# Modeles de Markov caches : theorie et techniques de base : partie I

T. Alani, Hadj Guellif

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

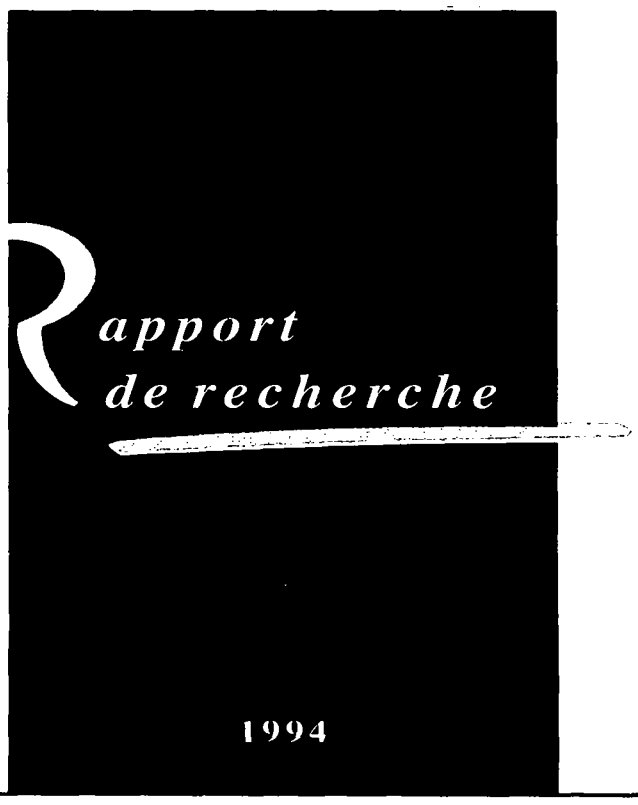
# *Modèles de Markov cachés théorie et techniques de base - Partie I*

Tarik ALANI
Hadj GUELLIF

**N° 2196**
Février 1994

PROGRAMME 5

Traitement du signal,
automatique et
productique



# HIDDEN MARKOV MODELS
# THEORY AND BASIC TECHNICS
## Part I

Tarik ALANI
Ecole Supérieure d'Ingénieurs en Electrotechnique et Electronique (E.S.I.E.E.)
Département Automatique
Cité DESCARTES 2 bd BLAISE PASCAL - B.P. 99
93162 NOISY-LE-GRAND CEDEX
Tél. (1) 45 92 65 98 Télex 231586F
Email : alanit@esiee.fr

Hadj GUELLIF
INRIA-Rocquencourt B.P. 105
78153 Le Chesnay Cedex
Tél. (1) 39 63 58 08 Télex 697 033F
Email : hadj.guellif@inria.fr

**Abstract**

This report aims at giving an introduction to the theory, the basic notions of the Hidden Markov Models (HMM) and an overview of recent advances in this domain. These models have been widely applied to Automatic Speech Recognition , where signals are encoded as temporal variation of a short time power spectrum, but still not popular in others fields like Pattern Recognition, Dynamic and Static Image Processing, Telecommunication, Robotics, Parameters Estimation of the Non-Stationnary Dynamic Models,...
An HMM is a double stochastic process with one underlying process that is not observable, but can only observed through another set of processes that produces a sequence of observations. These models are very rich in mathematical structure and have automatic training schemes that are corrective.

**Key-words** : Stochastic process; Markov chains; stochasic modeling.

# MODELES DE MARKOV CACHES
# THEORIE ET TECHNIQUES DE BASE
## Partie I

**Résumé**

Le but de ce rapport est d'introduire la théorie et les notions de base de Modèles de Markov Cachés "Hidden Markov Models (HMM)" et de donner un aperçu des progrès récents obtenus dans ce domaine. Ces modèles sont largement utilisés dans le domaine de la Reconnaissance Automatique de la Parole mais restent mal connus dans d'autres domaines tels que la Reconnaissance de Formes, Traitement d'Images Statiques et Dynamiques, Télécommunication, Robotique, Estimation des paramètres des systèmes dynamiques non stationnaires, etc.
Un modèle de Markov Caché est un processus doublement stochastique dont une composante est une chaîne de Markov non observable. Ce processus peut être observé à travers un autre ensemble de processus qui produit une suite d'observations. Les HMMs sont riches en propriétés et fondés sur des bases mathématiques solides. Ces modèles possèdent des techniques d'apprentissage automatique et correctif.

**mots-clés** : Processus stochastique; chaîne de Markov; modélisation stochastiques.

# Table des matières

# 1

# Introduction

La modélisation stochastique permet l'utilisation des modèles probabilistes pour traiter les problèmes à information incertaine ou incomplète. Ainsi, les modèles de Markov cachés connaissent actuellement un net regain d'intérêt tant dans ses aspects théoriques qu'appliqués. Actuellement, les modèles de Markov cachés sont largement appliqués dans les domaines de la reconnaissance de la parole, la reconnaissance de formes, le traitement d'images et des signaux, la robotique, ... Ces applications nécessitent une convergence d'approches du type compréhension, intelligence artificielle (IA), et celles du type traitement du signal, méthodes statistiques. Un modèle de Markov caché "Hidden Markov (HMM)" est caractérisé par un modèle Markovien à état fini et un ensemble de distributions de sortie. Les paramètres de transition dans la chaîne de Markov modélisent les variabilités temporelles, tandis que les paramètres des distributions de sortie modélisent les variabilités spectrales. Ces deux types de variabilités sont à la base de beaucoup de processus physiques tel que les signaux de la parole ou les signaux issus des systèmes dynamiques.
Les modèles de Markov Cachés doivent leurs succès à l'existance de plusieurs algorithmes élégants et efficaces. Pour l'apprentissage, l'algorithme "Forward-Backward" estime automatiquement et efficacement les paramètres de transition et de sortie. Cet algorithme est basé sur des concepts mathématiques rigoureux et est équivalent à l'algorithme " Expectation-Maximization (EM)". Ainsi, cet algorithme est très efficace et possède une complexité linéaire par rapport à la longueur de la suite de données du processus.
L'efficacité de l'algorithme "Forward-Backward" permet aux systèmes modélisés par des HMMs d'apprendre leurs paramètres à partir d'une grande base de données (corpus d'apprentissage).
L'inconvenient majeur de cet algorithme, est la supposition que les réalisations et leurs durées de rester dans un état dépendent seulement de l'état courant (processus markovien d'ordre un) et que cet état est conditionnellement indépendant de l'histoire finie du processus. Cependant, des solutions efficaces ont été proposées à ces problèmes (voir paragraphe 3.3.2).
La modélisation par HMM, comparée à d'autres types de modélisation, est plus générale et possède une base mathématique solide. Comparée aux approches à bases de connaissances, cette modélisation permet une intégration simple des sources des connaissances. Un des défauts de cette intégration est que les HMMs ne fournissent pas beaucoup plus de vision sur les processus de reconnaissance que les approches de (IA). Par conséquent, il est souvent difficile d'analyser les erreurs d'un système HMM dans le but d'améliorer ces

performances. Cependant, une incorporation prudente des connaissances peut améliorer les systèmes d'une façon significative.

## Organisation de ce rapport.

Au chapitre 2, nous avons rassemblé un nombre de définitions utilisées dans toute la suite de ce rapport; nous les avons unclu pour éviter au lecteur d'avoir à se repporter à d'autres ouvrages et pour préciser les notations que nous utilisons.
Au chapitre 3, sont présentées la théorie et les techniques de base des modèles de Markov cachés.
Le dernier paragraphe du chapitre 3 sera consacré à un panorama des principales variantes et améliorations des modèles de Markov cachés ainsi que leurs problèmes d'implementation.
Enfin, nous dégageons une conclusion sur ce thème.

# 2

# Processus stochastiques et chaîne de Markov discrète

## 2.1 Introduction

Les méthodes stochastiques dans le domaine de la modélisation des signaux présentent actuellement un net regain d'interêt. Ceci est dû principalement à deux raisons :

- La richesse de ces modèles réside dans leurs structures mathématiques, ce qui rend leur emploi utile dans certains domaines.
- En pratique, ces modèles, s'ils sont construits convenablement, donnent des résultats très intéressants.

Pour les modèles stochastiques, nous supposons que les processus physiques qui produisent l'observation peuvent être considérés comme un processus aléatoire, et que les paramètres de ce processus peuvent être estimés d'une façon précise et bien définie.
Le modèle décrit ces états à l'aide des probabilités de transition et des probabilités d'observation par état.
L'avantage des modèles Markoviens par rapport à d'autres types de modélisation stochastiques, réside dans le fait qu'ils sont fondés sur des bases théoriques fortes et rigoureuses.

## 2.2 Définitions

- Une source Markovienne peut être représentée par un ensemble de transitions, Fig. 2.1 :

  $$\Gamma = \{\tau_2, \tau_3, ..., \tau_T\}$$

  où $\tau_t$ est définie par la transition $q_{t-1} \Rightarrow q_t$, $t = 2, 3, ..., T$.

- Soient $Q$=$q_1, q_2, ..., q_T$, une suite d'états non directement observable (caché) et $O$=$\mathbf{O}_1, \mathbf{O}_2, ...$, une suite d'observations générée par un système. Alors la probabilité $p(O|Q)$ s'écrit:

$$p(O|Q) = p(O|\Gamma) = \prod_{t=1}^{T} p(\mathbf{O}_t|q_t) \tag{2.1}$$

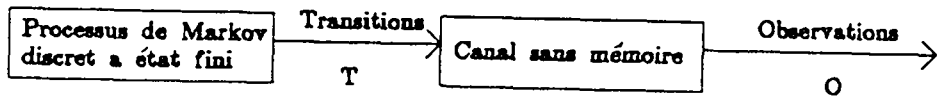



Figure 2.1: Source Markovienne.

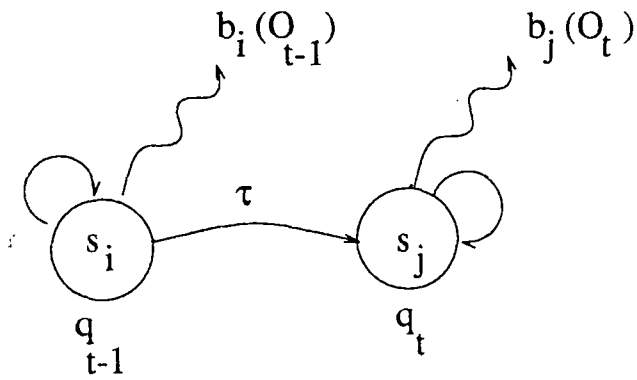



Figure 2.2: Cas où $\mathbf{O_t}$ dépend de $q_t$.

Deux cas peuvent se présenter:

- $\mathbf{O}_t$ dépend seulement de $q_t$, Fig. 2.2 :

$$p(O|Q) = p(O|\Gamma) = \prod_{t=1}^{T} p(\mathbf{O}_t|q_t) = \prod_{t=1}^{T} b_j(\mathbf{O}_t) \tag{2.2}$$

où

$$b_j(\mathbf{O}_t) = p(\mathbf{O}_t|q_t = s_j) \tag{2.3}$$

Dans certains systèmes (par exemple systèmes dynamiques à mémoire finie, systèmes de codage d'informations), l'état du système $q_t$ dépend essentiellement de l'évolution d'un paramètre stochastique $p_t \in \{0, 1, 2, ..., L-1\}$ sur une période $t = t-1, t-2, ..., t-m$ (le système possède une mémoire d'ordre m). Dans ce cas, on peut représenter le processus stochastique par un modèle du registre à décalage.

Modèle du registre à décalage, Fig.2.3

Dans ce cas, l'état non observable à l'instant t est défini par:

$$\Psi_t = f(p_{t-m}, ..., p_{t-1}),\ p_t \in 0, 1, 2, 3, ..., L-1 \tag{2.4}$$

où m est la longeur du registre à décalage. Soit la transition $\tau_t \Rightarrow \Psi_{t-1} \rightarrow \Psi_t$,

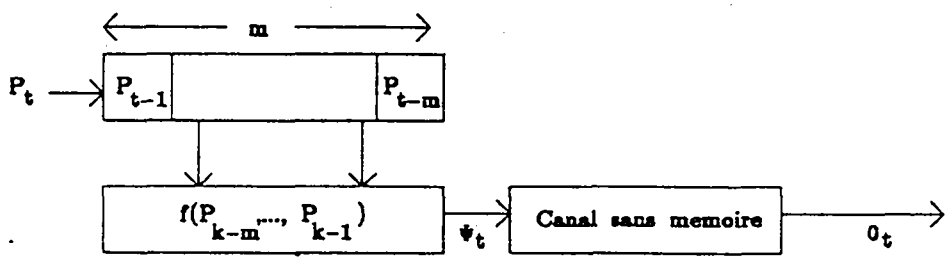



Figure 2.3: Modèle du registre à décalage.

la probabilité de transition correspondante est:

$$a_{ij} = p(\Psi_t = s_j | \Psi_{t-1} = s_i). \tag{2.5}$$

et dans ce cas

$$p(O|Q) = \prod_{t=1}^{T} p(\mathbf{O}_t | \Psi_t) \tag{2.6}$$

Le nombre tolal d' états, N, dans ce cas est égal à $L^m$.

- Dans certaines applications, la probabilité de l'observation dépend seulement de la transition $\tau_t$ $(q_{t-1} \rightarrow q_t)$, dans ce cas $b_j(\mathbf{O}_t)$ sera remplacé par, Fig. 2.4 :

$$b_{\tau_t}(\mathbf{O}_t) = p(\mathbf{O}_t | q_{t-1} = s_i, q_t = s_j)$$

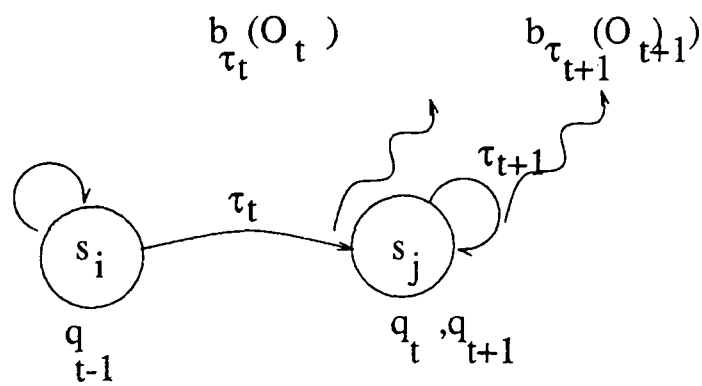



Figure 2.4: Cas où $\mathbf{O_t}$ dépend de $\tau_t$.

- Représentation du processus de Markov par un **graphe de transition d'état** ou par un **treillis** :
  L'évolution du processus de Markov peut être représenté soit par un graphe de transition d'état qui fait apparaître la structure de processus soit par un treillis décrivant l'évolution temporelle de ce processus (un état sera représenté par un nœud dans le diagrame de treillis), Fig. 2.5 et Fig. 2.6.

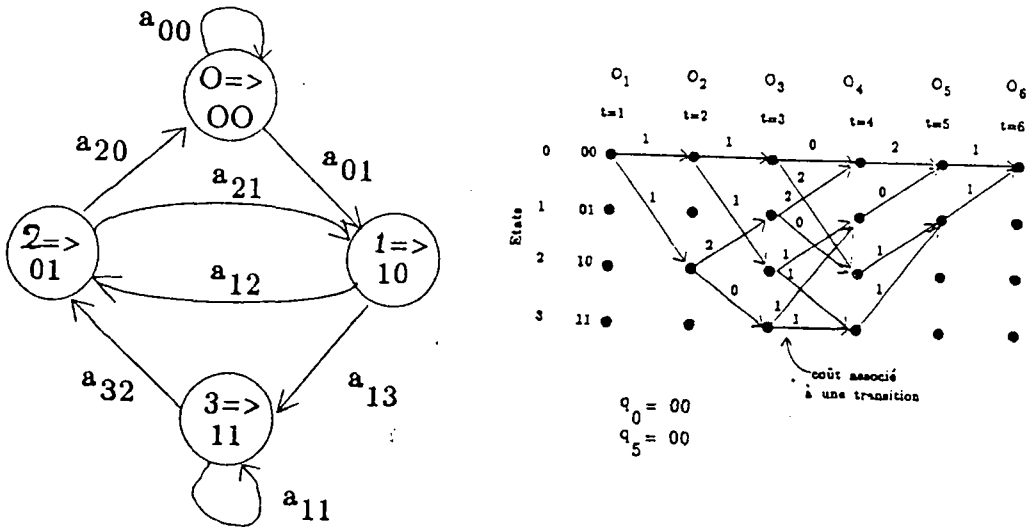
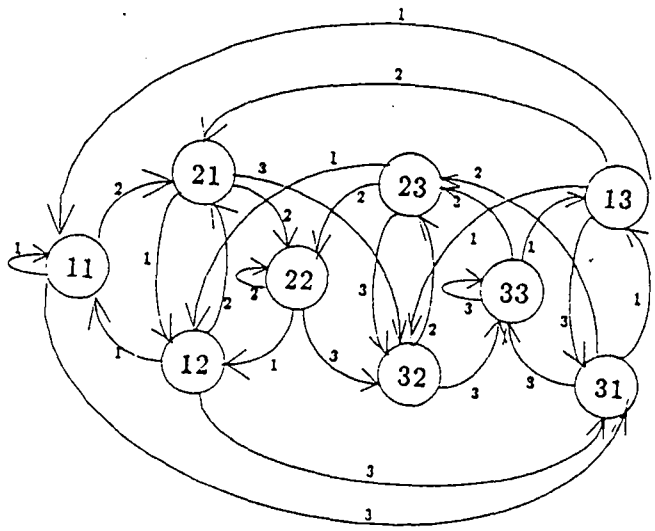



Figure 2.5: Modèle du registre à décalage, m=2, N=4, $p_t \in \{0, 1\}$.



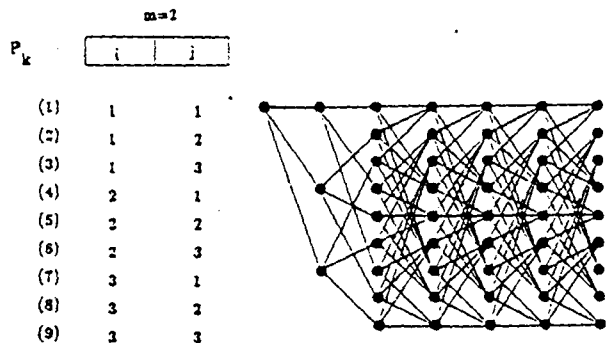



Figure 2.6: Modèle du registre à décalage, m=2, N=9, $p_t \in \{1, 2, 3\}$.

# 3

# Modèles de Markov cachés

Un modèle Markovien dans lequel chaque état correspond à un événement observable est très limité pour être appliqué à certains problèmes où les états ne sont pas directement observables. Nous étudions maintenant le principe de Markov pour inclure le cas où l'observation est une fonction probabiliste de l'état. Le modèle obtenu (Modèle de Markov Caché "Hidden Markov Modèle HMM") est un modèle doublement stochastique dont une composante est une chaîne de Markov non observable ( cachée ), mais elle peut être observée à travers un autre ensemble de processus stochastique qui produit une suite de symboles observables.

## 3.1 Historique

Voici l'historique de l'évolution des recherches sur les chaînes de Markov et leur applications dans des différents domaines :

- 1913 : La théorie des chaînes de Markov. Une première application a été développée par Markov pour analyser le langage [Mark 13];
- 1948..1951 : Les travaux de Shannon concernant la théorie de l'information utilisant les chaînes de Markov [Shan 48, Shan 51];
- 1957 : Les travaux de Bellman concernant la programmation dynamique [Bell 57];
- 1958..1961 : Les premières applications intéressantes. Par exemple :
  - Les modèles probabilistes d'urnes [Fell 58];
  - Calcul direct du maximum de vraisemblance [Hart 58];
  - L'observation de la suite d'états dans une chaîne de Markov [Bill 61].
- 1966..1974 : Dans cette période, des recherches très actives ont abouti à développer des algorithmes efficaces pour l'estimation des paramètres du modèle HMM d'une part et pour le décodage de la suite d'états cachés d'autre part :
  - Baum, Eagon, Soules, Weiss et Petrie [BaPe 66, BaEa 67, BaSe 68, Petr 69, BPSW 70, Baum 72] ont développé des algorithmes itératifs, basés sur le maximum de vraisemblance, pour l'estimation des états cachés et l'estimation des

paramètres du modèle. Des généralisations permettant d'inclure la notion de durée variable [Ferg 80b] et des densités de probabilités continues multivariables [Lipo 82] ont été développées.

- Des algorithmes efficaces de décodage utilisés en théorie de l'information ont été servis à l'estimation de la suite d'états cachés. Ces algorithmes ont été largement exploités au décodage de la parole. D'une part, l'algorithme de Viterbi [Vite 67, Forn 73], qui possède une complexité de calcul linéaire avec la longeur de la suite d'observations à décoder et qui permet d'estimer la suite des états du modèle correspondante au meilleur chemin. D'autre part, l'algorithme de Jelinek [Jeli 69] qui fait une exploration arborescente des chemins possibles en utilisant une pile.
- En 1970, l'utilisation des HMMs devient plus claire grâce à leur description par l'exemple des urnes de la part de Neuwirth qui a employé pour la première fois l'expression "Hidden Markov Model (HMM)" (modèle de Markov caché) au lieu de "probabilistic function of a Markov Chain" (fonction probabilistique d'une chaîne de Markov).

• 1975..1993 : Un très grand nombre d'applications, dominé par le domaine de la parole, a été développé. Ces applications ne se bornent pas, aux modèles Markoviens cachés de base, mais plusieurs notions, provenant des extensions théoriques de ces modèles et de leur variantes, ont été introduites dans le but d'améliorer les modèles (MMI, HMMC, mélange, durée, MDI, .. etc). Nous citons des exemples non exhaustifs :

- **Reconnaissance Automatique de la Parole (R.A.P.)**, [Rabi 89, Krio 90]:
  * 1975 : A cette époque les modèles HMM ont été introduits parallèlement par un groupe d'IBM [JeBM 75, BaJe 75] et par Baker à CMU [Bake 75a, Bake 75b, bake 75c] pour résoudre des problèmes de la R.A.P.
  * 1976..1982 : C'est une période de test et évaluation de cette technique pour la R.A.P. Nous citons par exemple :
    · Les travaux du groupe IBM-USA sur la R.A.P. continue [JeMB 76, Baki 76, Bahl 76, Bahl 79, Bahl 80, JeMe 80];
    · Les travaux sur l'apprentissage à partir des données insuffisantes [DeLR 77, JeMe 80].
  * 1983..1993 : Dans cette période, l'utilisation des HMMs, combinés parfois aux approches par réseaux de Neurone [BoWe 90], reste dominante, à titre d'exemple:
    · les mots isolés, en tant qu'alternative à la programmation dynamique [RaLS 83, Rabi 84, RJLS 86, JuRa 86, PoRi 86, Aver 86, EuWo 88];
    · les mots enchaînés [RaLe 85, RaWJ 86, RaWJ 87, MaRo 85];
    · la parole continue [BaJM 83, Bahl 89, Levi 87, LeHo 88, Lee 88, Schw 84, Schw 85, Chow 87, Kuba 88];
    · la localisation de mot (word spotting) [RoCo 87, DoPe 88, Dour 89, RRRG 89];
    · les petits vocabulaires [LeRS 83b, MaRo 85, RJLS 86, Well 86];

- les grands vocabulaires [MDES 87, Meri 88b];
- mono-locuteur [Meri 88, DeMe 89];
- multi-locuteurs [Boye 88];
- indépendantes du locuteur [RaLe 85, EuWo 88, Jouv 87, Lee 88].

- **Reconnaissance de l'écriture**, [NAWF 86, KuBa 88, VlKu 89, KuYP 89, KuHe 89, KuHB 91].
- **Modélisation des langages** , [CaNe 80, Katz 87].
- **Traitement d'images statiques et dynamiques** , [Devi 86, DeDe 87, XiNa 88, Devi 90, RaCh 91, Zhao 89, Veij 91].
- **Reconnaissance de formes** , [MaKu 90, YaAm 91a, YaAm 91b].
- **Robotique** , [ZHUQ 91, HaLe 91].
- **Théorie de codage** , [BCGR 74]
- **Traitement du signal** , [EsSh 77, RoRo 90, XiEv 91, CeCl 92, XiEv 93a, XiEv 93b, KrMC 93] .
- **Modélisation des finances** , [Cove 84].
- **Contrôle biologique** , [VaSK 85].
- **Biostatique** , [Ott 77].
- **Télécommunication** ,[KoVa 89, VALL 92a, VALL 92b].

## 3.2 Exemples introductifs

### 3.2.1 Exemple 1

Lancer de pièces, [RaJu 86]:

Le but de cette exemple est de représenter par un modèle de Markov le problème de lancée d'une pièce de monaie et d'expliquer la suite d'observations de " pile " ou de "face" obtenue. Il met en évidence ainsi:

- Le choix de la nature des états;
- Le choix du nombre d'états;
- La différence entre un modèle markovien caché et un modèle markovien observable;
- Le choix parmi plusieurs modèles possibles de celui qui répond le mieux à la suite d'observations en question.

Une série de "pile ou face" étant réalisée. L'opération de lancement et le nombre de pièces sont inconnus, seulement le résultat de chaque réalisation est révélé. Supposons que la suite d'observations obtenue est la suivante :

$$O = O_1 O_2 O_3 ... O_T = PPFFFPFFP...P$$

Plusieurs modèles markoviens peuvent être construits différemment:

- <u>Modèle 1</u>, Une pièce non biaisée Fig. 3.1

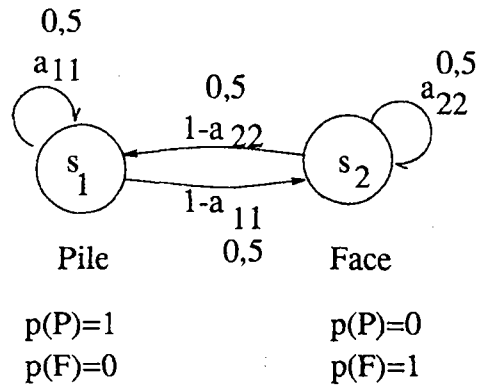



Figure 3.1: Un Modèle markovien observable (Modèle d'une pièce non biaisée), $O = PPFFFPFFFP...P$, $S$ = 11221211221...2.

  - Deux états : face 1 (pile) et face 2 (face);
  - Ce modèle est observable puisque la suite d'observations induit la suite d'états;
  - L'observation courante est non biaisée (les probabilités de pile ou de face dans ce cas p(P)=p(F)=0,5).

- <u>Modèle 2</u>, Une pièce biaisée Fig. 3.2

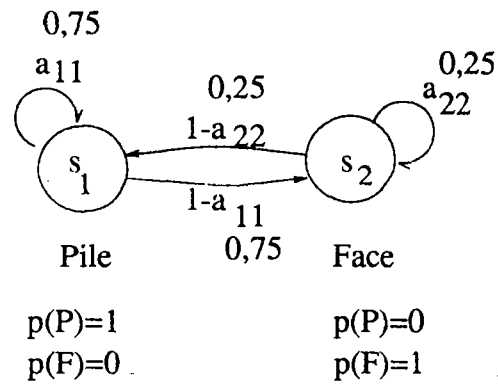



Figure 3.2: Un Modèle markovien observable (Modèle d'une pièce biaisée), $O = PPFFFPFFFP...P$, $S$ = 11221211221...2.

  - Deux états : face 1 (pile) et face 2(face);
  - Ce modèle est observable puisque la suite d'observations induit la suite d'états;
  - L'observation courante est donc biaisée, elle est en fonction de la valeur de biais (les probabilités de pile p(P) ou de face p(F) dans ce cas sont différentes).

- <u>Modèle 3</u>, Deux pièces non biaisées Fig. 3.3

  - Deux états : pièce 1 et pièce 2;

- Chaque état est caractérisé par une distribution de probabilité de piles ou de faces (p(P)=p(F)=0,5) et les transitions entre les états sont caractérisées par une matrice de transition d'états ($a_{ij} = 0,5, \forall i,j$). Dans ce cas, la statistique de la suite d'observations est indépendente de la matrice de transition. Ce modèle est donc caché puisqu'on ne peut pas savoir exactement quel est l'état qui a donné lieu à l'observation pile ou face. Malgré que ce modèle est caché, il est statistiquement indiscernable du modèle 1 vu précédement. La probabilité de transition est détérminée par le lancement d'une troisième pièce non biaisée utilisée indépendament des deux autres.

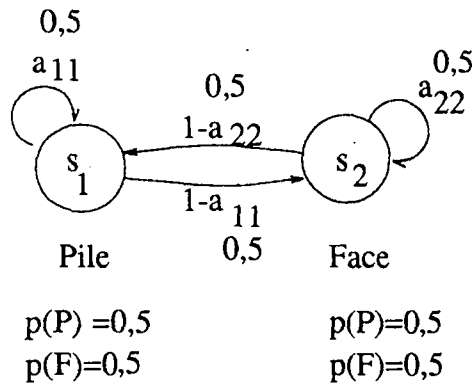



Figure 3.3: Un Modèle markovien caché (Modèle de deux pièces non biaisées), $O = PPFFFPFFFP...P$, $S = 21122212212...2$.

- <u>Modèle 4</u>, Deux pièces biaisées Fig. 3.4

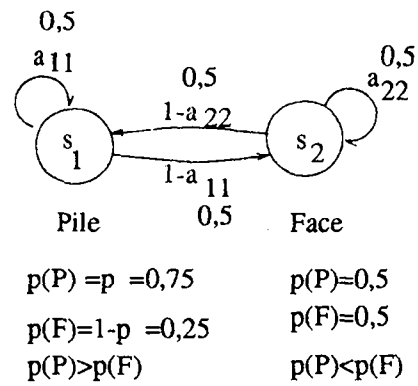



Figure 3.4: Un Modèle markovien caché (Modèle de deux pièces biaisées), $O = PPFFFPFFFP...P$, $S = 21122212212...1$.

- Deux états : pièce 1 et pièce 2;
- Chaque état est caractérisé par une distribution de probabilité de piles ou de faces ($p(P) \neq p(F)$) et les transitions entre les états sont caractérisées par une matrice de transition d'états ($a_{ij} = 0,5, \forall i,j$) grâce à une troisième pièce non biaisée ou autre événement probabiliste. Ce modèle est donc non observable. Il est intéressant de remarquer que la statistique à long terme de la suite

d'observations de ce modèle HMM est la même que celles des modèles 1,2 et 3.

- <u>Modèle 5</u>, Trois pièces biaisées Fig. 3.5

| | p(P) | p(F) |
|---|---|---|
| 1 | p =0,6 | $1\text{-}p_1$ =0,4 |
| 2 | p =0,25 | $1\text{-}p_2$ =0,75 |
| 3 | p =0,45 | $1\text{-}p_3$ =0,55 |

x

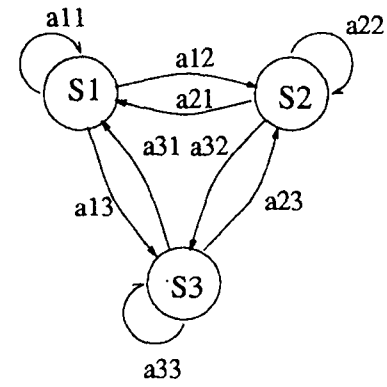



Figure 3.5: Un Modèle markovien caché (Modèle de trois pièces biaisées), $O = PPFFPFPPPFFP...F$, $S$ = 31233112313...2.

  - Trois états : pièce 1, pièce 2 et pièce 3;
  - Chaque état est caractérisé par une distribution de probabilité de piles ou de faces ($p(P) \neq p(F)$) et les transitions entre les états sont caractérisées par une matrice de transition d'états grâce à un autre événement probabiliste provenant d'un certain scénario.

Il est évident que le comportement de la suite d'observations produite par ce modèle, dépend fortement des probabilités de transition.
Nous pouvons conclure à partir de cet exemple que:

- L'étape la plus difficile est celle de déterminer le nombre d'états du modèle. Sans certaines connaissances a priori sur le modèle, cette décision est souvent difficile et demande plusieurs essais avant de décider du nombre d'états à prendre en considération.
  Théoriquement le modèle le plus approprié est celui qui possède le nombre d'états maximal (par conséquent le nombre maximal de paramètres) mais des considérations pratiques empêchent cette solution.
  Déterminer le nombre d'états réel du modèle nous amène à répondre à la question suivante: Comment peut-on choisir les paramètres du modèle (les probabilités de transition, les probabilité de symboles "pile ou face" dans chaque état) pour optimiser le modèle qui reflète de la meilleure façon la réalisation de la suite observée? La réponse à cette question fera l'objet du paragraphe 3.3.1.

- Un autre problème important concerne la longueur de la suite d'observations. Une suite plus longue permet une estimation plus fiable qu'une suite courte. La quantité de donnée d'apprentissage du modèle influe sur son optimalité.

### 3.2.2 Exemple 2

Modèle des urnes et des balles, [Pori 88, Ferg 80a, Krio 90] :
Cet exemple reflète parfaitement les deux composantes (l'état et l'observation) du processus stochastique d'un HMM.
Supposons que nous avons N urnes (états), Fig. 3.6 :

$$S = \{s_1, s_2, ..., s_N\}$$

Chaque urne a son propre mélange de balles colorées (symboles). Chaque balle peut être colorée avec M couleurs possibles ($1 \leq v_k \leq M$).
Soit $b_i(v_k)$, la fraction (la probabilité du symbole d'observation) dans l'urne (état) $s_i$, $1 \leq i \leq N$, où

$$\sum_{k=1}^{M} b_i(v_k) = 1, i = 1, 2, ..., N$$

Soit N+1 goblets:$G_0, G_1, ..., G_N$, chaque goblet a son propre mélange de pierres portant des marques. La marque sur une pierre est considérée comme étant "état 1" ou "état 2" ou ... ou "état N".
Soient $\pi_1, \pi_2, ..., \pi_N$ les fractions des pierres marquées "état i", $i = 1, 2, ..., N$, dans $G_0$.
Soient enfin $a_{1i}, a_{2i}..., a_{Ni}$ les fractions des pierres marquées "état i" respectivement dans $G_0, G_1, ..., G_N$
avec :

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

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

Générons une suite d'observations de couleurs de balles $O$=$\mathbf{O}_1\mathbf{O}_2...\mathbf{O}_T$, Fig. 3.6 .
Tirons aléatoirement une pierre du goblet $G_0$, sa marque est appelée "état i", $1 \leq i \leq N$.
Tirons ensuite une balle aléatoirement de l'urne i, sa couleur est $\mathbf{O}_1 = v_k, 1 \leq k \leq M$.
Maintenant, tirons une pierre aléatoirement du goblet $G_i$, sa marque est appelé "état j", $1 \leq j \leq N$. Continuons dans cette voie, en utilisant l'état courant pour obtenir à la fois l'observation courante et l'état suivant jusqu'à un total de T observations, Fig. 3.6. A chaque tirage dans une urne, la balle est remise dans la même urne. Le voile de Ferguson cache cet unique échantillonnage des goblets. L'observateur obtient seulement une information probabiliste concernant les pierres.
Ce mécanisme, génératif pour créer une suite d'observations est un processus stochastique avec une composante cachée: En générant la suite d'observations des couleurs O, une suite de pierres $Q = q_1q_2...q_T$ est aussi générée. Puisque la suite Q n'est pas observée, elle est alors une suite cachée (ou chemin caché).
Le paramètre vecteur du modèle stochastique est:

$$\lambda = [\pi_1\pi_2...\pi_N,\ a_{11}a_{12}...a_{NN},\ b_1(v_1)b_2(v_2)...b_N(v_M)]$$

Le vecteur de probabilité $\mathbf{\Pi} = [\pi_{01}\pi_{02}...\pi_{0N}]'$ est la distribution initiale des états.
La matrice stochastique $\mathbf{A} = [a_{ij}]$, où la ième rangée est associée au goblet i, est la matrice de transition d'états.
Ce modèle est un modèle markovien caché d'ordre un à N états. Il est un modèle du

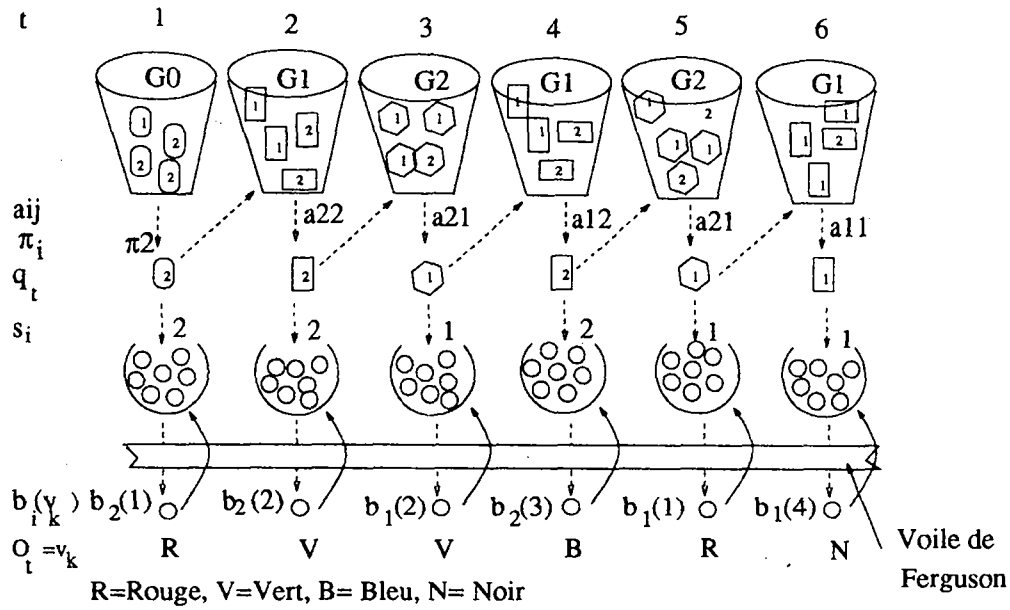



Figure 3.6: Un Modèle markovien caché (HMM) du premier ordre à deux états ($N = 2$) et un ensemble de symboles discrets appelé alphabet de sortie ($M = 4$, $v_k \in V = \{1, 2, 3, 4\}$). Les urnes sont supposées contenir un grand nombre de balles.

premier ordre puisque chaque état sélectionné comme une fonction probabiliste du dernier état prédécesseur.
Un modèle markovien caché est dit ayant un **alphabet de sortie** fini si les articles observés s'étendent à un ensemble fini de k éléments. A titre d'exemple:

- l'ensemble de couleurs de balles dans l'exemple des urnes;
- les vecteurs caractéristiques de l'alphabet d'une langue quelconque plus l'espace entre les mots;
- les symboles d'un dictionnaire de quantification vectorielle;
- primitives d'une image segmentée (primitives de contours et de regions);
- etc.

Pour chaque état i, le vecteur $\mathbf{b_i} = [b_i(1)b_i(2)...b_i(M)]'$ est appelé vecteur de probabilité de sortie pour l'état i. Ces probabilités de sortie tracent la suite d'états Q à partir de la suite d'observations O.
Alternativement, les observations peuvent s'étendre dans un ensemble continu (dénombrable fini).
Dans le choix de la sortie continue, chaque état i est associé à son propre paramètre de densité de probabilité $\mathbf{b_i}$.
Un HMM est représenté par son vecteur paramètre $\lambda = [\mathbf{\Pi}, \mathbf{A}, \mathbf{b_1}, \mathbf{b_2}, ..., \mathbf{b_N}]$.
Dans le cas d'un alphabet fini (discret), la matrice $\mathbf{B} = [\mathbf{b_1}, \mathbf{b_2}, ..., \mathbf{b_N}]$ s'appelle la matrice de probabilités d'observations et le modèle $\lambda$ devient $(\mathbf{\Pi}, \mathbf{A}, \mathbf{B})$.

## 3.3 Principe de Modèles de Markov Cachés (HMM's)

Les HMM's sont caractérisés par les paramètres suivants:

1. N, le **nombre des états** du modèle.
Nous désignons les états individuels par:

$$S = \{s_1, s_2, s_3, ..., s_N\} \tag{3.1}$$

et l'état au temps t par $q_t$, $q_t \in S$.

2. M, le **nombre des symboles d'observations distincts** dans le cas où l'observation $\mathbf{O_t}$ à la sortie physique du système est représentée sous forme discrète :
Ces symboles correspondent à la sortie physique du système. Nous désignons ainsi l'ensemble de symboles d'observation par:

$$\mathbf{O}_t = v_k,\ v_k \in V = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_M\} \tag{3.2}$$

3. **A**, la **distribution des probabilités des transitions des états**:

$$\mathbf{A} = \{a_{ij}\} \tag{3.3}$$

où

$$a_{ij} = p[q_{t+1} = s_j | q_t = s_i],\ 1 \leq i, j \leq N \tag{3.4}$$

et

$$\sum_{j=1}^{N} a_{ij} = 1,\ 1 \leq i \leq N \tag{3.5}$$

4. **B**, la **distribution des probabilités des observations dans chaque état j** :

$$\mathbf{B} = \{b_j(\mathbf{O}_t)\},\ j = 1, 2, ..., N \tag{3.6}$$

Dans le cas où l'observation est représentée sous forme continue, nous avons

$$\int_{-\infty}^{\infty} b_j(x)dx = 1,\ 1 \leq j \leq N \tag{3.7}$$

et dans le cas où l'observation est représentée sous forme discrète nous avons :

$$b_j(\mathbf{O_t} = v_k) = p[\mathbf{O_t} = \mathbf{v_k} | \mathbf{q_t} = \mathbf{s_j}],\ \mathbf{1 \leq j \leq N},\ \mathbf{1 \leq k \leq M} \tag{3.8}$$

avec

$$\sum_{k=1}^{M} b_j(\mathbf{O}_t = v_k) = 1,\ 1 \leq j \leq N \tag{3.9}$$

et dans ce cas **B** s'appelle la matrice de probabilités des symboles d'observations.

5. **Π**, la **distribution des probabilités initiales des états** :

$$\mathbf{\Pi} = \{\pi_i\} \tag{3.10}$$

où

$$\pi_i = p[q_1 = s_i],\ 1 \leq i \leq N \tag{3.11}$$

et

$$\sum_{i=1}^{N} \pi_i = 1 \tag{3.12}$$

On peut conclure que la spécification complète d'un HMM requiert:

- deux paramètres (N et M pour un HMM discret);
- définition des vecteurs d'observations;
- les distributions des probabilités **A**, **B** et $\mathbf{\Pi}$.

Nous désignons par:

$$\lambda = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi}) \tag{3.13}$$

pour indiquer un modèle complètement spécifié.

Etant donné des valeurs appropriées de N, M,**A**,**B** et $\mathbf{\Pi}$, le HMM peut être utilisé comme un générateur donnant une suite d'observations.

$$O = \mathbf{O}_1\mathbf{O}_2...\mathbf{O}_T \tag{3.14}$$

où

$$\mathbf{O}_t = v_k,\ v_k \in V,\ 1 \leq k \leq M \tag{3.15}$$

dans le cas où l'observation est représentée sous forme discrète.
T est le nombre d'observations dans la suite.
La procédure suivante génère une suite d'observations à partir d'un modèle HMM:

1. Choisir un état initial à l'instant t=1, $q_1 = s_i$ avec une distribution de l'état initial $\pi_i$;
2. Choisir $\mathbf{O}_t$ selon la distribution de probabilité de l'observation dans l'état $s_i$, c'est-à-dire $b_i(\mathbf{O}_t)$;
3. Si $t < T$ passer à un état $q_{t+1}=s_j$ avec la distribution de probabilité de transition d'état pour l'état $s_i$, c'est-à-dire $a_{ij}$;
4. Poser $t=t+1$.

Si $t \leq T$ retourner à l'étape 2
Sinon fin de la procédure.

### 3.3.1 Les trois problèmes fondamentaux d'un HMM

Etant donné un type de HMM, les trois problèmes à résoudre sont les suivants:

- Problème 1 : **Evaluation du modèle**

  Etant donné une suite d'observations $O = \mathbf{O}_1\mathbf{O}_2...\mathbf{O}_T$, et un modèle $\lambda = (\mathbf{\Pi}, \mathbf{A}, \mathbf{B})$, comment peut-on calculer efficacement $p(O|\lambda)$, la probabilité de la suite d'observation, sachant le modèle?

- Problème 2 : **Estimation de la suite d'états cachés**

  Etant donné une suite d'observations $\mathbf{O}_1\mathbf{O}_2...\mathbf{O}_T$, et un modèle $\lambda = (\mathbf{\Pi}, \mathbf{A}, \mathbf{B})$, comment peut-on choisir une suite d'états $Q = q_1q_2...q_T$ selon un critère convenable?

- Problème 3 : **Apprentissage**

  Comment peut-on ajuster le modèle $\lambda = (\mathbf{\Pi}, \mathbf{A}, \mathbf{B})$, pour maximiser $p(O|\lambda)$?

Pour comprendre comment ces trois problèmes se distinguent dans une application donnée, nous présentons un exemple d'application dans le domaine de la reconnaissance automatique de la parole pour des mots isolés.

Exemple d'application à la reconnaissance automatique de la parole des mots isolés, [Rabi 89]:

Soit $W=\{w_1 w_2 ... w_n\}$, un vocabulaire de mots. Pour chaque mot de w, nous construisons un modèle HMM de N états, Fig. 3.7.

Le signal de parole pour un mot donné est représenté, dans le temps, par une suite de

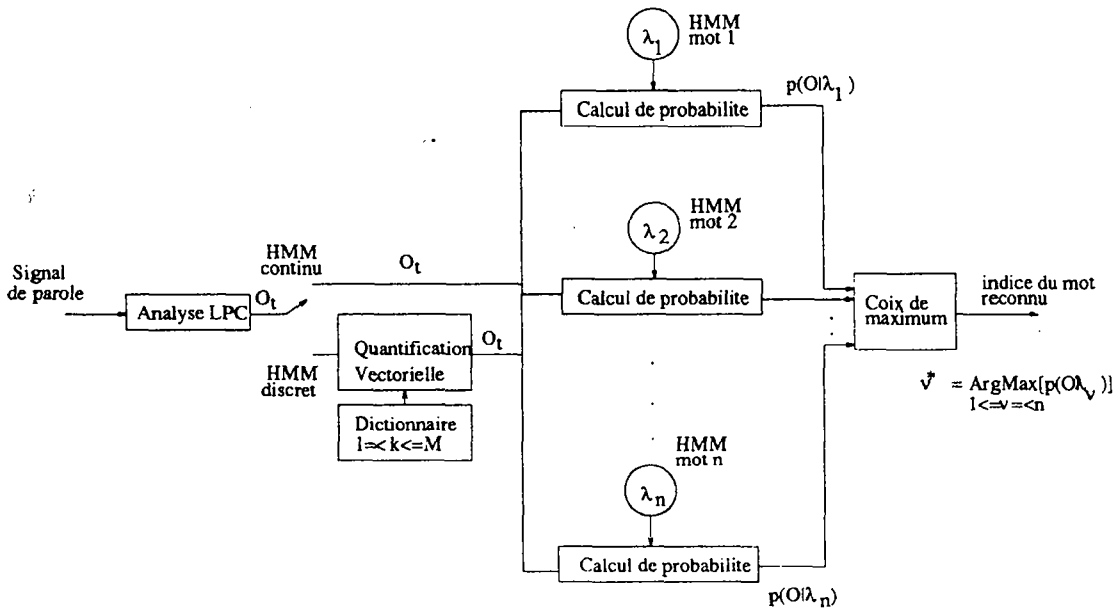



Figure 3.7: Reconnaissance de mots isolés.

vecteurs spectraux codés (coded spectral vectors) qui sont en fonction des coefficients de prédiction linéaire (LPC). Dans le cas où nous souhaitons construire un HMM discret , les vecteurs continus (vecteurs spectraux) sont remplacés par des dénsités discrètes de symboles d'observations. Nous supposons que le codage est réalisé en utilisant un dictionnaire (codebook) spectral composé de M vecteurs spectraux uniques. Chaque observtion est alors l'indice du vecteur spectral le plus proche (selon certain critère spectral) du signal de parole modèle. Cette technique de quantification vectorielle (QV) est une approche statistique qui consiste, dans l'espace des paramètres du signal, à définir à partir du nuage de points représentant les prononciations d'apprentissage par une technique d'analyse de données (clustring) quelques points (prototypes) représentatifs pour chaque mot à reconnaître [Gray 84].

La quantification vectoreille QV est bien adaptée aux systèmes multi-locuteurs, elle donne une représentation statistique de chaque mot du vocabulaire à partir des prononciations

d'un grand nombre de locuteurs. Elle est plutôt utilisée pour la reconnaissance de mots isolés. Cependant, par exemple, une extension à la reconnaissance de la parole continue était présentée dans [Gour 88].
Alors, pour chaque mot du vocabulaire, nous avons une séquence d'apprentissage composée d'un nombre de répétitons des suites d'indices du codebook correspondant aux mots prononcés (par plusieurs locuteurs).
Le système de reconnaissance automatique de la parole pour les mots isolés est construit en trois étapes:

- **Etape 1 Construction des modèles individuels pour chaque mot.** Cette étape est réalisée en utilisant la solution du problème 3 pour estimer d'une façon optimale, les paramètres du modèle de chaque mot.

- **Etape 2** Elle permet de **développer une connaissance du sens physique des états du modèle.** La solution du problème 2 est utilisée pour segmenter chacune des séquences des mots d'apprentissage en états, Fig. 3.8, puis étudier les propriétés des vecteurs spectraux qui donnent lieu aux observations émisent par chaque état. Le but ici est d'affiner le modèle (apporter des modifications sur le choix du nombre d'états, le choix de la dimension du codebook, etc.) pour améliorer sa capacité à modéliser le mot prononcé.

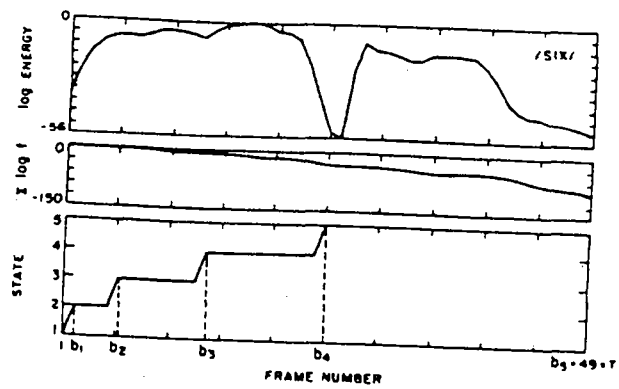



Figure 3.8: Exemple de segmentation du mot "six" en cinq états différents [Rabi 89].

- **Etape 3** Une fois les n modèles HMM construits et optimisés, la **reconnaissance du mot inconnu** est effectuée en utilisant la solution du problème 1 pour évaluer le modèle du chaque mot en se basant sur la suite d'observations courantes et de séléctionner le mot qui génère le meilleur score (Maximum de vraisemblance), Fig. 3.7.

#### 3.3.1.1 Problème 1: Evaluation du modèle

Etant donné une suite d'observations $\mathbf{O_1O_2...O_T}$, et un modèle $\lambda = (\mathbf{\Pi, A, B})$, comment peut-on calculer efficacement la probabilité que la suite d'observation O soit produite par $\lambda$, c'est-à-dire $p(O|\lambda)$. Autrement dit, comment évaluer le modèle afin de choisir parmi plusieurs celui qui génère le mieux cette suite d'observation. Plusieurs techniques permettent de résoudre ce problème: méthode d'évaluation directe, procédure "Forward-Backward" et Algorithme de Viterbi.

##### 3.3.1.1.1 Evaluation directe

La probabilité $p(O|\lambda)$ d'une suite d'observations O, sachant qu'un modèle $\lambda$ est donné, est la somme sur tous les chemins d'états, Q, possibles des probabilités conjointes de O et de Q par rapport à ce modèle:

$$p(O|\lambda) = \sum_Q p(O, Q|\lambda) = \sum_Q p(O|Q, \lambda)p(Q|\lambda)$$

où

$$Q = q_1 q_2 ... q_T,\ q_t = s_i,\ 1 \leq i \leq N$$

T est le nombre d'observations.

$$p(O|Q, \lambda) = b_{q_1}(\mathbf{O}_1)b_{q_2}(\mathbf{O}_2)...b_{q_T}(\mathbf{O}_T)$$

$$p(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} ... a_{q_{T-1} q_T}$$

$$p(O|\lambda) = \sum_Q \pi_{q_1} b_{q_1}(\mathbf{O}_1) a_{q_1 q_2} b_{q_2}(\mathbf{O}_2) a_{q_2 q_3} ... b_{q_{T-1}}(\mathbf{O}_{T-1}) a_{q_{T-1} q_T} b_{q_T}(\mathbf{O}_T)$$

- Initialement à $t = 1$ l'état initial est $q_1$ avec une probabilité $\pi_{q_1}$ et une observation $\mathbf{O_1}$ est générée avec une probabilité $b_{q_1}(\mathbf{O_1})$;
- à $t = t + 1$, $(t = 2)$, une transition est effectuée à l'état $q_2$ à partir de l'état $q_1$ avec une probabilité de transition $a_{q_1 q_2}$ et une observation $\mathbf{O_2}$ est générée avec une probabilité $b_{q_2}(\mathbf{O_2})$;
- Ce processus continue de la même manière jusqu'à la dernière transition $(t = T)$ de l'état $q_{T-1}$ à $q_T$ avec une probabilité de transition $a_{q_{T-1} q_T}$ et une observation $\mathbf{O_T}$ est générée avec une probabilité $b_{q_T}(\mathbf{O_T})$.

Pour calculer la probabilité $p(O|Q, \lambda)$ par cette méthode, il faut $(2T-1)N^T$ multiplications et $N^T - 1$ additions soit environ $2TN^T$ opérations. Cet ordre de calcul est non faisable même pour des petites valeurs de T et N. Par exemple, pour $N = 5$ et $T = 100$ on obtient environ $10^{72}$ opérations.

##### 3.3.1.1.2 Procédure Forward-Backward [Baum 67]

Dans cette approche, on considère que l'observation peut se faire en deux étapes : d'abord, l'émission de la suite d'observations $\mathbf{O_1O_2...O_t}$ et la réalisation de l'état $q_i$ au temps t, puis l'émission de la suite d'observations $\mathbf{O_{t+1}O_{t+2}...O_T}$ en partant de l'état $q_i$ au temps

t. Dans ce cas, l'évaluation de l'observation est:

$$p(O|\lambda) = \sum_i \alpha_t(i)\beta_t(i) \tag{3.16}$$

où $\alpha_t(i)$ est la probabilité d'émettre la suite $\mathbf{O_1O_2}...\mathbf{O_t}$ et d'aboutir à $q_i$ à l'instant t sachant le modèle, Fig. 3.9, et $\beta_t(i)$ est la probabilité d'émettre la suite $\mathbf{O_{t+1}O_{t+2}}...\mathbf{O_T}$ en partant de l'état $q_i$ à l'instant t sachant le modèle, Fig. 3.10.
Le calcul de $\alpha_t(i)$ se fait avec t croissant tandis que celui de $\beta_t(i)$ se fait avec t décroissant, d'où l'expression Forward-Backward.
Pour résoudre le problème 1, il suffit de calculer uniquement la partie Forward; le calcul de la partie Backward permettra de résoudre le problème 3.

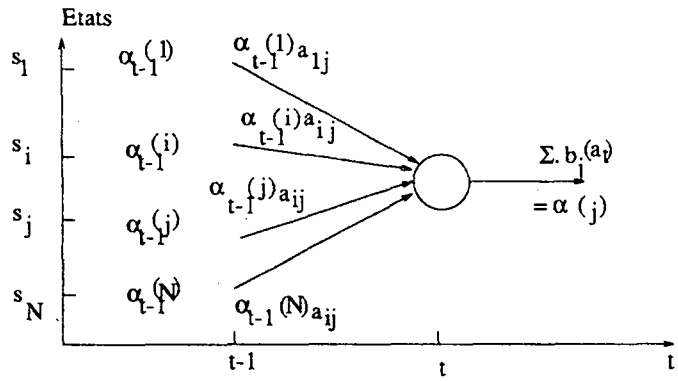



Figure 3.9: Suite partielle pour le calcul de $\alpha_t$.

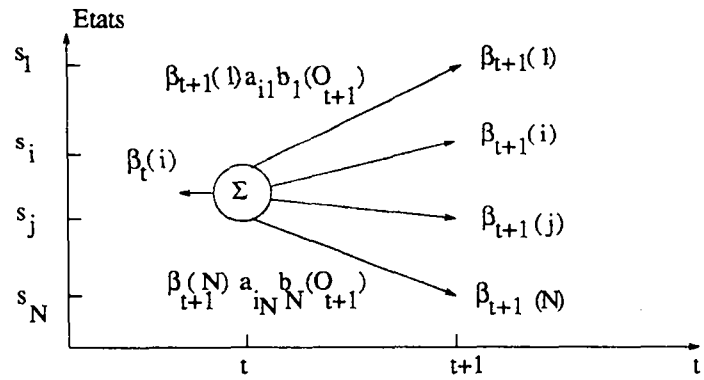



Figure 3.10: Suite partielle pour le calcul de $\beta_t$.

- **Calcul de $\alpha$**
  Soit la variable Forward $\alpha_t(j)$

$$\alpha_t(j) = p(\mathbf{O_1O_2}...\mathbf{O_t}, q_t = s_j|\lambda),\ 1 \leq j \leq N,\ 1 \leq t \leq T \tag{3.17}$$

## Algorithme Forward

1. Initialisation, $t = 1$

$$\alpha_1(i) = \pi_i b_i(\mathbf{O}_1),\ i = 1, 2, ..., N \tag{3.18}$$

Cette étape initialise la probabilité Forward. C'est la probabilité conjointe de l'état $s_i$, $i = 1, 2, ..., N$ et l'observation initiale $\mathbf{O}_1$.

2. Induction

$$\alpha_t(j) = [\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}] b_j(\mathbf{O}_t),\ j = 1, 2, ..., N,\ t = 2, 3, ..., T \tag{3.19}$$

Cette étape montre comment l'état $s_j$ peut être visité au temps $t + 1$ à partir de N états possibles $s_i, 1 \leq i \leq N$ au temps t.

3. Terminaison

$$p(O|\lambda) = \sum_{j=1}^{N} \alpha_T(j) \tag{3.20}$$

Pour calculer la probabilité de l'observation par cette méthode $N(N+1)(T-1)+N$ multiplications et $N(N-1)(T-1)$ additions soit environ $N^2T$ opérations sont effectuées. Par exemple, pour $N = 5$ et $T = 100$ on obtient environ 3000 opérations au lieu de $10^{72}$ opérations demandées par la méthode directe.
Toutes les transitions possibles entre les états peuvent être représentées sous forme de treillis, Fig. 3.11. Puisqu'il existe seulement N états (un nœud à chaque instant t), toutes les suites possibles d'états se fusionnent dans ces N nœuds quelque soit la longueur des suites d'observations.

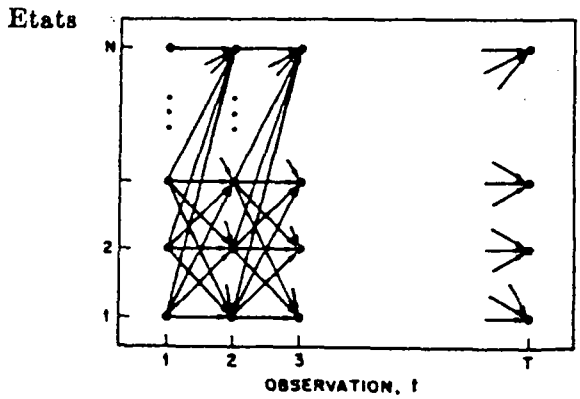



Figure 3.11: Implementation du calcule de $\alpha_t(i)$ ou $\beta_t(i)$ sous forme de treillis.

- **Calcul de $\beta$**
  Soit la variable Backward $\beta_t(i)$ définie par

$$\beta_t(i) = p(\mathbf{O}_{t+1}\mathbf{O}_{t+2}...\mathbf{O}_T|q_t = s_i, \lambda),\ 1 \leq i \leq N,\ T \leq t \leq 1 \tag{3.21}$$

**Algorithme Backward**

1. Initialisation, $(t = T)$

$$\beta_T(i) = 1,\ i = 1, 2, ..., N \tag{3.22}$$

Cette étape définie arbitrairement $\beta_T(i) = 1$ pour tous les états i.

2. Induction

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{O}_{t+1}) \beta_{t+1}(j),\ i = 1, 2, ..., N,\ t = T-1, T-2, ..., 1 \tag{3.23}$$

Pour être dans l'état $s_i$ au temps $t$, et pour tenir compte de la suite d'observation de $t+1$ à T, nous devons considérer tous les états possibles $s_j$ (toutes les transitions $a_{ij}$) aussi bien que l'observation $\mathbf{O}_{t+1}$ dans l'état j (les $b_j(\mathbf{O}_{t+1})$), puis de tenir compte de la suite d'observations partielle restante à partir de l'état j $(\beta_{t+1}(j))$.
Pour calculer la probabilité $p(O|\lambda)$ par cette méthode $N(N+1)(T-1)+N$ multiplications et $N(N-1)(T-1)$ additions soit environ $N^2T$ opérations sont effectuées. De même que l'algorithme Forward, toutes les transitions possibles entre les états peuvent être représentées sous forme de treillis, Fig. 3.11.

Les deux variables $\alpha_t(i)$ et $\beta_t(j)$ peuvent être utilisées pour calculer $p(O|\lambda)$ à chaque instant t, avec $1 \leq t \leq T$:

$$p(O|\lambda) = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i) \tag{3.24}$$

$$p(O|\lambda) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(\mathbf{O}_{t+1}) \beta_{t+1}(j) \tag{3.25}$$

Cette formule sera utilisée pour résoudre le problème 3.

Remarques

- la probabilité $p(O|\lambda)$ peut être calculée en posant $t = T$ dans l'équation (3.25) :

$$p(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.26}$$

- Les variables $\alpha_t$ et $\beta_t$ peuvent s'écrire sous forme matricielle :

Soit

$$\alpha_t = [\alpha_t(1)\, \alpha_t(2) \ldots \alpha_t(N)]' \tag{3.27}$$

et

$$\beta_t = [\beta_t(1)\, \beta_t(2) \ldots \beta_t(N)]' \tag{3.28}$$

alors

$$\alpha_t = \beta_t \mathbf{A}' \alpha_{t-1} \tag{3.29}$$

$$\mathbf{B}_t = \begin{pmatrix} b_1(\mathbf{O}_t) & & 0 & \\ & b_2(\mathbf{O}_t) & & \\ 0 & & \ddots & \\ & & & b_N(\mathbf{O}_t) \end{pmatrix} \tag{3.30}$$

$$\alpha_1 = \mathbf{B}_1\mathbf{\Pi} \tag{3.31}$$

$$\beta_t = \mathbf{A}\mathbf{B}_{t+1}\beta_{t+1} \tag{3.32}$$

$$\beta_T = \mathbf{1} \tag{3.33}$$

$$p(O|\lambda) = \beta_t'\alpha_t \tag{3.34}$$

Cas spécial:

$$t = 1 : p_1(O|\lambda) = \mathbf{\Pi}'\mathbf{B}_1\beta_1 \tag{3.35}$$

$$t = T : p_T(O|\lambda) = \mathbf{1}'\alpha_T = \mathbf{1}'\mathbf{B}_T\mathbf{A}'\mathbf{B}_{T-1}...\mathbf{A}'\mathbf{B}_1 \tag{3.36}$$

où (') signifie la transposée d'une matrice.
Dans chacune de ces formules la probabilité p peut être vue comme la trace d'une matrice $[1X1]$ qui est un produit de différentes matrices.

◇

#### 3.3.1.1.3 Algorithme de Viterbi [Vite 67]

Cette technique est basée sur l'algorithme de Viterbi qui fera l'objet du paragraphe suivant.

Algorithme

1. Initialisation, $t = 1$
Si $q_1$ est connu a priori ($q_1 = s_i$), alors

$$\delta_1(i) = 0$$

Autrement

$$\delta_1(i) = \pi_i b_i(\mathbf{O}_1)),\ i = 1, 2, ..., N \tag{3.37}$$

2. Induction, $t = 2, 3, ..., T$

$$\delta_t(j) = Max_{1\leq i\leq N}[\delta_{t-1}(i)a_{ij}]b_j(\mathbf{O}_t),\ 1 \leq j \leq N \tag{3.38}$$

3. Terminaison

$$p(O|\lambda) = \sum_{j=1}^{N} \delta_T(i) \tag{3.39}$$

### 3.3.1.2 Problème 2 : Estimation de la suite cachée

Etant donné une suite d'observations $\mathbf{O}_1\mathbf{O}_2...\mathbf{O}_T$, et un modèle $\lambda$, Comment peut-on choisir une suite d'états Q=$q_1q_2...q_T$ 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 possibles. Selon le choix du critère nous proposons trois solutions:

- **Estimation de l'état $q_t$ indépendamment des autres états**

Cette technique consiste à choisir l'état $q_t$ qui est la plus probable et ceci indépendamment des autres états; ce qui revient à choisir au temps t l'état qui maximise $p(q_t = s_i|O,\lambda)$. Ce critère d'optimalité permet de maximiser le nombre éspéré des états indépendants. L'un des problèmes posé par Baum [Baum 72] était de calculer l'estimation de $q_t$ pour $1 \leq t \leq T$, basée sur la réalisation de la suite d'observations O.
Sous le critère de la probabilité d'erreur minimale, il serait nécessaire de déterminer soit la vraisemblance conjointe:

$$\vartheta_t(s_i) = p(\mathbf{O}_1\mathbf{O}_2...\mathbf{O}_T, q_t = s_i|\lambda),\ i = 1,2,...,N \tag{3.40}$$

soit les probabilités a posteriori

$$\begin{aligned}
\tilde{\vartheta}_t(s_i) &= p(q_t = s_i|\mathbf{O}_1\mathbf{O}_2...\mathbf{O}_T,\lambda),\ i = 1,2,...,N,\ t = 1,2,...,T && (3.41)\\
&= p(q_t = s_i|\mathbf{O}_1\mathbf{O}_2...\mathbf{O}_t\mathbf{O}_{t+1}...\mathbf{O}_T,\lambda)\\
&= p(q_t = s_i, \mathbf{O}_1\mathbf{O}_2...\mathbf{O}_t|\lambda)p(\mathbf{O}_{t+1}...\mathbf{O}_T|q_t = s_i,\lambda)\\
&= \alpha_t(i)\beta_t(i),\ t = 1,2,...,T && (3.42)
\end{aligned}$$

On peut écrire

$$\gamma_t(i) = \frac{\tilde{\vartheta}_t(s_i)}{p(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)} \tag{3.43}$$

Le facteur de normalisation $p(O|\lambda)$ fait que

$$\sum_{i=1}^{N}\gamma_t(i) = 1 \tag{3.44}$$

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

$$q_t = argMax_{1\leq i\leq N}[\gamma_t(i)] \tag{3.45}$$

Remarques

- Les variables $\alpha$ sont calculées et stockées de façon récursive. Elles sont utilisées ensuite pendant l'étape de régression "Backward" pour calculer $\tilde{\vartheta}_t(s_i))$=$\alpha_t(i)\beta_t(i)$, $i$=$1,2,...,N$ et $t$=$T, T-1,...,1$.
- Il est possible de résoudre le problème 1 vu précédemment par la formule suivante:

$$p(O|\lambda) = \sum_{i=1}^{N}\alpha_T(i) = \sum_{i=1}^{N}\tilde{\vartheta}_t(i) \tag{3.46}$$

◇.

Bien que l'équation (3.45) maximise le nombre éspéré des états individuels en séléctionnant l'état le plus vraisemblable à chaque instant, cependant on peut avoir quelques

problèmes relatifs à la suite d'états produite par cette équation. Ainsi, quand 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 dûe à la solution de l'équation (3.45) qui détermine l'état le plus vraisemblable à chaque instant sans prendre en compte la probabilité d'occurence des suites d'états.

- **Prise en compte des transitions deux à deux ou trois à trois entre les états**

Dans certaines applications, nous choisissons des états qui ont le plus de chance deux à deux ou trois à trois. L'inconvenient de cette approche est qu'une partie des contraintes de transitions entre états ne sera pas prise en compte [Rabi 88].

- **Algorithme de Viterbi [Vite 67]**

Le critère le plus utilisé est celui de trouver l'unique trajectoire optimale de la suite d'états, c'est-à-dire Maximiser $p(Q|O,\lambda)$ ou Maximiser $p(Q,O|\lambda)$. Une technique formelle pour trouver le chemin optimal est basée sur les méthodes de programmation dynamique, c'est l'algorithme de Viterbi.
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 une trajectoire unique et optimale de la suite d'états, $Q = q_1 q_2 ... q_T$ produisant la suite d'observations $O = \mathbf{O}_1\mathbf{O}_2...\mathbf{O}_T$ nous définissons la quantité

$$\delta_t(i) = Max_{q_1 q_2 ... q_{t-1}} \ln p(q_1 q_2 ... q_t = i, \mathbf{O}_1\mathbf{O}_2...\mathbf{O}_t|\lambda),\ t \geq 2 \qquad (3.47)$$

qui représente le meilleur score (la probabilité maximale) correspondant à une trajectoire unique jusqu'au temps t et qui prend en compte les premieres "t-observations" et s'arrête à l'état $s_i$.
Par itération

$$\delta_t(j) = Max_{1 \leq i \leq N}[\delta_{t-1}(i) + ln a_{ij}] + ln b_j(\mathbf{O}_t),\ 1 \leq j \leq N \qquad (3.48)$$

Pour retrouver la suite optimale d'états, nous devons garder une trace des arguments qui maximise l'équation (3.48) pour chaque t et j.

### Principe de l'Algorithme de Viterbi

Soit la suite d'observation $O = \mathbf{O}_1\mathbf{O}_2...\mathbf{O}_T$, comment trouve-t-on une suite d'états $Q = q_1 q_2 ... q_T$ qui soit optimale en certain sens?
La réponse à cette question consiste à maximiser la probabilité conjointe $p(O,Q)$ :

$$Max_Q ln p(O,Q) \Rightarrow Q_{optimal} \qquad (3.49)$$

$$\begin{aligned} p(O,Q) &= p(Q)p(O|Q) \\ &= p(q_1 = s_l)p(\mathbf{O}_1|q_1 = s_l)\prod_{t=2}^{T} p(q_t = s_j|q_{t-1} = s_i)\prod_{t=2}^{T} p(\mathbf{O}_t|q_t = s_j) \\ &= \pi_l b_l(\mathbf{O}_1)\prod_{t=2}^{T} a_{ij} b_j(\mathbf{O}_t) \end{aligned} \qquad (3.50)$$

avec

$$\leq l \leq N, \leq i \leq N, \leq j \leq N$$

On a alors,

$$\ln p(O, Q) = ln(\pi_1 b_l(\mathbf{O}_1)) + \sum_{t=2}^{T} \delta(q_t = s_j) \tag{3.51}$$

qui représente le côut total pour le chemin Q, où $\delta$ est le côut d'un segment (une transition d'un état à un autre) de chemin Q :

$$\delta(q_t = s_j) = lna_{ij} + lnb_j(\mathbf{O}_t) \tag{3.52}$$

Nous définissons $\Psi_t(j)$ comme étant le chemin le plus court correspondant au nœud $q_t = s_i$ (surviveur). A chaque instant t, il existe N surviveurs (un pour chaque nœud). L'algorithme nécessite, à chaque instant t, la mémorisation de ces N surviveurs ainsi que leurs coûts.

Algorithme

1. Initialisation, t=1
Si $q_1$ est connu a priori, alors

$$\delta_1(i) = 0, \forall i \quad \text{(coût du surviveur i)} \tag{3.53}$$
$$\Psi_i = i \quad \text{(cette variable stocke l'état optimal à l'instant t)} \tag{3.54}$$

Autrement,
Si $q_1$ est inconnu a priori
Alors

$$\delta_1(i) = \ln(\pi_i b_i(\mathbf{O}_1)), i = 1, 2, ..., N \tag{3.55}$$
$$\Psi_i = 0 \tag{3.56}$$

2. Induction

$$\delta_t(j) = Max_{1 \leq i \leq N}[\delta_{t-1}(i)]b_j(\mathbf{O}_t), 1 \leq j \leq N, 2 \leq t \leq T \tag{3.57}$$
$$\Psi_t(j) = argMax_{1 \leq i \leq N}[\delta_t(i) + \ln a_{ij}], 1 \leq j \leq N, 2 \leq t \leq T \tag{3.58}$$

3. Terminaison

$$\ln p^* = Max_{1 \leq i \leq N}[\delta_T(i)] \tag{3.59}$$
$$q_T^* = argMax_{1 \leq i \leq N}[\delta_T(i)] \tag{3.60}$$

Chemin obtenu "Backtracking"

$$q_t^* = \Psi_{t+1}(q_{t+1}^*), t = T-1, T-2, ..., 1 \tag{3.61}$$

Remarques

- Dans certaines applications, la probabilité de l'observation dépend seulement de la transition $\tau_t \Rightarrow (q_{t-1} = s_i \rightarrow q_t = s_j)$ (voir paragraphe 2.2), dans ce cas $b_j(\mathbf{O}_t)$ est remplacée par $b_{ij}(\mathbf{O}_t)$ avec

$$b_{ij}(\mathbf{O}_t) = p(\mathbf{O}_t | q_{t-1} = s_i, q_t = s_j) \tag{3.62}$$

alors

$$p(O, Q) = ln\pi_1 + \sum_{t=2}^{T} \lambda(q_{t-1} = s_i, q_t = s_j) \tag{3.63}$$

avec

$$\lambda(q_{t-1} = s_i, q_t = s_j) = \ln a_{ij} + lnb_{ij}(\mathbf{O}_t) \tag{3.64}$$

Dans ce cas, il faut imposer $\delta_1(i) = 0$ dans l'équation (3.57) et utiliser $b_{ij}(\mathbf{O}_t)$ au lieu de $b_j(\mathbf{O}_t)$ dans cette équation.

- Si $q_1$ est connu ($q_1 = s_i$), alors $\pi_i = 1$.
- Si $a_{ij}$ est une constante, $\forall i, j$, ou inconnue a priori, alors le terme $\ln a_{ij}$ dans l'équation (3.58) est négligé.
- Une fois les $\delta_t(i)$ sont calculées, il n'est plus nécessaire de stocker l'observation $\mathbf{O}_t$ ce qui rend l'algorithme de Viterbi utilisable en temps réel.
- Compléxité de l'algorithme:
  * L'algorithme nécessite N cases mémoire, une case par état permettant de stocker $\delta_t(i)$ et $\Psi_t(q_t)$ de longueurs T symboles.
  * A chaque instant t, l'algorithme effectue $|\Gamma|$ additions (une addition pour chaque transition) et N comparaisons parmi le $|\Gamma|$ résultats.
  * Si l'algorithme de Viterbi est utilisé pour un processus du registre à décalage (voir paragraphe 2.2), alors $|\Gamma| = L^{m+1}$. La complexité augmente ainsi d'une manière exponentielle en fonction de la longueur du registre à décalage.

◇

### 3.3.1.3 Problème 3 : Optimisation des paramètres du modèle (Apprentissage).

Comment peut-on ajuster les paramètres du modèle $\lambda$=($\mathbf{\Pi}, \mathbf{A}, \mathbf{B}$) pour maximiser $p(\mathbf{O}_t|\lambda)$? Le fait que la longueur de la suite d'observations (données d'apprentissages) est finie, il n'exite pas de solutions analytiques directes (d'optimisation globale) pour construire le modèle.
Néanmoins, nous pouvons choisir $\lambda = (\mathbf{\Pi}, \mathbf{A}, \mathbf{B})$ tel que $p(O_t|\lambda)$ est un maximum local en utilisant une procédure itérative telle que celle de BAUM-WELCH [BaEa 67, Baum 72] (ou d'une façon équivalente l'algorithme d'identification de mélange de type EM (E pour Expectation , M pour Maximization) [DeLR 77] ou en utilisant les techniques de gradient telle que la méthode de Liporace [Lipo 82].
L'idée de l'application est donc d'utiliser des procédures de ré-estimation qui affinent le modèle petit à petit en suivant les étapes suivantes:

- Choisir un ensemble initial de paramètres $\lambda_0$;
- calculer $\lambda_1$ à partir de $\lambda_0$;

- répéter ce processus jusqu'à un critère de fin.

#### 3.3.1.3.1 Théorèmes de Baum

**Théorème 1** [BaEa 67, Baum 72]

Soit $\lambda = \{\pi_i, a_{ij}, b_j(\mathbf{O}_t = v_k)\}$ et soit $\tilde{\lambda} = \Upsilon\{\pi_i, a_{ij}, b_j(\mathbf{O}_t = v_k)\}$ est une transformation de l'ensemble

$$D = \{\lambda, \pi_i \geq 0, a_{ij} \geq 0, b_j(\mathbf{O}_t = v_k) \geq 0, \sum_{i=1}^{N} \pi_i = 1, \sum_{j=1}^{N} a_{ij} = 1,\ 1 \leq i \leq N, \sum_{k=1}^{M} b_j(\mathbf{O}_t = v_k) = 1\}$$

dans lui même. Alors la probabilité de l'observation

$$p = p(O|\lambda) = \sum_{Q} \pi_{q_1} b_{q_1}(\mathbf{O}_1) a_{q_1 q_2} b_{q_2}(\mathbf{O}_2) ... a_{q_{T-1} q_T} b_{q_T}(\mathbf{O}_T)$$

satisfait l'inégalité suivante :

$$p(O|\tilde{\lambda}) > p(O|\lambda)$$

sauf si $\tilde{\lambda} = \lambda$, qui est vrai si et seulement si $\lambda$ est un point critique de p.

**Théorème 2** [BaEa 67, Baum 72]

La probabilité $p(O|\lambda)$ s'exprime comme un polynome homogène à coefficients positifs.

Soit le polynôme homogène suivant:

$$p(Z_1, Z_2, ..., Z_n) = \sum_{\nu_1 \nu_2, ..., \nu_n} c_{\nu_1 \nu_2, ..., \nu_n} Z_1^{\nu_1} Z_2^{\nu_2}, ..., Z_n^{\nu_n}$$

avec

$$c_{\nu_1 \nu_2, ..., \nu_n} \leq 0$$

et

$$\sum_{\nu_1 \nu_2, ..., \nu_n} \nu_i = d$$

alors la transformation:

$$\Upsilon(Z_i) = \frac{Z_i \frac{\partial p}{\partial Z_i}}{\sum_j Z_j \frac{\partial p}{\partial Z_j}}$$

transforme l'ensemble:

$$D = \{Z_i, Z_i \geq 0, \sum_i Z_i = 1\}$$

dans lui même et vérifie:

$$p((\Upsilon(Z_i))_{i=1,...,n}) \geq p((Z_i)_{i=1,...,n})$$

L'inégalité est stricte sauf si $\{Z_i\}$ est un point critique de p dans D.
Pour le polynôme $p(O|\lambda)$, les $Z_i$ sont les $\pi_i$, $a_{ij}$ et les $b_j(k)$. Les $\Upsilon(Z_i)$ sont alors les $\tilde{\pi}_i$, $\tilde{a}_{ij}$ et les $\tilde{b}_j(k)$.

D'après le théorème 2, deux cas possibles peuvent se présenter :

- Le modèle initial, $\lambda = (\mathbf{\Pi}, \mathbf{A}, \mathbf{B})$, définit un point critique (ce qui est en général rare) de la fonction de vraisemblance $p(O|\lambda)$. Dans ce cas, nous avons: $\tilde{\lambda} = \lambda$;
- Le modèle ré-estimé, $\tilde{\lambda} = (\tilde{\mathbf{\Pi}}, \tilde{\mathbf{A}}, \tilde{\mathbf{B}})$ est plus probable que le modèle $\lambda$ dans le sens de l'inégalité du théorème 1:

$$p(O|\tilde{\lambda}) > p(O|\lambda)$$

De cette manière, nous avons trouvé un modèle à partir duquel la suite d'observations a plus de chance d'être produite.

**Théorème 3** [BPSW 70, Baum 72]

Soit R la fonction auxiliaire de $\tilde{\lambda}$ définie par

$$R(\lambda, \tilde{\lambda}) = \sum_{Q} p(O, Q|\lambda) log p(O, Q|\tilde{\lambda})$$

alors, dans le cas où $R(\lambda, \tilde{\lambda}) \geq (\lambda, \lambda)$ alors

$$p(O|\tilde{\lambda}) > p(O|\lambda)$$

sauf pour le point critique $p(O|\tilde{\lambda}) = p(O|\lambda)$.

#### 3.3.1.3.1.1 Méthode de BAUM-WELCH basée sur l'estimation par le maximum de vraisemblance, "Maximum Likelihood (MLE)" [BaEa 67, Baum 72]

Soit:

$$\zeta_t(i,j) = p[q_t = s_i, q_{t+1} = s_j|O, \lambda],\ t = 1, 2, 3, ..., T-1 \tag{3.65}$$

la probabilité de visiter l'état $s_i$ au temps t et l'état $s_j$ au temps t+1, sachant le modèle et la suite d'observations $O = \mathbf{O}_1\mathbf{O}_2...\mathbf{O}_T$, Fig. 3.12 .
Nous pouvons écrire:

$$\zeta_t(i,j) = \frac{p[q_t = s_i, q_{t+1} = s_j|O, \lambda]}{p(O|\lambda)},\ t = 1, 2, 3, ..., T-1 \tag{3.66}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(\mathbf{O}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(\mathbf{O}_{t+1}) \beta_{t+1}(j)} \tag{3.67}$$

et on peut démontrer que la formule (3.43) peut être écrite :

$$\gamma_t(i) = \sum_{j=1}^{N} \zeta_t(i,j),\ t = 1, 2, ..., T-1 \tag{3.68}$$

On peut remarquer que le nombre éspéré des transitions à partir de $s_i$ est donné par la formule suivante :

$$\gamma_i = \sum_{t=1}^{T-1} \gamma_t(i) \tag{3.69}$$

$$= \sum_{j=1}^{N} \gamma_{ij} \tag{3.70}$$

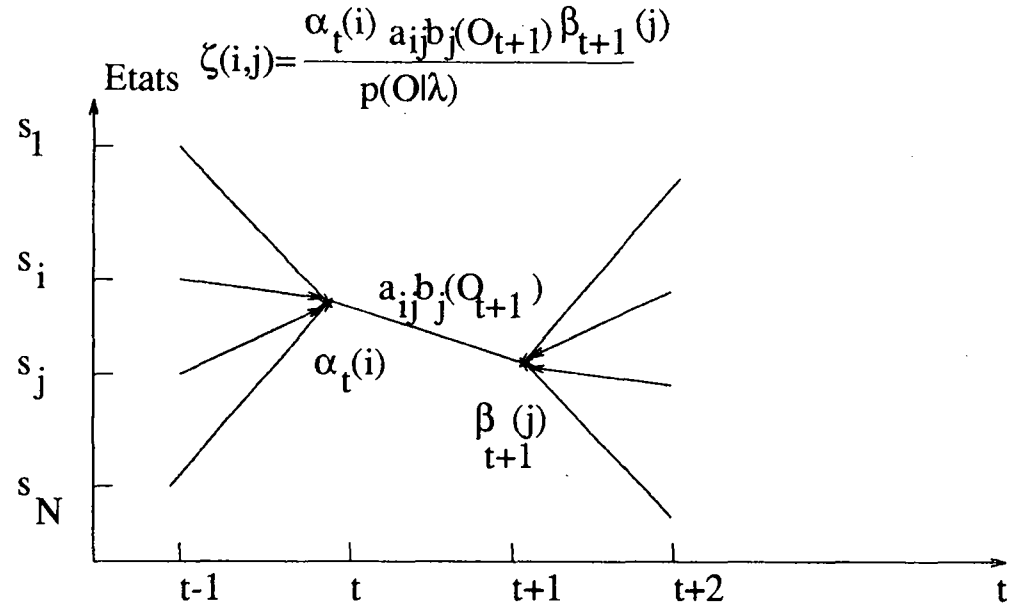



Figure 3.12: Séquences d'opérations nécessaires pour le calcul de l'événement conjoint pour que le système soit à l'état $s_i$ au temps t et à l'état $s_j$ au temps $t+1$.

où

$$\gamma_{ij} = \sum_{t=1}^{T-1} \zeta_t(i,j) \tag{3.71}$$

$$= \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(\mathbf{O}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(\mathbf{O}_{t+1}) \beta_{t+1}(j)} \tag{3.72}$$

$\gamma_{ij}$ est le nombre éspéré des transitions de $s_i$ vers $s_j$.
Cette méthode de Maximum de Vraisemblance est la plus utilisée dans les applications.

**Les formules de ré-estimation**

1. Nombre éspéré de fois d'être à l'état $s_i$ à t=1

$$\tilde{\pi}_i = \gamma_1(i),\ 1 \leq i \leq N \tag{3.73}$$

$$= \frac{\alpha_1(i)\beta_1(i)}{\sum_{i=1}^{N} \alpha_T(i)} \tag{3.74}$$

2. Nombre éspéré des transitions de $s_i$ vers $s_j$ sur le nombre éspéré de fois d'être dans l'état $s_j$

$$\tilde{a}_{ij} = \frac{\sum_{t=1}^{T-1} \zeta_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3.75}$$

$$= \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(\mathbf{O}_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)} \tag{3.76}$$

3. Dans le cas où l'observation est représentée sous forme discrète, le rapport entre le nombre éspéré de fois d'être dans l'état $s_j$ en observant le symbole $v_k$ et le nombre éspéré

de fois d'être dans l'état $s_j$

$$\tilde{b}_j(\mathbf{O}_t = v_k) = \frac{\sum^{T}_{\substack{t=1 \\ \exists \mathbf{O}_t = v_k}} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{3.77}$$

$$= \frac{\sum^{T}_{\substack{t=1 \\ \exists \mathbf{O}_t = v_k}} \alpha_t(j)\beta_t(j)x_t}{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j)} \tag{3.78}$$

ce rapport représente le nombre éspéré de fois d'être à l'état $s_j$ et observer le symbole $v_k$ sur le nombre éspéré de fois d'être à l'état $s_j$
C'est la fréquence d'occurrence de $v_k$ à l'état $s_j$ par rapport à la fréquence d'occurrence de n'importe quel symbole à l'état $s_j$.
Si l'observation est représentée sous forme continu, par exemple :

$$p(\mathbf{O}_t = x | q_t = s_j) = \frac{1}{(2\pi)^{\frac{1}{2}}\sigma_i} e^{\frac{(x_t - \mu_j)^2}{2\sigma_i^2}} \tag{3.79}$$

alors, nous avons le théorème suivant :

**Théorème 4** [BPSW 70, Baum 72]

Pour chaque ensemble $\lambda = \{\pi_i, a_{ij}, \mu_i, \sigma_i\}$, la fonction $R(\lambda, \tilde{\lambda})$ atteint un maximum global à un point critique. Ce point est la transformation $\Upsilon(\lambda)$ donnée par les formules (3.73) et (3.75) et les formules suivantes :

$$\mu_j = \frac{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j)x_t}{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j)} \tag{3.80}$$

$$\sigma_j^2 = \frac{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j)(x_t - \mu_j)^2}{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j)} \tag{3.81}$$

Les contraintes à respecter à chaque itération par ces quantités sont:

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{3.82}$$

$$\sum_{i=1}^{N} \pi_i = 1 \tag{3.83}$$

et

$$\sum_{k=1}^{M} b_j(\mathbf{O}_t = v_k) = 1,\ 1 \leq j \leq N \tag{3.84}$$

où

$$\int_{-\infty}^{\infty} b_j(x)dx = 1,\ 1 \leq j \leq N \tag{3.85}$$

En conclusion, nous utilisons $\tilde{\lambda}$ à la place de $\lambda$, et nous répétons la ré-estimation jusqu'à un certain point limite (en général, on choisit un nombre d'itérations convenable). Le modèle résultant est, ainsi, un HMM par le Maximum de Vraisemblance.

**L'algorithme de Baum-Welch** [Baum 70, Baum 72]

Cet algorithme peut être représenté sous la forme itérative suivante:

1. Fixer des valeurs initiales, k=0

$\lambda = \{\pi_i{}^0, a_{ij}^0, b_j{}^0(\mathbf{O}_t)\}$, *pour* $1 \leq i \leq N, 1 \leq j \leq N$

2. Calculer

$\zeta_t(i,j)$ et $\gamma_t(i)$ pour $1 \leq i \leq N$, $1 \leq j \leq N$

en utilisant les fonctions Forward et Backward.

3. Nouvelles estimations, k=1,2,3, ...

$\tilde{\lambda} = \{\tilde{\pi}_i^k, \tilde{a}_{ij}^k, \tilde{b}_j^k(\mathbf{O}_t)\}$, pour $1 \leq i \leq N$, $1 \leq j \leq N$

4. Recommencer en 2 jusqu'à un certain point limite.

Remarques :

- Le choix du modèle initial influe sur les résultats; toutes les valeurs nulles de **A** et de **B** au départ, restent à zero à la fin de l'apprentissage.
- l'algorithme converge vers des valeurs de paramètres qui forment un point critique de $p(O|\tilde{\lambda})$. Donc, nous obtenons un maximum local ou un point d'inflexion. D'où la nécessité d'un bon choix du modèle initial pour éviter les points d'inflexion.
- le test d'arrêt est en général le nombre des itérations qui est fixé empiriquemnt.
- Pour avoir une estimation convenable du modèle, les ré-estimations se font sur un ensemble de plusieurs suites d'observations appelées corpus d'apprentissage. Donc la taille du corpus d'apprentissage influe, elle aussi, sur les résultats.

◇

#### 3.3.1.3.2 Estimation de paramètres du modèle, vue comme un problème d'optimisation avec contraintes.

Les formules de ré-estimation permettent la mise à jour du modèle en respectant les contraintes (3.82), (3.83), (3.84) ou (3.85).
Le problème de ré-estimation peut donc être considéré comme un problème d'optimisation de la probabilité $p(O|\lambda)$ par rapport à ces contraintes que nous pouvons résoudre par la technique des multiplicateurs de Lagrange.

**Optimisation des paramètres**

Soit L le Lagrangien de $p(O|\lambda)$ par rapport aux contraintes (3.82), (3.83), (3.84) ou (3.85):

$$L(p(O|\lambda)) = p(O|\lambda) + \sum_{i=1}^{N} \lambda_i (\sum_{j=1}^{N} a_{ij} - 1) + \sum_{i=1}^{N} \mu_i (\sum_{i=1}^{N} \pi_i - 1) + (\sum_{i=1}^{N} \beta_i \sum_{j=1}^{N} b_j(\mathbf{O}_t = v_k) - 1)$$

dans le cas d'un HMM avec des densités d'observations discrètes, ou

$$L(p(O|\lambda)) = p(O|\lambda) + \sum_{i=1}^{N} \lambda_i (\sum_{j=1}^{N} a_{ij} - 1) + \sum_{i=1}^{N} \mu_i (\sum_{i=1}^{N} \pi_i - 1) + \sum_{i=1}^{N} \beta_i (\int_{-\infty}^{\infty} b_i(x)dx - 1)$$

dans le cas d'un HMM avec des densités d'observations continues. Les paramètres $\lambda_i$, $\mu_i$ et $\beta_i$ sont les multiplicateurs de Lagrange à déterminer.

**Optimisations par rapport à $a_{ij}$**

Dans ce cas, le Lagrangien L s'écrit:

$$L(p(O|\lambda)) = p(O|\lambda) + \sum_{i=1}^{N} \lambda_i (\sum_{j=1}^{N} a_{ij} - 1)$$

A un point critique de $p(O|\lambda)$ et à l'intérieur de la zone de contraintes définies par (3.82), (3.83), (3.84) ou (3.85) nous avons:

$$\frac{\partial L(p(O|\lambda))}{\partial a_{ij}} = \frac{\partial p(O|\lambda)}{\partial a_{ij}} + \lambda_i = 0,\ 1 \leq i,j \leq N \tag{3.86}$$

Multiplions (3.86) par $a_{ij}$ et faisons la somme sur j, nous obtenons:

$$\sum_{j=1}^{N} a_{ij} \frac{\partial p(O|\lambda)}{\partial a_{ij}} = -[\sum_{j=1}^{N} a_{ij}]\lambda_i = -\lambda_i = \frac{\partial p(O|\lambda)}{\partial a_{ij}} \tag{3.87}$$

(3.87) montre que $p(O|\lambda)$ est maximisée quand

$$\tilde{a}_{ij} = \frac{a_{ij} \frac{\partial p(O|\lambda)}{\partial a_{ij}}}{\sum_{k=1}^{N} a_{ik} \frac{\partial p(O|\lambda)}{\partial a_{ik}}} \tag{3.88}$$

Il est évident que l'equation (3.88) est analytiquement insoluble.

Rappelons l'equation (3.25):

$$p(O|\lambda) = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)a_{ij}b_j(\mathbf{O}_{t+1})\beta_{t+1}(j),\ 1 \leq t \leq T-1$$

$$\frac{\partial p(O|\lambda)}{\partial a_{ij}} = \sum_{t=1}^{T-1} \alpha_t(i)b_j(\mathbf{O}_{t+1})\beta_{t+1}(j) \tag{3.89}$$

Les équations (3.87), (3.88) et (3.89) donnent

$$\begin{aligned} a_{ij} &= \frac{\sum_{t=1}^{T-1} \alpha_t(i)a_{ij}b_j(\mathbf{O}_{t+1})\beta_{t+1}(j)}{\sum_{j=1}^{N}\sum_{j=1}^{N} \alpha_t(i)a_{ij}b_j(\mathbf{O}_{t+1})\beta_{t+1}(j)} \\ &= \frac{\sum_{t=1}^{T-1} \alpha_t(i)a_{ij}b_j(\mathbf{O}_{t+1})\beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\sum_{j=1}^{N} a_{ij}b_j(\mathbf{O}_{t+1})\beta_{t+1}(j)} \\ &= \frac{\sum_{t=1}^{T-1} \alpha_t(i)a_{ij}b_j(\mathbf{O}_{t+1})\beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)} \end{aligned} \tag{3.90}$$

(3.91)

Cette équation est identique à l'équation (3.76).

**Optimisations par rapport à $\pi_i$ , $b_j(\mathbf{O}_t = v_k)$**

Calculons ainsi

$$\frac{\partial p}{\partial \pi_i} = \sum_{j=1}^{N} b_i(\mathbf{O}_1)a_{ij}b_j(\mathbf{O}_2)\beta_2(j) \tag{3.92}$$

$$= b_i(\mathbf{O}_1)\beta_1(i) \tag{3.93}$$

et

$$\frac{\partial p}{\partial b_j(\mathbf{O}_t = v_k)} = \sum_{\substack{t=1 \\ \exists \mathbf{O}_t = v_k}}^{T}\sum_{i=1}^{N} \alpha_t(i)a_{ij}\beta_{t+1}(j) + \delta(\mathbf{O}_1, V_k)\pi_j\beta_1(j) \tag{3.94}$$

où $\delta$ est la fonction de Kronecker.
En substituant (3.93) et (3.94) dans leurs équations respectives:

$$\tilde{\pi}_i = \frac{\pi_i \frac{\partial p(O|\lambda)}{\partial \pi_i}}{\sum_{k=1}^{N} \pi_k \frac{\partial p(O|\lambda)}{\partial \pi_k}} \tag{3.95}$$

et

$$\tilde{b}_j(\mathbf{O}_t = v_k) = \frac{b_j(\mathbf{O}_t = v_k)\frac{\partial p(O|\lambda)}{\partial b_j(\mathbf{O}_t = v_k)}}{\sum_{k=1}^{M} b_j(\mathbf{O}_t = v_k)\frac{\partial p(O|\lambda)}{\partial b_j(\mathbf{O}_t = v_k)}} \tag{3.96}$$

nous obtenons les équations de ré-estimationsde $\pi_i$ et $b_j(\mathbf{O}_t = v_k)$ qui sont similaires aux équations (3.74) et (3.78).

### 3.3.2 Aspects pratiques des modèles HMM

L'objectif de ce paragraphe est d'introduire brièvement les différents aspects pratiques concernant les types de modèles HMM, l'amélioration du modèle et les problèmes d'implementation. Pour plus de détails et de synthèses relatifs aux différents thèmes de ce paragraphe, voir [AlGu 93].

#### 3.3.2.1 Type d'un HMM

Le type d'un HMM (Tableau 3.1) peut être spécifié selon :
1. les contraintes sur la matrice de transition **A**.
2. le type de densité de probabilité d'observations $b_j(\mathbf{O}_t)$.
3. la durée de séjour dans un état.

| *contraintes* | *Types du modèle HMM* | | | |
|---|---|---|---|---|
| *Par rapport à la matrice de transition A* | *HMM ergodique* | *HMM Gauche-droite* | | |
| Par rapport à la probabilité des observations | DHMM | CHMM | SCHMM | *SMHMM* |
| *Par rapport à la durée du séjour dans un état* | VDHMM | *CVDHMM* | | |
| *Par rapport à l'ordre de chaîne de Markov* | *HMM d'ordre r* | | | |

Tableau 3.1

1. **les contraintes sur la matrice de transition A**

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

- **Modèle ergodique**
  Ce Modèle est sans contraintes sur la matrice **A**. Chaque état peut être visité, à partir de n'importe quel état, en une seule transition , Fig. 3.13.

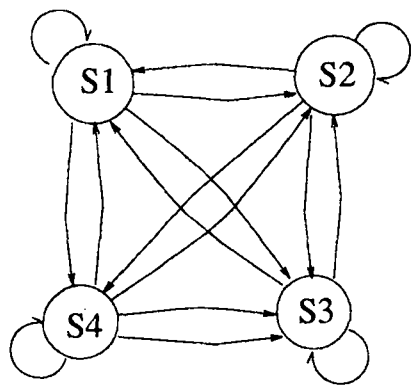



Figure 3.13: Modèle ergodique à 4 états.

- **Modèles Gauche-Droite (Left-Right Models)**, [Jeli 76, Baki 76]

  Ces modèles peuvent être utilisés pour modéliser des processus possédant des propriétés variantes dans le temps, telle que les signaux de la parole. Ils possèdent les

propriétés suivantes, Fig. 3.14, Fig. 3.15 et Fig. 3.16 :

1. La première observation est produite pendant que la chaîne de Markov est dans un état initiale, $q_1$, avec

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

2. La dernière observation est générée alors que la chaîne de Markov se trouve dans un état final (état absorbant), $q_N$, avec

$$\begin{aligned} \beta_T(j) &= 1, j = N \\ &= 0, j \neq N \end{aligned}$$

On choisit, en général, en plus $a_{ij} = 0$ si $j > i + \delta$ avec $\delta = 1, 2, \ldots$ , Fig. 3.14 et Fig. 3.15.

3. Dès que la chaîne de Markov quitte un état, cet état ne sera plus visité plus tard, c'est-à-dire $a_{ij} = 0$, $j < i$.

Les modèles Gauche-Droite du type série, Fig. 3.14 et fig. 3.15, procédent séquentiellement à travers les états, tandis que les modèles parallèles, Fig. 3.16, permettent des trajectoires multiples à travers les états où chaque trajectoire peut sauter un ou plusieurs états.

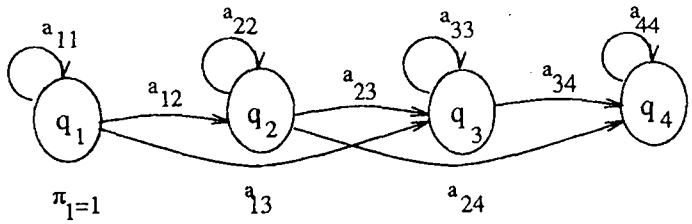



Figure 3.14: HMM avec contraintes séries à double transitions, $a_{ij} = 0$ pour $j < i$ et $j \geq i + 3$.

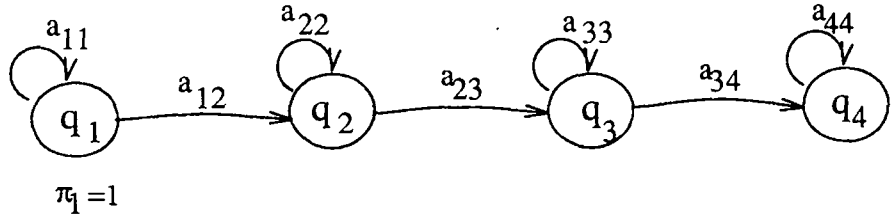



Figure 3.15: HMM avec contraintes séries à une seule transition, $a_{ij} = 0$ pour $j < i$ et $j \geq i + 2$ .

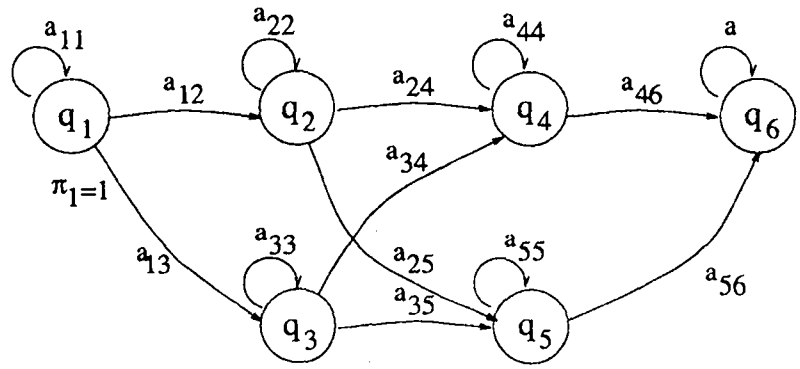



Figure 3.16: HMM avec contraintes parallèles.

## 2. Le type de densité de probabilité d'observations $b_j(O_t)$

Selon le type de densité de probabilité d'observations, discrète ou continue, nous pouvons construire deux types de modèles HMM : soit un HMM discret soit un HMM continu.

- **HMM discret "Discret Hidden Markov Models (DHMM)"**

  Les observations en général sont continues puisqu'elles proviennent de phénomènes physiques continus. Dans le cas d'un HMM discret, les observations continues sont quantifiées à l'aide d'un dictionnaire (voir équations (3.2) et (3.8) et paragraphe 3.3.1).

- **HMM continu "Continuous Hidden Markov Models (CHMM)"**

Bien qu'il soit possible de quantifier les observations continues, il y a une sérieuse dégradation d'information associée à cette quantification [Juan 85, JuLS 86]. 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, certaines réstrictions 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 "Scalar Gaussian (SG)"** [Baum 72], (voir paragraphe 3.3.1.3.1.1);
    * Fonction de densité de probabilité **Gaussienne Multivariable "Multivariable Gaussian (MG)"** [Lipo 82];
  - **Mélange fini** de densités de probabilité **Gaussiennes Multivariables "multivariable Gaussian mixture (GM)"** [Juan 85];
  - Fonctions **Gaussiennes Autoregréssives.** C'est un cas particulièrement approprié aux systèmes dynamiques et notamment aux signaux évoluants dans le temps. Il existe trois types de modèles :
    * HMM Autoregréssif d'une seule fonction Gaussienne par état "Gaussian Autoregressive (GAR)" [Pori 82];
    * HMM Autoregréssif de mélange de M fonctions Gaussiennes par état "Gaussian Autoregressive Mixture (GAM)" [JuRa 85a].
    * HMM de Mélange fini de densités de probabilité Gaussienne Autoregréssive avec décodage par la technique de la quantification vectorielle "Partioned Gaussian Autoregressive Mixture (PGAM)", [JuRa 85a].

Tableau 3.2 montre une comparaison, selon plusieurs critères, entre un HMM discret et un HMM continu.

| | A. HMM continu | B. HMM discret |
|---|---|---|
| Nombre de paramètres à estimer | un nombre élevé de paramètres | moins de paramètres que A |
| Précision de la classification | préçis | moins préçis que A |
| Hypothèses sur la nature des observations | importantes | moins importantes que A |
| Implementation | difficile et lent | plus façile et plus rapide que A |
| Nombre de corpus d'apprentissage | moyen | plus élevé que A |

Tableau 3.2

- **HMM Semi-Continu "Semi-continuous Hidden Markov Models (SCHMM)"**

Dans le cas d'un HMM discret, 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 le HMM discret au lieu du symbole

d'observation discret correspondant. Huang et Jack [HuJa 88] ont proposé une approche qui consiste à combiner le HMM discret et le HMM continu.
Le modèle obtenu, appelé HMM Semi-Continu "Semi-Continuous Hidden Markov Model (SCHMM)", place la distortion de la quantification vectorielle sous un cadre probabiliste.
Son principe consiste à remplacer les probabilités des symboles d'observations discrets par une combinaison des probabilités des symboles d'observations discrets et des fonctions de densités de probabilités continues dérivées du dictionnaire de la quantification vectorielle.
Les fonctions de densités de probabilités continues du dictionnaire sont utilisées dans l'algorithmes du décodage (algorithme de Viterbi, voir paragraphe 3.3.1.2) pour pondérer les vecteurs d'observations continues et les paramètres du HMM discret (le vecteur d'observation continue est utilisé directement dans la procédure de Viterbi).
Les résultats expérimentaux dans le domaine de la reconnaissance de la parole, [HuJa 88, HuJa 89a, HuJa 89b], ont montré que la précision de la reconnaissance, en utilisant les SCHMM's, est améliorée par rapport à celles des HMM discrets et celles des techniques du recalage temporel dynamique [Vint 68, BoWN 84].

- **Corrélation du temps explicite "Semi-Markovian Hidden Markov Models (SMHMM)"**

  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 à certains processus dynamiques réels. Une généralisation de cette hypothèse en définissant une nouvelle probabilité d'émission d'observation qui prend en compte la corrélation entre les vecteurs représentants les observations continues successives a été proposée [Well 87, cele 92]. Par conséquent, les formules de ré-estimation de Baum-Welch et l'algorithme de Viterbi ont été modifiées.

3. **La durée de séjour dans un état**

L'un des inconvenients des HMM's 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, à titre d'exemple la variabilité de la durée des sons dans la parole [CrHo 86]. Un modèle HMM permet de segmenter un mot prononcé en états (en utilisant l'algorithme de Viterbi, Voir paragraphe 3.3.1). Nous pouvons, alors, utiliser les durées mesurées pour qu'un état $q_i$ reste pendant une durée d avant la transition vers un autre état j avec une probabilité $a_{ij}$, Fig. 3.17. La densité de la durée d associée à l'état i, $p_i(d)$, avec un coefficient d'auto-transition $a_{ii}$ est donnée par l'equation :

$$p(O|\lambda, q_1 = s_i) = (a_{ii})^{d-1}(1 - a_{ii}) = p_i(d)$$

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 la plus courte.
Pour la majorité des phénomènes physiques, cette densité géométrique de la durée d'état,

qui décroît d'une façon exponentielle, est inadaptée [Ferg 80b, RuMO 85]. Ainsi diverses méthodes ont été proposées pour améliorer la prise en compte de cette durée [Ferg 80b, RuMO 85, RaLe 85, Rabi 88, Cook 88, BoWe 86, Levi 86].
Principalement, deux méthodes ont été dévéloppées, la méthode de Ferguson [Ferg 80b] basée sur un HMM de **Durée variable Discrète "Variable Duration Hidden Markov Model VDHMM"** et celle de Levinson [Levi 86] basée sur un HMM de **durée variable continue "Continuous Variable Duration Hidden Markov Model CVDHMM"**.

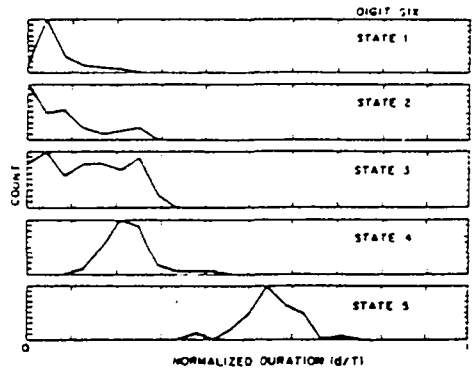



Figure 3.17: Exemple d'un histogramme de la densité de la durée normalisée d'un état pour les cinq états du mot "six" [Rabi 89].

4. **L'ordre de chaîne de Markov**

Une limitation des Modèles de Markov Cachés de base est de supposer que le processus markovien est d'ordre un ce qui n'est pas le cas dans de nombreuses applications.
Une application à la reconnaissance de l'écriture a était faite par Kundu et Bahl [KuBa 88]; ils ont montré l'avantage des HMMs du second ordre sur ceux du premier ordre. Ils ont signalé aussi la difficulé d'implementation des HMMs du second ordre dûe à l'indisponibilité de leurs formules de ré-estimation dans la litérature . Ils étaient alors obligés de calculer les probabilités de transition manuellement à partir du dictionnaire du langage.
He [He 88] et Kriouile [krio 90] ont développé et traité ainsi une technique d'extension des algorithmes de Viterbi et ceux de Baum-Welch pour un HMM du second ordre et d'ordre quelconque.
Les résultats, dans le domaine de la parole [Krio 90], de ces modèles du second ordre comparés à ceux du premier ordre, ont montré l'utilité et l'efficacité de cette approche.

### 3.3.2.2 Amélioration du modèle

#### 3.3.2.2.1 Critère d'optimisation pour les formules de ré-estimation.

La modélisation, basée sur le Modèle HMM, d'un processus réel, est efficace si les paramètres de ces modèles sont correctement estimés. Ces estimations, en pratique, sont inexactes dûesprincipalement à deux raisons : soit le processus n'obeit pas aux contraintes de HMM, ou bien à cause des difficultées d'obtenir des estimations fiables de tous les paramètres de HMM.
Diverses solutions ont été développées afin d'apporter des améliorations au niveau de l'apprentissage. Ces solutions sont des variantes et des alternatives des algorithmes de

Baum-Welch basés sur le Maximum de Vraisemblance (voir paragraphe 3.3.1.3.1.1). Il n'existe pas de réponses théoriques pour le choix d'une solution ou d'une autre.
Généralement, seul l'expérimentation permet de déterminer l'efficacité d'une solution par rapport à d'autres.

- **Technique de Maximum d'Information Mutuelle "Maximum Mutual Information (MMI)"**

  Cette technique [BBSM 86] consiste à construire plusieurs Modèles HMM, correspondant à plusieurs processus, en même temps de tel sorte que l'efficacité discriminante de chaque modèle soit maximale, c'est à dire, la capacité de chaque modèle de distinguer entre les suites d'observations générées par le modèle correcte et celles générées par les modèles alternatifs.
  Cette technique a donné de bonnes performances dans le domaine de la reconnaissance de la parole [Brow 87, Méri 88a, Méri 88b].

- **Technique de Minimum d'information de discrimination "Minimum Discrimination Information (MDI)"**

  C'est une technique de généralisation, à la fois de la technique de Maximum de Vraisemblance (MLE) et celle de Maximum d'Information Mutuelle (MMI).
  C'est un algorithme itératif permettant la construction des modèles HMM (de plusieurs processus) correspondant aux suites d'observations provenant des processus à modéliser [EpDR 87]. Son objectif est alors de choisir les paramètres de HMM qui minimisent l'information de discrimination (DI) ou l'entropie relative entre l'ensemble des densités de probabilité qui satisfont les mesures et l'ensemble des densités de probabilté de HMM.

- **Apprentissage Correctif**

  Cette approche [BBSM 87, BBSM 88] garde les mêmes idées de l'approche MMI tout en jugeant les valeurs des paramètres sur les erreurs des tests plutôt que sur la vraisemblance de corpus d'apprentissage. L'objectif est alors de minimiser le nombre d'erreurs de la reconnaissance en ajustant les valeurs de paramètres de façon que les processus corrects soient plus probables et les processus incorrects soient moins probables.

#### 3.3.2.2.2 Critère de dissimilarité entre deux modèles HMMs.

Etant donné deux modèles HMM, $\lambda_1$ et $\lambda_2$ quelle est leur mesure significative de dissimilarité?
Un point clé pour répondre à cette question est de définir un critère de dissimilarité. Juang et Rabiner [JuRa 85] ont proposé plusieurs critères de dissimilarité basés sur la notion de distance.

### 3.3.2.3 Problèmes d'implementation

Nous pouvons classer les différents problèmes d'implementation comme suit :
1. Problème de précision numérique "Underflow".
2. Problème de données insuffisantes pour l'apprentissage.
3. Problème de processus évoluants dans le temps.
4. Problème d'estimation initiale des modèles HMMs.
5. Problème d'estimation a priori du nombre d'états d'un modèle HMM.
6. Problème de stockage de l'historique complet des variables "Forward" et "Backward".

### 1. Problème de précision numérique "Underflow".

La mise en œuvre des récurrences dans un programme donnerait inévitablement lieu à des problèmes numériques "Underflow".
En effet les variables "Forward" et "Backward" tendent exponentiellement vers zero si t tend respectivement vers l'infini et un.
En pratique, les longueurs des suites d'observations aboutiraient à ces problèmes numériques si les équations (3.17) et (3.21) sont évaluées directement.
Pour résoudre ce problème plusieurs solutions ont été propsées et par conséquent les alogorithmes corespondant aux trois problèmes de HMM ont été modifiés [LeRs 83a, AsDe 81].

### 2. Problème de données insuffisantes pour l'apprentissage.

Baum et Petri [BaPe 66] ont montré que les paramètres du modèle HMM convergent vers les vraies valeurs quand le nombre d'observations tend vers l'infini. Or qu'alors la suite d'observations pour l'apprentissage est nécessairement finie. Par conséquent cette limitation provoque inévitablement un manque d'échantillons représentatifs des différents événements. Le problème de l'insuffisance du nombre des observations pour l'apprentissage, implique que certains événements de faible probabilité peuvent ne pas apparaître dans cet ensemble fini d'observations (probabilités nulles). Une solution théorique consiste à augmenter le corpus d'apprentissage.
Plusieurs solutions pratiques pour résoudre ce problème ont été proposées. Jelinek et Mercer [JeMe 80] ont utilisé des informations supplémentaires par rapport à celles de la suite d'observations pour estimer les petites probabilités. Levinson, Rabiner et Sondhi [LeRs 83a] ont utilisé une approche consistant à ajouter des contraintes supplémentaires aux paramètres des modèles pour empêcher les valeurs des paramètres de descendre en dessous d'un certain seuil.

### 3. Problème de processus évoluants dans le temps.

Dans un processus évoluant dans le temps, les états ont une nature transitoire. Par conséquent une séquence unique d'une suite d'observations n'est pas siffisante. Ainsi pour avoir une estimation fiable des paramètres du modèle HMM de ce processus, il est nécessaire d'utiliser des séquences multiples d'observations [LeRS 83a].

### 4. Problème d'estimation initiale des modèles HMMs.

Le problème d'estimation initiale des paramètres du modèle HMM a une influence directe

sur les formules de ré-estimation pour l'apprentissage. Une mauvaise estimation initiale provoque des solutions non optimales (problèmes des extremas locaux).
Il n'existe pas de solutions théoriques précises pour résoudre ce problème, seule l'expérimentation permet de porter une réponse. Il a été démontré que l'estimation des paramètres de matrice **B**, dans le cas d'un HMM continu, est très sensible au problème de l'estimation initiale [RJLS 85].
Plusieurs solutions ont été proposées pour résoudre ce problème. Paul et Jouvet [Paul 85, Jouv 87] ont proposé une méthode pour converger vers l'optimum global qui consiste à perturber aléatoirement les paramètres afin de s'éloigner du domaine de l'optimum local. Rabiner et al [RJLS 85, RJLS 86] ont proposé une méthode basée sur un bon choix initial de paramètres du modèle. Cette technique a deux avantages :

- réduction du nombre d'itérations au niveau de l'apprentissage;
- augmentation de la chance d'atteindre le maximum global.

Malheureusement, il n'existe pas une solution théorique permettant d'estimer les valeurs initiales de ces paramètres. Rabiner et al. ont montré, expérimentallement, que des estimations initiales aléatoires ou bien des estimations equiprobables des paramètres $\Pi i$ et du modèle sont adéquates pour obtenir une bonne ré-estimation de ces paramètres. Cependant, la matrice B nécessite des bonnes estimations initiales dans le cas d'un HMM discret et sont obligatoires dans le cas d'un HMM continu.

## 5. Problème d'estimation a priori du nombre d'états d'un modèle HMM.

Il n'existe pas de bonnes méthodes théoriques pour résoudre ce problème puisque les états ne sont pas nécessairement liés physiquement à un phénomène observable.
Le nombre d'état d'un HMM peut être déterminé par deux approches :

- Une approche expérimentale, en effectuant des tests sur plusieurs HMMs ayant un nombre différents d'états, afin de déterminer le nombre d'états optimal à partir des meilleurs resultats [Russ 86].
- Une approche donnant à un modèle HMM une explication physique explicite [Dour 89].

Cependant, Ponting [Pont 88] a utilisé un critère statistique pour déterminer automatiquement le nombre d'états optimal d'un processus à partir d'un corpus d'apprentissage.

## 6. Problème de stockage de l'historique complet des variables "Forward" et "Backward".

Les formules de ré-estimation de Baum-Welch pour l'identification des chaînes de Markov Cachées, font intervenir les équations "Forward" et "Backward" de Baum. LeGland [LeGl 92] a montré qu'il est possible de faire l'economie de ce mécanisme "Forward-Backward", et d'obtenir des formules de ré-estimations qui font intervenir seulement des équations "Forward". On évite ainsi le stockage de l'historique complet des variables "Forward" et "Backward". En revanche le temps de calcul augmente, puisqu'il y a maintenant autant de variables "Backward" que de paramètres à estimer. Un compromis entre mémoire et temps de calcul peut rendre cette méthode attractive dans le cas où le nombre de paramètres à estimer est réduit.

# 4

# Conclusion

Les Modèles de Markov Cachés ont prouvé dans des différents domaines qu'ils étaient des outils puissants. Sur la base théorique, les voies de recherches sont ouvertes et notamment au niveau de l'amélioration de la phase d'apprentissage. A ce niveau, les approches connexionnistes et les approches de recuits simulés peuvent apporter de nouvelles possibilités et de nouvelles améliorations. Sur la base pratique, l'efficacité des HMMs peut être nettement amélioré en utilisant les techniques de programmation parallèles et les techniques d'implementation sur une architecture massivement parallèle.
Enfin, nous considérons que les HMMs peuvent apporter des solutions à un vaste champ d'applications autre que la Reconnaissance Automatique de la Parole.

# 5

# Bibliographie

[AlGu 94] T. ALANI et H. GUELLIF. Modèles de Markov Cachés - Aspects pratiques. Rapport de Recherche, INRIA, à paraître.

[AsDe 81] M. Askar and H. Derin, A recursive algorithm for the Bayes solution of the smoothing problem, IEEE Transactions on Automatic Control, AC-26(2), pp. 558-561, 1981.

[Aver 86] A. Averbuch, L. R. Bahl, R. Bakis, P. F. Brown, A. Cole, G. Daggett, S. K. Das, K. Davies, S. V. DeGennaro, P. V. de Souza, E. A. Epstein, D. Fraleigh, F. Jelinek, S. Katza, B. L. Lewis, R. L. Mercer, A. J. Nadas, D Nahamoo, M. A. Picheny, G. Shichman and P. Spinelli. Experiments with the Tangora 20,000 word speech recognizer. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 701-704, Dallas, 1987.

[BaEa 67] L.E. Baum et J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. Bull. Amer. Soc. 73, pp. 360-363, 1967.

[Bake 75a] J. K. Baker. The DRAGON system - An overview. IEEE Trans. Acoust., Speech, Signal Processing, ASSP-23(1), pp. 24-29, 1975.

[Bake 75b] J. K. Baker. Stochastic Modeling as a Means of Automatic Speech Recognition. Thèse de Doctorat, Carnegie-Mellon University, April 1975.

[Bake 75c] J. K. Baker. Stochastic Modeling as a Means of Automatic Speech understanding. D. R. Reddy, ED., Speech Recognition, Academic Press, New York, 1975.

[Baki 76] R. Bakis. Continuous speech recognition via centisecond acoustic states. J. Acoustical Society Am. 59 Supp. 1, 1976.

[Bahl 76] L. R. Bahl, J. K. Baker, P.S. Cohen, N. R. Dixon, F. Jelinek, R. L. Mercer, et H. F. Silverman. Preliminary results on the performance of a system for the automatic recognition of continuous speech. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 425-429, 1976.

[Bahl 79] L. R. Bahl, R. Bakis, P.S. Cohen, A. G. Cole, F. Jelinek, B. Lewis et R. L. Mercer. Recognition results with several experimental acoustic processors. Proc. IEEE Int. Conf. Acoust., Speech, Signal processing, pp. 249-251, 1979.

[Bahl 80] L. R. Bahl, R. Bakis, P.S. Cohen, A. G. Cole, F. Jelinek, B. Lewis et R. L. Mercer. Further results on the recognition of continuously read natural corpus. Proc. IEEE Int. Conf. Acoust., Speech, Signal processing, pp. 872-875, 1980.

[Bahl 89] L. R. Bahl, R. Bakis, J. Bellegarda, P. F. Brown, D. Burshtein, S. K. Das, P. V. de Souza, P. S. Gopalakrishnan, F. Jelinek, D. Kanevsky, R. L. Mercer, A. J. Nadas, D. Nahamoo and M. A. Picheny. Large vocabulary natural langage continuous speech recognition. Proc. IEEE Int. Conf. Acoust., Speech, Signal processing, pp. 465-467, Glasgow, Scotland, 1989.

[BaJe 75] L. R. Bahl, F. Jelinek. Decoding for channels with insertions, deletions and substitutions, with applications to speech recognition. IEEE Trans. Inform. Theory, IT-21, pp. 404-411, July 1975.

[BaJM 83] L. R. Bahl, F. Jelinek and R. L. Mercer. A maximum likelihood approach to continuous speech recognition. IEEE Trans. PAMI., PAMI-5(20), pp. 179-190, 1983.

[BaPe 66] L.E. Baum et T.Petrie. Statistical inference for probabilistic functions of finite state Markov chains. Ann. Math. Stat. 37, pp. 1554-1563. 1966.

[BaSe 68] L.E. Baum et G. R. Sell. Growth transformations for functions on manifolds. Pac. J. Math., 27, pp. 211-227, 1968.

[BBSM 86] L. R. Bahl, P. F. Brown, P. V. DE SOUZA, and R. L. Mercer. Maximum mutual information estimation of hidden Markov model parametrs for speech recognition. Proc. ICASSP'86 (Tokyo), pp.49-52, Apr. 1986.

[BBSM 87] L. R. Bahl, P. F. Brown, P. V. DE SOUZA, and R. L. Mercer. Estimating HMM parametrs so as to maximise speech recognition accuracy. Research Report RC-13121. IBM TJ Waston Research Center, 9/10/1987.

[BBSM 88] L. R. Bahl, P. F. Brown, P. V. DE SOUZA, and R. L. Mercer. A new algorithm for the estimation of hidden Markov model parametrs. Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing, pp. 493-496, New York. 1988.

[BPSW 70] L.E. Baum, T.Petrie, G.Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann. Math. Stat. 41(1), pp. 164-171, 1970.

[Baum 72] L.E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. Inequalities, 3, pp. 1972.

[BCGR 74] L. R. Bahl, J. Cocke,F. Jelinek and J.Raviv. Optimal decoding of linear codes

for minimizing symbol error rate. IEEE Trans. Inform. Theory, IT-20, pp. 275-320, 1975.

[Bell 57] R. Bellman. Dynamic Programming. Princeton, N.F., Univ. Press, Princeton, 1957.

[Bill 61] P. Billingsley. Statistical inference for Markov processes. Univ. of Chicago Press, Chicago, 1961.

[Boye 88] A. Boyer, J. Di Martino, P. Divoux, J. P. Haton, J. F. Mari and K. Smaili. Statistical methods in multi-speaker automatic speech recognition. Forth International Symposium, Nancy, Dec. 1988.

[BoWe 86] Herve Bourlard and christian Wellekens.Connected speech recognition by statistical methodes. Eurasip short course on speech recognition, Brussel, Feb. 1986.

[BoWe 90] Herve Bourlard and christian Wellekens. Links between Markov Models and Multilayer perceptrons. IEE Trans. On Pattern Analysis and Machine Inteligence, vol. 12, No. 10, pp. 1-4, SanDiego, CA, 1984.

[BoWN 84] Herve Bourlard, christian Wellekens and H. Ney. Connected digit recognition using vector quantization. Proc. Int. Conf. Acoust., Speech and Signal Processing, vol. 26, No. 12, Dec. 1990.

[Brow 87] P. F. Brown. The acoustic-modeling problem in automatic speech recognition. PhD Dissertation. Carnegie Mellon University, May 1987.

[CaNe 80] R. L. Cave et L. P. Neuwrith. Hidden Markov Models for English. Proc. of the Symposium on the Applications of Hidden Markov Models to Text and Speech, IDA-CRD, pp. 16-56, Princeton, NJ, 1980.

[CeCl 92] G. CELEUX, J. CLAIRAMBAULT. Estimation de chaînes de Markov cachées: méthodes et problèmes. GDR 134, Traitement du signal et images, JOURNEES THEMATIQUES, Approches Markoviennes en signal et images, pp. 5-19, 1992.

[Chow 87] Y. L. Chow, M. O. Dunham, O. A. Kimball, M. A. Krasner, G. F. Kubala, J. Makhoul, P. J. Price, S. Roucos and R. M. Schwartz. BYBLOS : the BBN continuous speech recognition system. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 89-92, Dallas, 1987.

[Cook 88] A.E. Cook. Experimental evaluation of algorithms for connected speech recognition using hidden Markov models. 7th FASE Symposium, Edinburg, 1988.

[Cove 84] THOMAS M. COVER. An Algorithm for Maximizing Expected Log Investment Return. IEEE Transactions on Information Theory, Vol. IT-30, No. 2, March 1984.

[CrHo 86] T. H. Crystal and A. S. House. Characterisation and modeling of speech-segment durations.Proc. Int. Conf. Acoust., Speech and Signal Processing, pp. 2791-

2794, Tokyo 1986.

[DeDe 87] Pierre A. Devijver and M. Dekesel. Learning the parameters of a hidden Markov random field image model: A simple exemple. Pattern Recognition Theory and Applications, P.A. Devijver and J. Kittler Eds., Heidelberg: Springer-Verlag, 1987.

[DeLR 77] A. P. Dempster, N. M. Laird and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. B 39, pp. 1-38, 1977.

[DeMe 89] A. M. Derouault and Mérialdo. Improuving speech recognition accuracy with contextual phonemes and MMI training. Proc. IEEE Int. Conf. Acoust., Speech, Signal processing, Glasgow, Scotland, 1989.

[Devi 86] Pierre A. Devijver. Segmentation of binary images using Third-order Markov Mesh Image Models. Proc. 8th Internat. Conf. Pattern Recognition, Paris, Octobre 1986.

[Devi 90] Pierre A. Devijver. Real-Time Modeling of Image Sequences based on Hidden Markov Mesh Random Field Models. Proc. Internat. Conf. Pattern Recognition, IEEE 1990.

[DoPe 88] C. Dours and Pérennou. The role of intermediary scores in word-spotting. Seventh FASE symposium, Edinburgh, 1988.

[Dour 89] C. Dours. Contribution à l'étude du décodage acoustico-phonétique pour la reconnaissance automatique de la parole. Thèse de Doct. Univ. Paul Sabatier de Toulouse, 1989.

[EpDR 87] Y. Ephraim, A. Dembo, and L. R. Rabiner. A minimum discrimination information approach for hidden Markov modeling. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 25-28, Dallas, 1987.

[EuWo 88] S. Euler and D. Wolf. Continuous Hidden Markov Models in speaker independent isolated word recognition. Proc. 4e EUSIPCO, pp. 1185-1188, Grenoble, France, Sept. 1988.

[Fell 58] W. Feller. An introduction to probability theory and its applications. John Wiley, 2nd Edition, vol. I, New York, 1958.

[Ferg 80a] J. D. Ferguson. Hidden Markov analysis : an introduction. Proc. of the Symp. on the Applications of Hidden Markov Models to text and speech, IDA-CRD, pp. 8-15, Princeton, NJ, 1980.

[Ferg 80b] J. D. Ferguson. Variable duration models for speech. Proc. of the Symp. on the Applications of Hidden Markov Models to text and speech, IDA-CRD, pp. 8-15, Princeton, NJ, 1980.

[Forn 73] G. D. Forney. The Viterbi algorithm. Proc. IEEE, pp. 268-278, Mar 1973.

[Gour 88] A. Gourinda. Codage et reconnaissance de la parole par quantification vectorielle. Thèse de Doct. Univ. de NANCY, 1988.

[Gray 84] R. M. Gray. Vector quantization. IEEE ASSP magazine, pp 4-29, Avril 1984.

[HaLe 91] Blake Hannaford, Paul Lee. Hidden Markov Model Analysis of Force/Torque Information in Telemanipulation. The International Journal of Robotics Research, pp. 528-540, 1991.

[Hart 58] H. O. Hartley. Maximum likelihood estimation from incomplete data. Biometrics, 14, pp. 174-194, 1958.

[HuJa 88] X. D. Huang and M. A. Jack. Semi-Continuous Hidden Markov Models in Isolated Word Recognition. Proc. IEEE 9th Int. Conf. on Pattern Recognition, pp. 406-408, Rome, Italy, 1988.

[JeBM 75] F. Jelinek, L. R. Bahl, and R. L. Mercer. Design of a linguistic statistical decoder for the recognition of continuous speech. IEEE Trans. IT, IT-21(No 3), May 1975.

[Jeli 69] F. Jelinek. A fast sequential decoding algorithm using a stack. IBM J. of Res. and Develop., 13, pp. 675-685, 1969.

[Jeli 76] F. Jelinek. Continuous Speech Recognition by statistical méthods. Proc. IEEE ,vol. 64, pp. 532-536, April, 1976.

[JeMB 76] F. Jelinek, R. L. Mercer, et L. R. Bahl. Continuous speech recognition : statistical methods. C.S.R. group, IBM T.J., 1976.

[JeMe 80] F. Jelinek, R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. Pattern recognition in practice, E. S. Gelsema et L. N. Kanals, Eds Amsterdam, pp. 381-402, The Netherlands : North Holland, 1980.

[Jouv 87] D. Jouvet. Reconnaissance de mots connectés indépendamment du locuteur par des méthodes statistiques. AFCET-INRIA, 1987.

[Juan 85] B. H. Juang. Maximum likelihood estimation for mixture multivariate stochastic observations of Markov chains. AT&T Tech. J., vol. 64, no. 6, pp. 1235-1249, July-Aug. 1985.

[JuRa 85a] B. H. Juang, S. E. Levinson and M. M. Sondhi. Mixture autoregressive hidden Markov models for speech signals. IEEE Trans. Acoust., Speech Signal Processing, vol. ASSP-33, no. 6, pp. 307-309, Mar. 1986.

[JuRa 85b] B. H. Juang, S. E. Levinson and M. M. Sondhi. Maximum likelihood esti-

mation for multivariate mixture observations of Markov chains. IEEE Trans. Inform. Theory, vol. IT-32, no. 2, pp. 1404-1413, Dec. 1985.

[JuRa 86] B. H. Juang, L. R. Rabiner. Mixture Autoregressive hidden Markov models for speaker independent isolated word recognition. IEEE-ICASSP, pp. 41-44, Tokyo, 1986.

[Katz 87] S. M. Katz. Estimation of probabilities from sparse data for the language model component of speech recognizer. IEEE-ASSP-35, pp. 400-401, 1987.

[KoVa 89] H. KOREZLIOGLU et R. VALLET. Identification des caînes de Markov cachées: Applications aux canaux non-linéaires. Douzieme colloque GRETSI, pp. 149-152, 1989.

[KrMC 93] Vikram Krishnamurthy, John B. Moore, and Shin-Ho Chung.Hidden Markov Model Signal Processing in Presence of Unknown Deterministic Interferences. IEEE Transactions on Automatic Control, Vol. 38, No. 1, January 1993.

[Kuba 88] F. Kubala, Y. Chow, A. Derr, M. Feng, O. Kimball, J. Makhoul, P. Price, J. Rohlicek, S. Roucos, R. Schwartz and J. Vandegrift. Continuous speech recognition results of the BYBLOS system on the DARPA 1000-word resource management database. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 291-294. New York, 1988.

[KuBa 88] AMLAN KUNDU and PARAMRIR BAHL. Recognition of handwritten script: A Hidden Markov Model based approach. ICASSP, PP. 928-931, 1988.

[KuHe 89] AMLAN KUNDU and YANG HE. On optimal order in modeling sequence of letters in words of common language as a Markov chain. Pattern Recognition, vol. 24, No. 7. PP. 603-608, 1991.

[KuHB 89] AMLAN KUNDU, YANG HE and PARAMRIR BAHL.Recognition of handwritten word: First and second order Hidden Markov Model based approach. Pattern Recognition, vol. 22, No. 3. PP. 283-297, 1989.

[KuYP 91] AMLAN KUNDU and YANG HE . On Optimal order in modeling sequences of letters in words of common language as a Markov chain. Pattern Recognition, vol. 24, No. 7. PP. 603-608, 1991.

[Krio 88] A. Kriouile. La reconnaissance dautomatique de la parole et les modèles Markoviens cachés. Thèse de Doctorat, Univérsité de Nancy I, Octobre 1990.

[LeGl 92] François LeGland. Algorithme EM pour les chaînes de Markov cachées. GDR 134, Traitement du signal et images, JOURNEES THEMATIQUES, Approches Markoviennes en signal et images, pp. 5-19, 1992.

[LeHo 88] K. F. Lee and H. W. Hon. Large-vocabulary speaker-independent Continuous speech recognition using HMM. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 123-126, New York, 1988.

[Lee 88] K. F. Lee. Large-vocabulary speaker-independent Continuous speech recognition: The SPHINX System, PhD thesis. CMU-CS-88-148,Carnegie-Mellon University, April 1988.

[LeRS 83a] S. E. Levinson, L. R. Rabiner and M. M. Sondhi. An introduction to the application of the theory of probabilistic fonctions of Markov process to automatic speech recognition. The Bell System Technical journal, 62(4), 1983.

[LeRS 83b] S. E. Levinson, L. R. Rabiner and M. M. Sondhi. Speaker independent isolated digit recognition using hidden Markov models. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 1049-1052, 1983.

[Levi 86] S. E. Levinson. Continuously Variable Duration Hidden Markov Models for Automatic Speech Recognition. Comp. Speech and Lang., 1, pp. 29-46, 1986.

[Levi 87] S. E. Levinson. Continuous speech recognition by means of acoustic/phonetic classification obtained from a hidden Markov model. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 93-96, Dallas, 1987.

[Lipo 82] L. R. Liporace. Maximum likelihood estimation for multivariate observations of Markov sources. IEEE. Trans. Inform. Theory, IT-28, pp. 729-734, 1982.

[MaKu 90] W. D. Mao and S.Y. Kung. An object recognition system using stochastic knowledge source and VLSI parallel architecture. Proc. Int. Conf. on PATTERN RECOGNITION, pp. 382-386, June, 1990.

[MaRo 85] J. F. Mari et S. Roucos. Speaker independent connected digit recognition using hidden Markov models. Speech tec., pp. 22-24, New York, April 1985.

[Meri 88a] B. Mérialdo. Phonetic recognition using Hidden Markov Models and Maximum Mutual Information training. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 111-114, New York, 1988.

[Meri 88b] B. Mérialdo. Apprentissage des modèles Markoviens par maximum d'inforamtion mutuelle. 17e JEP-SFA, 1988.

[Mark 13] A. A. Markov. An example of statistical investigation in the text of 'eugene onyegin' illustrating coupling of 'tests' in chains. Proc. Acad. Sci. St. Petersburg VI Ser. 7, pp. 153-162, 1913.

[MDES 87] B. Mérialdo, A. M. Derouault, M. El Beze et S. Soudoplatoff. Reconnaissance de parole avec un très grand vocabulaires. 16e JEP-SFA, 1987.

[NAWF 86] R. Nag, K.H. Wong, F. Fallside Script Recognition using Hidden Markov Models. ICASSP, Tokyo, pp. 2071-2074, 1986.

[Ott 77] J. Ott. Counting methodes (EM algorithme) in human pedigree analysis : linkage

and segregation analysis. Ann. Human Genetics, 40, pp. 443-454, 1977.

[Paul 85] D. B. Paul. Training of HMM recognizers by simulated annealing. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 13-16, 1985.

[Petr 69] T. Petrie. Probabilistic functions of finite state Markov chains. Ann. Math. Stat. 40, pp. 97-115, 1969.

[Pont 88] K. M. Ponting. A statistical approach to the determination of hidden Markov models structure. 7th FASE Symposium, Edinburgh, 1988.

[Pori 82] A. B. Poritz. Linear predictive hidden Markov models and the speech signal. Proc. ICASSP 82, pp. 1291-1294, Paris, May 1982.

[Pori 88] A. B. Poritz. Hidden Markov Models : A Guided Tour. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 7-13, 1988.

[PoRi 86] A. B. Poritz and A. G. Richter. On hidden Markov models in isolated word recognition. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 705-708, Tokyo, 1986.

[RaLe 85] L. R. Rabiner, L. E. Levinson. A speaker-independent, syntax-directed, connected word recognition system based on hidden Markov models and level Bulding. IEEE-ICASSP, 33 pp. 561-573, June, 1985.

[RaLS 83] L. R. Rabiner, L. E. Levinson and M. M. Sondhi. On the application of vector quantization and hidden Markov models to speaker-independent isolated word recognition. Bell Sys. Tech. J., 62, pp. 1075-1105, 1983.

[Rabi 84] L. R. Rabiner. On the application of energy contours to the recognition of connected word sequences. Bell Sys. Tech. J. 63, pp. 1981-1995, Nov 1984.

[Rabi 88] L. R. Rabiner. Mathematical fondations of Hidden Markov Models. H. Niemann, M. Lang et G. Sagerer editors, Recent advances in speech understanding and Dialog Systems, pp. 183-205, Springer Verlag, 1988.

[Rabi 89] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.IEEE,pp. 267-296, 1989.

[RaCh 91] Raymond D. Rimey and Christopher M. Brown. HMMs and Vision: Representing Structure and Sequences for Activw Vision using Hidden Markov Models. The University of Rochester, Computer Science Departement, Technical Report 366, January 1991.

[RaLS 84b] L. R. Rabiner, L. E. Levinson and M. M. Sondhi. On the use of hidden Markov models for speaker independent recognition of isolated words from a medium size vocabulary. Bell Sys. Tech. J. 63, pp. 627-642, April 1984.

[RaJu 86] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. IEEE ASSP Magazine, pp. 4-16, Jan. 1986.

[RaWJ 86] L. R. Rabiner, J. G. Wilpon and B. H. Juang. A model-based connected-digit recognition system using either hidden Markov models or templates. Computer Speech and Langage, 1(2), pp. 167-197, Dec. 1986.

[RaWJ 87] L. R. Rabiner, J. G. Wilpon and B. H. Juang. A performance evaluation of connected digit recognizer. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, Dallas, 1987.

[RJLS 85] L. R. Rabiner, B. H. Juang, S. E. Levinson and M. M. Sondhi. Some properties of continuous hidden Markov model representations. AT&T Tech. J., 64(6), pp. 1251-1270, July-Aug, 1986.

[RJLS 86] L. R. Rabiner, B. H. Juang, L. E. Levinson and M. M. Sondhi. Recognition of isolated digits using hidden Markov models with continuous mixture densities. Bell Sys. Tech. J., 64(6), pp. 1211-1222, July-Aug. 1986.

[RRRG 89] J. R. Rohlicek, W. Russell, S. Roucos and H. Gish. Continuous hidden Markov modeling for speaker independent word spotting. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, 1989.

[RoCo 87] A. E. Rosemberg et A. M. Colla. A connected speech recognition system based on spotting diphone-like segments-preliminary results. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 85-87, Dallas, 1987.

[RoRo 90] ROY L. STREIT and ROSS F. BARRETT. Frequency Line Tracking usig Hidden Markov Models. IEEE Trans. on Acoust., Speech, Signal Processing. Vol. 38, No. 4, April 1990.

[RuMo 85] M. J. Russel and R. K. Moore. Explicit models for automatic speech recognition. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing. pp. 5-8, Tampa,Florida, 1985.

[RuCo 86] M. J. Russel and A. E. Cook. Experiments in speecker-dependent isolated digit recognition using Hidden Markov Models. Proc. Inst. of Acoust. 8, Part 7, pp. 291-298, 1986.

[Schw 84] R. Schwartz, Y. L. Chow, O. Kimball, S. Roucos, M. Krasner and J. Makhoul. Improuved hidden Markov modeling of phonemes for continuous speech recognition. Proc. Int. Conf. Acoust., Speech, Signal Processing, 1984.

[Schw 85] R. Schwartz, Y. L. Chow, S. Roucos, M. Krasner and J. Makhoul. Context-dependent modeling for acoustic-phonetic recognition of continuous speech. Proc. Int. Conf. Acoust., Speech, Signal Processing, pp. 1205-1208, 1985.

[Shan 48] C. C. Shannon. A mathematical theory of communications. Bell Sys. Tech. J. 27, pp. 379-423,623-656, 1948.

[Shan 51] C. C. Shannon. Prediction and entropy of printed english. Bell Sys. Tech. J. 30, pp. 50-64, 1951.

[VALL 92a] Robert VALLET. Utilisation des modèles de Markov cachés en communications numériques. GDR 134, Traitement du signal et images, JOURNEES THEMATIQUES, Approches Markoviennes en signal et images, pp. 185-193, 1992.

[VALL 92b] Robert VALLET. Joint optimum phase and channel parameters estimation for QAM digital modulation. GDR 134, Traitement du signal et images, JOURNEES THEMATIQUES, Approches Markoviennes en signal et images, pp. 194-197, 1992.

[VaSK 85] Y. VARDI, L. SHEPP and L. KAUFMAN. A Statistical Model for Positron Emission Tomography. Journal of the American Statistical Association, pp. 8-20, Vol. 80, No. 389, Applicatoins, March 1985.

[Veij 91] Ari Veijanen. A Simulation-Based Estimator for Hidden Markov Random Fields. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 13, No. 8, August 1991.

[Vent 68] J. K. Vintsyuk. Recognition of words of oral speech by dynamic programming methodes. Kibernetica, vol. 81, no. 8, 1968.

[Vite 67] A. J. Viterbi. Error bounds for convolutional codes and asymptotically optimum decoding algorithm. IEEE Trans. Inform. Theory, IT-13, pp. 260-269, April 1967.

[VlKu 89] J.A. Vlontzos, S.Y. Kung. Hidden Markov Models for character recognition. ICASSP pp. 1719-1722, 1989.

[Well 86] C. J. Wellekens. Global connected digit recognition using Baum-Welch algorithm. IEEE-ICASP, pp. 1081-1084, Tokyo, 1986.

[Well 87] C. J. Wellekens. Explicit time correlation in Hidden Markov Models for speech recognition. Proc. IEEE Int. conf. Acoust. Speech, Signal Processing, pp. 384-386, Dallas, 1987.

[XiEv 91] Xianya Xie and Robin J. Evans. Multiple Target Tracking usig Hidden Markov Models. IEEE Transaction on Signal Processing. Vol. 39, No. 12,pp. 2659-2676 December 1991.

[XiEv 93a] Xianya Xie and Robin J. Evans. Frequency-Wavenumber Tracking usig Hidden Markov Models. IEEE Transaction on Signal Processing. Vol. 41, No. 3,pp. 1391-1394 March 1993.

[XiEv 93b] Xianya Xie and Robin J. Evans. Multiple Frequency Line Tracking with Hidden Markov Models-Further Results. IEEE Transaction on Signal Processing. Vol. 41,

No. 1,pp. 334-343 January 1993.

[XiNa 88] Xiao Gong and Nal-Kuan Huang. Texture Segmentation using Iterative Estimate of Energy States. Proc. Internat. Conf. Pattern Recognition, 1988.

[YaAm 91a] Yang He and Amlan Kundu. Planar Shape Classification using Hidden Markov Model. Proc. Internat. Conf. Pattern Recognition, IEEE 1991.

[YaAm 91b] Yang He and Amlan Kundu. 2-D Shape Classification using Hidden Markov Model. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 13, No. 11, Novembre 1991.

[Yang 88] Yang He. Etended Viterbi algorithm for second order Hidden Markov Process. Proc. IEEE 9th Int. Conf. on Pattern Recognition, pp 718-720, Rome, Italy, 1988.

[Zhao 89] Yunxin Zhao, Lars S. Andersen and Lee E. Atlas. Parameters estimation and restoration of noisy images using Gibbs distribution in Hidden Markov Models. Proc. Internat. Conf. Pattern Recognition, IEEE 1989.

[ZHUQ 91] Qiuming Zhu. Hidden Markov Model for Dynamic Obstacle. IEEE Transaction on Robotics and Automation, pp. 390-397, 1991.

Unité de Recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

Unité de Recherche INRIA Lorraine Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)
Unité de Recherche INRIA Rennes IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399