède des facteurs de production qui peuvent influer la politique économique ; c'est à dire, ces facteurs s'influencent et s'échangent les conséquences. Parmi ces facteurs de production nous donnons l'exemple des indices des prix à la consommation et du salaire moyen ; chacune de ces variables peut résulter de la variation de l'autre ou, par transition, de sa propre variation. nous considérons ces variations comme étant des états dans notre modèle d'économie et nous constatons que ces états peuvent donner, comme conséquences , des phénomène sociologiques et économiques, voire l'inflation et le chômage. Cette situation peut être modélisée comme une chaîne de Markov caché à éspace d'états fini (indice des prix à la consommation, salaires moyens) et à chacun des état est associée une fonction probabilistique pour l'émission des conséquences (le chômage, l'inflation).

Cependant, au lieu de restreindre l'application à un seul domaine, nous allons présenter et traiter le problème d'une manière général. Notons que une fois un modèle est construit, dans le cas général, il est facile de l'adapter au domaine dans lequel se pose le même problème.

Considérons un phénomène disposant d'un certain nombre d'états. La manifestation de ce phénomène s'observe à travers des réalisations d'un certain événement sans se rendre compte de ses états. Il est à noter que chaque état du système permet la réalisation de cet événement, ce qui le rend non observable (sous-jacent). Donc, nous assistons à un événement dû à la manifestation d'un phénomène sans connaître les états qui ont donné ces réalisations. Le problème qui se pose dans ce cas est le suivant :

Étant donné une observation de l'événement secondaire, comment retrouver l'état du phénomène qui a permis la réalisation de cet événement.

L'objectif de notre travail est de retrouver les états cachés d'un certain phénomène en se basant sur l'analyse statistique des réalisations de l'événement observé, généré par le phénomène. L'approche statistique que nous proposons s'appuie sur les chaînes de Markov cachées.

4.2.2 Présentation du modèle

Nous considérons un jet aléatoire de trois pièces de monnaies (équitable ou biaisées); la face de la pièce i peut apparaître avec une probabilité p_i et le pile de la pièce i avec une probabilité $q_i = 1 - p_i$.

Dans notre exemple d'application, nous considérons les observations comme des réalisations de Bernoulli et par conséquent, nous notons succès par 1 et échec par 0. Ainsi, la séquence d'observations est la suite constituée de l'ensemble de$\{0, 1\}$; on note la première pièce "1" et la deuxième pièce "2",la troisième pièce "3".

en jetant aléatoirement les pièces de monnaie, nous obtenons une séquence d'observations telle que :

0 0 0 1 1 0 1 0 1 0 1 1 1 0 0 0 1 0

nous avions joué avec les trois pièces dans un ordre tel que

1 3 3 1 2 1 1 2 1 1 2 3 2 1 2 3 2

La première séquence étant observée, la deuxième est la séquence cachée. Nous constatons :

- trois états : pièce "1" et pièce "2",pièce "3".

- chaque état est caractérisées par une distribution de probabilité de pile ou face (échec ou succès) qui ne sont pas généralement égales $(p\,(echec) \neq p\,(succès))$, les transitions entre les états sont caractérisées par une matrice de transitions d'états dont les probabilités sont déterminer par un événement spécifique.

4.2.3 Démarche à suivre

Nous abordons dans cette partie les deux étapes suivantes :

- Dans un premier temps, nous supposons que les paramètres du modèle sont connus, à savoir les probabilité de transition et les paramètres de la lois, et nous essayons de retrouver la séquence des états la plus probable à partir d'une séquence observée de réalisations. Donc cette étape consiste à retrouver la suite d'états qui suivra le système pour produire la séquence de l'observation observée ;

- Dans un deuxième temps, nous estimons les paramètres du modèle sur la base de la séquences de l'observations d'entraînement, dont nous connaissons le chemin d'états qui aura produit cette séquences d'observations, c'est la phase d'apprentissage par Baum-welch.

Dans cette section, nous étudions le comportement pratique des résultats du programme Baum-welch, en fixant

1. La longueur de la séquence L, nous travaillons sur une séquence de taille $L = 5$.

2. Le nombre de paramètres d : notre chaîne de Markov est à trois états et deux alphabets de sortie chacun de ces états émet selon une loi de Bernoulli de probabilité p .l'alphabets considérés est à deux lettres 0 (échec) et, 1 (succès), la matrice de probabilités de transitions étant une matrice 3×3, et la matrice d'émissions est aussi 3×2 et deux alphabets de sortis "0" et"1"d'ou $d = 17$.

3. Les valeurs du modèle initial $\lambda^{(0)}$ à estimer.

Dans l'apprentissage Baum-welch, nous avons choisi arbitrairement un modéle initiale $\lambda_1^{(0)}$. Puis nous avons étudié le comportement des paramètres de ce modèles après M itérations.

4.3 Apprentissage du modèle

Nous partont d'une CMC suivante, dont les paramètres on été données de manière arbitraire $\lambda_0 = (A, B, \Pi)$.

Avec $n = 3,\ m = 2, L = 5,\ k = 1$

$$A = \begin{pmatrix} 0.45 & 0.35 & 0.20 \\ 0.10 & 0.50 & 0.40 \\ 0.15 & 0.25 & 0.60 \end{pmatrix}; B = \begin{pmatrix} 1.0 & 0.0 \\ 0.5 & 0.5 \\ 0.0 & 1.0 \end{pmatrix}; \pi = \begin{pmatrix} 0.5 \\ 0.3 \\ 0.2 \end{pmatrix};$$

Soit $O = (0, 1, 1, 0, 0)$.

4.3.1 Les trois problèmes de CMC

$\star$ Le premier problème à résoudre est le calcul de $P(O/\lambda_0)$?

pour cela on à oppter pour l'évaluation par les fonctions forward-backward qui s'avère une méthode plus rapide que celle de la méthode directe (chapitre 3).

⋆ Le deuxième problème est le calcul du chemin optimal ou estimation de la partie cachée du modèle?

il s'agit ici de trouver dans le modèle la suite d'états qui maximise $P(Q/O, \lambda_0) \Leftrightarrow P(Q, O/\lambda_0)$

⋆ Le troisième problème qui se pose est celui de l'apprentissage : comment ajuster les paramètres du modèle λ_0 pour maximiser $P(O/\lambda_0)$, à partir de la séquences d'apprentissage dont on sait qu'elles ont été émises par ce modèle?

Pour la programmation, nous utilisons le logiciel Delphi6 voir (l'annexe A) pour l'organigramme détaillé .

4.3.2 Résultat d'application

1)-Evaluation par les fonction Forward-Backward

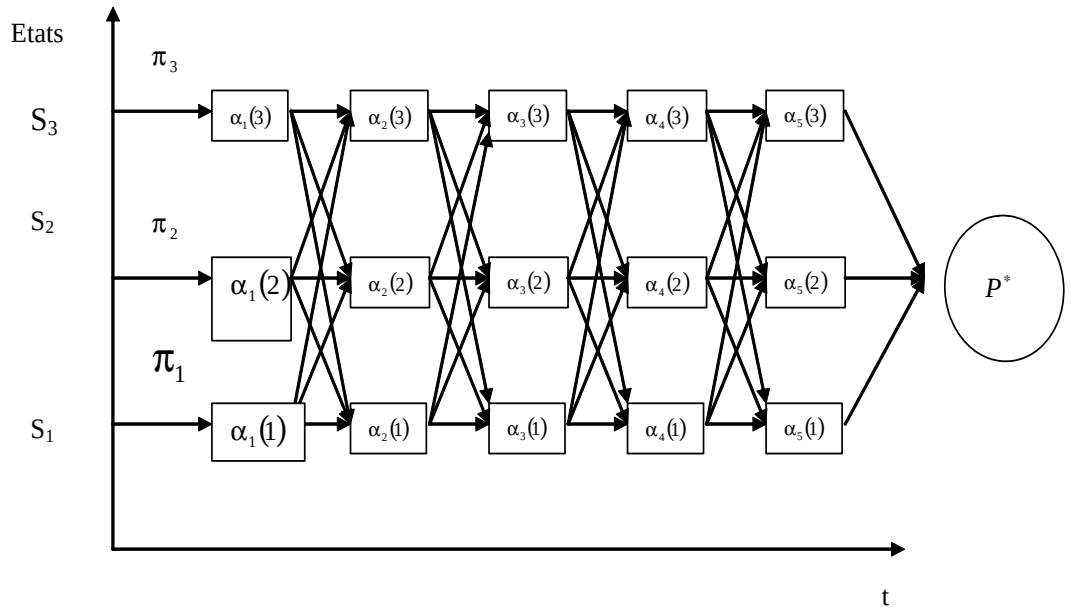
Le calcul de $\alpha(t)$ pour l'observation O nous donne la matrice suivante :

$$\alpha = \begin{pmatrix} 0.5 & 0 & 0 & 0.27025 & 0.152675 \\ 0.15 & 0.125 & 0.05125 & 0.031625 & 0.012495 \\ 0 & 0.06 & 0.146 & 0 & 0 \end{pmatrix}$$

avec la probabilité d'observation :

$$P^* = 0.0277625$$

la figure ci dessous illustre le calcul de α pour la suite d'observation donnée :



Calcul de la probabilité α_t

et le calcul de $\beta(t)$ pour l'observation O nous donne la matrice suivante :

$$\beta = \begin{pmatrix} 0.0364375 & 0.05375 & 0.3425 & 0.625 & 1 \\ 0.063625 & 0.0925 & 0.15 & 0.35 & 1 \\ 0.0723125 & 0.10125 & 0.1375 & 0.275 & 1 \end{pmatrix}$$

avec aussi la probabilité d'observation :

$$P^* = 0.0277625$$

2)- calcul du chemin optimal

le calcul des quantités δ, Ψ, P^*et q^* données par les matrices suivante :

$$\delta = \begin{pmatrix} 0.5 & 0 & 0 & 0.009 & 0.00405 \\ 0.15 & 0.0875 & 0.021875 & 0.0075 & 0.001875 \\ 0 & 0.1 & 0.06 & 0 & 0 \end{pmatrix}$$

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

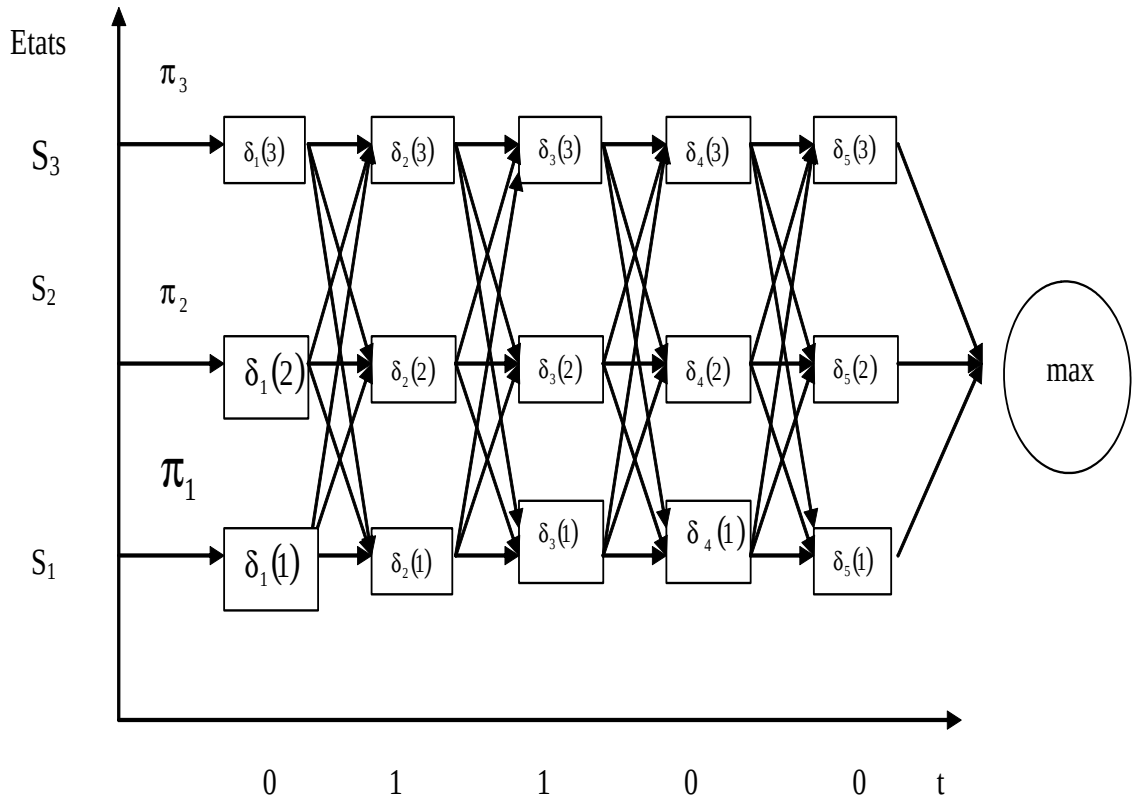
avec :

$$P^* = 0.0277625$$

et

$$Q = 1,\ 3,\ 3,\ 1,\ 1$$

et la figure ci dessous qui illustre le calcul de δ pour la suite d'observation donnée :



3)-Apprentissage

Si on prend comme ensemble d'apprentissage cette seule observation, l'application de l'algorithme de Baum-welch doit augmenter sa probabilité de reconnaissance .

Après une ré estimation on trouve $\lambda_1^{(1)}$ suivant:

$$A = \begin{pmatrix} 0.346 & 0.365 & 0.289 \\ 0.159 & 0.514 & 0.327 \\ 0.377 & 0.259 & 0.364 \end{pmatrix} ; \; B = \begin{pmatrix} 1 & 0 \\ 0.631 & 0.369 \\ 0 & 1 \end{pmatrix} ; \; \pi = \begin{pmatrix} 0.656 \\ 0.344 \\ 0 \end{pmatrix}$$

et

$P\left(O/\lambda_1\right) = 0.0529$

Après 15 itérations on trouve $\lambda_1^{(15)}$ suivant:

$$A = \begin{pmatrix} 0 & 0 & 1 \\ 0.212 & 0.788 & 0 \\ 0 & 0.515 & 0.485 \end{pmatrix} ; \; B = \begin{pmatrix} 1 & 0 \\ 0.969 & 0.031 \\ 0 & 1 \end{pmatrix} ; \; \pi = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

et

$P\left(O/\lambda_{15}\right) = 0.2474$

Après 150 itérations on trouve $\lambda_1^{(150)}$ qui réalise la convergence :

$$A = \begin{pmatrix} 0 & 0 & 1 \\ 0.18 & 0.82 & 0 \\ 0 & 0.5 & 0.5 \end{pmatrix} ; \; B = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} ; \; \pi = \begin{matrix} 1 \\ 0 \\ 0 \end{matrix}$$

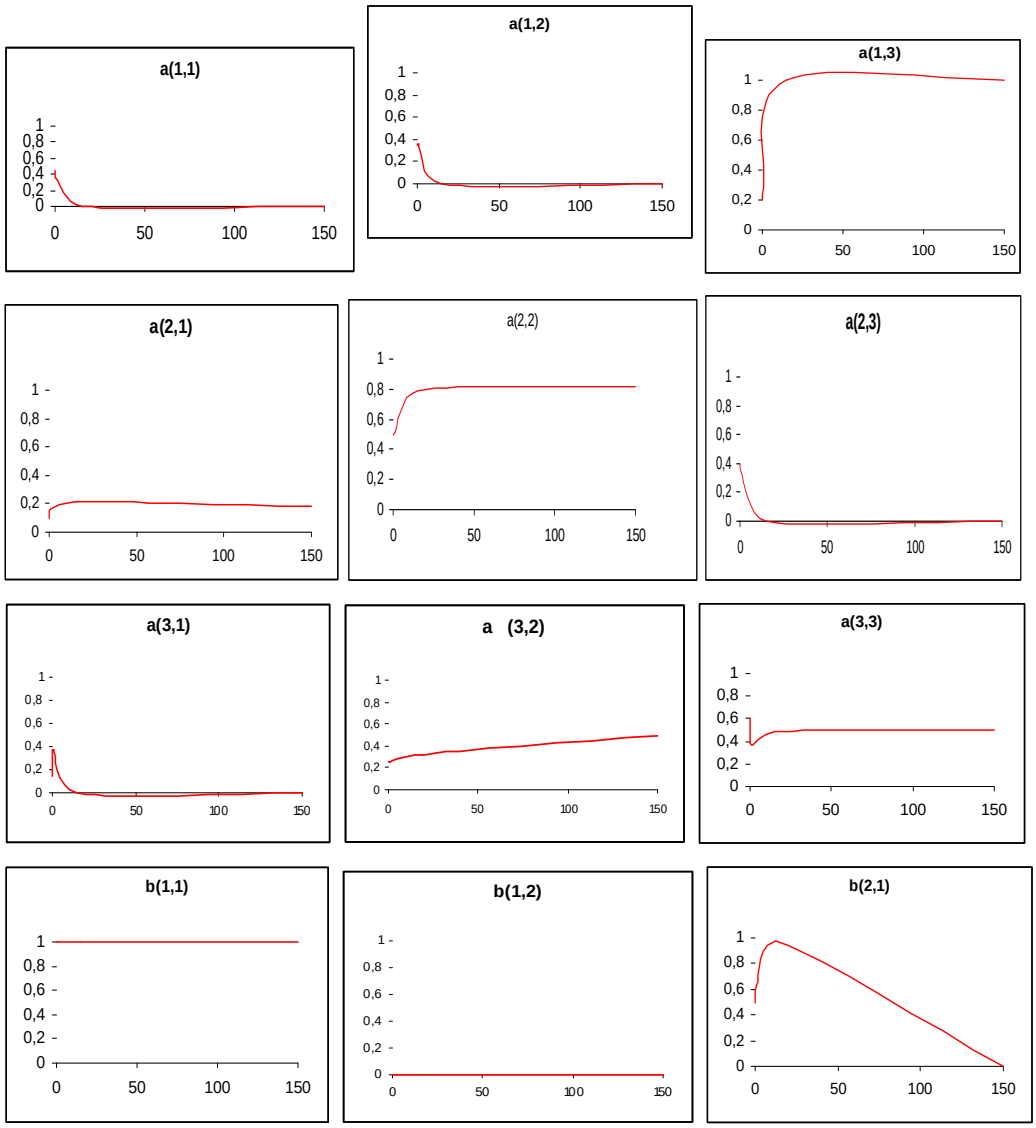
et

$P\left(O/\lambda_{150}\right) = 0.25$

Nous résumons les résultats précédent dans le tableau suivant:

L=5	$\lambda_1^{(0)}$	n=1, $\widehat{\lambda}_1 = \lambda_1^{(n)}$	n=15, $\widehat{\lambda}_1 = \lambda_1^{(n)}$	n=150, $\widehat{\lambda}_1 = \lambda_1^{(n)}$
$a(1,1)$	0.45	0.346	0	0
$a(1,2)$	0.35	0.365	0	0
$a(1,3)$	0.20	0.289	1	1
$a(2,1)$	0.10	0.159	0.212	0.18
$a(2,2)$	0.50	0.514	0.788	0.82
$a(2,3)$	0.40	0.327	0	0
$a(3,1)$	0.15	0.377	0	0
$a(3,2)$	0.25	0.259	0.315	0.5
$a(3,3)$	0.60	0.364	0.485	0.5
$b(1,1)$	1	1	1	1
$b(1,2)$	0	0	0	0
$b(2,1)$	0.5	0.631	0.969	1
$b(2,2)$	0.5	0.369	0.031	0
$b(3,1)$	0	0	0	0
$b(3,2)$	1	1	1	1
$Forward$	0.0277625	0.0529	0.2474	0.25

et les graphe associes aux paramètres est :



4.3.3 Conclusion

Dans l'apprentissage par Baum-Welch du modèle initial $\lambda_1^{(0)}$ générant la suite d'observation O on constat que le calcul de la variable forward (vraisemblance) est monotone après 150 itérations.

Les valeurs ré-estimées des paramètres du modèle $\lambda_1^{(0)}$ sont différentes et éloignées des vrais valeurs des paramètres du modèle initiale

Chapitre 5

Conclusion générale

Nous avons présenté dans ce travail; les traitements statistiques fondés sur les chaîne de Markov Cachées, qui présentes des qualités exceptionnelles, découlant de leur aptitude à prendre en compte , de façon souvent élégante et mathématiquement rigoureuse, l'ensemble de l'information disponible, pour des problèmes à caractère aléatoire.

Beaucoup de problèmes réels dans le monde ont la caractéristique de changer dans le temps d'où l'utilisation fréquente de ce modèle dans différents domaines d'application. Dans notre document, nous nous sommes intéressés à l'aspect probabiliste des chaînes de Markov cachées ; nous avons effectué une revue de la théorie de ces modèles, et nous avons appliqué les méthodes d'estimation citées dans la partie théorique sur des données choisies arbitrairement.

Un bon modèle de Chaîne de Markov caché est celui qui fournit une valeur de la vraisemblance la plus élevé (maximale).Toutefois, un calcul direct a des coûts de stockage et de calcul exponentiel très élevés et par conséquent, nous avons appliquer l' algorithme Forward et Backward qui plus facile et moins complexe en espace de stockage.

Par ailleurs, les Chaînes de Markov cachées sont connus par le manque de données concernant les états et par conséquent l'estimation des paramètres du modèle par la méthode du maximum de vraisemblance devient impossible.

Plusieurs méthodes d'estimation de ces modèles ont été développées. Il s'agit des algorithms itératifs qui consistent à estimer récursivement les paramètres du modèle jusqu'à un

critère d'arrêt prédéfini. Pour notre modèle nous avons appliquer l'algorithme Baum-welch qui est la variante de l'algorithme EM dans le cas de chaînes de Markov cachées.

Le choix de l'application s'est porté sur le jet de trois pièces de monnaies selon certaines lois de probabilité et les réalisations ainsi obtenues suivent des lois de Bernoulli de paramètres p. Dans l'application nous avons développé un mini logiciel comportant les codes informatiques des algorithmes des différentes méthodes citées dans la partie théorique, nous avons appliqué ces derniers sur les données quelconques des paramètres attribuée initialement et nous avons constaté que les résultats différent selon les valeurs des paramètres, la longueur de la séquence traiter et le nombre d'itérations.

L'application des programme d'apprentissage nous à permis de relever les constatations (résultats) suivantes:

Apprentissage Baum-welch

Les résultats obtenus montrent bien les performances obtenues en phase d'apprentissage

Les valeurs trouvée représentes le $P(O/\lambda)$ moyen (la vraisemblance ou la variable Forward)

On remarque aussi l'accroissement de $P(O/\lambda)$; jusqu'a un certain seuil, c'est - à dire que l'algorithme d'apprentissage de Baum-welch converge vers un maximum local, point d'inflèction).

De la on constate que la phase d'apprentissage est une composante majeure de tout système, lui permettant d'améliorer ses performances.

La précision numérique

La mise en oeuvre des récurrences manipulant des nombres inférieurs à un dans un programme donne forcément des problème numériques ou l'underflow. En effet, les résultats des procédures Forward et Backward tendent exponentiellement vers zéro quand la longueur de la séquence tend vers l'infini. Pour y remédier, plusieurs solutions ont été proposé et ce en modifiant les algorithmes correspondants comme le cas de l'utilisation de logarithme dans l'algorithme de Viterbi.

Le manque de données

Nous avons montré dans le chapitre 3 que les paramètres du modèle tendent vers leurs vraies valeurs si la longueur de la séquence est très grande voire infinie (taille de l'échantillon

$n \to \infty$), alors qu'à l'apprentissage nous utilisons des observations de longueur finie. Cette insuffisance du nombre d'observations donne un mauvais échantillon représentatif du problème à traiter. Le problème de l'insuffisance du nombre d'observations pour l'apprentissage, fait que certains événements généralement de faibles probabilités peuvent ne pas apparaître dans cet ensemble fini d'observations, c'est à dire que l'apprentissage donne des probabilités nulles.

Estimation initiale de la CMC

Le problème de l'estimation initiale des paramètres de la CMC a une influence directe sur les formules de ré estimation pour l'apprentissage. Le problème du maximum local est souvent le résultat d'une mauvaise estimation initiale du modèle. Une solution exacte pour ce problème n'existe pas, seules l'intuition et l'expérience aident à trouver le bon estimateur du vecteur du modèle initial.

Stockage de l'historique

Les formule de ré-estimation de Baum-welch pour identification des chaînes de Markov cachés, utilisant les procédures "Forward" et "Backward". À chaque appel, ces derniers font intervenir toutes les observations jusqu'au temps t (l'instant du calcul) d'où la nécessité du stockage à chaque exécution.

Dans notre application, nous avons utilisé une seule séquence d'apprentissage alors qu'il est souhaitable d'utiliser plusieurs séquences d'apprentissage (corpus) pour avoir de bons résultats; le paramètre estimé à partir de ce corpus sera égal à la moyenne de tous les paramètres issus de toutes les séquences d'apprentissage.

En terme de perspective pour une recherche ultérieur, Ce modèle peut être enrichit par plusieurs choses, car ici nous avons seulement mis le point sur l'algorithme de Baum-welch auquel nous avons effectué le programme informatique et l'application et nous avons négligé par ailleurs les autres méthodes itératives utilisées dans le cas de manque de données tel que l'algorithme SEM, EM à la Gibbs et l'estimation bayesienne dont le principe consiste à augmenter les données des paramètres du modèle. Ces méthodes sont développées dans la littérature et méritent d'être traitées, pour pouvoir comparer les résultats des différentes méthodes pour tirer conclusion sur l'estimation de chacune d'elles. Une autre proposition

consiste à appliquer toutes ces méthodes sur plusieurs lois statistiques pour étudier le comportement de chaque méthode selon la loi de distribution choisie.

Annexes A

Organigramme de l'apprentissage

Dans cette annexe, nous présentons l'organigrame des algorithme d'apprentissage par la chaîne de markov cachée .

Voici la légende des notion utilisées:

• $\alpha\,[i,t]$: Matrice N*T qui désigne la probabilité conjointe de la séquence partielle de d'observations de l'instant initial 1 à l'instant t avec l'hypothése que le processus est dans la classe ω_i à l'instant t

$$\alpha_t\,(i) = P(X_t = \omega_i, Y_1 = y_1, ..., Y_t = y_t)$$

• $B\,[i, O\,[]]$: Vecteur qui représente la probabilité que l'on observe la réalisation O alors que le modèle se trouve dans la classe $\omega_{i,}$ soit:

• $\Pi\,[i]$: Matrice représente la probabilité que l'état de dépard (initiale) du modèle soit la classe ω_i, $1 \leq i \leq k$

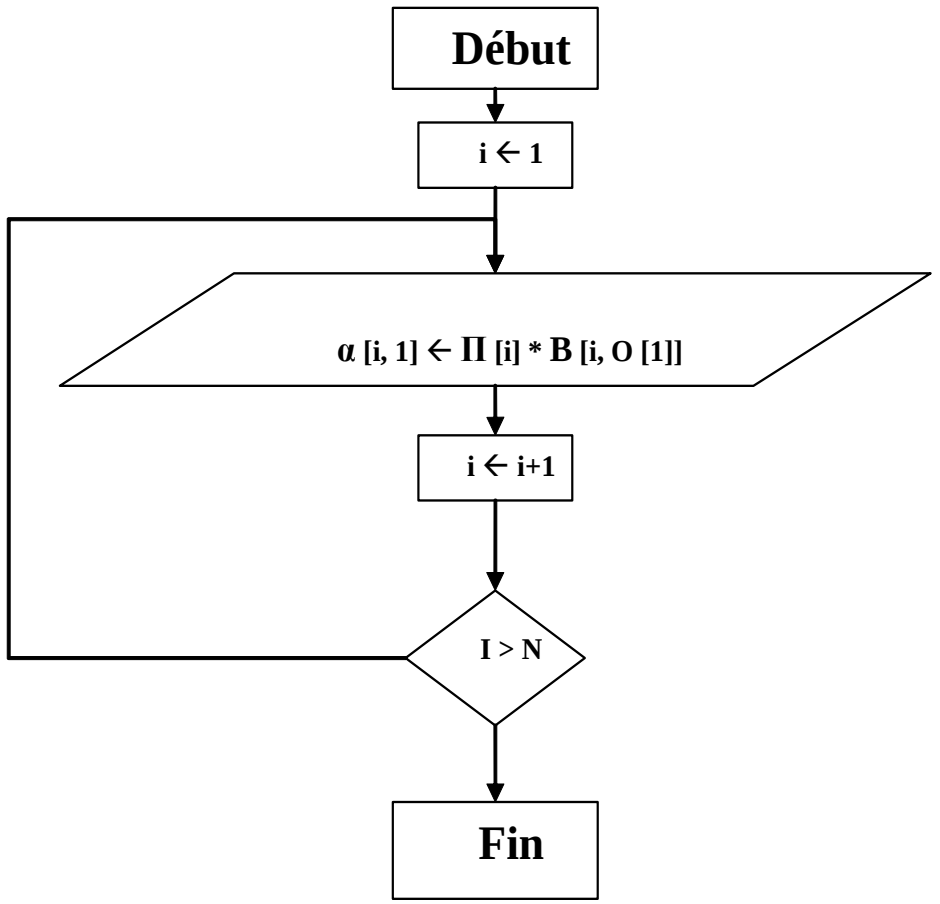
• $O\,[]$: Vecteur de la une suite d'observations $O = (o_1, ..., o_T)$

• $\beta\,[i,T]$: Matrice N*T qui désigne la probabilité conjointe de la séquence partielle de d'observations de l'instant $t+1$ à l'instant T avec l'hypothése que le processus est dans la classe ω_i à l'instant t.

• $\delta\,[1,1]$: Matrice N*T qui represente $argmax\Psi$

• $\Psi\,[i,1]$: Matrice N*T qui représente la probabilité conjointe d'emprunter le meilleur chemin pour atteindre la classe ω_j à l'instant t et d'avoire la sous-séquence d'observations allant de l'instant initial à l'instant t.

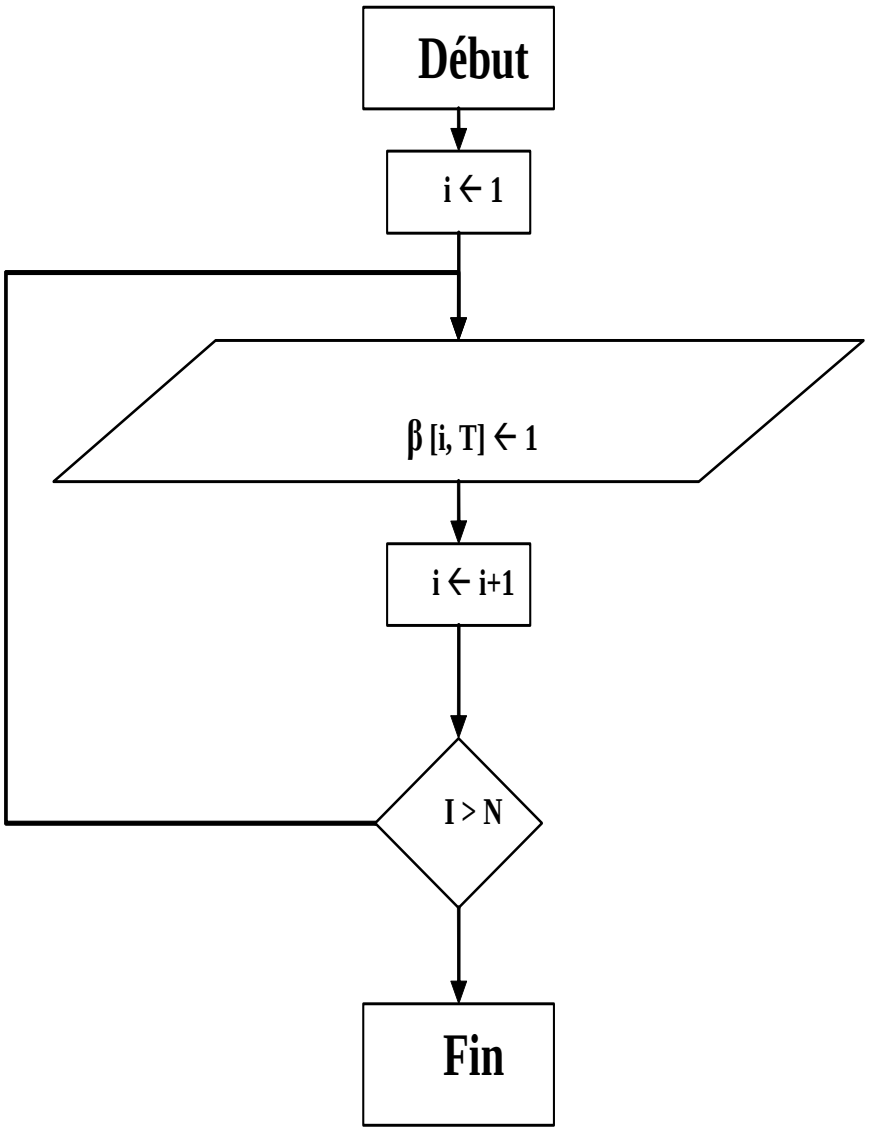
Initialisation



- N : le nombre d'observations .
- T : la longueure de l'observation.

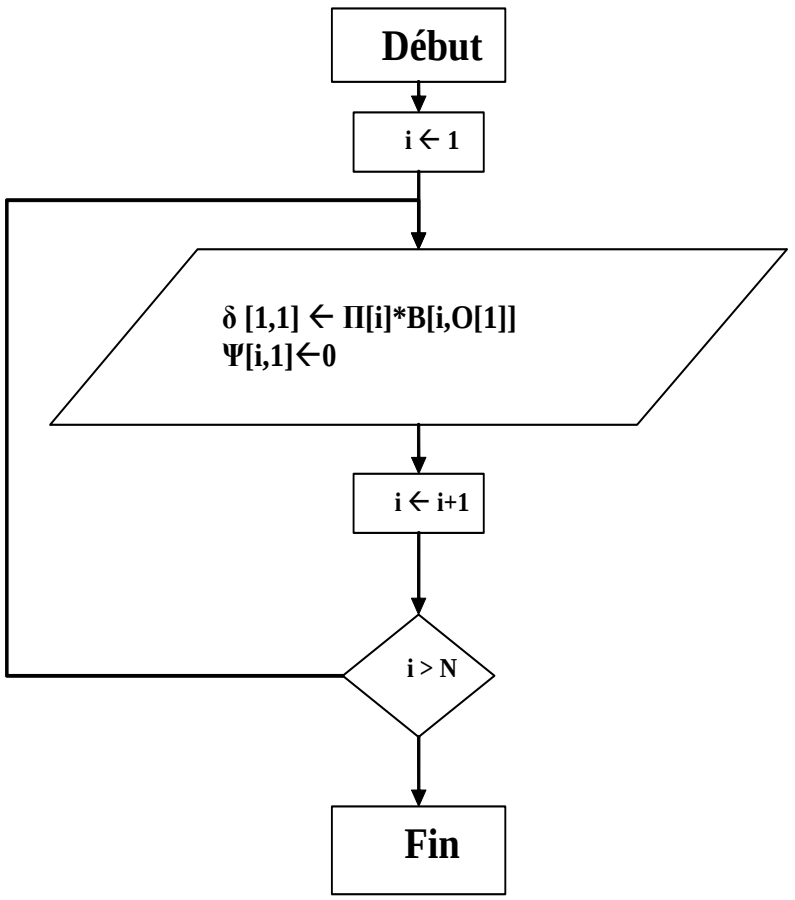
Calculer la matrice β

Initialisation



Calculer la matrice δ et Ψ

Initialisation



Bibliographie

[1] AISSANI, A. (1992). Modèles Stochastique de la Théorie de Fiabilité. Edition OPU. Alger.

[2] ALANI, T. et GUELLIF, H. (1994). Modèles de Markov Cachés Théories et Techniques de base. Traitement du signal automatique et productique. Rapport de stage. No. 2196. Programme 5. INRIA.

[3] ALANI, T. et HAMAM, Y.(1999). Extension of Poritz Linear Predictive HMMs to Single Input Multi decorrelated Output Dynamic Systems. Technical Report SC 21 Laboratory. Control Departement. ESIEE.

[4] ALBERT, P.S. (1994). A Markov Model for Sequences of Ordinal Data from a Relapsing-Remiting Disease. Biometrics. No. 50, 51-60

[5] ALBIN, J.M.P. (1992). Stochastic Process and their Applications. An Official Journal of The Bernoulli Society of Mathematical Statistics and Probability. Vol. 42. No. 1, 126-143.

[6] Anigbogu, J.C.(1992). "Reconnaissance de caractères imprimés multifontes à l'aide de modèles stochastiques et métriques. Thèse de doctorat, Univ. Nancy I, 1992.

[7] ASKAR, M. et DERIN, H. (1981). A Recursive Algorithme of Bayes Solution of the Smmoothing Problem. IEEE Transaction on Automatic Control. Vol. AC 26, No. 2, 558-561.

[8] ATTWOOD, T.K. et PARRY-SMITH, D.J.(1999). Introduction to Bioinformatics. Ad¬$disonWesleyLongmanLimited, England.$

[9] BAHL, L.R., BAKIS, R., COHEN, P.S. , COLE, A.G., JELINEK, F., LEWIS, B.L. et MERCER, R.L. (1979). Recognition Results with Several Experimental Acoustic Pro¬$cessors.IEEE, InternationalConferenceonAcousticSpeechandSignalProcessing$, 249−251.

[10] BAHL, L.R. et JELINEK, F. (1975). Decoding of Channels with Insertion, Deletion and Substitution with Applications to Speech Recognition. IEEE Transaction on Information theory. Vol. IT 21. No. 4, 404-411.

[11] BAKER, J.K. (1975). The Dragon System. IEEE Transaction on Acoustic Speech and Signal processing. Vol. ASSP 23. No. 1, 24-29.

[12] BAKRY, D., MILLAND, X. et VANDERKERKOVE, P.(1997). Statistique de Chaîne de Markov Cachées à Espace d'états Fini : le cas non stationnaire. C. R. Acad. Scie. Paris. Tome 325. Serie I, 203-206.

[13] BALDI, P., CHAUVIN, Y., HUNKAPILLER, T. et McCLURE, M.A. (1994). Hidden Markov Models of Biological Primary Sequence Information. Proceeding of National Academic Science. Vol. 91, 1059-1063.

[14] BARRAS, C. (1996). Reconnaissance de la Parole Continue : adaptation au locuteur et contrôle temporel dans les Modèles de Markov Cachés. Thése de doctorat d'état. Universite de Paris VI. Spécialité Lnformatique, pp. 23-38.

[15] BAUM, L.E. et EAGON, J.A. (1966). An Equality with Application to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology. communicated by R. C. Back, Bull. Amen See. No. 73, 360-363.

[16] BAUM, L.E. et PETRIE, T. (1966). Statistical Inference for Probabilistic Function of Finite State Markov Chains. Annals of Mathematical Statistics. No. 37, 1554-1562.

[17] BAUM, L.E., PETRIE, T., SOULE, G. et WEISS, N.(1970). A Maximization Technique Occuring in the Statistical Analysis of Probabilistic Function of Markov Chain. Anal. Math. Stat., Vol. 41. No. 1, 164-171.

[18] BAUM, L.E et SELL1, G. R. (1969). Growth Transformation for Functions on Manifolds. Pacifical Journal of Mathematics. Vol. 27. No. 2, 211-227.

[19] B.Benlmiloud,A. Peng, W.Piieczynski, "Estimation conditionnelle itérative dans les chaîne de Markov Cachées et Segmentation statitique non supervisée d'images". Acte du quatorzième quolloque GRETSI, septembre1993.

[20] B. Benlmiloud , "chaîne de markov cachée et segmentation statistique non supervisée d'image ". thése, Université Paris VII, 1994.

[21] BILLINGSLEY, P. (1960). Statistical Methods in Markov Chains. Annals of Mathe¬$maticalStatistics.No.32, 12-24$.

[22] BOROVKOV, A. (1987). Statistiques Mathematiques. Edition MIr Moscou. p. 120.

[23] Bramer, M. A. (1986). Expert Systems : the version and Reality . Thames Polytchnic.

[24] BREMAUD, P. (1991). Markov Chains : Gibbs fields, Monté Carlo Simulation and queues. Edition Springer ; 95-117.

[25] CHURCHILL, G. (1989). Stochastic Models for Heteregenous DNA Sequences. Bulletin of Mathematical Biology, No. 51 (1), 79-94.

[26] çINLAR, E. (1975). Introduction to Statistic Process. Northwestern University, Prentice-Hall, Inc. England Cliffs, New Jersey, 1-16.

[27] COVER, T.M. (1984). An Algorithm for Maximizing Expected Log Investment Re¬$turn.IEEETransactiononInformationTheory.Vol.IT-30.No.2, 369-373$

[28] DEMPSTER, A.P., LAIRD, N.M. et RUBIN, D.B. (1976). Maximum Likelihoode from Incomplete Data via the EM Algorithm. Havard University and Educational Testing Service, 1-38.

[29] DEVORE, J.L. (1976). A Note on the Estimation of parameters in a Bernoulli Models with Dependence. The Annals of Statistics. Vol. 4. No. 5, 990-992.

[30] DOUC, R. et MATIAS, C. (2000). Propriétés As.symptotiquees de l'Estimateur de Maximum de Vraissemblance pour des Modèles de Markov Cachés Généraux. C. R. Acad. Scien.,. Paris, Tome 330. Serie I. Statistiques, 135-138.

[31] ELLIOT, R.J. (1995). Recursive Estimation for Hidden Markov Models : a Dependent cases ; Departement of Statistics and Applied Probability. Stochastic Analysis and Application. No. 13 (4), 437-460.

[32] ELLIOTT, R.J., AGGOUN, L. et MOORE, J. B. (1991). Hidden Markov Models : Estimation and Control. Edition Springer-Verlag, New Work.

[33] EPHRAIN, Y., DEMBO, A. et RABLNER, L R. (1987). A Minimum Discrimination Information Approach from Hidden Markov Modeling. IEEE International Conference on Acoustic, Speech and Signal Processing; 25-28.

[34] FELLER, W. (1958). An Introduction to Probability Theory and its Application. John Wiley. 2éme Edition. Vol. 1, New York.

[35] FORNEY, G., DAVID, J. R (1973). The Viterbi Algorithm ; Invited paper. Processing of the IEEE. Vol. 61. No. 3, 268-278

[36] GIUDICI and T. RYDÈN and P. VANDEKERKHOVE. (2000). Likehood-ratio tests for Hidden Markov Modéls. Biometrics 56,pp. 742-747.

[37] GUIKHMAN, I., SKOROP, A. (1980). Introduction a la Théorie de Processus Aléatoires. Edition MIR, Moscou, 41-42.

[38] HARTLEY, H.O. (1958). Maximum Likelihhod Estimation From Incomplete Data. Biometrics. No. 14, 174-194.

[39] HAUSLER, D. KROGH, A. MIAN, S. and SJÖlANDER. K. (1993). Protein Modeling Using Hidden Markov Models : Analysis of Globins. IEEE. Computer Society Press, vol. 1, pp 792-802.

[40] HUGHES, J.P. (1997). Computing the Observes Information in the Hidden Markov Model using EM Algorithm. Stat. Prob. Letters. No. 32. Department of Biostatistics. University of Washington, Seattle, 107-114.

[41] JAMSHIDIAN, M. et JENNRICH, R. (1993).Conjugate Gradient Acceleration of the EM Algorithm. Journal of the Americain Statistical Association. Vol. 88. No. 421, 221-228.

[42] JUANG, X.D., et JETACK, M. A. (1988). Semi-Continuous HMMs in Isolated Word Recognition. IEEE; International Conference on Pattern Recognition. Rome Italy, 406-406.

[43] CH. Kaddouri, B.Aissa, Compression des images fixes par fractrales basée sur la triangulation de launay et la quantification vectoielle, P.E.F.,USTHB,1998.

[44] KATZ, S.M. (1987). Estimation of Probabilities from Sparase data for the language Model Component of a Speech Recognizer. IEEE Transaction on Acoustic Speech and Signal Processing. Vol. ASSP 35. No. 3, 400-401.

[45] KERDEM, B. (1976). Sufficient Statistics Asssociated with a Two-state Second-Order Markov Chain. Biometrika. No. 63, 127-132.

[46] KLOTZ, J. (1973). Statistical Inference in Bernoulli Trials with Dependence. The Annals of Statistics. Vol. 1. No. 2, 373-379.

[47] Kodratoff, Y. (1986). Leçons d'Apprentissage Symbolique Automatique. Cépadues, Paris.

[48] Kodratoff, Y. (1988). Introduction to Machine Learning. Piuman Publishishing Co.

[49] KRIS, J. et Lars, K. ((1998). C /C++ La bible du programmeur. Edition Eyrolles.

[50] KRISHNAMURTHY, V., Moore, J.B. et Chung, S. (1993). Hidden Markov Model Signal Processing in Presence of Unknown Deterministic Interference. IEEE Transaction on automatic Control. Vol. 38. No.1, 146- 152.

[51] Laurière, J.-L. (1986). UN langage Déclaratif SNARK. TSI, Paris.

[52] LEROUX, B.G. (1992). Maximum Likelihood Estimation for Hidden Markov Models. Stochastic Processes and their Applications. No. 40, 127-143.

[53] LEVINSON, S.E., RABINER, L.R. et SONDHI, M.M. (1982). An Introduction to the Application of the Theory of Probabilistic Function of a Markov Process to Automatic Speech Recognition. The Bell System Technical Journal. Vol. 62. No. 4, 1035-1074.

[54] LOUIS, T. (1982). Finding the Observed Information Matrix when using the EM Algorithm. Journal of Royal Staistics Society Ser. B. No 44 (2), 226-233.

[55] LEVY, B.C., BENVENISTE, A. et NIKOUKHAH, R. (1996). High-Level Primitives for Recursive Maximum Likelihood Estimation. IEEE Transaction on Automatic Control. Vol. 41. No. 8, pp1125-1144.

[56] LJUANG, L. (1977). Analysis of Recursive Stochastic Algorithms. IEEE, Trans. on Autom. Control. Vol. AC-22. No. 4, 551-575.

[57] MARKOV, A.A. (1913). An Exemple of Statistical Investigation in the Text of "Eugene Onyegin" Illustrating Coupling of "Test" in Chains, Proce. of Academic Science St. Petersburg VI. Ser.Pp 153-162.

[58] MacDONALD, I.L. et ZUCCHINI, W. (1997). Hidden Markov Models for Discret Valued Time Series. Edition CHAPMAN & Hall / CRC. 137-206.

[59] MLONE, J.R., & McGregor, D.R. (1988). Intlligence architecture and infernce : VLSI generalized associative memory devices. In Hayes, J.E., D. Michie & J. Richards (Eds.). Machine Intelligence, Vol. 11 Logic and the Acquisition of Knowledge. pp. 333-343, Oxford University Press.

[60] MENG, X. et RUBIN, D. (1991). Using EM Algorithm to Obtain Asymptotic Variance-Covariance matrices : The SEM Algorithms. J. Americain Association. No 86 (416), 899-909.

[61] MINOUX, M. (1983). Programmation Matématique, Théorie et Algorithmes. Tome 2. Publication CNET et ENST. Edition Dunod, 107-154.

[62] Nicolas. J. (1986). Les stratégies de contrôle dans l'apprentissage à partir d'exemples .journées Françaises sur l'Apprentissage. JFA.

[63] OMURA, J.K. (1969). On the Viterbi Decoding Algorithm. IEEE Transaction on Information Theory, 177-179.

[64] PARZEN, E. (1962). On Estimation of Probability Density Function and Mode; Annals of Mathematical statistics. vol. 33, 1065-1076.

[65] PERENOU, G. (1994). Communication Orale ; Rapport Communication Orale Homme - machine. Université Paul Sabatier. IRIT Département CHMPT, URA CNRS. Toulouse.

[66] PETRIE, T. (1969). Probabilistic Function of Finite State Markov Chains. The Annals of Mathematical Statistics. Vol. 40. No. 1, 97-115.

[67] W. Pieczynski, "Mixture of distribution, Markov random fields and unsupervised baysien segmentation of images". Rapport technique n° 122, L.S.T.A,Université paris VI,1990.

[68] W. Pieczynski, "Parameter estimation in the case of hidden data". 16 th biennial Symposium on Communication, Kingstion,Canada, mai 1992.

[69] W. Pieczynski, "Champs de Markov cachés et estimation conditionnelle itérative". Traitement du Signal, Vol. 11 n° 2, 1994.

[70] Pitrat, (1985). Quand l'ordinateur apprend. Micro Système.

[71] PORITZ, P.B. (1988). Hidden Markov Models : A Guided Tour. IEEE International Conference on Acoustic, Signal and Speech Recognition., 7-13.

[72] Quinlan, J.R., (1990). Learning logical definitions from relations. Machine Learning, 5, pp. 239-266. Kluwer Academic Publishers, Boston, USA.

[73] RABINER, L.R. (1989). A tutorial on Hidden Markov Models and Selected Applica¬$tioninSpeechRecognition.IEEE, Proceedingof IEEE.Vol.77.No.2, 257-285$.

[74] RABINER, L.R. et JUANG, B.H. (1986). An introduction to Hidden Markov Models. IEEE ASSP Magasine, 4-16.

[75] REDNER, R. et WALKER, H. (1984). Mixture Densities, Maximum Likelihood and the EM Algorithm. SIAM Review. No. 26 (2), 195-239.

[76] ROBERT, C., CELEUX, G., et DIEBLOT, J. (1993). Bayesian Estimation of Hidden Markov Chaîns : A Stochastic Implementation. Statistic of Probability Letters. No 16, 77-83.

[77] ROLLAND, C. (1999). Latex par la pratique. Edition O'Reilly, Paris.

[78] RYDEN, T. (1994). Estimating the Order of HMMs; Statistic, N. 26, Lund Institute of Technology Sweden, Oversea Published Association, pp.345-354.

[79] RYDEN, T. (1995). On Recursive Estimation for Hidden Markov Models; Depar¬$tementofStatistics; universityofcalifornia.Evanshalls.CA94720USA.StochasticProcessesandt$ 96.

[80] SAT : Traitement en Analyse des Sequences; Action Informatique,Mathérmatiques et Physique pour le Génome. Universite d'Every-Val d'Essonne. 22-24 Novembre 2000.

[81] SHANNON, C.C. (1948). A Mathematical Theory of Communication. The Bell System Technical Journal. No. 27, 379-423 et 623-656.

[82] SHANNON C. C. (1951). Prediction and Entropy of Printed English. The Bell System Technical Journal. N. 30, 50-64.

[83] Simon, H.A. and Lea, G. (1983). Why should machines learn? Machine Larning, PP. 3-23, Morgan Kaufman Publishers, Los Altos, USA.

[84] M.Slimane, J.P.Asselin de Bauville, "Introduction aux modèles de Markov Cachées du premier ordre, première partie", Rapport interne, No.171,LI EIII, Tours,36p, 1994.

[85] Tutorial Stochastic Modeling Techniques : Understanding and Using Hidden Markov Midels ; cf leslie@sce.ucsc.edu et rph@cse.ucsc.edu.

[86] TOUIOUI, N. et SIAD, A. (1998) ; Reconnaissance de la Parole Basés sur les Modèles de Markov Cachés (Application a la langue arabe) ; Mêmoire de fin d'ètude. Institut National d'Informatique, 30-42.

[87] VITERBI, A.J. (1967). Error Bounds for Convolutionnal Codes and Assymptotically Optimum Decoding Algorithm. IEEE Transaction on Information Theory. Vol. IT-13. No. 2, 260-269.

[88] Waltz, D. (1984). L'Intelligence artificielle. In. L'Intelligence de l'Informatique. bibliothèque pour la Science. pp. 139-154

[89] WATERMAN, M.S. (1995). Introduction to Computational Biology. Edition Chapman & Hall, 261-262.

[90] WATERMAN, M.S. (1996). Introduction to Computational Biology : Maps, Sequence and Genomes. Edition Chapman & Hall / CRC, USA. .

[91] WELLEKENS, C.J. (1986). Global Connected Recognition using Baum-Welch Algorithm. IEEE International Conference On acoustic and Speech recognition. Tokyo 1986, 1081-1084.

[92] WU, C. (1983). On the Convergence Properties of the EM Algorithm. The anals of Statistics. No. 11 (1), 95-103.

[93] ZEGER, S.L. et QAQISH, B. (1988). Markov Regression Models for Time Series : A Quasi-likelihood Approach. Biometrics. No. 44, 1019-1031.

[94] ZHU, Q. (1991) ; Hidden Markov Model for Dynamic Obstacle Avoidance fo Mobile Robot Navigation. IEEE Transaction on Robotics and Automation. Vol. 7. No. 3, 390-397.