Guillaume Gravier

Analyse statistique à deux dimensions pour la modélisation segmentale du signal de parole - application à la reconnaissance.

Thèse présentée le 10 Janvier 2000
pour obtenir le grade de docteur
de l'Ecole Nationale Supérieure des Télécommunications
Spécialité Signal et Images

Composition du jury

Jean-Paul Haton	Président
Régine André-Obrecht	Rapporteurs
Bernard Chalmond	
Laurent Younes	Examinateur
Gérard Chollet	Directeurs
Marc Sigelle	

A la mémoire de mon père...

Remerciements

L'aventure relatée dans ce document a débuté le 1er Avril 1995 : non, ce n'est pas une blague! Depuis, bien des gens ont traversé mon univers et la plupart m'ont aidé, peut-être même sans le savoir, à mener à bien ce travail, aussi bien par des apports au niveau scientifique que personnel. C'est donc toutes ces personnes qu'il faudrait remercier mais la liste exhaustive serait bien évidemment trop longue. Je pourrais (peut-être même devrais-je) m'arréter là mais certains méritent quand même une mention spéciale.

L'aventure a donc commencé au fond d'une vallée suisse, dans un café, par une discussion en apparence anodine avec Gérard Chollet. Quelques années plus tard, en cherchant un thème qui pourrait servir de sujet à ma thèse, Gérard a ressorti cette idée et je tiens à le remercier pour m'avoir fait confiance, pour m'avoir fournit les clés nécessaires à la réalisation de cette thèse[1] et pour bien d'autres choses encore. Il me faut également remercier Marc Sigelle qui a bien voulu se joindre à nous dans l'aventure en nous fournissant la clé des champs et en suivant avec une patience exemplaire ce travail jusqu'au bout. Merci a tout les deux!

Les membres du Jury, qui ont eu le courage de se plonger dans la lecture (passionnante?) de ce travail et de fournir de précieux conseils sont également à remercier. Par galanterie, je commencerai par Régine André-Obrecht pour avoir accepter de rapporter ce document, pour les éloges contenues dans le rapport et pour son soutien. Ensuite, ma gratitude va a Bernard Chalmond qui, sans le savoir, est à la base de cette thèse pour m'avoir initié il y a longtemps au secret des champs de Markov et qui a passé une partie de ses vacances de Noël à lire et à annoter ce document. Je tiens également à remercier Laurent Younes pour l'ensemble de ses remarques, toujours pertinentes et constructives, sur mes travaux. Enfin, merci à Jean-Paul Haton qui m'a fait l'honneur de présider mon jury de thèse.

La liste de ceux qui méritent une mention spéciale dans ce document inoubliable continue avec les (parfois) collègues et (souvent) amis que j'ai côtoyé dans les coulisses des divers congrès et autres manifestations scientifiques que nous fréquentons. La place me manque pour écrire une mention spéciale à chacun, mais je tiens néanmoins à en dresser la liste non-exhaustive : la sympathique équipe de Rennes avec Fred, Mohamadou et Raphaël; la terrible équipe

1. Entre autre, une partie de la clé réside dans les 478 999 articles (chiffre non contractuel) qui ont atteris sur mon bureau depuis 5 ans.

d'Avignon avec Jeff, Corinne, Téva, Sylvain, Fred, Pascal, ...; Ivan qu'on sait même plus dans quelle équipe il joue tellement il change souvent; l'équipe "*Ca va ou bien!*" Suisse Romande avec le grand Doms, réfugié politique en Californie, et le petit Johnny; Christophe de l'équipe Nancéene et Fabrice de l'équipe "Parisienne-mais-pas-locale" pour nos longues discussions sur les bières et les HMM; François mon co-équipier local qui a accepté de me supporter après mon transfuge d'après-thèse; les élans de l'informatique (comprenne qui pourra) et en particulier Pierre-Yves et Christophe; Patrick, digne représentant de l'équipe Belge; et plein d'autres encore, jouant dans les diverses équipes mondiales qui font augmenter la connaissance et baisser le niveau des fûts de bières!

Bien évidemment, je ne pouvais m'arréter sans un grand merci pour l'équipe locale des thésards du département Signal (puis TSI): Stéphanie and Co., notre groupe de "rock francais international", Marine, Gabriel, Olivier, Lisa, David, Florence. A tous, merci du fond du coeur pour les bons moments qu'on a passé à ramer sur la même galère! Il convient également d'ajouter un grand merci pour les copains qui, pour la plupart, ne font pas augmenter la connaissance mais qui aident tous à faire baisser le niveau des fûts: Poukse, Mims, Momo, Bubu-Agnès-&-Alban, Boblon, la famille Fize2, Pat & Isa, Lise, ... Bref tout les amis qui ont supporté pendant ces années initiatiques mes crises de haine, d'angoisse, de joie, d'enthousiasme, de pessimisme, de folie, ... (rayez les mentions inutiles).

Enfin, last but not least, je voudrais remercier publiquement du fond du coeur Mónica qui m'accompagne depuis le milieu de cette aventure. Mil gracias por tu amor y tu ayuda, y por soportarme!

Résumé

Une modélisation statistique de la parole est utilisée dans la plupart des applications de reconnaissance de la parole et du locuteur. En effet, le cadre statistique se prête bien à la modélisation des variabilités temporelle et fréquentielle de la parole. Les modèles les plus populaires sont bien sûr les modèles de Markov cachés que l'on peut voir comme la superposition de deux processus aléatoires pour modéliser les deux axes de variabilité. Les modèles de Markov cachés sont en principe associés à une représentation cepstrale du signal de parole. Un des avantages de cette représentation est qu'elle est moins variable comparée à la représentation temps/fréquence. En revanche, les traitements de débruitage sont plus difficiles au niveau du cepstre et le passage du spectre au cepstre est accompagné d'une réduction d'information.

Dans ce travail, nous étudions la modélisation de segments de parole dans le plan temps/fréquence à l'aide du formalisme des champs de Markov. A partir de la formulation d'une chaîne de Markov comme distribution de Gibbs, nous proposons un modèle paramétrique qui peut-être vu comme un modèle de Markov multi-bandes dans lequel une modélisation de la synchronie entre les bandes est ajoutée. Nous définissons une procédure d'estimation des paramètres, au sens du maximum de vraisemblance, ainsi que des stratégies de décodage associées au formalisme des distributions de Gibbs. L'estimation des paramètres est basée sur une généralisation stochastique de l'algorithme EM et s'applique à toutes les distributions de Gibbs pour lesquels les potentiels sont linéaires par rapport aux paramètres. Cette procédure d'estimation des paramètres est appliquée au modèle proposé et validée sur des données simulées. Enfin, le modèle est appliqué à la reconnaissance de mots isolés. Les performances obtenues dans le cas mono-bande sont semblables à celles des HMM. Dans le cas multi-bande, les expériences montrent qu'il est necéssaire de disposer d'un bon modèle *a priori* du processus caché, ce dernier servant de régularisation pour la segmentation. La modélisation de la synchronie inter-bande, proposée comme une première approche pour la modélisation de la parole par champs de Markov, ne s'avère pas suffisante comme modèle *a priori* pour la régularisation et demande à être améliorée. Cependant, elle permet de limiter la baisse du taux de reconnaissance en présence de bruit additif lorsque le nombre de bandes est élévé. L'intérêt principal de ce travail réside dans la formalisation d'un nouveau cadre théorique et des procédures associées pour la modélisation segmentale de la parole.

Summary

Statistical modeling of speech is nowadays used in most of the speech and speaker recognition applications, the stochastic appoach providing an elegant framework to model the variabilities of speech in the time and frequency domains. The most commonly used models are the hidden Markov models which can be seen as the superposition of two stochastic processes to model the two axes of variability. The hidden Markov models are in principle used along with a cepstral representation of the signal. One of the advantages of such a representation is that it is less variable compared to a time/frequency one. On the other hand, denoising is more difficult to implement in the cepstral domain and some information is lost when projecting the spectral representation on the cepstral domain.

In this work, segmental modeling of speech in the time/frequency domain using a Markov random field based approach is studied. Starting from the formulation of a Markov chain in terms of Gibbs distribution, we propose a parametric model that can be seen as a mutli-band model in which a modeling of the synchrony between the bands is added. A maximum likelihood parameter estimation procedure as well as decoding strategies for the random field approach are proposed. The parameter estimation procedure is based on a stochastic generalisation of the EM algorithm and is valid for any Gibbs distribution whose potentials are linear with respect to the parameters. This algorithm is applied to the proposed random field model and validation is performed on simulated data. Finally, the random field model is applied to isolated word recognition. In the mono-band case, the performances of the proposed approach are similar to the ones obtained with hidden Markov modeling. In the multi-band case, the experiments pointed out the fact that a good model of the *a priori* process is needed when the observations become more variable. The prior model is used for regularisation in the segmentation process. Modeling the inter-band synchrony, as proposed in this first approach to random field based speech modeling, turned out to be insufficient as a regularisation prior. However, the decrease of performance with additive noise when the number of bands is high can be limited thanks to this approach. The main interest of this work lies in the formulation of a new theoretical framework and of the associated algorithms for the segmental modeling of speech.

Table des matières

Table des figures

Chapitre 0

Préambule

0.1 Motivations

Deux qualificatifs peuvent s'appliquer au signal de parole: variabilité et redondance. Ces deux aspects de la parole ne sont pas indépendants. En effet, le signal de parole a pour but d'assurer une communication entre deux personnes ou encore entre une personne et une machine. Ce signal étant intrinsèquement variable, de par la physiologie des articulateurs utilisés pour la production de la parole et de par la diversité des locuteurs, et la communication se devant d'être robuste aux conditions environnementales, la redondance de l'information est alors nécessaire.

La variabilité de la parole se retrouve dans le signal à deux niveaux: temporel et fréquentiel. La variabilité temporelle est principalement due aux différentes vitesses d'élocution tandis que la variabilité fréquentielle provient des différentes stratégies articulatoires. Nous illustrons ces variabilités figure 0.1 à travers une représentation temps/fréquence de la parole, le spectrogramme. Les spectrogrammes situés sur la même ligne correspondent aux mêmes mots. Cette figure illustre aussi qu'un segment de parole peut-être considéré comme une image. Bien que variables, les images correspondant aux même mots présentent heureusement des similitudes entre elles.

L'existence de deux axes de variabilités nous a amené à considérer une modélisation bi-dimensionnelle, dans le plan temps/fréquence, du signal de parole. Les techniques classiques de modélisation de la parole s'appuient sur une approche stochastique qui permet de traiter de manière élégante le problème de la variabilité des observations. En revanche, aucune méthode ne s'applique directement aux deux axes de variabilité. Les modèles de Markov cachés, largement utilisés en reconnaissance de la parole, traitent bien évidemment le problème des variabilités temporelle et fréquentielle mais en superposant deux modèles aléatoires: la chaîne de Markov pour la modélisation temporelle et les densités de probabilité associées aux états pour la modélisation fréquentielle. On est alors en droit d'attendre qu'une meilleure modélisation des variabilités permettent d'améliorer les systèmes de reconnaissance de la parole. Il est également en-

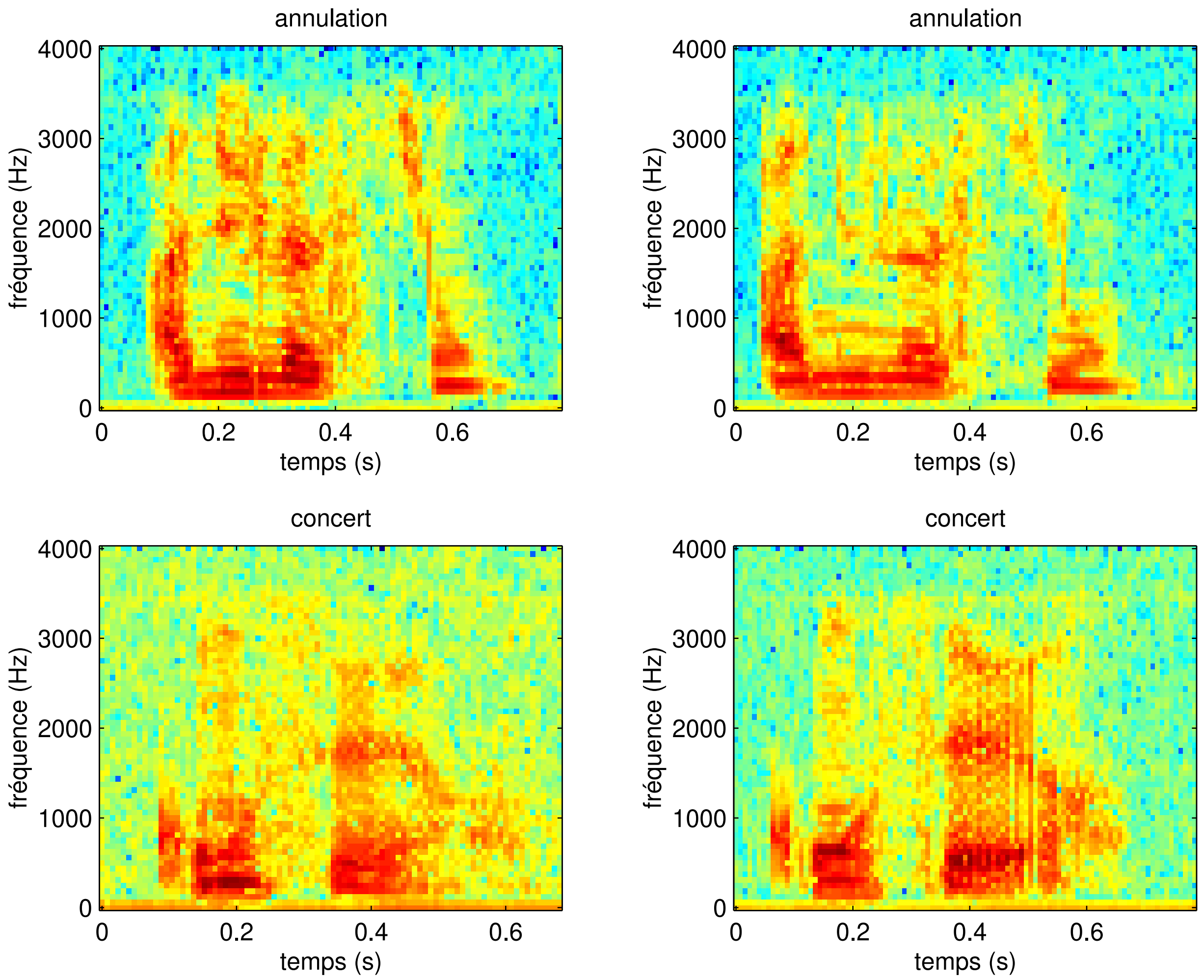


FIG. 0.1 – *Exemples de spectrogrammes pour les deux mots "annulation" et "concert".*

visageable de travailler directement sur la représentation temps/fréquence de la parole pour accroître la robustesse des systèmes de reconnaissance aux variations des conditions d'acquisition et de transmission du signal à l'aide de techniques de traitement d'image.

L'idée du travail présenté dans ce document est donc de modéliser simultanément dans un même processus les deux axes de variabilités. La modélisation doit donc porter sur une représentation temps/fréquence du signal de parole, ce qui présente l'avantage de ne pas s'appuyer sur une projection de cette représentation dans un espace plus restreint comme l'espace cepstral utilisé avec les modèles de Markov cachés. En effet, si la projection dans un sous-espace entraîne une réduction de la variabilité, elle entraîne également une réduction de l'information. Comme nous le mentionnions précédemment, la représentation temps/fréquence nous permet de considérer la parole comme une image. Il est dès lors possible d'envisager d'appliquer des techniques de traitement d'images pour améliorer la qualité de cette représentation d'une part et pour la modélisation d'autre part. En particulier, le cadre théorique des champs de Markov, largement utilisé en traitement d'image pour résoudre des problèmes de segmentation et de débruitage, permet d'envisager une modélisation stochastique des représentations temps/fréquence du signal de parole.

0.2 Organisation du document

Les travaux présentés dans ce document ont donc pour but de développer les techniques permettant la modélisation de la parole dans le plan temps/fréquence en utilisant le formalisme des champs de Markov.

Le document peut se diviser en deux parties. Dans une première partie théorique, nous présentons d'abord les avantages de la modélisation statistique de la parole, ainsi qu'une analyse critique des modèles généralement utilisés pour la reconnaissance de la parole et du locuteur et en particulier, des modèles de Markov cachés. Nous présentons ensuite le formalisme des champs de Markov ainsi que les algorithmes permettant de résoudre les deux problèmes que sont la recherche d'une segmentation optimale et l'estimation des paramètres d'un modèle à partir d'exemples. Nous concluons cette partie théorique en rappelant les liens entre chaînes et champs de Markov.

La deuxième partie du document est consacrée à la définition et à l'étude d'un modèle segmental de la parole basé sur les champs Markoviens. Nous présentons au chapitre 3 un modèle, basé sur une extension du modèle multi-bandes, dans lequel une modélisation de la synchronisation entre les bandes de fréquence est introduite. Après avoir défini des procédures d'estimation des paramètres de ce modèle, nous étudions et comparons ces dernières sur des données simulées. Finalement, le modèle proposé est appliqué à la reconnaissance de mots isolés au chapitre 5 et nous proposons un ensemble de perspectives en guise de conclusion.

Chapitre 1

Modélisation segmentale stochastique de la parole

1.1 De l'intérêt des modèles statistiques

Depuis la fin des années 70, le traitement automatique de la parole, et en particulier la reconnaissance automatique, se fait principalement à l'aide d'une modélisation statistique de segments de paroles. Dans la chaîne de traitement traditionnelle d'un système de reconnaissance de la parole ou du locuteur, la première étape consiste à extraire des caractéristiques pertinentes du signal, en fonction de la tâche visée. Cette phase de paramétrisation du signal permet aussi d'obtenir une représentation plus compacte de l'information contenue dans le signal. Pour s'affranchir des problèmes de non-stationnarités du signal, on a recours à une analyse spectrale ou cepstrale à l'aide d'une fenêtre glissante. A intervalles de temps réguliers, typiquement toutes les 10 ms., on représente la portion de signal considéré par un vecteur acoustique. La plupart des techniques d'analyse acoustique du signal s'appuient sur des hypothèses de production du signal et on trouvera dans [74] un comparatif de la plupart de ces techniques pour la reconnaissance automatique de la parole. En reconnaissance du locuteur, plusieurs représentations du signal sont présentées dans [92] ou encore dans [50]. Les systèmes de reconnaissance n'utilisent donc pas le signal lui-même mais plutôt la suite de vecteurs acoustiques $y = \{y_t \; t = 1, \ldots, T\}$, considérée comme la réalisation d'un processus aléatoire Y.

Les avantages d'une telle conceptualisation sont multiples. Le principal avantage réside dans le fait de pouvoir exprimer intrinsèquement la variabilité de la parole par un modèle mathématique du processus aléatoire Y, ce qui n'est pas possible avec une approche de type appariement de formes où, pour modéliser la variabilité, on a recours à des formes de références multiples ou bien à des distances stochastiques comme la distance de Mahalanobis. Un modèle stochastique sera donc capable de rendre compte des variabilités intra et inter locuteurs. De plus, en supposant que l'on dispose d'un modèle du processus aléatoire, il est alors possible de calculer la vraisemblance d'une observation (*i.e.*

d'une réalisation du processus aléatoire) pour un modèle donné, ce qui permet alors d'exprimer de manière simple les problèmes de reconnaissance de la parole ou du locuteur comme nous le verrons par la suite. Un dernier avantage de ce type d'approche réside dans la possibilité d'avoir un modèle paramétrique du processus aléatoire. Une forme peut donc être définie par un nombre restreint de paramètres plutôt que par une ou plusieurs formes de référence. L'ensemble de ces avantages justifient la prépondérance de l'approche statistique dans la mise en œuvre des systèmes de traitement automatique de la parole.

1.1.1 Approche statistique en reconnaissance de la parole

En reconnaissance de la parole, on cherche à trouver le message prononcé connaissant l'observation, comme illustré par la figure 1.1. Un locuteur prononce une séquence de mots w^* qui donne lieu à une réalisation acoustique $Y = y$. Le décodeur cherche la séquence de mots $\hat{w}$ qui approxime le mieux, selon un critère donné, w^* connaissant l'observation. Dans le cadre d'une approche statistique,

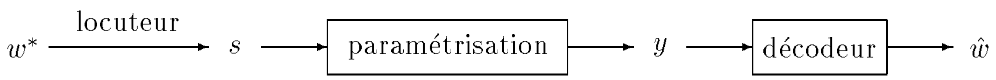


FIG. 1.1 – *Schéma générique d'un système de reconnaissance automatique de la parole*

le critère le plus approprié est naturellement celui du *maximum a posteriori*. On cherche donc la séquence $\hat{w}$ pour laquelle la probabilité *a posteriori* de la séquence w connaissant l'observation $Y = y$ est maximale, ce qui se traduit mathématiquement par

$$\begin{aligned} \hat{w} &= \arg\max_w P[W = w | Y = y] \\ &= \arg\max_w P[Y = y | W = w] \, P[W = w] \ , \end{aligned} \tag{1.1}$$

la deuxième formulation faisant apparaître deux termes distincts, le premier, $P[Y = y|W = w]$, étant appelé score acoustique tandis que le second correspond au score linguistique. En effet, le premier terme représente la probabilité de l'observation (*i.e.* de la suite de vecteurs acoustiques) pour une séquence de mots donnée tandis que le deuxième terme est la probabilité *a priori* de la séquence de mots. Les modèles statistiques de segments de parole[1] rendent possible le calcul du score acoustique comme nous le verrons par la suite dans le cadre de la modélisation par modèles de Markov cachés. Le score linguistique

1. On entend par segment une unité quelconque allant du phone ou encore d'une unité infra-phonétique jusqu'au mot. La notion précise de segment, ou encore d'unité acoustique, sera explicitée ultérieurement.

est donné par un modèle de langage. Ce document traitant principalement de la modélisation acoustique, nous ne parlerons pas plus des modèles de langage et le lecteur pourra trouver une étude sur le sujet dans [33] ou encore dans [76].

1.1.2 Approche statistique en reconnaissance du locuteur

En reconnaissance automatique du locuteur, on distingue généralement deux types d'applications : l'identification, en ensemble ouvert ou fermé, et la vérification d'identité [2]. Nous ne rentrerons pas dans le détail de ces différentes approches et nous nous contenterons de montrer comment l'approche statistique est utilisé en identification et en vérification, le lecteur étant invité à consulter [35] pour de plus amples détails. On distingue aussi généralement les systèmes dépendant du texte pour lesquelles le texte prononcé est connu, des systèmes indépendant du texte.

Identification

Le problème de l'identification est de donner l'identité correspondant à un segment de test $Y = y$, la solution étant déterminée parmi un ensemble d'identités possibles. La figure 1.2 illustre la problématique de l'identification. Comme dans le cas de la reconnaissance de la parole, le module de décision cherche l'identité $\hat{\lambda}$ qui correspond le mieux, selon un critère donné, à un des locuteurs connu du système. A nouveau, on utilise le critère du *maximum a posteriori*

$$w^* \xrightarrow{\text{locuteur } (\lambda^*)} s \longrightarrow \boxed{\text{paramétrisation}} \longrightarrow y \longrightarrow \boxed{\text{décision}} \longrightarrow \hat{\lambda}$$

FIG. 1.2 – *Schéma générique d'un système d'identification du locuteur*

pour prendre cette décision en cherchant l'identité pour laquelle la probabilité *a posteriori* d'une identité λ est maximale connaissant l'observation $Y = y$. De manière formelle, on a donc la règle de décision suivante

$$\begin{aligned} \hat{\lambda} &= \arg\max_{\lambda} P[\Lambda = \lambda | Y = y] \\ &= \arg\max_{\lambda} P[Y = y | \Lambda = \lambda]\, P[\Lambda = \lambda] \ . \end{aligned} \quad (1.2)$$

Dans la deuxième formulation, on voit encore une fois apparaître deux termes, l'un correspondant au score acoustique, le second terme étant lié à la probabilité *a priori* qu'un locuteur donné se présente. Notons que si tous les locuteurs

2. Notons que cette distinction n'est pas aussi nette, la vérification pouvant être considérée comme un problème d'identification en ensemble ouvert, pour un ensemble ne contenant qu'un seul locuteur.

sont équiprobables, la décision se fait alors suivant un critère de maximum de vraisemblance. Le score acoustique, ou encore la vraisemblance de l'observation, pour un locuteur donné est calculé à l'aide de modèles statistiques. Généralement, on utilise des modèles de segments lorsqu'on travaille en mode dépendant du texte, tandis que, dans le cas contraire, on utilise un modèle “global” de la parole.

Vérification d'identité

La vérification d'identité est la tâche qui consiste à dire si un segment de parole a été prononcé par une personne donnée. Contrairement à l'identification, une identité supposée est fournie au système qui doit rejeter ou accepter le locuteur comme étant celui qui a prononcé le segment. La figure 1.3 illustre le fonctionnement d'un tel système. Un locuteur λ^* émet une séquence de vecteurs acoustiques $Y = y$. On cherche alors à déterminer si le segment de test a été prononcé par un locuteur $\Lambda = \lambda$ ou pas. Traditionnellement, la partie

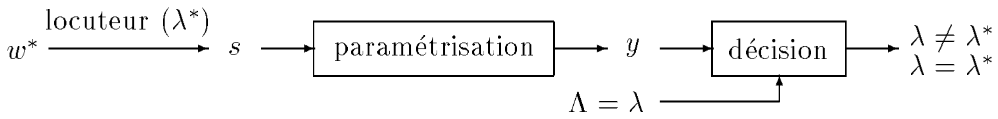


FIG. 1.3 – *Schéma générique d'un système de vérification d'identité par la parole.*

décision est mise en œuvre comme un test statistique binaire d'hypothèse, où H_0 représente l'hypothèse que l'identité supposée correspond à l'identité réelle, H_1 étant l'hypothèse inverse. Selon la théorie bayésienne de la décision, la décision se base sur la comparaison du rapport de vraisemblance à un seuil fixé en fonction des performances recherchées, ce qui nous donne

$$\frac{P_{\lambda=\lambda^*}[Y=y]}{P_{\lambda\neq\lambda^*}[Y=y]} \underset{H_1}{\overset{H_0}{\gtrless}} \beta \ . \tag{1.3}$$

On note ici $P_{\lambda=\lambda^*}[Y=y]$ (resp. $P_{\lambda\neq\lambda^*}[Y=y]$) la vraisemblance de l'observation sous l'hypothèse H_0 (resp. H_1). L'équation (1.3) met clairement en évidence l'intérêt d'avoir un modèle statistique de la parole pour pouvoir calculer les vraisemblances du segment de test sous les deux hypothèses. Comme dans le cas de l'identification, on utilisera généralement des modèles de segments en mode dépendant du texte et un modèle “global” en mode indépendant du texte.

1.1.3 Bilan

Les trois applications présentées précédemment montrent clairement l'intéret d'une modélisation statistique de la parole pour formuler de manière simple les tâches de reconnaissance de la parole et du locuteur. Implicitement, on a défini deux types de modèles appelés jusqu'à maintenant modèles *segmentaux* et modèles *globaux*. Avant de poursuivre, il convient de définir plus précisément ces notions.

En reconnaissance de la parole, par exemple, on doit définir les unités de base, ou segments, que l'on cherche à reconnaitre dans le signal. Lorsqu'on a un vocabulaire limité, les unités de bases peuvent être les mots. Il est bien évident que lorsque la taille du vocabulaire augmente, il n'est plus possible d'avoir un modèle pour chacun des mots du vocabulaire puisque cela nécessiterait alors des corpus d'apprentissage trop grand. On utilise donc des segments plus courts, tels que le phone ou la syllabe, les mots étant décrits comme une suite de ces segments.

On peut utiliser des approches à base de segments en reconnaissance du locuteur lorsqu'on travaille en mode dépendant du texte (voir par exemple [39]). Dans le cas de modèles segmentaux, la technique la plus couramment utilisée est celle des chaînes de Markov cachés (HMM pour Hidden Markov Model) que nous détaillerons par la suite (cf. section 1.2). En mode indépendant du texte, il n'est pas possible d'avoir directement recours à une modélisation segmentale puisqu'on ne connait pas *a priori* les segments présents dans la parole. Deux solutions sont alors possibles. La première consiste à recourir à une modélisation "globale" de la parole, c'est à dire à des modèles plus simples qui tentent de décrire la loi de répartition des vecteurs acoustiques en les considérant indépendants. Le modèle le plus populaire est sans conteste le modèle de mélange de gaussiennes (GMM pour Gaussian Mixture Model) [93]. Une autre approche consiste à se ramener au cas segmental en faisant précéder la reconnaissance du locuteur par une phase de reconnaissance de la parole permettant de déterminer les segments présents [79] ou encore par une phase de classification de segment déterminé à partir du signal [104, 23].

Rappelons enfin que lorsqu'on a défini un modèle paramétrique, il reste deux grands problèmes à résoudre pour pouvoir l'utiliser. Le premier problème est le problème du décodage et consiste à pouvoir calculer la vraisemblance d'une observation pour un modèle donné. Lorsque ce calcul s'avère trop compliqué, ou trop coûteux, on a recours à des approximations comme c'est le cas pour les HMM pour lesquels on utilise l'algorithme de Viterbi [105]. Le deuxième problème à résoudre est celui de l'estimation des paramètres du modèle à partir d'exemples à l'aide de critères tels que le maximum de vraisemblance, le minimum d'erreur de classification ou l'information mutuelle maximale dans le cas d'un apprentissage discriminant [58].

1.2 Chaînes de Markov cachées

Comme mentionné précédemment, la technique la plus utilisée pour la modélisation segmentale du signal de parole se base sur l'utilisation des chaînes de Markov cachés [71, 6, 5, 55, 57]. Le but de cette partie est d'introduire le formalisme des HMM et de passer rapidement en revue les algorithmes permettant de résoudre les deux problèmes évoqués plus haut.

1.2.1 Modélisation

Le modèle HMM suppose que la suite de vecteurs d'observation $\{y_1, \dots, y_T\}$ est stationnaire par morceaux, ce qui signifie que, par morceau, les vecteurs acoustiques suivent la même loi de probabilité. On associe donc au processus Y un processus caché X où X_t est une indicatrice de la loi correspondant à Y_t. Pour modéliser l'évolution temporelle de la parole, la loi du processus X est donné par une chaîne de Markov homogène, généralement d'ordre 1. On représente habituellement le processus X sous la forme d'un automate stochastique comme illustré figure 1.4, une densité de probabilité étant associée à chacun des états de l'automate. Cette représentation de la chaîne de Markov est pratique mais a tendance à cacher le fait que X est un processus aléatoire. Pour mettre ce fait en évidence, on peut également utiliser la représentation sous forme de graphe de dépendance comme le montre la figure 1.5 où les traits continus représente les dépendances entre les variables. Cette représentation fait clairement apparaître X comme la suite de variables aléatoires $X_1, \dots, X_T$, la valeur prise par X à l'instant t ne dépendant que de l'état du processus à l'instant $t-1$. On remarque aussi que l'observation à l'instant t ne dépend que de l'état du processus caché au même instant. Il s'agit d'une hypothèse classique d'indépendance conditionnelle des variables Y_t. De manière plus formelle, un modèle de

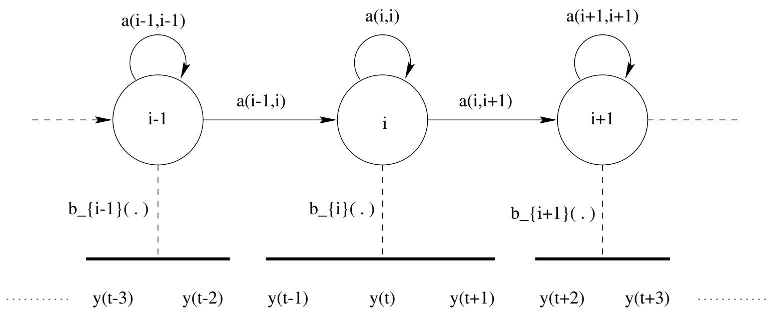


FIG. 1.4 – *Représentation d'un HMM sous la forme d'un automate stochastique.*

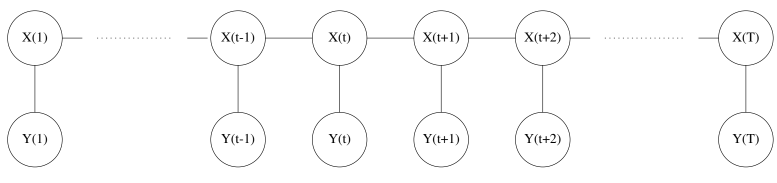


FIG. 1.5 – *Représentation d'un HMM sous la forme d'un graphe de dépendance.*

Markov cachés est défini par le nombre d'états de l'automate N, et l'ensemble de paramètres suivants

$$\theta_N = \{\pi_{i=1,\dots,N} \quad A = (a_{i,j=1,\dots,N}) \quad b_{i=1,\dots,N}(.)\} \ ,$$

où π_i est la probabilité que $X_1 = i$, $a_{i,j}$ la probabilité de passer de l'état i à l'état j, soit $P[X_t = j|X_{t-1} = i]$, et $b_i(.)$ la fonction de densité de probabilité associée à l'état i. Les fonctions de densités associées aux états sont en général des mélanges de gaussiennes mais on trouve des lois discrètes [55, 7] ou encore des HMM hybrides où l'on associe des perceptrons multi-couches aux états [1, 48]. Dans ce document, nous nous intéressons uniquement aux mélanges de gaussiennes. La densité associée au *i-ème* état est donnée par l'équation (1.4) où $\mathcal{N}(.|\mu, \Sigma)$ représente la densité gaussienne de moyenne μ et de matrice de covariance Σ.

$$b_i(y_t) = \sum_{m=1}^{M} \gamma_m \ \mathcal{N}(y_t; \mu_{i,m}, \Sigma_{i,m}) \tag{1.4}$$

Nous avons vu en introduction de ce chapitre que l'approche statistique de la reconnaissance de la parole et du locuteur reposait sur le calcul du terme $P[Y = y|W = w]$ où, en reconnaissance de la parole par exemple, w est une suite de modèles d'unités acoustiques. Dans le cas des chaînes de Markov , le calcul de ce terme est donnée par

$$\begin{aligned} P[Y = y|W = w] &= \sum_{x\in\Omega} P[Y = y|W = w, X = x]\, P[X = x|W = w] &(1.5)\\ &= \sum_{x\in\Omega} b_{x1}(y_1)\pi(x_1) \prod_{t=2}^{T} a(x_{t-1}, x_t) b_{x_t}(y_t) \ , &(1.6)\end{aligned}$$

où Ω représente toutes les séquences d'états de longueur T possibles[3]. Notons que lorsque les densités de probabilités associées aux états sont des mélanges de gaussiennes, l'équation 1.5 ne correspond plus à une probabilité mais à une vraisemblance. Nous conserverons cependant la notation sous forme de probabilité. En pratique, on approxime la somme sur $x \in \Omega$ par le terme prédominant de la somme afin de simplifier les calculs d'une part et de définir la réalisation du processus X correspondant le plus vraisemblablement à l'observation. Connaissant la valeur du processus, on peut alors faire de la segmentation en plus de la reconnaissance, c'est à dire qu'on est en plus capable de donner les frontières des mots reconnus.

1.2.2 Décodage par l'algorithme de Viterbi

Le premier problème que l'on doit résoudre est le calcul du score acoustique $P[Y = y|W = w]$ donné normalement par l'équation (1.5). L'algorithme de

3. On ajoute en pratique des états fictifs non-émetteurs pour représenter le début et la fin du processus. On limite alors l'ensemble Ω à l'ensemble des séquences d'états de longueur $T + 2$, commençant dans l'état initial à $t = 0$ et se terminant dans l'état final à $t = T + 1$.

Viterbi [105, 5] permet d'approximer cette équation par

$$P[Y = y|W = w] \propto \max_{x \in \Omega} P[Y = y|W = w, X = x]\, P[X = x|W = w] \ , \quad (1.7)$$

la segmentation optimale étant donnée par le terme pour lequel le maximum est atteint. Le calcul se fait de manière itérative sur le logarithme de la vraisemblance. En effet, le produit sur le temps de l'équation (1.5) devient alors une somme et on peut maximiser l'ensemble par un ensemble de maximisations locales. Si on définit $\phi_i(t)$ comme étant la log-vraisemblance le long du meilleur chemin finissant dans l'état i à l'instant t, on montre alors aisément que

$$\phi_j(t+1) = \ln b_j(y_{t+1}) + \max_i \left(\phi_i(t) + \ln a_{ij}\right) \ , \quad (1.8)$$

où la maximisation se fait sur l'ensemble des états prédécesseurs possibles pour l'état j. Cette récursion permet de calculer l'approximation (1.7), le score final étant donné par $\max_i \phi_i(T) + \ln a_{i,F}$, $a_{i,F}$ étant la probabilité de "*sortir*" de la chaîne de Markov à partir de l'état i. On peut aussi mémoriser à chaque instant t l'état à partir duquel $\phi_j(t)$ a été maximisé pour retracer le chemin optimal par la suite. On pourra trouver une description plus formelle de l'algorithme dans [90] ou encore dans [57].

1.2.3 Estimation des paramètres

Dans cette partie, on s'intéresse au problème de l'estimation des paramètres d'un modèle à partir d'exemples. Il existe plusieurs solutions à ce problème. On peut utiliser un apprentissage "heuristique" basé sur l'algorithme de Viterbi (connu sous le nom de k-moyenne segmental) ou bien optimiser les paramètres selon un critère donné comme par exemple la maximum de vraisemblance ou encore le minimum d'erreur de classification. La technique la plus populaire, que nous expliquons par la suite, se base sur un critère de maximum de vraisemblance et utilise l'algorithme EM [25].

Apprentissage par Viterbi

L'estimation des paramètres d'un modèle peut se faire à l'aide de l'algorithme de Viterbi associé à des estimateurs empiriques [91]. En effet, si l'on considère un exemple d'apprentissage, l'algorithme de Viterbi permet de déterminer la séquence d'états cachée x^* la plus vraisemblable connaissant cette observation. On peut alors réestimer les moyennes et les variances des gaussiennes pour un état i avec des estimateurs empiriques en prenant en compte les observations associées à l'état i le long du chemin de Viterbi. Par exemple dans le cas où les densités associées aux états sont des mono-gaussiennes, on a pour la moyenne de l'état i

$$\mu_i = \frac{\sum_{t=1}^{T} y_t \, \delta(x_t^* = i)}{\sum_{t=1}^{T} \delta(x_t^* = i)} \;, \tag{1.9}$$

où $\delta(x_t = i)$ est la fonction de Kronecker définie par

$$\delta(x_t = i) = \begin{cases} 1 & \text{si } x_t = i \\ 0 & \text{sinon} \end{cases} \;. \tag{1.10}$$

Cette stratégie d'initialisation est appliquée itérativement jusqu'à convergence de la vraisemblance.

L'algorithme EM

L'algorithme Expectation-Maximization [25] permet d'estimer de manière itérative les paramètres d'un modèle au sens du maximum de vraisemblance lorsqu'on ne peut pas maximiser directement la fonction de vraisemblance. En effet, dans le cas des HMM, on cherche l'ensemble de paramètres $\hat{\theta}$ tel que

$$\hat{\theta} = \arg\max_{\theta} P_\theta[Y = y] \;, \tag{1.11}$$

ce qui n'est pas possible directement pour la fonction de vraisemblance (1.5) du fait de la somme sur toutes les réalisations possibles de X. On peut voir ce problème comme un problème en données incomplètes. En effet, on sait résoudre l'estimation au sens du maximum de vraisemblance pour le processus conjoint (X, Y) mais pas pour Y seul. Le processus Y est donc "incomplet" pour ce problème. L'algorithme EM travaille sur les données complètes et le principe en est de former la fonction auxiliaire (étape E) définie par

$$Q(\theta, \theta^{(n)}) = E_{\theta^{(n)}}[\ln p_\theta(x, y) | Y = y] \;, \tag{1.12}$$

où $\theta^{(n)}$ représente une estimation courante à l'étape n des paramètres du modèle et $p_\theta(x, y)$ la vraisemblance des données complètes, et de maximiser la fonction auxiliaire par rapport aux paramètres θ (étape M). On peut montrer que cet algorithme converge vers un maximum local de la vraisemblance des données [100]. Dans le cas des chaînes de Markov cachées et pour une seule observation, l'équation (1.12) devient

$$Q(\theta, \theta^{(n)}) = \sum_{i=1}^{N} \ln(\pi_i)\gamma_i(1) + \sum_{i,j=1}^{N} \ln(a_{ij}) \sum_{t=2}^{T} \gamma_{ij}(t) + \sum_{i=1}^{N}\sum_{t=1}^{T} \ln(b_i(y_t))\gamma_i(t) \;, \tag{1.13}$$

avec

$$\gamma_i(t) = P_{\theta^{(n)}}[X_t = i | Y = y] \tag{1.14}$$

$$\gamma_{ij}(t) = P_{\theta^{(n)}}[X_{t-1} = i, X_t = j | Y = y] \;. \tag{1.15}$$

L'étape E de l'algorithme consiste donc à estimer ces deux fonctions ce qui se fait de manière exacte grâce à l'algorithme *forward-backward* [8, 90]. Pour l'étape de maximisation, l'annulation des dérivées partielles de l'équation (1.13) par rapport aux paramètres sous les contraintes

$$\sum_{i=1}^{N} \pi_i = 1$$
$$\sum_{j=1}^{N} a_{i,j} = 1 \quad \forall i \in [1, N] \ ,$$

nous donne, dans le cas mono-gaussien et pour une observation unique, les équations de remise à jour suivantes:

$$\pi_i^{(n+1)} = \frac{\gamma_i(1)}{\sum_j \gamma_j(1)} \tag{1.16}$$

$$a_{ij}^{(n+1)} = \frac{\sum_t \gamma_{ij}(t)}{\sum_k \sum_t \gamma_{ik}(t)} \tag{1.17}$$

$$\tag{1.18}$$

$$\mu_i^{(n+1)} = \frac{\sum_t y_t \gamma_i(t)}{\sum_t \gamma_i(t)} \tag{1.19}$$

$$\Sigma_i^{(n+1)} = \frac{\sum_t (y_t - \mu_i^{(n+1)})(y_t - \mu_i^{(n+1)})' \gamma_i(t)}{\sum_t \gamma_i(t)} \ . \tag{1.20}$$

Les équations de remise à jour des paramètres se généralisent très facilement au cas plus réaliste pour lequel les densités de probabilités associées aux états sont des mélanges de gaussiennes et lorsqu'on dispose de plusieurs observations pour l'apprentissage.

1.2.4 Commentaires sur le modèle HMM

En dehors de ses bonnes performances, il existe plusieurs autres raisons à la popularité et au succès des chaînes de Markov cachés. En effet, comme nous l'avons vu, on dispose d'algorithmes efficaces pour traiter les problèmes de reconnaissance de la parole à l'aide de ces modèles. Notamment, l'algorithme de Viterbi peut s'adapter pour rechercher la meilleure solution (ou le meilleur chemin) en construisant dynamiquement l'espace de recherche et en intégrant le modèle de langage [77, 82]. Les modèles de Markov peuvent également se combiner aisément pour pouvoir faire, par exemple, un modèle de mot à partir de modèles de phones ou encore d'allophones (voir par exemple [57] pour plus de précision). Ces différentes raisons, entre autres, justifient l'utilisation des chaînes de Markov dans les systèmes de reconnaissance grand vocabulaire (voir par exemple [4, 17]). Parmi les autres raisons, on pourra citer la possibilité de partager des paramètres entre états (technique connue sous le nom de "*state tying*") ainsi que l'adaptation au locuteur et/ou à l'environnement selon un critère de maximum *a posteriori* [37] ou de régression linéaire par maximum de vraisemblance (Maximum Likelihood Linear Regression) [67].

Malgré cela, la modélisation par chaîne de Markov cachée admet des limitations et quelques inconvénients. Le premier problème concerne la modélisation des durées. En effet, un HMM d'ordre 1 fournit une mauvaise modélisation de la durée d'un événement acoustique. De manière implicite, on a pour chaque état un modèle de durée exponentiel défini par les probabilités de transitions. D'une part un modèle exponentiel n'est pas forcément approprié et, d'autre part, il n'a pas un très grand rôle, l'influence des probabilités de transition dans le score étant quasiment négligeable par rapport aux vraisemblances des observations[4]. Pour résoudre ce problème de modélisation de la durée on peut utiliser des HMM d'ordre supérieur à 1 [70, 32] pour avoir, en théorie, un meilleur modèle de durée, ou ajouter explicitement une loi de durée aux états [16, 94]. Les deux solutions mentionnées n'ont cependant pas apporté d'amélioration significative des performances des HMM. Une explication possible est d'une part que le modèle exponentiel n'est pas pertinent et d'autre part que les probabilités *a priori* n'ont pas une grande influence même lorsqu'on introduit une modélisation explicite de la durée.

La formulation par HMM utilise l'hypothèse d'indépendance conditionnelle des observations par rapport au processus caché X. Sans pour autant considérer les observations comme strictement indépendantes, le modèle n'utilise pas explicitement la corrélation entre deux trames successives. Quelques tentatives pour remédier à ce problème, comme les HMM auto-regressifs [88, 59] ou la modélisation explicite [106, 63] ou implicite [85] de ces corrélations, n'ont pas non plus donné de résultats satisfaisants. L'utilisation des coefficients cepstraux dérivés [34] est la seule solution retenue qui permet de prendre en compte de manière indirecte les corrélations entre trames successives. Kenny *et al.* [61] montrent que la modélisation explicite des corrélations entre trames successives à l'aide d'un modèle de prédiction linéaire n'améliore pas les taux de reconnaissance lorsqu'on utilise les dérivés des coefficients cepstraux.

Enfin, dans le but de réduire le nombre de paramètres des HMM, on utilise une représentation cepstrale du signal. En effet, les coefficients cepstraux sont connus pour être décorrélés ce qui permet d'utiliser des densités gaussiennes dont les matrices de covariance sont diagonales. Cependant, la représentation cepstrale s'appuie sur une hypothèse source-filtre de production du signal et n'est pas forcément la mieux adaptée au problème de la reconnaissance de la parole. En particulier, la représentation cepstrale est sensible aux bruits et n'est donc pas robuste face à la diversité des canaux de transmissions.

---

4. Voir l'anecdote relatée dans [13] à ce propos.

1.3 Variantes autour des chaînes de Markov

1.3.1 Modèles mutli-flux

Depuis quelques années, on a vu apparaître dans la littérature une nouvelle extension des HMM : les modèles de Markov multi-flux [15, 14, 49, 19]. Ce modèle consiste à représenter la parole par plusieurs flux parallèles indépendants, chaque flux étant modélisé par un HMM. A un certain niveau, soit régulièrement, soit à un état donné, les scores sur chacun des chemins sont recombinés. La figure 1.6 illustre un tel modèle. Les flux utilisés peuvent être de nature différente [56] mais la procédure la plus courante consiste à diviser la bande passante du signal en sous bandes qui sont ensuite traitées indépendemment. Cette dernière approche trouve sa justification dans les travaux de Fletcher sur la perception humaine [2]. On retrouve cette approche en reconnaissance de la parole, principalement dans les applications où l'environnement est un facteur limitatif [15, 49], aussi bien qu'en reconnaissance du locuteur [10, 3]. Ce modèle

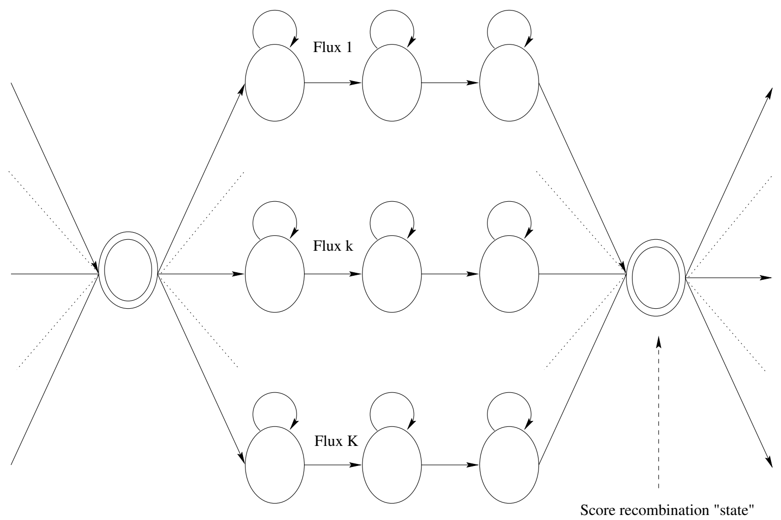


FIG. 1.6 – *Schéma générique d'un modèle de Markov caché multi-flux.*

peut aussi être vu comme un HMM de topologie non strictement gauche-droite pour lequel les états seraient "*partagés*". En effet, on peut regrouper les états dans des combinaisons d'états pour obtenir une chaîne de Markov factorielle comme dans Jordan [41]. Le principe est de construire une nouvelle chaîne dont un état est en fait la combinaison des états des chaînes correspondants à chacun des flux. Par exemple, pour un modèle bi-flux, être simultanément dans l'état i du premier HMM et dans l'état j du deuxième correspond à un état, que l'on pourrait noter ij, de la chaîne partagée équivalente. On trouvera dans [20] une description plus précise de ce principe. On obtient alors un espace d'état plus complexe.

De manière plus formelle, pour un modèle à K flux, on dispose d'une observation $\{y_{t,k} \;\; t=1,\ldots,T,\, k=1,\ldots,K\}$. La vraisemblance d'une observation $Y=y$ pour un modèle W est alors donnée par

$$P[Y=y|W=w] = \prod_{k=1}^{K} P[Y_k = \{y_{1,k}\ldots y_{T,k}\}|W=w]\, P[E_k|W=w] \;\; , \qquad (1.21)$$

le terme $P[E_k|W=w]$ mesurant la fiabilité du décodage dans la bande k. La fiabilité peut-être mesurée en fonction du rapport signal sur bruit ou en fonction de l'information véhiculée dans les différentes bandes. On peut encore apprendre d'autres formes de recombinaison, notamment en utilisant des réseaux de neurones [15] où dans ce cas, l'équation (1.21) prend la forme générale

$$\ln P[Y=y|W=w] = f(B, \{\ln P[Y_k = \{y_{1,k}\ldots y_{T,k}\}|W=w], \forall k\}) \;\; , \qquad (1.22)$$

B étant l'ensemble des paramètres de recombinaison et $f(.)$ une fonction quelconque. L'étape de recombinaison est un des points cruciaux de l'approche multi-bande comme le montrent les résultats de Hermansky [49] reportés dans le tableau 1.1. Ces résultats portent sur la reconnaissance de 13 mots isolés avec un reconnaisseur à deux sous-bandes ([111,1330] et [1220,4000] Hz), pour de la parole de qualité téléphonique.

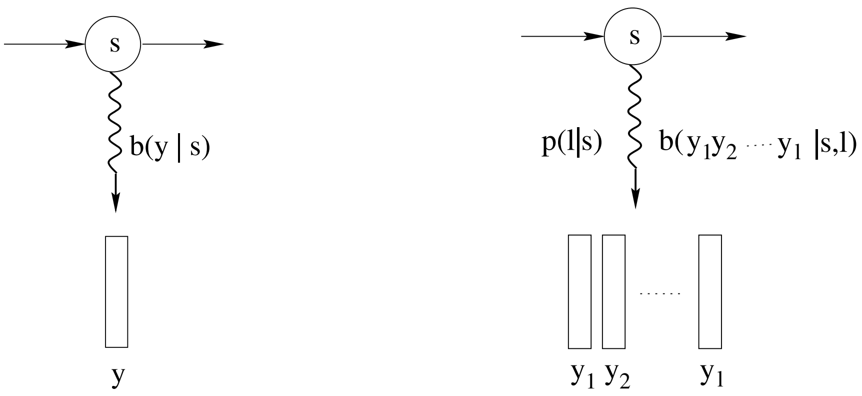
Reconnaisseur	*Erreur (en %)*
Système de référence	3.85
sous-bande 1 ([111,1330] Hz)	14.0
sous-bande 2 ([1220,4000] Hz)	10.65
Combinaison linéaire	4.2
MLP	2.73

TAB. 1.1 – *Intérêt de la recombinaison de scores dans les modèles multi-bandes pour la reconnaissance de mots isolés (d'après [49])*

Parmi les questions ouvertes sur les HMM multi-bandes, le choix du nombre de bandes et leurs bandes passantes respectives est important. On trouve la plupart du temps des systèmes à 3, 5 ou 7 bandes. La recombinaison des sous-bandes par réseau de neurones (MLP pour Multi Layer Perceptron), qui semble jusqu'ici la meilleure solution, est aussi confrontée à la nécessité de disposer de données de validation afin d'apprendre les poids du réseau. Enfin, malgré l'étape de recombinaison, les HMM multi-bandes traitent les différentes sous-bandes de manière indépendante, ce qui nous parait être une hypothèse fausse.

1.3.2 Modèles de segments

Si les modèles multi-bandes permettent une modélisation acoustique plus robuste aux bruits additifs, ils n'apportent pas de solutions aux faiblesses de la modélisation par HMM évoquées précédemment. En effet, les modèles multi-bandes reposent toujours sur l'hypothèse d'indépendance conditionnelle des trames et sur un modèle de durée lié à la modélisation par chaîne de Markov. Une nouvelle famille de modèles, appelés modèles de segments ou modèles



(a) (b)

FIG. 1.7 – *Principe des modèles de segments (d'après [84]). La figure (a) illustre le principe des HMM tandis que la figure (b) représente la modélisation par modèle de segment.*

de trajectoire, a été récemment proposée dans la littérature [84]. Dans cette modélisation, la parole est décrite comme une suite de segments. Un modèle est associé à chaque segment et la suite des segments est définie comme un processus Markovien d'ordre 1. On a donc un segment plutôt qu'une seule trame associé à un état de la chaîne de Markov, comme illustré figure 1.7. Des segments de la taille d'un phone sont en général utilisés bien qu'il soit possible de modéliser des segments de taille quelconque comme dans [60] ou [42]. Un segment est modélisé par une loi de durée explicite ou implicite, $p(l|s)$, et par l'ensemble des densités d'emmission $b(y_1, \ldots, y_l|s, l)$. La technique communément utilisée consiste à diviser un segment en régions en associant à chaque région une densité d'emmission dont les paramètres sont constants. Il est alors nécessaire de définir une correspondance entre les trames et les régions. La correspondance peut se faire soit de manière déterministe, par l'intermédiaire d'une table de correspondance ("*look-up* table") ou d'une trajectoire échantillonnée[5], ou encore de manière dynamique en utilisant les algorithmes de programmation dynamique pour mettre en correspondance une trajectoire donnée avec un nombre de régions fixé à l'avance. Enfin, plusieurs possibilités sont offertes pour la forme des densités d'un ensemble de trames pour un segment et une région donnés. Certains modèles supposent l'indépendance conditionnelle des données au sein d'une région [83, 27], d'autres utilisant la formulation Gauss-Markov [26, 30] ou encore une formulation de la trajectoire sous la forme d'un système linéaire dynamique [31].

Les modèles de trajectoire ont été utilisé avec succès dans des systèmes de reconnaissance grand vocabulaire.

5. On appelle trajectoire le chemin suivi par les vecteurs de paramètres acoustiques.

1.4 Bilan

Ce premier chapitre nous a permis de souligner l'intérêt des modèles stochastiques qui offrent de nombreuses facilités pour résoudre les problèmes de reconnaissance de la parole ou du locuteur. Entre autre, l'intérêt principal d'une telle approche vient du fait que les modèles stochastiques permettent de prendre en compte les variabilités intra- et inter-locuteurs, ces variabilités se traduisant aussi bien dans le domaine fréquentiel que temporel. Les modèles de Markov cachés sont le plus souvent utilisés pour la modélisation de segments de parole mais nous avons vu l'émergence de nouvelles techniques. L'approche multi-bande montre qu'une meilleure modélisation de la parole dans le domaine fréquentiel, réalisée dans ce modèle par la division du signal en sous-bandes, permet une moins grande sensibilité des modèles au bruit. Les résultats obtenus avec les modèles de trajectoire montrent également qu'une meilleure modélisation de la parole est importante. Il est donc naturel de s'intéresser à la modélisation de la parole dans le plan temps/fréquence par un processus stochastique à deux dimensions. Les champs de Markov, présentés au chapitre suivant, sont adaptés à ce genre de modélisation.

Chapitre 2

Champs de Markov

Dans ce chapitre, nous introduisons la théorie des champs Markoviens d'une manière générale et présentons les algorithmes de simulation et d'optimisation. L'objectif du chapitre n'est pas de faire un cours sur les champs Markoviens et les distributions de Gibbs mais plutôt d'introduire le formalisme et les algorithmes que nous utiliserons par la suite. Pour une présentation plus formelle et plus complète des champs de Markov nous renvoyons le lecteur à [28] ou bien encore à [99] dont nous nous sommes largement inspirés.

Les champs de Markov ont fait leur apparition en traitement d'image vers le milieu des années 80 avec les travaux de Geman et Geman [38]. Initialement issu de la physique statistique [54], ce modèle fait maintenant parti des outils classiques du traitement d'image. Il s'agit d'un modèle statistique de l'image ou d'un processus caché dont l'image à proprement parler est l'observation. Dans le premier cas, les champs de Markov sont utilisées, par exemple, pour la détection de texture. Dans le cas où l'observation est liée à un processus caché, les champs de Markov sont utilisés pour résourdre des problèmes tels que la segmentation ou le débruitage d'images. Par la suite, nous nous intéresseront principalement au cas des champs cachés dans la mesure où le problème qui nous intéresse s'apparente à un problème de segmentation.

2.1 Notations, définitions

2.1.1 Définition d'un champ de Markov

On considère un champ aléatoire X défini sur un treillis fini S de taille $|S|$, la variable aléatoire X_s associée au point du treillis, ou site, s prenant ses valeurs dans un ensemble discret fini E. Les réalisations de X appartiennent donc à $\Omega = E^{|S|}$. Un tel champ sera dit Markovien si la probabilité d'observer une valeur donnée en un site connaissant les valeurs prises sur les autres sites du treillis ne dépend que d'un nombre limité de sites dits "voisins". Cette définition informelle indique en fait que le champ X est défini par un ensemble de relations locales contraintes par un système de voisinage V sur S vérifiant les conditions suivantes

$$V_s = \{t\} \ \ t.q. \left\{ \begin{array}{l} s \notin V_s \\ t \in V_s \Longrightarrow s \in V_t \end{array} \right. \quad ,$$

en notant V_s l'ensemble des sites voisins de s. Le système de voisinage étant défini, un champ Markovien vérifie donc la relation

$$P[X_s = x_s | X_{S\backslash s} = x_{S\backslash s}] = P[X_s = x_s | X_{V_s} = x_{V_s}] \ \ ,$$

$X_{S\backslash s}$ désignant le champ X sur S privé du site s, $x_{S\backslash s}$ la réalisation associée, X_{V_s} étant le champ restreint au voisinage du site s.

Pour un système de voisinage donné, on obtient un système de cliques, une clique étant soit un singleton, soit un ensemble de sites mutuellement voisins. La figure 2.1 illustre la notion de clique dans le cas de voisinages en 4-connexité et en 8-connexité. On notera $\mathcal{C}$ l'ensemble des cliques associées à un système de voisinage.

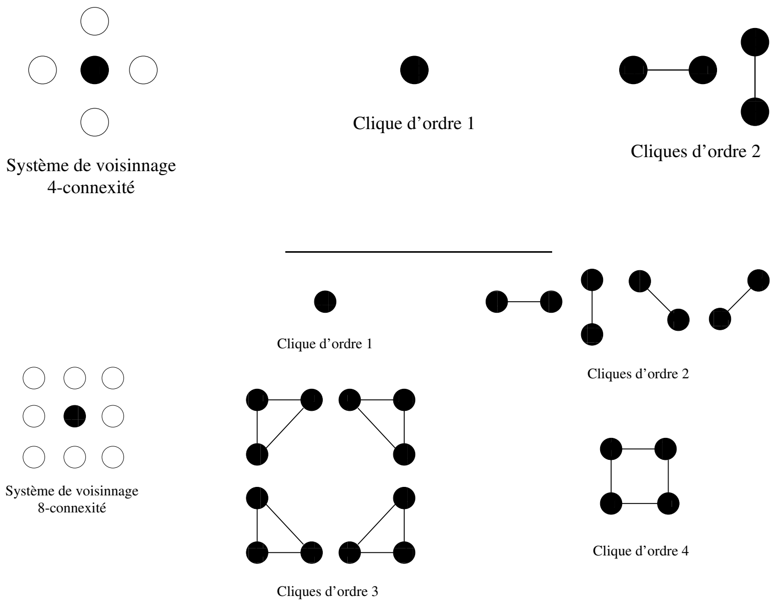


FIG. 2.1 – *Cliques associées à des systèmes de voisinage en 4 et 8 connexité.*

2.1.2 Probabilités d'une configuration

Probabilité *a priori*

Le théorème de Hammersley-Clifford [11] établit le lien fondamental entre champs de Markov et distributions de Gibbs, ce qui permet d'exprimer la probabilité d'une configuration pour un champ de Markov en terme de distribution de Gibbs.

Théorème 1 (Hammersley-Clifford) *En supposant S fini ou dénombrable, un système de voisinage V borné et un espace d'états E discret, alors les deux propositions suivantes sont équivalentes :*

- *X est un champ de Markov relativement à V tel que $P[X=x]>0\ \forall x \in \Omega$*
- *X est un champ de Gibbs de potentiel associé à V*

La probabilité d'une configuration pour un champ de Markov est alors donnée sous la forme d'une distribution de Gibbs définie par

$$P[X=x] = \frac{1}{Z}\exp\left(-U(x)\right) = \frac{1}{Z}\exp\left(-\sum_{c\in\mathcal{C}} U_c(x)\right) \quad , \tag{2.1}$$

la constante Z étant un terme de normalisation appelé fonction de partition et défini par

$$Z = \sum_{x\in\Omega} \exp\left(-U(x)\right) \quad . \tag{2.2}$$

Le terme $U(x)$, correspondant à l'énergie associé à la configuration x, est en fait la somme des énergies locales d'interactions $U_c(x)$ associées à chaque clique définie par le système de voisinage. Une énergie de clique est une fonction quelconque des variables x_s pour s appartenant à la clique c[1]. Notons que l'énergie d'une configuration est inversement proportionelle à sa probabilité, une énergie forte indiquant une configuration instable. Pour un site s donné, on montre que la probabilité d'observer une valeur donnée en ce site est définie en terme d'énergie locale par

$$P[X_s = x_s | X_{S\backslash s} = x_{S\backslash s}] = \frac{\exp -U_s(x_s|x_{V_s})}{\displaystyle\sum_{i\in E} \exp -U_s(i|x_{V_s})} \quad , \tag{2.3}$$

l'énergie locale $U_s(.|x_{V_s})$ étant la somme sur toutes les cliques contenant s des énergies de clique, soit

$$U_s(x_s|x_{V_s}) = \sum_{c \text{ t.q.} s\in c} U_c(x) \quad .$$

Probabilité *a posteriori*

Comme nous l'évoquions en introduction de ce chapitre, on suppose parfois que l'observation dont on dispose est une version dégradée d'une réalisation d'un champ de Markov. On parle alors de champs de Markov cachés. En effet, la configuration Y dont on dispose est considérée comme une version dégradée, ou bruitée, d'un champ de Markov X, le lien entre les deux champ étant fourni

1. La notation $U_c(x)$ est donc un abus puisque l'énergie de clique ne dépend pas de x mais de sa restriction aux points du treillis appartenant à la clique.

par un terme d'attache aux données. Dans la plupart des cas, on pose l'hypothèse d'indépendance conditionnelle des observations, c'est-à-dire que l'on suppose que les variables Y_s sont indépendantes conditionnellement au champ X. On peut également se munir d'un modèle pour la dégradation en associant une densité de probabilité $p[Y_s = y_s | X_s = x_s]$, notée $p(y_s|x_s)$ par mesure de simplicité. La vraisemblance d'une observation y conditionnellement à X est alors donnée par

$$\begin{aligned} p(Y = y | X = x) &= \prod_{s \in S} p(y_s | x_s) \\ &= \exp\left(-U(y|x)\right) \quad , \end{aligned}$$

en posant

$$U(y|x) = \sum_{s \in S} -\ln p(y_s|x_s) \quad . \tag{2.4}$$

La probabilité *a posteriori* du champ X est donnée par

$$P[X = x | Y = y] \propto \exp\left(-U(x) - U(y|x)\right) \quad . \tag{2.5}$$

Cette formulation montre donc que, sous l'hypothèse d'indépendance conditionnelle des observations qui permet d'écrire $P[Y = y | X = x]$ sous la forme d'une distribution de Gibbs, la distribution *a posteriori* pour le champ X reste une distribution de Gibbs, ce qui signifie que le champ conditionnel à l'observation est également un champ de Markov. Ce dernier résultat, très utilisé dans le cadre de la restauration et de la segmentation d'image, indique qu'il est possible d'utiliser les algorithmes classiques de simulation et d'optimisation des camps de Markov aussi bien pour la distribution *a priori* que pour la distribution *a posteriori*.

2.2 Algorithmes de simulation et d'optimisation

Nous nous intéressons dans cette partie aux algorithmes qui permettent soit de faire un tirage aléatoire suivant la loi donnée par une distribution de Gibbs soit, dans le cas d'un champ caché, de trouver la configuration la plus vraisemblable pour une observation donnée.

2.2.1 Simulation d'une configuration

Nous disposons de deux algorithmes pour réaliser un tirage aléatoire selon une distribution de Gibbs, l'échantillonneur de Gibbs [38] et l'algorithme de Metropolis [73], les deux algorithmes étant basés sur la construction itérative d'une suite d'images, en choisissant en chaque site une valeur suivant la loi conditionnelle locale.

Echantillonneur de Gibbs

Dans cet algorithme itératif, on construit une suite de réalisations $x^{(n)}$ ($n \in \mathbb{N}$), cette dernière étant construite à partir de $x^{(n-1)}$ en choisissant un site s quelconque et en tirant aléatoirement sa valeur en fonction des probabilités conditionnelles locales. On a donc les étapes suivantes pour passer de $x^{(n-1)}$ à $x^{(n)}$:

- choix d'un site s à mettre à jour (de manière aléatoire ou suivant un ordre de visite prédéfini)
- calcul de $P[X_s = i|x_{V_s}^{(n-1)}]$ $\forall i \in E$ donné par l'équation (2.3).
- choix d'une nouvelle valeur pour X_s selon la loi $P[X_s = i|x_{V_s}^{(n-1)}]$

On peut montrer que la suite $x^0, \ldots, x^{(n)}$ est la réalisation d'un processus aléatoire X dont la loi est celle d'une chaîne de Markov homogène dont la loi d'équilibre n'est autre que la loi globale du champ. Si l'on balaye l'ensemble des sites une infinité de fois, on réalise donc bien un tirage aléatoire du champ suivant la distribution de Gibbs donnée.

Algorithme de Metropolis

Le principe de l'algorithme de Metropolis est très voisin de celui de l'échantillonneur de Gibbs. Il s'agit également d'un algorithme itératif qui construit une suite de champs $x^{(n)}$, ce dernier étant construit à partir de $x^{(n-1)}$ en utilisant la dynamique suivante :

- choix d'un site s à mettre à jour
- tirage aléatoire d'une valeur $i \in E$ selon une loi uniforme
- calcul de la variation d'énergie $\Delta U = U_s(i|x_{V_s}^{(n-1)}) - U_s(x_s^{(n-1)}|x_{V_s}^{(n-1)})$ liée au changement $x_s^{(n-1)} \to i$
- Si $\Delta U < 0$ alors $x_s^{(n)} = i$, sinon

$$x_s^{(n)} = \begin{cases} i \text{ avec la probabilité } p = \exp(-\Delta U) \\ x_s^{(n-1)} \text{ avec la probabilité } 1-p \end{cases}$$

L'algorithme de Metropolis est donc relativement similaire à celui de Gibbs, la différence principale résidant dans le choix *a priori* d'une valeur pour le site plutôt que de considérer toutes les valeurs possibles. A chaque étape, le nombre de calculs est beaucoup moins élevé pour l'algorithme de Metropolis ce qui le rend souvent plus rapide. Cependant, la taux d'acceptation [2] étant plus faible que pour l'échantillonneur de Gibbs, la convergence vers la loi d'équilibre peut être plus longue. Le principe de convergence est le même que pour l'algorithme de Gibbs mais pour un noyau de transition de la chaîne de Markov différent.

2. La taux d'acceptation est le taux de changement entre $x_s^{(n-1)}$ et $x_s^{(n)}$.

2.2.2 Optimisation d'une configuration

Dans ce deuxième problème, on se place dans le cadre des champs cachés et on cherche à déterminer la meilleure configuration pour X connaissant l'observation $Y = y$. On entend ici par meilleure configuration, la configuration qui maximise la probabilité *a posteriori* du champ X. Pour déterminer une configuration d'énergie minimale, c'est-à-dire de probabilité maximale, nous disposons de l'algorithme des modes conditionnels itérés (ICM pour Iterated Conditionnal Modes) proposé par Besag [12] ainsi que du recuit simulé initialement proposé par Kirkpatrick [64] et repris dans le cadre des champs Markoviens par Geman et Geman [38].

Il est à noter qu'il existe bien évidemment d'autres critères que celui du *maximum a posteriori* (MAP). En effet, si on définit un estimateur $x = f(y)$ de $\Omega \longrightarrow \Omega$ et qu'on se munit d'une fonction de coût $L(.,.)$ de $\Omega \times \Omega \longrightarrow \mathbb{R}$ vérifiant

$$\forall x \in \Omega,\ \forall x' \in \Omega, \quad L(x,x') \geq 0 \ \text{ et } \ L(x,x') = 0 \Longleftrightarrow x = x' \ ,$$

on peut alors définir l'estimateur optimal (ou encore la fonction d'estimation $\hat{f}(.)$ optimale) comme minimisant l'espérance du coût $E[L(x,f(y))|Y = y]$. L'estimateur MAP correspond à la fonction de coût

$$L(x,x') = \begin{cases} 1 \text{ si } x \neq x' \\ 0 \text{ sinon} \end{cases}$$

mais d'autres estimateurs, comme le *maximum posterior mean* (MPM) ou le *thresholded posterior mean* (TPM) peuvent être envisagés [72]. Nous utiliserons notamment l'estimateur MPM qui correspond à la fonction de coût

$$L(x,x') = \sum_{s \in \Omega} \delta(x_s = x'_s) \ .$$

Cependant, pour des raisons qui seront évoquées plus tard, nous utiliserons principalement le critère MAP par la suite et nous illustrerons donc uniquement les algorithmes associés à cet estimateur.

Algorithme ICM

L'algorithme ICM est une version déterministe de l'échantillonneur de Gibbs présenté précédemment. De manière similaire, on construit une suite de champs $x^{(n)}$, le passage de $x^{(n-1)}$ à $x^{(n)}$ se faisant en choisissant un site et en modifiant sa valeur de manière à maximiser la probabilité conditionnelle locale. L'algorithme se résume ainsi :

- choix d'un site s à mettre à jour
- calcul de $P[X_s = i|x_{V_s}^{(n-1)}]\ \forall i \in E$ donnée par l'équation (2.3).
- $X_s = \arg\max_i P[X_s = i|x_{V_s}^{(n-1)}]$

Il est facile de démontrer que l'énergie globale diminue à chaque itération et donc que l'algorithme ICM converge vers un minimum local de l'énergie, c'est à dire vers un maximum local de la probabilité *a posteriori*. Malgré cet inconvénient, cet algorithme présente l'avantage de converger très rapidement et de ne nécessiter que peu de calculs, d'où son utilisation lorsque l'on dispose d'une bonne stratégie d'initialisation de la solution $x^{(0)}$.

Recuit simulé

Le recuit simulé est en fait un algorithme d'échantillonnage d'une distribution de Gibbs. La différence avec les échantillonneurs que nous avons vu précédemment vient de l'introduction d'un terme de température dans la fonction énergie. En effet, pour un champ d'énergie $U(x)$ on considère la distribution de Gibbs définie pour $T > 0$ par

$$P_T[X = x] = \frac{1}{Z(T)} \exp\left(-\frac{U(x)}{T}\right) \quad , \tag{2.6}$$

ce qui correspond à la distribution de Gibbs associée à la fonction énergie $U(x)/T$. On peut voir que lorsque T tend vers l'infini, la loi $P_T[X = x]$ tend vers la probabilité uniforme sur Ω puisque $U(x)/T$ tend vers zéro. A l'inverse, lorsque T tend vers zéro, on peut montrer que la loi de X est la loi uniforme sur toutes les configurations d'énergie totale minimale. De manière plus formelle, si on note U^* une telle énergie et Ω^* l'ensemble des configurations d'énergie U^*, on a alors

$$P_0[X = x] = \begin{cases} 0 \ \text{ si } x \notin \Omega^* \\ \frac{1}{|\Omega^*|} \ \text{ si } x \in \Omega^* \end{cases} \quad .$$

En utilisant une dynamique d'échantillonnage et en faisant descendre progressivement la température, on obtient donc une série de réalisations qui converge vers la configuration, ou l'une des configurations, d'énergie minimale. L'algorithme construit itérativement une suite de réalisations $x^{(n)}$ à la température $T^{(n)}$ à partir d'une configuration initiale $x^{(0)}$ et d'une température de départ $T^{(0)}$ "élevée". Le passage de $x^{(n-1)}$ à $x^{(n)}$ se fait par les étapes suivantes

- choix de la température $T^{(n)} < T^{(n-1)}$
- simulation de $x^{(n)}$ par un échantillonneur de Gibbs ou de Metropolis en prenant $x^{(n-1)}$ comme solution initiale et la distribution de Gibbs définie par $U(x)/T^{(n)}$.

On peut montrer que si $T^{(n)} > c/\ln(1+n)$, où c est une constante grande, alors l'algorithme converge vers un minimum global de $U(x)$ [38]. Cette démonstration s'appuie, comme dans le cas des algorithmes de simulation, sur la construction d'une chaîne de Markov, à la différence que la chaîne est maintenant inhomogène du fait du changement de température entre deux itérations. En pratique cependant, on utilise une loi de descente de température géométrique plutôt que logarithmique, cette dernière étant de décroissance trop longue.

2.3 Estimation des paramètres en segmentation

2.3.1 Problématique

On s'intéresse dans cette partie au problème de l'estimation des paramètres d'une distribution de Gibbs dans le cas d'un champ caché, à l'aide du critère de maximum de vraisemblance (MV). Notons qu'il existe d'autres critères, qui seraient mieux adaptés dans certains cas que le maximum de vraisemblance. Cependant, pour des raisons historiques, nous focaliserons nos travaux sur le critère MV. Le problème d'estimation des paramètres au sens du maximum de vraisemblance reste encore un problème non complètement résolu dans la littérature sur les champs Markoviens du fait de la forme de cette distribution. Un certain nombre d'approches ont été proposées et peuvent se ranger dans deux catégories : les approches basées sur l'algorithme EM et les approches de type gradient stochastique. Nous présentons d'abord la problématique avant de passer en revue les approches basées sur le gradient stochastique puis les approches EM. Enfin, nous présenterons des variantes de ces algorithmes, parfois basées sur un critère différent du maximum de vraisemblance.

Dans un souci de simplicité, nous considérerons l'estimation des paramètres d'un champ caché X à partir d'une seule observation y [3]. Nous noterons θ l'ensemble des paramètres de la distribution *a priori* de X et λ les paramètres d'attache aux données. Par exemple, dans le cas d'une chaîne de Markov cachée, θ correspond à la matrice de transition et aux probabilités initiales tandis que λ correspond aux moyennes et variances des densités gaussiennes associées aux états. En considérant les deux types de paramètres comme indépendant, il est possible d'envisager différentes stratégies d'estimation pour chacun d'eux. Notons que la plupart des algorithmes proposés dans la littérature s'intéressent uniquement à l'estimation des paramètres de la distribution de Gibbs en considérant les paramètres d'attache aux données connus. Nous nous proposons de voir en détail l'estimation simultanée des deux paramètres. La vraisemblance de l'observation par rapport aux paramètres de la distribution *a priori* est donnée par

$$\begin{aligned} L(\theta) &= P[Y = y, \Lambda = \lambda | \Theta = \theta] \\ &= \sum_{x \in \Omega} P[Y = y | X = x, \Theta = \theta] P[X = x | \Theta = \theta] \ , \end{aligned} \tag{2.7}$$

les paramètres λ étant supposés connus. Comme dans le cas des HMM, on ne peut pas maximiser directement la vraisemblance en dérivant par rapport à θ à cause de la somme sur toutes les configurations possibles. On a donc recours à des algorithmes itératifs.

3. Ce cadre est typique en traitement d'image dans les problèmes de segmentation et de restauration pour lesquels on cherche l'image originale à partir de sa version dégradée. L'extension à des observations multiples du même modèle ne pose pas de difficultés majeures.

2.3.2 Le gradient stochastique

L'algorithme de gradient stochastique a d'abord été proposé par Younes dans le cas de données complètes [108] et généralisé ensuite au cas des données incomplètes [109]. On suppose dans un premier temps les paramètres d'attache aux données λ connus pour s'intéresser uniquement à l'estimation des paramètres θ de la loi *a priori*. On étudiera ensuite le cas de l'estimation de λ.

Estimation des paramètres de la loi *a priori*

En notant $U_\theta(x)$ le potentiel associé à la distribution de Gibbs *a priori* et $U_\lambda(y|x)$ l'énergie d'attache aux données, la vraisemblance est donnée par

$$L(\theta) = \frac{1}{Z_\theta} \sum_{x \in \Omega} \exp\left(-U_\theta(x) - U_\lambda(y|x)\right) \quad . \tag{2.8}$$

En remarquant que la somme sur $x \in \Omega$ n'est autre que la fonction de partition $Z_{\theta,\lambda}$ de la distribution *a posteriori* de X [98], la vraisemblance est alors donnée par le rapport des deux fonctions de partition

$$L(\theta) = \frac{Z_{\theta,\lambda}}{Z_\theta} \quad . \tag{2.9}$$

En dérivant la log-vraisemblance par rapport aux paramètres θ, on obtient

$$\frac{d \ln L(\theta)}{d\theta} = E[U'_\theta(x)] - E[U'_\theta(x)|Y=y] \quad , \tag{2.10}$$

où $U'_\theta(x)$ est la dérivé de la fonction énergie $U_\theta(x)$ par rapport à θ et $E[\,.\,]$ (resp. $E[\,.\,|Y=y]$) l'espérance mathématique sous la loi *a priori* (resp. *a posteriori*). Il est important de noter que les opérateurs espérances de l'équation (2.10) dépendent des paramètres θ bien que ceci n'apparaisse pas dans la notation par souci de clarté. L'estimateur au sens du maximum de vraisemblance, $\hat{\theta}$ doit donc vérifier l'équation stochastique

$$h(\hat{\theta}) = E[U'_{\hat{\theta}}(x)] - E[U'_{\hat{\theta}}(x)|Y=y] = 0 \quad . \tag{2.11}$$

L'équation $h(\theta) = 0$ n'étant pas analytiquement calculable, l'idée principale est d'utiliser un algorithme de gradient pour la résoudre. En appliquant directement un schéma de Newton-Raphson, on construit la suite d'estimateurs $\theta^{(n)}$ donnée par

$$\theta^{(n+1)} = \theta^{(n)} - J^{-1}(\theta^{(n)})(E[U'_{\theta^{(n)}}(x)] - E[U'_{\theta^{(n)}}(x)|Y=y]) \quad , \tag{2.12}$$

où J est la matrice de Jacobi donnée par $\left(\frac{dh(\theta)}{d\theta}\right)$. En dérivant une seconde fois l'équation (2.8) par rapport à θ, on obtient pour la matrice de Jacobi dans le cas général

$$\frac{d^2 l(\theta)}{d\theta^2} = \left(E[U''_\theta(x)] - E[U''_\theta(x)|Y=y]\right) - \left(var(U'_\theta(x)) - var(U'_\theta(x)|Y=y)\right) \quad , \tag{2.13}$$

où l'opérateur *var*() désigne la variance[4], $U''_\theta(x)$ étant la dérivée seconde de la fonction énergie par rapport à θ.

Il est intéressant de noter que dans le cas où la fonction énergie $U_\theta(x)$ est linéaire par rapport aux paramètres θ, soit $U_\theta(x) = \theta\varphi(x)$, la dérivée première $U'_\theta(x)$ vaut alors $\varphi(x)$ et que la dérivée seconde est nulle. Le détail des dérivations est donné en annexe A, où nous illustrons également le cas de potentiels linéaires par rapport aux paramètres.

La résolution directe de l'équation (2.12) reste bien évidemment incalculable et il est nécessaire de recourir à des approximations. Une première approche consiste à approximer les espérances par des estimations empiriques calculées à partir de tirages aléatoires suivant les lois *a priori* et *a posteriori* [68]. Pour évaluer avec précisions les espérances, il faut réaliser un grand nombre de tirages aléatoires ce qui peut alors poser des problèmes de temps de calcul. On aura donc de préférence recours à des méthodes de type Monte-Carlo [100] pour la simulation, les moyennes étant alors estimées avant convergence des échantillonneurs.

Dans le cadre du gradient stochastique, les moyennes sont remplacées par la valeur obtenue avec un seul échantillon. Partant d'une réalisation selon la loi *a priori* $x^{(0)}_{\text{prior}}$, d'une autre selon la loi *a posteriori* $x^{(0)}_{\text{post}}$ et d'une valeur initial des paramètres $\theta^{(0)}$, l'algorithme proposé dans [109] suit la dynamique suivante pour le passage de $\theta^{(n)}$ à $\theta^{(n+1)}$:

- simuler $x^{(n+1)}_{\text{prior}}$ selon la loi définie par $U_{\theta^{(n)}}(x)$ à partir de $x^{(n)}_{\text{prior}}$
- simuler $x^{(n+1)}_{\text{post}}$ suivant la loi définie par $U_{\theta^{(n)}}(x) + U_\lambda(y|x)$ à partir de $x^{(n)}_{\text{post}}$
- calculer $\theta^{(n+1)}$ selon

$$\theta^{(n+1)} = \theta^{(n)} + \frac{1}{(n+1)\,V}\ \left(U'_{\theta^{(n)}}(x^{(n+1)}_{\text{prior}}) - U'_{\theta^{(n)}}(x^{(n+1)}_{\text{post}})\right)$$

Le pas du gradient $\frac{1}{(n+1)V}$ remplace ici le Hessien et assure la convergence presque sûre de l'algorithme. Il est dû à la substitution des espérances par une seule réalisation.

Estimation des paramètres d'attache aux données

On s'intéresse maintenant à l'estimation simultanée des paramètres d'attache aux données. La vraisemblance par rapport à λ est également donnée par l'équation (2.9). Par un calcul similaire au précédent, on en déduit aisément que la dérivée de $l(\lambda) = \ln(L(\lambda))$ par rapport à λ donne

$$\frac{dl(\lambda)}{d\lambda} = -E[U'_\lambda(y|x)|Y = y] \tag{2.14}$$

4. ou la matrice de covariance lorsque θ est un vecteur)

où, à nouveau, l'espérance est donnée suivant la loi *a posteriori* et où $U'_\lambda(y|x)$ désigne la dérivée de la fonction énergie par rapport aux paramètres λ. Suivant la forme de l'attache aux données, on obtient différents types d'estimateurs basés sur des espérances *a posteeriori*. Ces espérances ne sont bien évidemment pas calculables mais on peut, comme pour l'estimation des paramètres de l'énergie *a priori*, les estimer à partir de tirages aléatoires du champ suivant la loi *a posteriori*.

Pour illustrer la méthode générale que nous venons d'exposer, nous nous intéressons au cas où l'attache aux données est modélisée par une gaussienne de dimension 1 dont les paramètres dépendent de la valeur du champ caché. Dans ce cas, on a $\lambda = \{\mu_i\ ,\ \sigma_i\ \ \forall i \in E\}$ et l'énergie d'attache aux données (2.4) est donnée par

$$U_\lambda(y|x) = \sum_{s\in S} \frac{1}{2}\left(\frac{y_s - \mu_{x_s}}{\sigma_{x_s}}\right)^2 + \ln(\sqrt{2\pi}\sigma_{x_s})\ ,$$

que l'on peut également réécrire sous la forme suivante

$$U_\lambda(y|x) = \sum_{i\in E}\left[\sum_{s\in S}\left(\frac{1}{2}\left(\frac{y_s-\mu_i}{\sigma_i}\right)^2 + \ln(\sqrt{2\pi}\sigma_i)\right)\delta(x_s = i)\right]\ , \qquad (2.15)$$

afin de faire apparaître de manière directe les paramètres μ_i et σ_i. En dérivant cette dernière équation par rapport à μ_i et σ_i et en résolvant les équations de maximisation ainsi obtenues, on obtient alors

$$\mu_i = \frac{\sum_{s\in S} y_s P[X_s = i|Y = y]}{\sum_{s\in S} P[X_s = i|Y = y]} \qquad (2.16)$$

$$\sigma_i = \frac{\sum_{s\in S}(y_s - \mu_i)^2 P[X_s = i|Y = y]}{\sum_{s\in S} P[X_s = i|Y = y]}\ . \qquad (2.17)$$

On notera la similarité entre ces deux formules et les équations (1.19) et (1.20) obtenues pour les HMM. Comme nous le verrons par la suite, l'approche EM donne également les mêmes formules de remise à jour des paramètres d'attache aux données. Notons que ces équations ont été obtenues en supposant les paramètres θ de la loi *a priori* connus ce qui n'est pas la cas en pratique. Dans ce cas, on réestime ces paramètres à chaque itération du gradient stochastique en estimant les probabilités $P[X_s = i|Y = y]$ à partir du ou des échantillons dont on dispose pour la loi *a posteriori* .

Il convient de faire attention lors de l'estimation à contraindre les variances à ne pas descendre en dessous d'un certain seuil. En effet, dans le cas Gaussien, la vraisemblance n'est pas bornée et une variance nulle donnerait une vraisemblance infinie mais un très mauvais estimateur des paramètres. En d'autres termes, le seuillage de la variance permet d'éviter les problèmes de sur-apprentissage en imposant des restrictions sur l'espace des paramètres.

2.3.3 Approche EM

Dans cette partie, nous étudions l'approche EM [75, 25, 100] pour obtenir une estimation des paramètres au sens du maximum de vraisemblance. Nous présentons d'abord la formulation exacte de l'algorithme EM dans le cadre des distributions de Gibbs. Malheureusement, à l'exception de quelques cas particuliers (par exemple [9, 86]), l'algorithme EM ne donne pas de solutions analytiques pour l'estimation des paramètres. Lorsque c'est cependant le cas, les espérances mise en jeu par l'EM ne sont pas explicitement calculables, à l'inverse des HMM pour lesquelles on dispose de l'algorithme *forward-backward*, et on a alors recours à une version stochastique de l'EM [18]. Après la présentation générale du principe EM dans le cas des champs de Markov, nous discutons les solutions et les approximations rapportées dans la littérature.

L'approche exacte ...

Comme nous l'avons vu au chapitre 1.2.3 dans le cas des chaînes de Markov cachés, l'algorithme EM repose sur le calcul de la fonction auxiliaire (1.12). Dans le cas d'un champ caché, cette fonction s'écrit

$$\begin{aligned} Q(\theta,\lambda;\theta^{(n)},\lambda^{(n)}) &= -E[U_\theta(x)|Y=y,\theta^{(n)},\lambda^{(n)}] - \\ &\quad E[U_\lambda(y|x)|Y=y,\theta^{(n)},\lambda^{(n)}] - \ln Z_\theta \end{aligned} \tag{2.18}$$

en notant $E[\,.\,|Y=y,\theta^{(n)},\lambda^{(n)}]$ l'espérance selon la loi *a posteriori* pour les valeurs courantes $\theta^{(n)}$ et , $\lambda^{(n)}$ des paramètres. La maximisation par rapport aux paramètres θ de la loi *a priori* donne

$$E[U'_\theta(x)|\theta] - E[U'_\theta(x)|Y=y,\theta^{(n)},\lambda^{(n)}] = 0 \tag{2.19}$$

et on obtient pour les paramètres d'attache aux données λ

$$E[U'_\lambda(y|x)||Y=y,\theta^{(n)},\lambda^{(n)}] = 0 \ . \tag{2.20}$$

Le détail de ce calculs est donné en annexe A.

Pour les paramètres de la loi *a priori*, l'étape de maximisation de l'algorithme EM consiste donc à résoudre (2.19) ce qui n'est pas possible dans la plupart des cas. On peut approximer l'espérance *a posteriori* par des méthodes de Monte-Carlo mais il reste cependant difficile de résoudre cette équation sans l'aide d'un algorithme de gradient stochastique. Un algorithme allant de ce sens est proposé dans [110] mais n'est pas mis en œuvre. Notons aussi les travaux de Lange [66] où l'étape M de l'algorithme est remplacée par un pas de gradient.

Pour les paramètres λ dans le cas gaussien, on retrouve naturellement les mêmes équations que pour les chaînes de Markov cachés (1.19)-(1.20), le calcul des probabilités *a posteriori* devant également être réalisé par des méthodes de Monte-Carlo.

... et ses approximations

L'approche exacte de l'algorithme EM met clairement en évidence les difficultés rencontrées avec cette méthode. La solution la plus courante pour contourner ces problèmes, proposée par Chalmond [21] sous le nom d'EM Gibbsien, consiste à remplacer la vraisemblance des données par la pseudo-vraisemblance, qui présente l'avantage d'être explicitement calculable, dans l'expression de la fonction auxiliaire. La pseudo-vraisemblance est définie comme le produit sur l'ensemble des sites des probabilités conditionnelles locales, soit

$$l_p(\theta) = \sum_{s \in S} \ln P[X_s = x_s | X_{S \backslash s} = x_{S \backslash s}] \ , \tag{2.21}$$

la probabilité conditionnelle locale étant donnée par (2.3). La maximisation de la nouvelle fonction auxiliaire par rapport à θ nous donne l'équation suivante

$$E[\sum_{s \in S} \left(E[U'_\theta(x_s|V_s)|x_{S\backslash s}] - U'_\theta(x_s|V_s) \right) |Y = y, \theta^{(n)}] = 0 \ . \tag{2.22}$$

La résolution de cette équation n'est pas entièrement triviale car l'expression de $E[U'_\theta(x_s|V_s)|x_{S\backslash s}]$ par rapport aux paramètres peut-être compliquée mais on peut obtenir une formulation plus simple lorsque le cardinal de l'ensemble des configurations possibles Ω est relativement petit. Nous renvoyons le lecteur à l'article d'origine [21] ou encore à [110] pour une formulation complète de l'algorithme.

Comme mentionné précédemment, le calcul exact des espérances *a posteriori* dans l'approche EM standard n'est pas possible sauf dans quelques rares cas particuliers. On utilise donc des techniques de Monte-Carlo, ce qui nous donne des algorithmes de type SEM [18] lorsque la maximisation de la fonction auxiliaire fournit une forme analytique, ou des algorithmes de gradient stochastique sur la fonction auxiliaire dans le cas contraire. Dans [114], Zhang propose une autre méthode basée sur la théorie du champ moyen [22] pour approximer ces espérances [113].

2.3.4 Autres techniques

De nombreuses techniques basées sur des critères d'estimations différents du maximum de vraisemblance ont également été proposées. On peut par exemple mentionner l'estimation conditionnelle itérative de Pieczynski [86] où l'on suppose que l'on est capable de trouver la valeur des paramètres à partir des données complètes (*i.e.* on dispose d'une fonction du type $\hat{\theta} = \hat{\theta}(X, Y)$) pour construire une suite d'estimateurs. Le principe est d'approximer l'estimateur $\hat{\theta}$ par rapport à l'espace d'observation Y au sens de l'erreur quadratique minimum. Partant d'une valeur $\theta^{(0)}$, on construit l'estimateur $\theta^{(n+1)} = E_{\theta^{(n)}}[\hat{\theta}|Y = y]$. Lorsque l'espérance par rapport à $\theta^{(n)}$ n'est pas explicitement calculable, l'auteur propose l'utilisation d'une approximation stochastique. Bien que le principe de cette méthode soit fondamentalement différent des principes EM et SEM, on aboutit dans le cas du mélange de gaussiennes à des formules relativement similaires [5].

Dans les problèmes d'estimation et de segmentation simultanées, on construit en général de manière itérative une solution x^* en parallèle de l'estimation des paramètres. En complétant l'observation y par l'estimation courante de la solution x^*, on se ramène dans le cas des données complètes pour lequel on dispose d'algorithmes efficaces comme celui du gradient stochastique [108]. Par exemple, Lakshmanan et Derin [65] proposent de maximiser la loi jointe de l'observation et de la solution courante, soit

$$\hat{\theta} = \arg\max_{\theta} P[X = x^*, Y = y | \Theta = \theta, \Lambda = \lambda]$$

ce qui donne un estimateur vérifiant la relation $E_{\hat{\theta}}[U'(x)] = U'(x^*)$. Pour les paramètres de la loi *a priori* , un tel estimateur est en fait l'estimateur du maximum de vraisemblance pour la distribution *a priori* avec la donnée complète x^*. De ce schéma découle une procédure itérative alternant construction de la solution x^* et réestimation des paramètres du modèle [112, 62]. On trouve également comme critère la probabilité du résultat conditionnellement à l'observation, soit

$$\hat{\theta} = \arg\max_{\theta} P[X = x^* | Y = y, \Theta = \theta, \Lambda = \lambda] \ ,$$

ce qui donne, pour les paramètres θ, l'estimateur au sens du maximum de vraisemblance pour la loi *a posteriori* et pour la donnée complète x^*.

Nous clôturons ici notre petit tour d'horizon des méthodes d'estimation des paramètres pour les champs de Markov cachés. Il existe bien évidemment d'autres méthodes d'estimation, comme la technique basée sur les méthodes Monte-Carlo Markov Chain Maximum Likelihood de Descombes [29] ou encore l'analyse en cumulants de Sigelle [98], mais les méthodes exposées jusqu'ici montrent que le problème de l'estimation des paramètres n'est pas simple. En effet, on voit que beaucoup de méthodes sont développées dans des cas particuliers ou en s'appuyant sur des hypothèses restrictives et toutes font appel à de nombreuses approximations stochastiques. Nous nous interesserons par la suite aux approches de type gradient stochastique, couplées à l'algorithme EM, qui présentent l'avantage d'être développées dans un cadre assez général.

2.4 Distribution de Gibbs et chaîne de Markov

Pour conclure ce chapitre sur les champs de Markov, nous établissons certains liens entre les chaînes de Markov et les distributions de Gibbs. Entre autre, nous montrons que la mesure associée à une chaîne de Markov peut-être définie comme une distribution de Gibbs et que, à l'inverse, certaines distributions de Gibbs sont équivalentes à des chaînes de Markov.

5. Notons quand même que les formules de réestimation des paramètres ne sont pas les mêmes pour tout ces algorithmes même si elle partage un grand nombre de traits communs.

2.4.1 Chaîne de Markov comme distribution de Gibbs

Dans ce sens, il est aisé de montrer qu'une chaîne de Markov homogène peut se récrire sous la forme d'une distribution de Gibbs. En effet, considérons une chaîne à N états sur $S = \mathbb{Z}$, de matrice de transition $P = (P(i,j))_{i,j=1,\ldots,N}$. Pour un ensemble $A = [t_1, t_2]$ ($A \in S$), la probabilité de X_A conditionnelle à $X_{S\setminus A}$ est donnée par

$$P[X_A = x_A | x_{S\setminus A}] = \prod_{t=t_1}^{t_2} P(x_{t-1}, x_t) \ , \tag{2.23}$$

ce que l'on peut aisément récrire comme

$$P[X_A = x_A | x_{S\setminus A}] = \exp\left(- \sum_{t=t_1}^{t_2} - \ln P(x_{t-1}, x_t) \right) \ . \tag{2.24}$$

Cette dernière formulation correspond à une distribution de Gibbs (2.3) pour le voisinage en 2-connexité sur $S = \mathbb{Z}$ si la matrice de transition P est strictement positive (*i.e.* $P(i,j) > 0 \ \ \forall i,j \in [1,N]$). En effet, la fonction de partition vaut alors

$$\begin{aligned} Z &= \sum_{x \in E^{|A|}} \exp\left(- \sum_{t=t_1}^{t_2} - \ln P(x_{t-1}, x_t) \right) \\ &= \sum_{x \in E^{|A|}} P[X_A = x | x_{S\setminus A}] \ , \end{aligned}$$

soit 1 dans ce cas là. Les potentiels U_c associés aux cliques du système de voisinage $V_t = \{t-1, t+1\}$ sont alors donnés par

$$U_c(x) = \begin{cases} 0 & \text{si } c = \{s\} \\ -\ln P(x_{s_1}, x_{s_2}) & \text{si } c = \{s_1, s_2\} \end{cases} \ . \tag{2.25}$$

Lorsque la matrice de transition P n'est pas strictement positive, le théorème de Hammersley-Clifford ne s'applique plus puisqu'il existe des configurations x telles que $P[X = x] = 0$. Nous discuterons ce cas particulier dans la partie 3.2.2.

On vient donc de montrer que la mesure associée à une chaîne de Markov homogène de matrice de transition positive est une distribution de Gibbs dont les énergies de cliques sont données par (2.25). Cependant, il est évident que plusieurs spécifications de Gibbs peuvent correspondre à la même chaîne de Markov. En effet, si l'on ajoute une constante C aux potentiels U_c donnés par l'équation (2.25), on définit une nouvelle spécification Gibbsienne correspondant également à une chaîne de matrice de transition P, la contribution de la constante dans le calcul de (2.24) s'éliminant du numérateur avec la fonction de partition qui vaut alors $\exp -C$ au lieu de 1. Nous allons maintenant établir l'équivalence entre une spécification de Markov homogène positive et une chaîne de Markov en rappelant le résultat établi dans [40], chapitre 3.

2.4.2 Spécification de Markov homogène positive

Georgii, dans [40], établit une correspondance entre l'ensemble des spécifications de Markov homogènes positives et l'ensemble des matrices stochastiques positives sur E. Soit g une spécification [6] définie sur $S = \mathbb{Z}$ et à valeurs dans E. La notion de spécification de Markov homogène positive est alors donnée par la définition suivante:

Définition 1 *La spécification g est une spécification de Markov homogène positive si il existe une fonction $u(.,.,.) > 0$ de $E^3 \mapsto \mathbb{R}$ telle que*

$$g_{\{t\}}(X_t = i | X_{S\backslash t} = x_{S\backslash t}) = u(x_{t-1}, i, x_{t+1})$$

pour tout $t \in S$ et $i \in E$.

Soit μ_P la distribution associée à la chaîne de Markov stationnaire de matrice de transition P. Le théorème 2 montre que chaque spécification de Markov homogène positive admet une unique mesure de Gibbs qui correspond à une chaîne de Markov ergodique dont la matrice de transition peut-être exprimée en fonction de g.

Théorème 2 *Il existe une correspondance $g \longleftrightarrow P$ entre l'ensemble de toutes les spécifications de Markov homogènes positives et l'ensemble des matrices stochastiques positives sur E. Pour P donné, la spécification correspondante est donnée par*

$$g_A(X_A = \xi | X_{S\backslash A} = x_{S\backslash A}) = \mu_P(X_A = \xi | X_{\partial A} = x_{\partial A}) \qquad \forall \xi \in E^{|A|} \ . \tag{2.26}$$

Inversement, la matrice P est donnée en fonction de u, la fonction déterminante de g, par

$$P(i,j) = \frac{Q(i,j)r(j)}{qr(i)} \qquad (i,j \in E) \ , \tag{2.27}$$

avec $Q(i,j) = g(a,i,j)/g(a,a,j)$ pour $a \in E$ arbitrairement fixé, q étant la plus grande valeur propre de $Q = (Q(i,j))_{i,j\in E}$ et r le vecteur propre associé.

Notons que comme, par construction, la matrice Q est positive, le théorème de Perron-Frobenius assure que q est strictement positif, que toutes les autres valeurs propres sont inférieures en module à q, et qu'il existe un vecteur propre associé $r \in]0,\infty[^{|E|}$.

Enfin, un corollaire à ce théorème permet de définir plus clairement ce qu'est une spécification de Markov homogène positive. Le potentiel pour le système

6. On appelle *spécification* l'ensemble des distributions conditionnelles Gibbsienne définissant une mesure de Gibbs.

de voisinage en 2-connexité sur $\mathbb{Z}$ est dit homogène si il existe deux fonctions $f_1 : E \longrightarrow \mathbb{R}$ et $f_2 : E \times E \longrightarrow \mathbb{R}$ telles que

$$U_c = \begin{cases} f_1(x_t) & \text{si } c = \{t\} \\ f_2(x_{t-1}, x_t) & \text{si } c = \{t-1, t\} \end{cases} .$$

Corollaire 1 *Une spécification g est une spécification de Markov homogène positive si et seulement si g est Gibbsien pour un potentiel homogène en 2-connexité.*

En résumé, le théorème 2 permet en particulier de définir, grâce à l'équation (2.27), la matrice de transition de la chaîne de Markov correspondant à une spécification de Markov homogène positive lorsque la fonction déterminante est connue.

2.5 Bilan

Dans ce chapitre, nous avons présenté le formalisme des champs Markoviens et des distributions de Gibbs ainsi que les algorithmes de simulation, d'optimisation et d'estimation des paramètres associés. En particulier, on a mis en évidence l'équivalence, sous certaines conditions, entre une chaîne de Markov et une distribution de Gibbs. Ce dernier résultat sera largement utilisé par la suite. Nous allons maintenant utiliser le formalisme et les outils introduits dans ce chapitre pour définir un modèle segmental de la parole ainsi que les algorithmes associés.

Chapitre 3

Définition d'un modèle de champ Markovien pour la parole

3.1 Introduction historique

Bien que les champs de Markov soient couramment utilisés en traitement d'images, il est rare de voir ce modèle utilisé dans d'autres domaines de recherche. Dans le domaine du traitement automatique de la parole, les quelques études présentées dans la littérature s'appuient principalement sur la formulation d'une chaîne de Markov comme une distribution de Gibbs.

Chronologiquement, la première étude est due à Zhao *et al.* et porte sur la reconnaissance de mots isolés [115]. Les auteurs récrivent une chaîne de Markov caché sous la forme d'une distribution de Gibbs et définissent un algorithme EM d'estimation des paramètres. Une procédure itérative de calcul de la fonction de partition pour une distribution de Gibbs sur un voisinage en 2-connexité permet de ne pas faire d'approximation dans l'étape E de l'algorithme EM d'une part, et de calculer la probabilité *a posteriori* de manière exacte d'autre part. Comme nous l'avons vu au chapitre précédent, l'étape de maximisation n'admet pas de forme analytique directe et est remplacée par un algorithme de gradient appliqué à la fonction auxiliaire. Les résultats expérimentaux en reconnaissance de mots isolés sur la base TI Digit montrent un très léger avantage de la formulation en terme de distribution de Gibbs par rapport à la formulation HMM.

En 1994, Noda et Shirazi s'appuient sur la formulation en terme de distribution de Gibbs d'une chaîne de Markov pour la réalisation d'un décodeur parallèle basé sur l'algorithme ICM plutôt que sur l'algorithme de Viterbi [81], l'algorithme ICM étant alors utilisé pour l'estimation de la séquence d'état caché optimale. Ils étendent également la procédure au cas des HMM prédictifs [80][1] pour lesquels l'hypothèse d'indépendance conditionnelle des données n'est plus valable, ce qui rend alors l'utilisation directe de l'algorithme ICM impossible. Aussi bien pour les HMM classiques que pour les HMM prédicitifs, les auteurs rapportent des taux d'erreur comparables entre l'algorithme de Viterbi et l'ICM, sans toutefois préciser comment l'ICM est initialisé.

Le premier modèle vraiment 2D est proposé par Lucke en 1995 [69]. Il propose un modèle basé sur une distribution de Gibbs qui permet de prendre en compte de manière explicite les interactions entre les coefficients cepstraux en utilisant un voisinage en 4-connexité. Pour cela, deux fonctions potentiels différentes sont utilisées pour les cliques horizontales et verticales. L'espace d'état étant discret, ces fonctions sont en fait deux matrices et aucun *a priori* sur le type d'interactions modélisées n'est utilisé. L'estimation de ces paramètres se fait en utilisant une procédure relativement complexe de "*coarse graining*". Les résultats présentés montrent de bonnes performances sur un tâche de discrimination entre deux phonèmes mais la taille du corpus de test (5 occurrence de chaque phone) ne permet pas de conclure de manière significative.

Enfin, Huo et Chan proposent l'utilisation de distribution de Gibbs pour modéliser des dépendances bi-directionnelles [52, 53]. En effet, on peut considérer qu'un HMM ne modélise des dépendances que dans le sens des temps croissants (*i.e.* en ne considérant que $p(x_t|x_{t-1})$). Les auteurs mettent en évidence l'intérêt d'une modélisation bi-directionnelle en réalisant des expériences de reconnaissance de lettres isolés à partir de HMM dont les paramètres sont estimés soit directement à partir des occurrences d'apprentissage, soit à partir de ces mêmes occurrences prises à l'envers. Ils montrent qu'en prenant les séquences à l'envers, les mêmes taux de reconnaissance peuvent être obtenus que dans le cas standard, la combinaison linéaire des deux reconnaisseurs permettant d'améliorer les résultats. Le modèle proposé étend la procédure de quantification vectorielle contextuelle (CVQ) proposée dans [51] en considérant l'ensemble des probabilités $p_{ijk} = P(X_t = i|X_{t-1} = j, X_{t+1} = k)$, X_t étant le code (ou état) auquel l'observation à l'instant t est associé. Ils utilisent alors l'algorithme ICM et des estimateurs empiriques pour l'apprentissage de ces probabilités. Ce modèle donne des performances similaires à celles obtenues avec la CVQ mais supérieures aux performances obtenues avec un HMM discret ou avec une procédure LBG [107] pour la quantification.

Pour conclure cette introduction historique sur l'utilisation des champs de Markov en modélisation de la parole, notons que des techniques basées sur les distributions de Gibbs font également leur apparition dans d'autres domaines connexes. Ainsi, Shahshahani [96] utilise les champs de Markov pour l'adaptation au locuteur de modèles acoustiques, les sites sur lequel le champ est défini étant l'ensemble des paramètres des modèles acoustiques. Citons également les travaux de Della Pietra *et al.* [87], dans le domaine du traitement du langage naturel, où les champs de Markov sont utilisés pour une tâche de classification morphologique de mots.

1. Les résultats présentés dans cette introduction sont en fait issus de ce dernier article, le premier ([81]), disponible en japonais uniquement, étant cité pour des raisons historiques.

3.2 Le modèle RFM-*sync*1

Dans cette partie, nous présentons une première approche de modélisation de segments de parole à l'aide de champ de Markov. Nous définissons un modèle, RFM-*sync*1, et étudions les algorithmes associés dans le cadre de la reconnaissance de mots isolés. Nous discuterons au chapitre suivant la valeur théorique des algorithmes proposés tandis que le chapitre 5 sera consacré à l'application de ce modèle sur des signaux réels de parole. Notons que l'intérêt du modèle présenté ici réside plus dans la validation des techniques champ de Markov appliquées à la modélisation de la parole que dans les performances du modèle. Nous discuterons ultérieurement les défauts de cette première approche et les moyens d'y remédier.

3.2.1 Motivations

Nous avons vu, au chapitre 1.3.1, les modèles de Markov cachés multi-bandes. Dans cette approche, les différents canaux fréquentiels sont traités de manière indépendante ce qui permet de limiter artificiellement la contribution des canaux bruités à l'étape de recombinaison des scores sur chacun des chemins. Cependant, l'hypothèse d'indépendance des canaux fréquentiels est bien évidemment fausse et nous nous proposons d'introduire une certaine forme de dépendance entre les différentes bandes. Une manière naturelle de procéder consiste à modifier le modèle pour le processus caché qui, dans l'approche traditionnelle, est défini par les différentes chaînes de Markov parallèles. En effet, la probabilité d'être dans un état donné dans la bande k à l'instant t dépend uniquement de l'état observé à l'instant précédent dans la même bande pour les HMM parallèles. Pour ajouter une interaction entre les bandes, on peut considérer que cette probabilité dépend en plus des états observés dans les autres bandes au même instant. On définit ainsi implicitement une relation de voisinage, le processus caché pouvant alors être modélisé par une distribution de Gibbs pour prendre en compte de telles interactions. Dans le modèle que nous présentons, les interactions entre les bandes de fréquences sont en fait des mesures de synchronie. On considère que deux bandes sont synchrones si les zones spectralement stables qui, en théorie, devraient correspondre à un état dans une chaîne de Markov cachée sont observées en même temps.

Dans l'approche HMM classique, les différents canaux fréquentiels sont bien évidemment totalement synchrones tandis que dans l'approche multi-bande, les canaux sont totalement asynchrones. Des recherches ont montré qu'il est possible que les canaux fréquentiels soient asynchrones [102, 47] pour différentes raisons (canal de transmission, mécanisme de production, ...). Les approches multi-bandes permettent de prendre en compte cette asynchronie et Tomlinson [102] propose d'utiliser la décomposition par HMM [103, 36] pour autoriser un degré d'asynchronie entre bandes. L'idée de modéliser la synchronie entre les canaux fréquentiels est donc motivée par la volonté de trouver un juste milieu entre les approches synchrones et asynchrones. En effet, si la synchronie est une hypothèse fausse[2], l'asynchronie totale est également une hypothèse fausse, les articulateurs n'étant pas totalement asynchrones! De plus, les phénomènes d'asynchronie qui pourraient être dus au canal de transmission ne sont pas intéressants pour la reconnaissance de la parole et il peut s'avérer utile d'éliminer ou de corriger ces effets.

Nous allons maintenant définir ce modèle de manière paramétrique, sous la forme d'une distribution de Gibbs.

3.2.2 Définition des potentiels

Notons (t, k) le point du treillis correspondant à l'instant t dans la bande k. On désignera dorénavant les sites en fonction de l'indice temporelle t et de l'indice de bande k plutôt que par l'indice générique s comme au chapitre précédent. Comme mentionné dans l'introduction, nous considérons que la variable aléatoire $X_{t,k}$ dépend de la valeur observée dans la même bande à l'instant $t-1$ et des valeurs observées dans les autres bandes au même instant, ce qui correspond au système de voisinage $V_{t,k}$ défini par

$$V_{t,k} = \{(t-1, k), (t+1, k), (t, l) \;\; \forall l \neq k\}$$

et illustré par la figure 3.1.

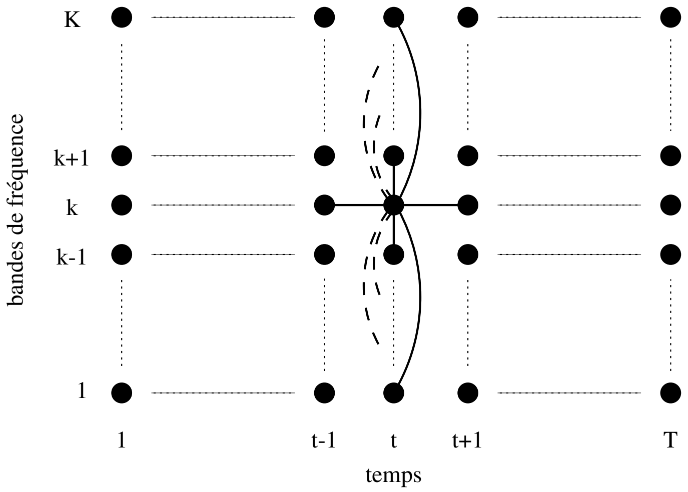


FIG. 3.1 – *Système de voisinage $V_{t,k}$ associé au modèle.*

Deux types de cliques sont associées à ce système de voisinage, les premières, du type $\{(t-1, k), (t, k)\}$, intervenant dans la modélisation temporelle, les secondes, du type $\{(t, k), (t, l)\}$, reflétant l'interaction entre les bandes k et l à l'instant t. Nous appellerons les cliques du premier type horizontales tandis que celles du deuxième type seront appelées verticales. Le système de voisinage étant fixé, il nous reste à définir les fonctions potentiel associées aux cliques.

2. Malgré les bons résultats que l'on peut obtenir ...

Nous utiliserons des potentiels horizontaux correspondant à une chaîne de Markov et des potentiels verticaux permettant de contrôler la synchronie entre deux bandes.

Si l'on considère un modèle à K bandes (*i.e.* $k \in [1, K]$), le modèle peut être vu comme un ensemble de K HMM interagissant[3]. Le lien entre le processus caché et l'observation se fait de manière classique en associant un mélange de gaussiennes à chaque état dans chacune des sous-bandes.

Potentiels horizontaux

Les potentiels horizontaux sont définis de manière à ce que le modèle associé à chaque bande, considérée indépendamment des autres, corresponde à une chaîne de Markov. En effet, une chaîne de Markov est, comme nous l'avons vu précédemment, un cas particulier de champ de Markov et l'on peut exprimer la probabilité d'une réalisation suivant une distribution de Gibbs.

Si $X = \{X_1, \dots, X_T\}, X_t \in [1, N]$ est un processus de Markov, la distribution de Gibbs associée au voisinage en 2-connexité est donnée par

$$P[X = x] = \frac{1}{Z} \exp\left(-\sum_{t=1}^{T} U_1(x_t) - \sum_{t=2}^{T} U_2(x_{t-1}, x_t) \right) \quad . \tag{3.1}$$

Cette dernière équation est l'extension de l'équation (2.24) au cas où la chaîne est définie sur $S = \{1, \dots, T\}$ plutôt que sur $S = \mathbb{Z}$. Les potentiels associés aux singletons permettent de prendre en compte les probabilités initiales de la chaîne tandis que les potentiels des cliques d'ordre deux sont liés aux probabilités de transition. Par exemple, pour une chaîne de distribution initiale $\pi = (\pi_i)_{i \in [1,N]}$ et de matrice de transition $P = (P(i,j))_{i,j \in [1,N]}$, on a

$$U_1(x_t) = -\ln \pi_{x_t} \delta(t = 1)$$

pour les cliques d'ordre 1 et

$$U_2(x_{t-1}, x_t) = -\ln P(x_{t-1}, x_t)$$

pour les cliques d'ordre 2. En utilisant un état initial artificiel non-émetteur, noté 0, et en étendant le processus pour avoir $X_0 = 0$, on peut reformuler l'équation (3.1) en fonction des cliques d'ordre 2 ce qui présente l'avantage d'avoir des potentiels homogènes[4]. En pratique, on utilise aussi un état artificiel non-émetteur final pour contraindre l'espace de réalisation dans le cas de chaînes gauche-droite.

On note que l'énergie associée à une transition de probabilité nulle est infinie, ce qui correspond à une énergie barrière. En effet, le théorème de Hammersley-Clifford (1) ne s'applique normalement que si toutes les configurations sont de

3. Cette dernière formulation est purement illustrative puisque lorsqu'on ajoute des interactions entre les bandes, le modèle intra-bande n'est plus équivalent à une chaîne de Markov cachée. En effet, la distribution ne correspond plus à une spécification de Markov homogène (cf. corollaire 1) et le théorème 2 ne s'applique donc plus.

4. On a alors $P(0, i) = \pi_i, \ \forall i \in [1, N]$.

probabilité non nulle, ce qui n'est pas le cas lorsqu'il existe des transitions de probabilité nulle. Moussouris [78] montre qu'une condition suffisante pour que l'équivalence entre champs de Markov et distributions de Gibbs définie par le théorème reste valable lorsqu'il existe des configurations de probabilité nulle, est que de telles configurations admettent une énergie infinie appelée énergie barrière. Cette condition étant vérifiée dans notre cas, on peut donc définir n'importe quelle chaîne de Markov en terme de distribution de Gibbs. Une autre stratégie, que nous évoquerons plus tard, consiste à travailler sur un espace Ω restreint aux seules configurations de probabilité non nulle.

A partir de cette équivalence entre chaîne de Markov et champ de Markov, nous définissons le potentiel associé à la clique $\{(t-1,k),(t,k)\}$ par

$$U_k^{(h)}(x_{t-1,k}, x_{t,k}) = a^{(k)}_{x_{t-1,k}, x_{t,k}} \quad . \tag{3.2}$$

Les chaînes de Markov considérées étant homogènes, nous notons un tel potentiel $U_h^{(k)}$ puisqu'il ne dépend pas du temps t. D'après ce que nous avons vu, le paramètre $a_{i,j}^{(k)}$ est homogène à l'opposé du logarithme de la probabilité de la transition $i \rightarrow j$ dans la chaîne de Markov associée à la bande k. Comme nous l'avons noté précédemment, nous n'associons pas de potentiel aux cliques d'ordre 1 dans ce modèle.

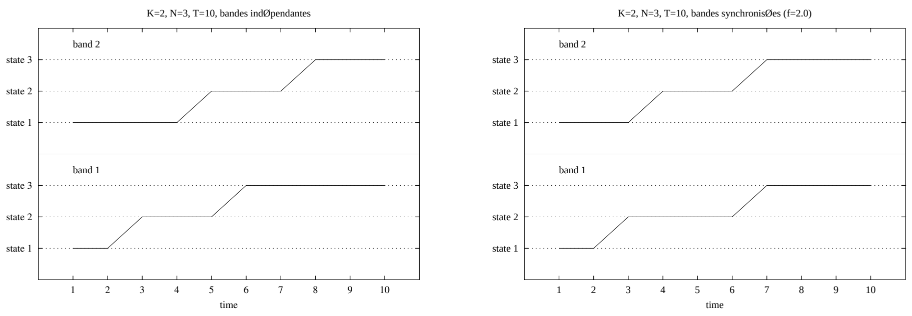
Potentiels verticaux

Le potentiel vertical entre deux bandes k et l a pour but de contrôler la synchronie entre ces deux bandes. Deux bandes seront considérées comme synchrones lorsque les zones spectralement stables apparaissent aux mêmes instants. Du fait de la modélisation par HMM des bandes, une zone spectralement stable correspond à l'occupation d'un état du HMM puisque tous les vecteurs acoustiques de cette zone suivent la même distribution. On peut donc traduire la contrainte de synchronie au niveau du processus caché plutôt qu'au niveau de l'observation en favorisant les processus cachés correspondant à deux bandes synchrones à avoir une réalisation voisine, c'est-à-dire à avoir des changements d'états synchrones. La figure 3.2 illustre le concept de synchronisation au niveau du processus caché, le schéma *(b)* montrant deux bandes dont les réalisations sont très voisines du fait de la synchronisation.

Si les deux chaînes ont le même nombre d'états, on peut traduire la synchronisation en terme d'énergie de clique en associant à la clique $\{(t,k),(t,l)\}$ le potentiel donné par

$$U_{k,l}^{(v)}(x_{t,k}, x_{t,l}) = f_{k,l} \, |x_{t,k} - x_{t,l}| \quad , \tag{3.3}$$

le paramètre $f_{k,l}$ contrôlant la synchronisation des deux bandes. En effet, si $f_{k,l}$ est grand, les configurations pour lesquelles la différence $|x_{t,k} - x_{t,l}|$ est petite seront plus vraisemblable, ce qui correspond bien au caractère synchrones des deux chaînes. On voit donc qu'une valeur élevée pour ce paramètre favorise la synchronisation des bandes. La figure 3.2 montre la validité de ce modèle par



(a) *(b)*

FIG. 3.2 – *Exemples de réalisations pour un modèle à deux bandes sans (a) et avec (b) synchronisation entre les bandes.*

rapport aux objectifs fixés puisque les réalisations représentées sont en fait issues de tirages aléatoires avec $f = 0$ et $f = 2$ respectivement, où $f = 0$ correspond au cas où les deux bandes sont indépendantes. Notons que dans ce modèle la synchronisation est homogène, ce qui signifie qu'elle ne dépend pas du temps, d'où la notation $U^{(v)}_{k,l}(.,.)$. Cette hypothèse peut paraître vraisemblable pour des modèles correspondant à des unités acoustiques courtes où la synchronisation ne va pas changer entre le début et la fin. Par contre, pour des unités plus longues, il est évident qu'une telle formulation est fausse et limitative. Nous reviendrons sur ce point dans la discussion finale.

Après avoir définies les relations de voisinage ainsi que les énergies de cliques pour ce système de voisinage, nous pouvons maintenant formuler les énergies associées aux lois *a priori* et *a posteriori* pour le champ complet.

3.2.3 Energies des lois *a priori* et *a posteriori*

Le modèle RFM-*sync*1 est défini par le nombre de bandes, K, le nombre d'états dans les bandes, N, et l'ensemble de paramètres suivant

$$\theta_{NK} = \{A^{(k)} \;\; k \in [1,K], \;\; F, \; ; b_{i,k}(.) \;\; i \in [1,N] \text{ et } k \in [1,K]\} \;\; ,$$

$A^{(k)}$ étant la matrice $(N+2, N+2)$ de poids de transition associée à la bande k [5] et F la matrice symétrique (K,K) de synchronisation contenant les paramètres $f_{k,l}$. La fonction $b_{i,k}(.)$ désigne la densité de probabilité associée à l'état i de la bande k.

A partir des deux fonctions potentiels (3.2) et (3.3), nous pouvons exprimer la probabilité *a priori* d'une configuration. L'énergie totale d'une configuration x est donnée par

$$U(x) = \sum_{t=1}^{T}\sum_{k=1}^{K} a^{(k)}(x_{t-1,k}, x_{t,k}) + \sum_{t=1}^{T}\sum_{k=1,l>k}^{K} f_{k,l}\, |x_{t,k} - x_{t,l}| \;\; ,$$

ce que l'on peut encore écrire sous la forme paramétrique suivante

$$U(x) = \sum_{k=1}^{K}\sum_{i,j=1}^{N} a^{(k)}_{i,j}\; \varphi^{(k)}_{i,j}(x) + \sum_{k=1,l>k}^{K} f_{k,l}\; \psi_{k,l}(x) \;\; . \tag{3.4}$$

Dans cette dernière formulation de l'énergie *a priori* , qui présente l'avantage de faire apparaître la linéarité de la fonction énergie par rapport aux paramètres, on définit deux fonctions de comptage données par

$$\varphi^{(k)}_{i,j}(x) = \sum_{t=1}^{T} \delta(x_{t-1,k} = i)\, \delta(x_{t,k} = j) \tag{3.5}$$

$$\psi_{k,l}(x) = \sum_{t=1}^{T} |x_{t,k} - x_{t,l}| \;\; . \tag{3.6}$$

5. Rappelons que la dimension $(N+2, N+2)$ de la matrice est due aux états artificiels non-émetteur.

La fonction $\varphi_{i,j}^{(k)}(x)$ compte le nombre de transitions de l'état i vers l'état j dans la bande k pour le champ x et est donc directement reliée au poids de transition $a_{i,j}^{(k)}$. La seconde fonction, $\psi_{k,l}(x)$, qui mesure "l'écart cumulé" entre les bandes l et k pour le champ x, est naturellement utilisée en association avec les poids de synchronisation.

Comme nous l'avons vu au chapitre 2.1.2, la fonction énergie selon la loi *a posteriori* , est donnée par $U(x|Y=y) = U(x)+U(y|x)$ où, dans le cas présent, $U(x)$ est bien évidemment donné par (3.4) tandis que la forme générale pour l'attache aux données $U(y|x)$ est donnée par

$$U(y|x) = \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{i=1}^{N} -\ln(b_{i,k}(y_{t,k}))\delta(x_{t,k}=i) \quad . \tag{3.7}$$

Dans le cas mono-gaussien que nous étudierons par la suite, on retrouve l'équation (2.15).

3.3 Estimation des paramètres

Nous présentons dans cette partie les algorithmes du chapitre 2 appliqués au modèle RFM-*sync*1. En particulier, nous présentons en détail les algorithmes d'estimation des paramètres du modèle. Dans toute cette partie, nous nous placerons par mesure de simplicité dans le cas d'une observation unique et d'une loi d'attache aux données mono-gaussienne. L'extension au cas plus général, pour lequel les lois d'attache aux données sont des mélanges de gaussiennes et pour lequel plusieurs observations sont disponibles, ne présente pas de difficultés majeures et nous donnons en annexe B les équations obtenues dans ce cas ainsi que des détails concernant l'implémentation de l'algorithme proposé.

3.3.1 Estimation des paramètres par l'algorithme EM généralisé

L'approche que nous avons choisie pour l'estimation des paramètres consiste à utiliser l'algorithme EM couplé avec un algorithme de gradient stochastique pour maximiser la fonction auxiliaire comme mentionné au chapitre 2.3.3. Nous décrivons ici en détail la procédure proposée dans [44] en nous placant dans le cas d'une observation unique, la généralisation à des observations multiples étant directe. En remplaçant dans l'équation (2.18) les énergies par leurs expressions données dans la section précédente, on obtient pour la fonction auxiliaire

$$\begin{aligned} Q(\theta,\theta^{(n)}) \quad = \quad & -\sum_{k}\sum_{i,j} a_{i,j}^{(k)} E[\varphi_{i,j}^{(k)}(x)\,|\,Y,\theta^{(n)}] - \\ & \sum_{k,l>k} f_{k,l} E[\psi_{k,l}(x)\,|\,Y,\theta^{(n)}] - \\ & \ln Z_{\Theta} - \\ & \sum_{t,k}\sum_{i} E[\delta(x_{t,k}=i)|Y=y,\theta^{(n)}]\ln b_{i,k}(y_{t,k}) \quad , \end{aligned} \tag{3.8}$$

Z_θ étant la fonction de partition associée à la loi *a priori* . Par analogie avec la notation adoptée pour l'algorithme EM appliqué aux chaînes de Markov cachées, nous noterons $\gamma_{t,k}(i)$ l'espérance *a posteriori* de la fonction de comptage $\delta(x_{t,k} = i)$. L'énergie $U(x)$ étant linéaire par rapport aux paramètres, on obtient, d'après (2.19) la relation suivante pour les paramètres $a_{i,j}^{(k)}$

$$E[\varphi_{i,j}^{(k)}(x)|\theta] - E[\varphi_{i,j}^{(k)}(x)|Y, \theta^{(n)}] = 0 \tag{3.9}$$

et pour $f_{k,l}$

$$E[\psi_{k,l}(x)|\theta] - E[\psi_{k,l}(x)|Y, \theta^{(n)}] = 0 \ . \tag{3.10}$$

Lorsque la densité d'attache aux données est une gaussienne, on obtient pour la remise à jour des moyennes et variances les équations (1.19) et (1.20) établies dans le cas des chaînes de Markov cachées. Nous allons voir comment résoudre ces différentes équations à l'aide d'approximations stochastiques.

Poids de transitions

Les poids de transitions n'étant bien évidemment pas indépendants, la résolution directe de l'équation 3.9 à l'aide d'un algorithme de gradient stochastique n'a aucun sens et nous devons regrouper les paramètres dépendants dans une seule variable (ou vecteur) afin de résoudre simultanément les équations de maximisation. Si l'on considère les poids de transitions $a_{i,j}^{(k)}$ indépendants dans les différentes bandes et pour les différents états de départ (i.e. les $a_{i,j}^{(k)}$ sont indépendants pour différentes valeurs de i et de k), on peut alors considérer le vecteur de paramètres $a_{i,(k)} \in \mathbb{R}^N$, regroupant les $a_{i,j}^{(k)}$. Notons qu'une telle considération est analogue à la contrainte de somme à 1 des probabilités dans le cas d'une chaîne de Markov. Les vecteurs $a_{i,(k)}$ sont deux à deux indépendants.

Pour toutes les valeurs possibles de i et de k, il est nécessaire de résoudre l'ensemble des équations

$$h_{i,j}^{(k)}(a_{i,(k)}) = 0 \qquad j = 1, \ldots, N \ , \tag{3.11}$$

où la fonction $h_{i,j}^{(k)}(a_{i,(k)})$ correspond à l'équation (3.9) prise en $\theta = \theta^{(n)}$, soit

$$h_{i,j}^{(k)}(a_{i,(k)}) = E[\varphi_{i,j}^{(k)}(x)|\theta^{(n)}] - E[\varphi_{i,j}^{(k)}(x)|Y, \theta^{(n)}] \ .$$

Si on applique un schéma de Newton-Raphson à cet ensemble d'équations, on obtient la formule de mise à jour des paramètres suivante

$$a_{i,(k)}^{(n+1)} = a_{i,(k)}^{(n)} - J_{i,(k)}^{-1}(a_{i,(k)}^{(n)}) \ h_{i,(k)}(a_{i,(k)}^{(n)}) \ , \tag{3.12}$$

où $h_{i,(k)}(a_{i,(k)})$ est un vecteur regroupant les fonctions $h_{i,j}^{(k)}(a_{i,(k)})$, soit

$$h_{i,(k)}(a_{i,(k)}) = \begin{pmatrix} h_{i,1}^{(k)}(a_{i,(k)}) \\ \vdots \\ h_{i,N}^{(k)}(a_{i,(k)}) \end{pmatrix} \ , \tag{3.13}$$

et où $J_{i,(k)}(a_{i,(k)})$ est la matrice de Jacobi définie par

$$J_{i,(k)}(a_{i,(k)}) = \begin{pmatrix} \frac{\partial h_{i,1}^{(k)}(a_{i,(k)})}{\partial a_{i,1}^{(k)}} & \dots & \frac{\partial h_{i,1}^{(k)}(a_{i,(k)})}{\partial a_{i,N}^{(k)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{i,N}^{(k)}(a_{i,(k)})}{\partial a_{i,1}^{(k)}} & \dots & \frac{\partial h_{i,N}^{(k)}(a_{i,(k)})}{\partial a_{i,N}^{(k)}} \end{pmatrix} . \tag{3.14}$$

Les dérivées partielles intervenant dans la matrice de Jacobi sont en fait les dérivées partielles secondes de la fonction auxiliaire puisque les fonctions $h_{i,j}^{(k)}$ sont, par définition, obtenues en dérivant la fonction auxiliaire par rapport aux paramètres $a_{i,j}^{(k)}$. On peut montrer que l'on a

$$\begin{aligned} \frac{\partial h_{i,m}^{(k)}(a_{i,(k)})}{\partial a_{i,n}^{(k)}} &= \frac{\partial^2 Q(\theta, \theta^{(n)})}{\partial a_{i,m}^{(k)} \partial a_{i,n}^{(k)}} \\ &= -cov_\theta(\varphi_{i,m}^{(k)}, \varphi_{i,n}^{(k)}) \ . \end{aligned} \tag{3.15}$$

Dans l'équation précédente, l'opérateur cov_θ désigne la covariance sous la loi *a priori* avec les paramètres θ. Notons que le terme lié à la densité *a posteriori* disparaît puisqu'il ne dépend pas des *vrais* paramètres θ mais de leur estimation courante $\theta^{(n)}$. La matrice de Jacobi est donc par construction une matrice définie négative ce qui permet de garantir l'accroissement presque certain de la vraisemblance à chaque itération [66].

Dans l'ensemble de ces équations, les espérances et les covariances mises en jeu ne sont bien évidemment pas explicitement calculables et sont remplacées par des estimations empiriques calculées à partir de tirages aléatoires selon les lois *a priori* et *a posteriori* effectués pour la valeur courantes $\theta^{(n)}$ des paramètres. L'étape de maximisation de l'algorithme EM est remplacée par un seul pas de l'algorithme de gradient stochastique tel que nous venons de le présenter. La procédure de remise à jour des poids de transitions consiste à appliquer l'équation (3.12) pour $k = 1, \dots, K$ et $i = 1, \dots, N$, après avoir estimé les espérances nécessaires à partir d'échantillons obtenus pour la valeur courante des paramètres.

Poids de synchronisation

On suppose que tous les paramètres de synchronisation $f_{k,l}$ sont indépendants. On peut donc appliquer directement un pas de l'algorithme de gradient stochastique pour résoudre l'équation (3.10) pour obtenir une nouvelle estimation des paramètres. En notant que

$$\frac{\partial^2 Q(\theta, \theta^{(n)})}{\partial^2 f_{k,l}} = -var_\theta(\psi_{k,l}(x)) \ ,$$

on obtient la formule de remise à jour suivante

$$f_{k,l}^{(n+1)} = f_{k,l}^{(n)} + \frac{E[\psi_{k,l}(x)|\theta^{(n)}] - E[\psi_{k,l}(x)|Y, \theta^{(n)}]}{var_{\theta^{(n)}}(\psi_{k,l})} \tag{3.16}$$

où, à nouveau, les espérances et les variances sont estimées à partir de tirages aléatoires suivant les lois *a priori* et *a posteriori* pour la valeur courante des paramètres.

Paramètres d'attache aux données

Nous utiliserons des gaussiennes comme densités de probabilité pour l'attache aux données dans toutes les expériences présentées dans la suite. Les formules d'estimation des moyennes et variances sont dans ce cas similaires à celles données par les équations (1.19) et (1.20). En revanche, contrairement au cas des HMM, les probabilités $\gamma_{t,k}(i)$ ne sont pas explicitement calculables et sont, à nouveau, remplacées par des estimations à partir de tirages aléatoires réalisés selon la loi *a posteriori* pour la valeur courante des paramètres. Par exemple, si l'on dispose de P échantillons $x^{(1)}$ à $x^{(P)}$ du champ selon la loi *a posteriori* , on remplacera alors $\gamma_{t,k}(i)$ par la moyenne empirique, soit

$$\gamma_{t,k}(i) \simeq \frac{1}{P}\sum_{p=1}^{P}\delta(x_{t,k}^{(p)} = i) \ .$$

En pratique, un seuillage de la variance est utilisé pour contraindre l'espace des paramètres afin d'éviter des valeurs improbables des paramètres qui donnent cependant une vraisemblance forte (voire infinie).

3.3.2 Comparaison avec l'approche gradient stochastique

Nous comparons dans cette partie l'approche EM généralisée avec l'algorithme de gradient stochastique dans le cas général où l'on dispose de M observations $y^{(1)}$ à $y^{(M)}$ pour l'estimation des paramètres.

Equations de maximisation

Dans l'approche gradient stochastique, les équations à résoudre sont données par

$$E[\varphi_{i,j}^{(k)}(x)|\theta] - E[\varphi_{i,j}^{(k)}(x)|Y,\theta] = 0 \tag{3.17}$$

pour les poids de transitions et

$$E[\psi_{k,l}(x)|\theta] - E[\psi_{k,l}(x)|Y,\theta] = 0 \ . \tag{3.18}$$

pour les poids de synchronisation. Ces deux équations sont très similaires aux équations (3.9) et (3.10) obtenues avec l'approche EM, la différence provenant du fait que les espérances *a posteriori* dépendent des vrais paramètres θ et non plus de l'estimation courante $\theta^{(n)}$. Pour les poids de transition, l'équation de remise à jour (3.12) s'applique alors au vecteur $h_{i,(k)}(a_{i,(k)}^{(n)})$ regroupant les équations (3.17) prise pour la valeur courante $\theta^{(n)}$ des paramètres, les éléments de la matrice de Jacobi étant alors donnés par

$$\frac{\partial h_{i,m}^{(k)}(a_{i,(k)})}{\partial a_{i,n}^{(k)}} = cov_\theta(\varphi_{i,m}^{(k)}, \varphi_{i,n}^{(k)}|Y = y) - cov_\theta(\varphi_{i,m}^{(k)}, \varphi_{i,n}^{(k)}) \quad . \tag{3.19}$$

A la différence de la formulation EM, le terme dû aux espérances sous la loi *a posteriori* ne disparaît pas. Les mêmes remarques s'appliquent pour la formule de maximisation des poids de synchronisation et le dénominateur de l'équation (3.16) devient alors $var_{\theta^{(n)}}(\psi_{k,l}) - var_{\theta^{(n)}}(\psi_{k,l}|Y = y)$. En effet, la différence entre les deux algorithmes résident uniquement dans la matrice de Jacobi puisque l'algorithme EM repose en partie sur le fait que la différence entre la vraisemblance et la fonction auxiliaire, $L(\theta) - Q(\theta, \theta^{(n)})$, admet un minimum pour $\theta = \theta^{(n)}$, soit

$$\left.\frac{\partial L(\theta)}{\partial \theta}\right|_{\theta^{(n)}} = \left.\frac{\partial Q(\theta, \theta^{(n)})}{\partial \theta}\right|_{\theta^{(n)}} \quad .$$

Nous nous intéresserons uniquement aux poids de transition par la suite, le cas des poids de synchronisation étant tout à fait similaire.

Approximations stochastiques

Plaçons nous maintenant dans le cas général où l'on dispose de M observations, $y^{(1)}$ à $y^{(M)}$, pour l'estimation des paramètres. L'extension des équations de maximisation (3.17) et (3.18) au cas multi-observations est directe et suit le même schéma que dans le cas de l'EM (cf. annexe B). Pour chaque exemple d'apprentissage, on réalise P tirages aléatoires selon les lois *a priori* et *a posteriori* pour approximer les espérances. En notant $x_{\text{prior}}^{(m,p)}$ (resp. $x_{\text{post}}^{(m,p)}$) l'échantillon p associé à la m-ième observation selon la loi *a priori* (resp. *a posteriori*), les covariances intervenant dans la matrice de Jacobi sont alors données par

$$\begin{aligned} cov(\varphi_{i,q}^{(k)}, \varphi_{i,r}^{(k)}) \quad &\simeq \quad \frac{1}{MP} \sum_{m,p} \varphi_{i,q}^{(k)}(x_{\text{prior}}^{(m,p)})\, \varphi_{i,r}^{(k)}(x_{\text{prior}}^{(m,p)}) - \\ &\quad \frac{1}{(MP)^2} \left[\sum_{m,p} \varphi_{i,q}^{(k)}(x_{\text{prior}}^{(m,p)})\right] \left[\sum_{m,p} \varphi_{i,r}^{(k)}(x_{\text{prior}}^{(m,p)})\right] \end{aligned} \tag{3.20}$$

pour la loi *a priori* et par

$$\begin{aligned} cov(\varphi_{i,q}^{(k)}, \varphi_{i,r}^{(k)}|Y = y^{(m)}) \quad &\simeq \quad \frac{1}{P} \sum_{p} \varphi_{i,q}^{(k)}(x_{\text{post}}^{(m,p)})\, \varphi_{i,r}^{(k)}(x_{\text{post}}^{(m,p)}) - \\ &\quad \frac{1}{P^2} \left[\sum_{p} \varphi_{i,q}^{(k)}(x_{\text{post}}^{(m,p)})\right] \left[\sum_{p} \varphi_{i,r}^{(k)}(x_{\text{post}}^{(m,p)})\right] \end{aligned} \tag{3.21}$$

pour la loi *a posteriori* .

Si l'on ne dispose que d'une seule séquence d'apprentissage, soit $M = 1$, et que l'on fixe $P = 1$ alors les covariances estimées valent 0. Cette valeur reflète en réalité le fait que les moments d'ordre 2 ne sont pas estimables à partir d'un seul échantillon. Dans ce cas, les éléments de la matrice de Jacobi sont remplacés par une constante arbitraire positive et on a également recours à un pas en $1/(n+1)$ pour assurer la convergence de l'algorithme (cf. section 2.3.2). Notons que les paramètres du modèle sont alors forcément considérés comme indépendants puisqu'il n'est pas possible d'estimer les relations qui les lient.

Dans le cas où l'on dispose de plusieurs séquences d'apprentissage, il est toujours possible d'estimer les covariances sous la loi *a priori* (3.20) du fait de la sommation sur m. En revanche, les covariances sous la loi *a posteriori* (3.21) sont estimables uniquement si $P > 1$. Lorsque $P = 1$, les covariances selon la loi *a posteriori* ne sont pas prises en compte dans la matrice de Jacobi et on retrouve alors, à condition de ne pas utiliser de pas de convergence, le schéma de l'approche EM. Si l'on considère l'ensemble des paramètres plutôt que les seuls poids de transition, on peut montrer que si $P = 1$ alors les formules de maximisation des deux algorithmes sont similaires. Notons qu'il est également possible d'utiliser un pas de convergence en $1/(n+1)$ dans l'algorithme EM généralisé ou d'approximer le Jacobien par n'importe quelle matrice (voir à ce sujet [101] où la matrice de Jacobi est remplacée par l'opposé de la matrice d'information de Fisher des données complètes) comme c'est le cas pour l'algorithme de gradient stochastique lorsque $M = 1$.

3.3.3 Initialisation des paramètres

Les algorithmes d'estimation des paramètres dont nous disposons étant connus pour converger vers un maximum local de la vraisemblance, il est nécessaire de partir d'une estimation $\theta^{(0)}$ la plus proche possible de la solution. Nous avons vu au chapitre 1 comment déterminer les paramètres d'un HMM à partir d'un alignement de Viterbi grâce à l'algorithme des k-moyennes segmental. Nous proposons d'utiliser un schéma similaire pour l'initialisation des paramètres du modèle RFM-*sync*1 à partir d'une segmentation au sens du *maximum a posteriori*[6], le principe de cet algorithme étant d'estimer une segmentation et de réestimer les paramètres de manière itérative.

En effet, pour une observation donnée $Y = y$ et une estimation courante $\theta^{(n)}$ des paramètres, on peut déterminer la réalisation

$$x^* = \arg\max_x P_{\theta^{(n)}}[X = x|Y = y] \ .$$

Comme nous l'avons vu au chapitre précédent, x^* peut-être déterminé à l'aide d'un algorithme de recuit simulé ou bien par l'algorithme ICM.

6. Notons qu'il est également possible, comme nous le verrons, de remplacer le critère de *maximum a posteriori* par un critère du type moyenne *a posteriori* maximum. Cependant, l'utilisation d'autres critères peut poser des problèmes dans le cas de topologies admettant des énergies barrières (cf. [90], section 6.4).

Les moyennes et variances des gaussiennes associées aux états sont alors réestimées en utilisant des estimateurs empiriques pour les données associées à l'état considéré (cf. eq. (1.9)). En ce qui concerne les poids de transitions, on peut estimer la probabilité d'une transition $i \longrightarrow j$ dans la bande k à partir de la fonction de comptage $\varphi_{i,j}^{(k)}$ donnée par l'equation (3.5). En effet, connaissant la segmentation x^*, la probabilité empirique est donnée par

$$\hat{P}_{ij}^{(k)} = \frac{\varphi_{i,j}^{(k)}(x^*)}{\sum_{j=1}^{N} \varphi_{i,j}^{(k)}(x^*)} \ .$$

D'après la formulation d'une chaîne de Markov en terme de distribution de Gibbs établie précédemment, nous pouvons alors estimer les poids de transition $a_{i,j}^{(k)}$ par

$$a_{i,j}^{(k)} = -\ln \hat{P}_{ij}^{(k)} = \ln \left(\sum_{j=1}^{N} \varphi_{i,j}^{(k)}(x^*) \right) - \ln \varphi_{i,j}^{(k)}(x^*) \ . \tag{3.22}$$

Les poids de synchronisation ne sont pas initialisés par cette procédure. On pourrait cependant envisager de le faire en se basant sur la mesure de $E[|x_{t,k}^* - x_{t,l}^*|]$. En effet, si cette espérance le long de la solution MAP est faible pour les deux bandes k et l, cela signifie donc que ces bandes sont synchronisées et on devrait donc avoir un poids de synchronisation inversement proportionnel à cette espérance.

3.3.4 Apprentissage heuristique

Suite à la remarque précédente concernant l'estimation des poids de synchronisation et en raison de la forte similarité entre le modèle RFM-*sync*1 et les HMM, nous proposons un apprentissage heuristique des paramètres du modèle [45]. En effet, on peut estimer les paramètres des HMM de manière indépendante dans chaque bande en utilisant classiquement l'algorithme de Baum-Welch, et en déduire les paramètres de la distribution de Gibbs, comme précédemment. On peut également définir par une heuristique les poids de transitions, soit

$$f_{k,l} = \frac{\gamma \, T}{\sum_t |x_{t,k}^* - x_{t,l}^*|} \ ,$$

T étant le nombre de trame de l'observation, les valeurs $x_{t,k}^*$ étant obtenus, de manière indépendante dans chaque bande, par l'algorithme de Viterbi. Cet estimateur est bien inversement proportionnel à l'espérance que nous approximons ici par la moyenne empirique. L'hyper-paramètre γ permet d'accorder plus ou moins d'importance aux potentiels verticaux, donc à la synchronisation, par rapport aux potentiels horizontaux. En effet, lorsque γ augmente, les poids de transition deviennent plus grands et ont donc plus d'importance dans l'énergie *a priori* totale.

3.4 Stratégies de décodage

Rappelons que le décodage de la parole repose en partie sur le calcul du score acoustique $P[Y = y|W = w]$ donné par l'équation (1.5). Comme pour les HMM, le calcul direct est impossible et nous avons recours à des approximations. L'idée consiste à travailler sur les données complètes plutôt que sur les observations uniquement en approximant la vraisemblance de l'observation y par

$$P[Y = y|W = w] \simeq P[Y = y|X = x^*, W = w] \, P[X = x^*|W = w] \ , \qquad (3.23)$$

le champ x^* correspondant à la segmentation la plus vraisemblable, soit $x^* = \arg\max_x P[X = x|Y = y]$. Ce critère n'est en fait autre que le critère de Viterbi (1.7) puisque l'on a

$$\begin{aligned} P[Y = y|W = w] &\simeq \max_x P[X = x, Y = y|W = w] \\ &\simeq P[Y = y|W = w] \max_x P[X = x|Y = y, W = w] \ . \end{aligned}$$

Deux algorithmes sont utilisables pour estimer le champ caché le plus vraisemblable: l'algorithme ICM et le recuit simulé. L'ICM converge vers un maximum local de la probabilité et requiert donc une bonne initialisation de la solution. On peut initialiser la solution par une segmentation uniforme ce qui ne fournit pas forcement un bon point de départ. Une autre solution consiste à utiliser l'algorithme de Viterbi indépendamment dans chaque bande pour donner une première estimation de la segmentation. En effet, en considérant les bandes comme indépendantes (*i.e.* en négligeant les potentiels de synchronisation), le théorème 2 nous permet de trouver le HMM équivalent dans chaque bande en vue de la segmentation par l'algorithme de Viterbi. Notons qu'en l'absence de couplage entre les bandes, la solution obtenue est directement la meilleure. Dans le cas du recuit simulé, il est raisonnable de partir d'une segmentation uniforme mais il reste à déterminer le choix de la température initiale et le facteur d'atténuation de la température.

Le calcul de la vraisemblance (3.23) n'est pas possible, même dans le cas des données complètes (x^*, y) du fait de la fonction de partition dans le calcul de la probabilité de $P[X = x^*|W = w]$. Nous remplacerons donc la log-vraisemblance par la log pseudo-vraisemblance définie par l'équation (2.21).

La figure 3.3 illustre la stratégie de décodage proposée. L'évolution de la log pseudo-vraisemblance (log-PV) est tracée en fonction du nombre d'itérations de l'algorithme de recuit simulé pour deux lois géométriques de descente de température de raison respective 0.097 et 0.095. Pour le recuit simulé à partir d'une segmentation uniforme, on voit que la log-PV converge vers la log-PV de la meilleure solution donnée par l'algorithme Viterbi tandis que l'algorithme ICM donne une solution moins vraisemblable.

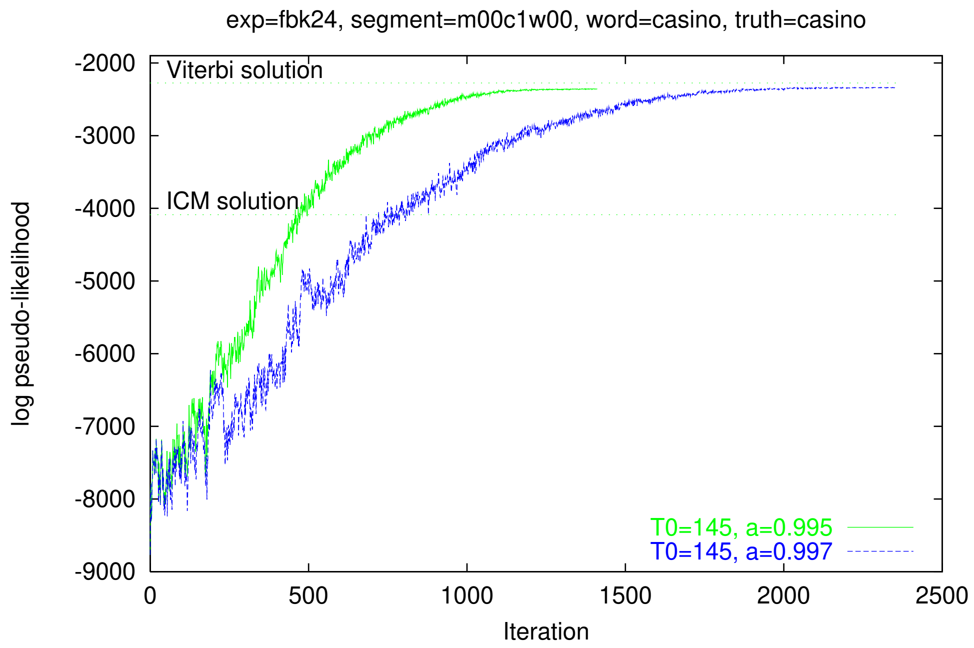


FIG. 3.3 – *Illustration de la stratégie de décodage (cas de 24 bandes non couplées).*

3.5 Bilan

A partir de l'équivalence entre chaîne de Markov et distribution de Gibbs, nous avons proposé un modèle qui permet de représenter la synchronie (ou l'asynchronie) entre les bandes de fréquences dans une approche multi-bande. Nous avons également proposé une généralisation de l'algorithme EM pour l'estimation des paramètres du modèle et montré les liens entre ce dernier et l'algorithme de gradient stochastique. Nous nous proposons maintenant d'étudier pour le modèle proposé les algorithmes d'initialisation et d'estimation des paramètres définis dans ce chapitre, à l'aide de données simulées.

Chapitre 4

Estimation des paramètres sur des données simulées

4.1 Introduction

4.1.1 Objectifs, motivations

Dans ce chapitre, nous étudions les algorithmes d'initialisation et d'estimation des paramètres proposés au chapitre précédent à l'aide de simulations. Pour un modèle dont les paramètres sont connus, il est possible de réaliser des tirages aléatoires d'observations qui sont utilisées pour l'initialisation puis pour l'estimation des paramètres du modèle. Il est alors possible d'étudier la validité des algorithmes proposés à partir de données suivant la distribution définie par le modèle. En particulier, nous nous intéresserons au cas des chaînes de Markov cachées pour lesquelles nous disposons d'algorithmes connus. L'étape de validation des algorithmes nous parait essentielle dans la mesure où, pour l'algorithme EM généralisé d'estimation des paramètres, aucun résultat théorique sur la convergence n'a été établi. Les expériences présentées dans ce chapitre permettent également d'illustrer les définitions théoriques du chapitre précédent.

Deux points nous semblent particulièrement importants dans ces simulations. Le premier point important est la convergence des algorithmes vers une solution, cette dernière n'étant pas forcément la meilleure solution. En effet, si, par construction, les algorithmes d'initialisation des paramètres finissent forcément par se stabiliser autour d'une solution, il n'en va pas forcément de même pour les algorithmes d'estimation des paramètres qui font appel à des approximations stochastiques sans introduire de pas de convergence. Nous étudierons simplement la convergence en traçant l'évolution des paramètres au cours des itérations. Le deuxième point important est la validité des paramètres obtenus. Comme nous l'avons vu, plusieurs ensembles de paramètres peuvent définir la même distribution de Gibbs puisque seules les différences d'énergies sont importantes (cf. section 2.4.1). Afin de pouvoir comparer les paramètres obtenus avec les paramètres ayant été utilisés pour le tirage aléatoire, il est nécessaire d'avoir une représentation unique de la distribution de Gibbs étudiée. La nécessité d'une représentation unique intervient pour les poids de transition et pour les poids de synchronisation. Cette remarque nous renforce dans le choix de travailler sur des HMM puisque nous disposons alors du théorème 2 qui nous permet de retrouver la matrice de transition d'une chaîne de Markov équivalente à une distribution de Gibbs homogène positive. Pour un modèle multi-bandes, on pourra définir une matrice de transition associée à chaque bande en négligeant l'influence de la synchronisation. Nous ne disposons malheureusement pas d'une telle représentation unifiée pour les poids de synchronisation et nous nous contenterons alors d'une analyse qualitative et subjective.

Pour maintenir la lisibilité de ce chapitre, une partie des figures de convergence a été placée en annexe C.

4.1.2 Définition des modèles

Les simulations présentées s'appuient sur les trois modèles suivants:

- une chaîne de Markov ergodique à deux états (*n2*),
- une chaîne de Markov gauche-droite à trois états (*n3*), et
- une distribution de Gibbs sur deux bandes simulant deux chaînes de Markov gauche-droite couplées (*k2*).

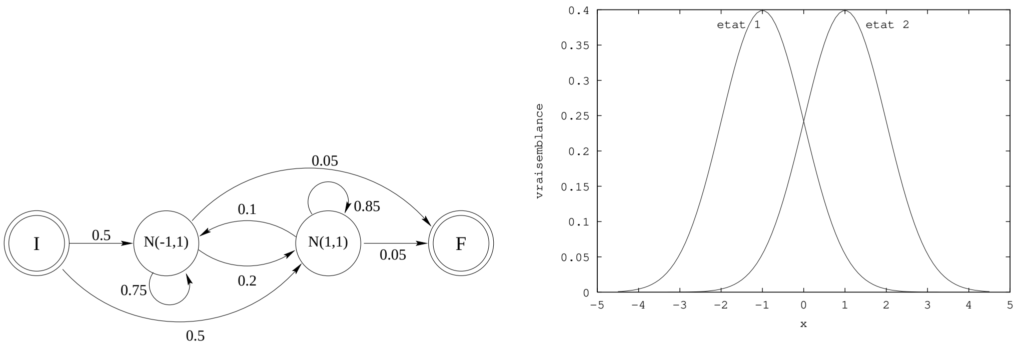
Pour tout ces modèles, l'espace des observations dans une bande est monodimensionnel (*i.e.* $y_{t,k} \in \mathbb{R}$) et les densités de probabilités associées aux états sont des mono-gaussiennes de variance unité.

Le modèle n2

Le premier modèle, que nous noterons *n2*, correspond à une chaîne de Markov ergodique à deux états dont les probabilités de transitions sont données sur la figure 4.1*(a)*. Notons que du fait des états artificiels initiaux et finaux, l'ergodicité n'est vrai que pour les états émetteurs. Les gaussiennes associées aux états sont de moyennes respectives 1 et -1 comme le montre la figure 4.1*(b)* donnée ici pour illustrer la séparabilité des deux classes du mélange. L'intérêt du modèle de mélange ergodique vient principalement du fait qu'il n'y a pas d'énergie barrière dans la fonction potentiel ce qui signifie que l'on peut appliquer sans difficultés majeures les résultats sur l'équivalence entre distribution de Gibbs et chaîne de Markov.

Nous avons effectué le tirage aléatoire de 200 réalisations à partir du HMM équivalent [1], la durée des observations étant comprise entre 1 et 106 trames avec une moyenne de 20 trames.

1. cf. le descriptif de la commande HSource dans [111] pour plus de détail sur la procédure de tirage aléatoire



(a) Topologie et paramètres *(b)* densités de probabilité

FIG. 4.1 – *Paramètres du modèle* n2

Le modèle n3

Le second modèle, noté *n3* et représenté sur la figure 4.2, correspond à une chaîne de Markov cachée gauche-droite à trois états. Les moyennes des gaussiennes ont été fixées respectivement à -2, 0 et 2 de manière à ce que la séparabilité des observations entre deux états consécutifs soit la même que pour le modèle *n2*. L'intérêt de ce modèle est évidemment la topologie gauche-

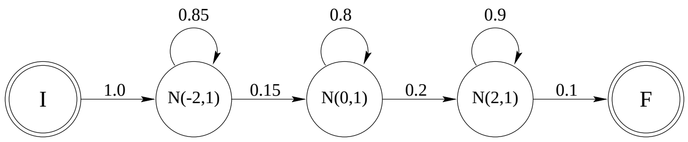


FIG. 4.2 – *Paramètres du modèle* n3

droite contrainte de la chaîne de Markov sous-jacente, topologie classiquement utilisée en parole pour modéliser l'évolution temporelle du signal. L'application du théorème 2 pose parfois des problèmes dans ce cas, notamment des problèmes numériques lorsque les énergies barrières sont trop élevées.

Comme pour le modèle précédent, nous avons réalisé le tirage aléatoire de 200 observations à partir de la formulation HMM du modèle, le nombre de trames dans une observation étant compris entre 4 et 81 avec une moyenne de 22 trames.

Le modèle k2

Les deux modèles précédents étant mono-bande, ils ne peuvent être utilisés pour étudier l'estimation des poids de synchronisation. Pour cela, nous utiliserons un modèle gauche-droite à deux bandes avec trois états par bande. La première bande correspond au modèle gauche-droite défini précédemment, tandis que dans la deuxième bande, les probabilités de bouclage sur les états sont respectivement de 0.95, 0.6 et 0.8 pour des moyennes de -1.5, 0 et 1.5. Différentes valeurs de paramètres de synchronisation peuvent être utilisées et nous avons mené des expériences avec $f_{1,2} = 0,\ 0.5,\ 1$ et 2.

A nouveau, pour chaque valeur du paramètre de synchronisation entre les deux bandes, 200 échantillons ont été tirés en utilisant l'échantillonneur de Metropolis, la longueur d'une observation étant déterminée suivant une loi gaussienne de moyenne 22 trames et d'écart-type 10. Le choix de la moyenne et de l'écart-type provient des statistiques obtenues sur les tirages aléatoires effectués avec le modèle *n3*.

4.2 Initialisation des paramètres

4.2.1 Cas du modèle n2

La convergence des paramètres pour différentes techniques d'initialisation est donnée figure 4.3 pour le modèle *n2*. Nous comparons ici les approches ICM et recuit simulé dans le calcul de la solution x^* utilisée pour l'estimation empirique des paramètres.

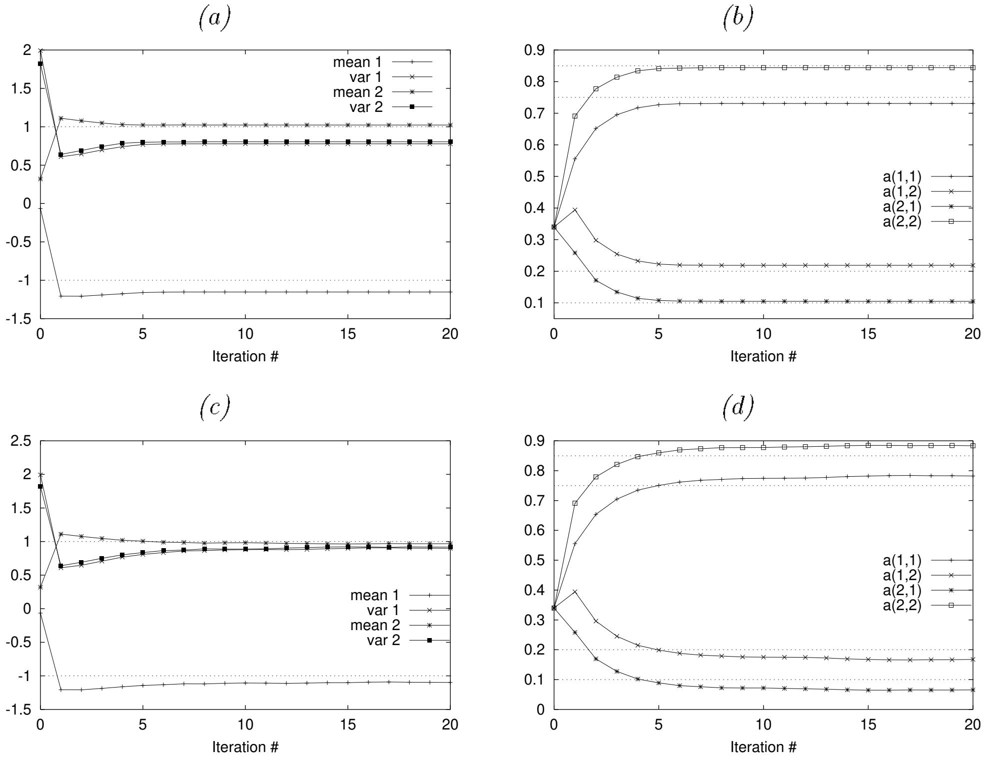


FIG. 4.3 – *Initialisation des paramètres du modèle n2 par ICM (a et b) et par recuit simulé (c et d). Les figures a et c correspondent aux paramètres des densités tandis que b et d correspondent aux probabilités de transitions.*

On note que les deux algorithmes convergent très vite vers une solution, la convergence étant un peu plus longue pour le recuit simulé qui converge en une dizaine d'itérations tandis que l'ICM se stabilise après 5 itérations. Les probabilités de transitions convergent vers les bonnes valeurs dans les deux cas et, pour les paramètres des densités gaussiennes, l'initialisation à base de recuit simulé donne une estimation plus précise des paramètres. Ceci s'explique par le fait que les segmentations obtenues par l'algorithme de recuit simulé sont plus *lisses*, dans le sens où cet algorithme favorise des grandes régions homogènes du fait de sa globalité. Au contraire, la segmentation fournie par l'algorithme ICM a tendance à donner des estimateurs des densités pour lesquels la variance est minimale, au détriment des moyennes. Sur les mêmes données, l'algorithme de Viterbi converge très lentement en une soixantaine d'itérations vers des valeurs proches des valeurs théoriques. Notons cependant que la procédure d'initialisation par Viterbi utilisée ne remet à jour que les paramètres des densités gaussiennes sans remettre à jour les probabilités de transitions [2]. Lorsque les deux composantes du mélange sont moins séparables, c'est l'initialisation par l'algorithme ICM qui donne alors les meilleurs résultats, même si les variances restent sous-estimées au détriment des moyennes. Lorsque la différence entre les deux moyennes est de 1, la variance étant de 1 pour les deux composantes gaussiennes, l'initialisation par recuit simulé ne converge plus vers les bonnes valeurs. En effet, dans ce cas, les points d'une observation étant confondus quelque soit la distribution dont ils sont issus, ils sont tous classés comme correspondant au même état par l'algorithme de recuit simulé afin d'avoir une segmentation la plus homogène possible. On a donc un état qui cumule les deux distributions, l'autre étant marginalement utilisé pour quelques points. Les courbes correspondants à cette expérience sont données en annexe C, figure C.1.

Enfin, l'espace des réalisations du champ caché étant non contraint, il est possible d'estimer le champ x^* avec un critère de moyenne *a posteriori* maximale (MPM) en remplacement du critère de maximum *a posteriori* . Les résultats obtenus avec un estimateur MPM sont similaires à ceux obtenus dans le cas des modes conditionnels itérés, avec cependant un temps de convergence légèrement plus long et une estimation des probabilités de transitions plus précises.

4.2.2 Cas du modèle n3

Pour le modèle *n3*, des courbes similaires aux précédentes sont données sur figure 4.4.

Les deux techniques d'initialisation convergent en quelques itérations. La convergence est plus rapide que dans le cas d'une chaîne ergodique ce qui ce justifie par la technique utilisée pour établir les paramètres initiaux du modèle. En effet, on note que les moyennes à l'itération 0, qui correspond en quelque sorte à l'initialisation de l'initialisation, sont plus proches de leurs vraies valeurs dans le cas d'une chaîne gauche droite que dans le cas ergodique, les moyennes et variances étant tout d'abord estimées sur la base d'une segmentation uniforme. Bien que non optimale, la segmentation uniforme est plus réaliste pour une chaîne gauche-droite que pour une chaîne ergodique, ce qui explique la convergence rapide. A nouveau, on remarque que l'initialisation basée sur le recuit simulé donne une estimation plus précise des paramètres. L'initialisation classique des HMM à l'aide de l'algorithme de Viterbi donne également de bons résultats avec une convergence rapide. Dans le cas de données moins séparables, c'est à dire lorsque les moyennes de deux états consécutifs sont proches, les deux algorithmes se comportent bien, le recuit simulé donnant toujours de meilleurs résultats. Le maintien des bonnes performances du recuit simulé sur

2. Il est également fort possible que le logiciel utilisé, HTK v1.4, comporte un *bug* dans le cas de l'initialisation des paramètres d'une chaîne ergodique.

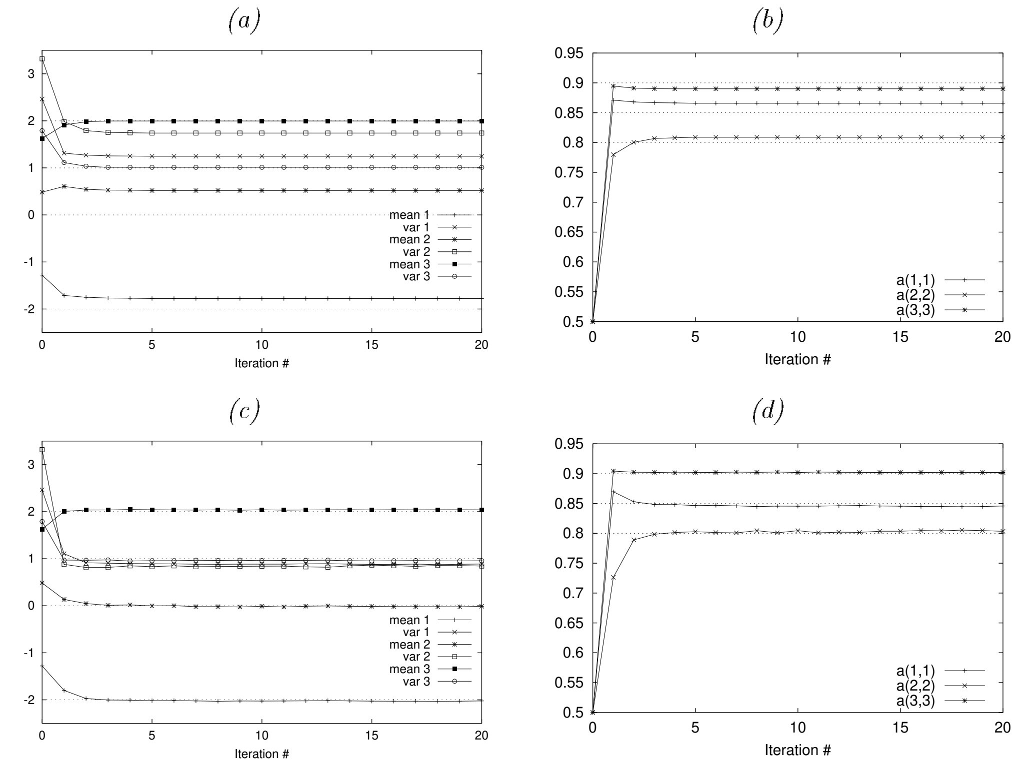


FIG. 4.4 – *Initialisation des paramètres du modèle n3 par ICM (a et b) et par recuit simulé (c et d). Les figures a et c correspondent aux paramètres des densités tandis que b et d correspondent aux probabilités de transitions.*

des données difficilement séparables s'explique par la topologie contrainte du modèle. La convergence de l'initialisation à base de recuit simulé est évidemment plus lente dans ce cas. Les courbes illustrant le cas difficilement séparable sont données en annexe C, figure C.2.

4.2.3 Cas du modèle k2

Lorsque les deux bandes sont indépendantes (*i.e.* $f_{1,2} = 0$), les résultats pour la première bande sont bien sur les mêmes que pour le modèle *n3*. En revanche, avec l'algorithme ICM, on observe bien la convergence des paramètres dans la deuxième bande mais vers des valeurs relativement éloignées des valeurs théoriques, notamment en ce qui concerne la moyenne associée à l'état 2. Ce problème s'explique par le fait qu'on dispose de peu d'observations associées à cet état, d'une part parce que la probabilité de bouclage $a_{2,2}$ est faible et, d'autre part, parce que les données sont plus difficilement séparables dans cette bande, ce qui rend l'algorithme ICM moins performant. La mauvaise estimation de cette moyenne se répercute naturellement au niveau de l'estimation des variances et des probabilités de transitions. Lorsqu'on ajoute une synchronisation des deux bandes, le problème se répercute alors sur la première bande et les paramètres sont alors mal estimés dans les deux bandes. L'utilisation du recuit simulé permet de résoudre en partie ces problèmes grâce à une meilleure segmentation dans la deuxième bande. Les paramètres des densités sont alors correctement initialisés quelque soit la valeur du poids de synchronisation. Les figures sont données en annexe C, figure C.3.

4.2.4 Exemples de segmentation

Pour conclure cette partie sur les algorithmes d'initialisation, nous donnons figure 4.5 quelques exemples de segmentations obtenues après convergence des algorithmes. Dans cette figure, les lignes correspondent aux observations, les pointillés aux valeurs théoriques des moyennes et les losanges donnent la segmentation. Pour clarifier le dessin, les marques de segmentation (les losanges) ont été placées sur la moyenne théorique associée à l'état.

Comme nous le mentionnions auparavant, on remarque que la segmentation obtenue par recuit simulé est plus réaliste que celle obtenue par ICM, ce qui n'est pas surprenant. Sur l'observation correspondant au modèle *n2*, on note que le recuit favorise les zones de segmentation homogène par rapport à l'ICM. Sur les quelques cas que nous avons regardés, la segmentation obtenue par recuit simulé est similaire à la segmentation donnée par l'algorithme de Viterbi. Un estimateur MPM associé au modèle *n2* donne également des segmentations similaires à celles du recuit simulé après convergence de l'algorithme d'initialisation des paramètres. La meilleure qualité de la segmentation par recuit simulé explique les meilleurs résultats obtenus avec cette méthode lorsque les données sont facilement séparables ou lorsque la segmentation est contrainte.

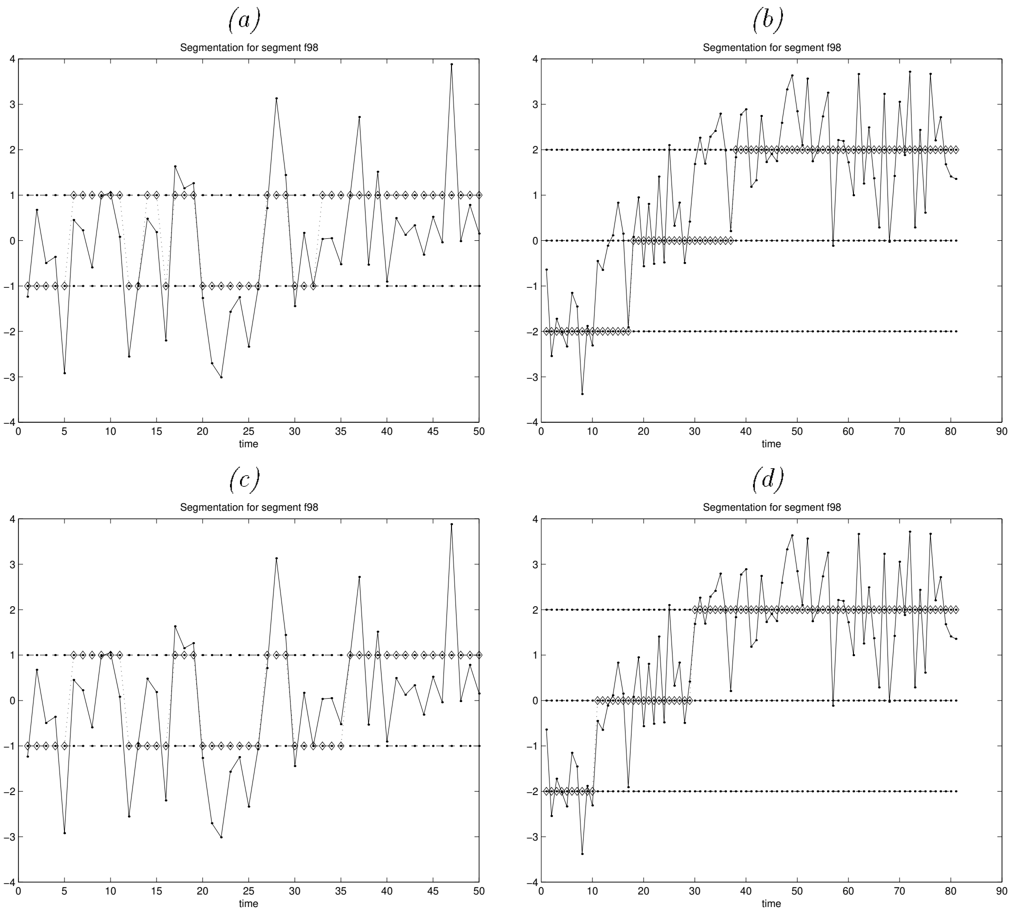


FIG. 4.5 – *Exemple de segmentations après initialisation par ICM (a et b) et par recuit simulé (c et d) pour les modèles n2 et n3.*

4.3 Estimation par l'algorithme EM

Sauf mention contraire, dans toutes les expériences présentées ci-dessous, les espérances ont été estimées à partir de 4 échantillons, c'est à dire qu'à chaque exemple d'apprentissage, 4 échantillons suivant la loi *a priori* et 4 suivant la loi *a posteriori* sont tirés et utilisés pour l'approximation des espérances (cf. annexe B, $M = 4$).

4.3.1 Réestimation des paramètres

Comme nous venons de le voir, les paramètres convergent vers des valeurs proches des valeurs théoriques avec les algorithmes d'initialisation présentés précédemment. Lorsqu'on applique l'algorithme EM généralisé aux modèles initialisés, les paramètres convergent très rapidement vers leur valeur théorique. Notons cependant qu'après convergence, les paramètres estimés oscillent faiblement autour de leurs valeurs théoriques respectives mais ne se stabilisent jamais complètement. La stabilisation des estimateurs pourrait être obtenue en ajoutant un pas de convergence à l'étape de gradient de l'algorithme. Lorsque les paramètres initialisés sont éloignés des valeurs théoriques, comme c'est le cas pour le modèle à deux bandes initialisé avec l'algorithme ICM ou pour les modèles à données difficilement séparables, l'algorithme EM généralisé converge également vers la solution théorique, cette convergence étant relativement lente. Pour le modèle *k2*, *i.e.* dans le cas de bandes asynchrones, la convergence se fait en une quinzaine d'itérations tandis que lorsque les bandes sont synchronisées, la vitesse de convergence augmente de manière inversement proportionnelle au poids de synchronisation. On note que lorsque la synchronisation entre les deux bandes augmente, les poids de transitions ne convergent pas tout a fait vers les valeurs théoriques ce qui peut s'expliquer par le fait que les probabilités de transition sont obtenues sous l'hypothèse que les bandes sont indépendantes.

Enfin, la procédure de réestimation des paramètres permet aussi d'estimer les poids de synchronisation qui n'interviennent pas dans la phase d'initialisation. La figure 4.6 illustre la convergence du paramètre $f_{1,2}$ avec l'algorithme EM généralisé après initialisation des modèles par ICM. Le poids de synchronisation est bien estimé pour des valeurs faibles de ce dernier. En revanche, pour des valeurs élevées, on ne retrouve pas la valeur d'origine. Nous rappelons que cela ne signifie pas que le paramètre est mal estimé puisque ce sont les différences d'énergies qui sont importantes plutôt que les valeurs absolues de ces énergies. Notons enfin que $f = 2$ correspond à un cas où les deux bandes sont quasiment totalement synchrones.

4.3.2 Estimation directe des paramètres

En dernier lieu, nous présentons quelques résultats obtenus en appliquant directement l'algorithme EM généralisé sans initialisation. Ces résultats sont donnés dans le but de mieux comprendre cet algorithme. En effet, sur les simula-

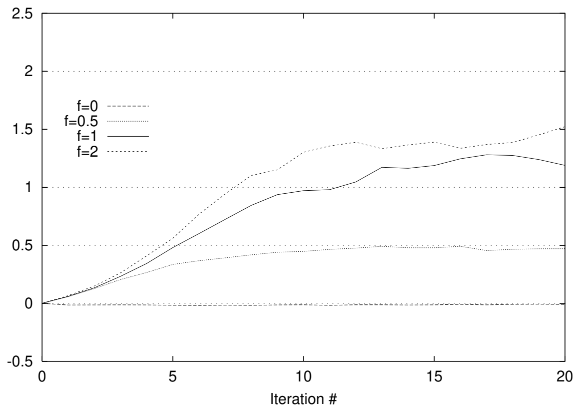


FIG. 4.6 – *Convergence du poids de synchronisation dans le modèle k2 après initialisation par ICM.*

tions précédentes, les procédures d'initialisation donnent une solution "*proche*" de la solution théorique, ce qui ne permet pas vraiment d'étudier le comportement de l'algorithme d'estimation des paramètres et notamment la qualité des estimateurs obtenus.

Dans toutes les simulations effectuées, nous avons observé la convergence de l'algorithme pour tous les paramètres, la convergence étant cependant longue dans le cas de la chaîne ergodique. Les résultats pour le cas de données facilement séparables sont donnés figure 4.7. L'estimation des paramètres d'attache aux données par l'algorithme EM donne toujours des valeurs proches des valeurs théoriques, ce qui n'est pas le cas pour les probabilités de transition qui sont mal estimées. Nous n'avons malheureusement pas d'explication à la mauvaise estimation des poids de transition. En effet, on peut supposer que cette mauvaise estimation est due à l'approximation des espérances à partir de 4 tirages aléatoires par exemple d'apprentissage ($M = 4$) mais, lorsqu'on augmente le nombre de tirages aléatoires réalisés jusqu'à $M = 10$ afin de mieux estimer les espérances, l'estimation des probabilités de transition ne s'améliore pas. Une explication possible réside dans la mauvaise modélisation des dépendances entre les poids de synchronisation. En effet, des résultats à peu près similaires sont obtenus avec le modèle *n2* en considérant les poids de transition comme indépendant. Si les probabilités de transition sont mal estimées, on note cependant pour le modèle *n2* que lorsque l'on considère les poids de transition sortant du même état comme dépendants, la différence entre $a_{2,1}$ et $a_{2,2}$ est plus grande. Pour le modèle *n3*, du fait de la forme très particulière de la matrice de Jacobi, les poids de transitions sont toujours considérés comme indépendants. Enfin, lorsqu'on approxime les espérances par une seule réalisation ($M = 1$), ce qui correspond à un algorithme de gradient stochastique, nous avons observé sur les deux modèles *n2* et *n3* que les paramètres convergent vers les mêmes valeurs que pour $M = 4$, la convergence étant plus longue pour le modèle ergodique. La différence réside dans le fait qu'après convergence, les paramètres oscillent

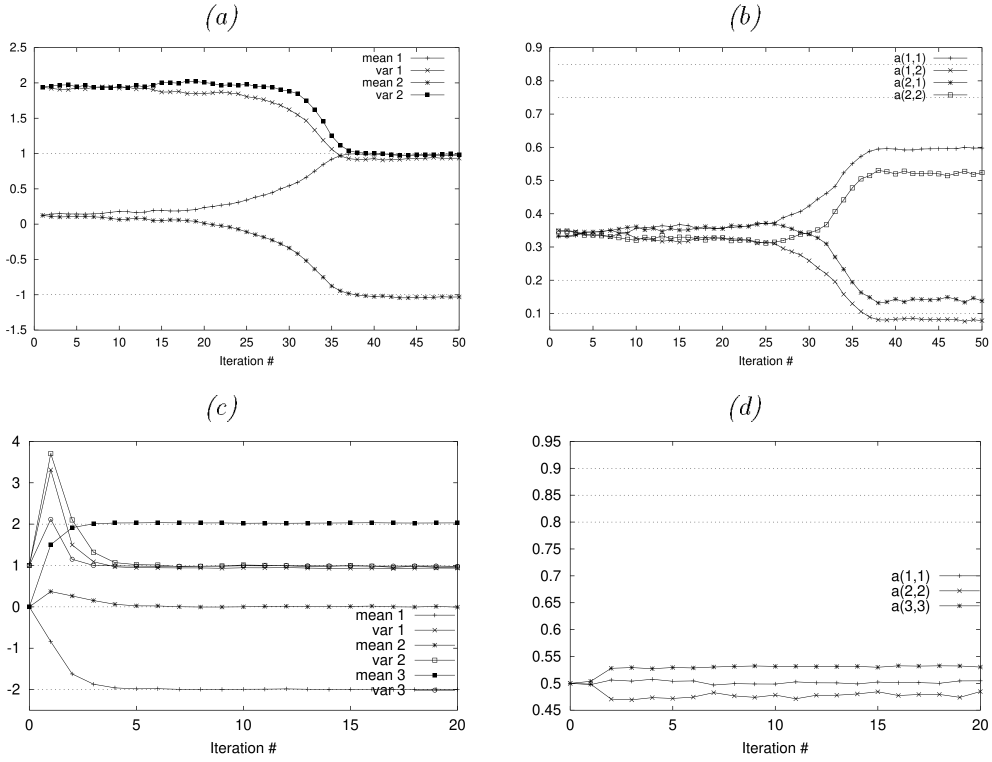


FIG. 4.7 – *Convergence de l'algorithme EM pour les modèles n2 (a et b) et n3 (c et d). Les figures a et c correspondent aux paramètres des densités tandis que b et d correspondent aux probabilités de transitions.*

autour des "vraies" valeurs dans un intervalle plus grand lorsqu'un seul échantillon est utilisé pour approximer les espérances. Les courbes de convergence pour le modèle *n2* avec $M = 1$ et en considérant les poids de transitions comme indépendant sont données en annexe, figures C.4 et C.5.

Dans le modèle multi-bande, les mêmes difficultés se retrouvent et l'estimation des poids de transition n'est pas bonne alors que les paramètres des densités sont correctement estimés quelque soit la valeur de la synchronisation. Il est intéressant de noter la différence entre les résultats obtenus pour la première et la deuxième bande dans l'estimation. En effet, dans la deuxième bande, les probabilités de transition estimées sont plus "*écartées*" que dans la première bande, ce qui correspond à la réalité. Il semblerait donc que les écarts entre probabilité de bouclage dans un modèle gauche-droite soient respectés tandis que l'ordre de grandeur de ces probabilités n'est pas estimé correctement. Enfin, comme dans le cas où l'EM était appliqué après initialisation des modèles, les poids de transition sont correctement estimés lorsque la valeur du poids de synchronisation n'est pas trop élevée, comme le montre la figure 4.8.

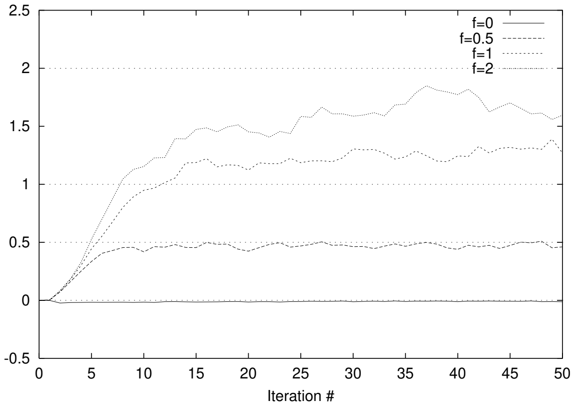


FIG. 4.8 – *Convergence du poids de synchronisation dans le modèle k2 avec l'algorithme EM généralisé.*

4.4 Bilan

Les simulations présentées dans ce chapitre ont permis de valider et de mieux connaître les algorithmes présentés de manière théorique au chapitre 3. Nous avons ainsi pu mettre en évidence les bonnes performances des algorithmes d'initialisation des paramètres et plus particulièrement de l'algorithme basé sur le recuit simulé. L'initialisation à l'aide de l'algorithme ICM donne cependant de meilleures estimations lorsque les données sont peu séparables et présente l'avantage d'être plus rapide et de ne dépendre d'aucuns paramètres, à l'inverse du recuit simulé qui dépend de la température initiale et de la loi de descente de la température. Malgré des limites évidentes, en particulier concernant l'estimation des poids de transition, l'algorithme EM généralisé s'est avéré performant et se comporte de manière favorable par rapport aux algorithmes de gradient stochastique. Il reste néanmoins nécessaire de l'utiliser en complément d'un des algorithmes d'initialisation proposé. Enfin, le problème de la mesure de la convergence de ces algorithmes n'a pas été abordé dans cette partie. Nous nous contenterons par la suite de fixer le nombre d'itérations mais un critère basé sur la variation des paramètres est également envisageable et reste à étudier.

Chapitre 5

Reconnaissance de mots isolés

Ce chapitre est dédié à l'utilisation du modèle RFM-*sync*1 pour la reconnaissance de mots isolés. Nous y présentons les expériences effectuées sur de la parole téléphonique et commentons les résultats obtenus.

5.1 Présentation de l'application

5.1.1 Protocole expérimental

Toutes les expériences présentées dans ce chapitre portent sur la reconnaissance mono-locuteur de mots isolés et ont été effectuées sur un petit sous-ensemble de la base de données PolyVar [24]. PolyVar est une base de données téléphonique enregistrée à l'IDIAP dans le but de travailler sur la modélisation des variabilités inter et intra locuteur. En particulier, afin d'étudier la variabilité intra-locuteur, chaque locuteur a réalisé plusieurs appels téléphoniques au serveur, en prononçant à chaque appel plusieurs mots-clés choisis parmi une liste de 17 mots-clés. Pour réaliser les expériences, nous avons sélectionné un locuteur masculin ayant un nombre d'appels suffisant ainsi qu'une liste de 10 mots parmi les 17. La liste des mots utilisés est donnée dans le tableau 5.2. Pour chaque mot, nous disposons de 100 répétitions, les 50 premières, dans l'ordre chronologique, étant utilisées pour l'estimation des paramètres des modèles, tandis que les 50 dernières servent aux tests. Ces deux ensembles, apprentissage et test, ont chacun été extrait d'une cinquantaine de sessions enregistrées sur une période de 5 mois, entre Février et Juin, pour le corpus d'apprentissage et entre Août et Octobre pour le corpus de test.

Au total, les taux de reconnaissance que nous allons présenter sont donc calculés sur un ensemble de 500 tests. La lecture et l'interprétation des résultats ne pouvant être faits sans la définition d'un intervalle de confiance, nous donnons tableau 5.1 les intervalles de confiance à 95% et à 99% associés à différentes valeurs du taux de reconnaissance. Ces intervalles ont été calculés en supposant que le taux de reconnaissance suit une loi de Bernoulli [95]. Afin de ne pas surcharger les tableaux de résultats, nous ne rappellerons pas systématiquement les intervalles de confiance et nous invitons le lecteur à se référer à ce tableau.

taux de reconnaissance	intervalle à 95%	intervalle à 99%
90	$[81.5, 97.1]$	$[79.1, 98.7]$
80	$[72.5, 86.3]$	$[70.3, 87.7]$
40	$[36.3, 43.1]$	$[35.2, 43.8]$

TAB. 5.1 – *Intervalles de confiance à 95% et à 99%*

5.1.2 Système de référence

Afin de comparer la modélisation par champs de Markov aux techniques classiques, nous utiliserons un système de référence basé sur la modélisation par chaînes de Markov cachées. Dans ce système noté HMM_{cep}, 12 coefficients cepstraux, calculés à partir de la sortie d'un banc de 24 filtres régulièrement répartis entre 0 et 4000 Hz, sont utilisés pour représenter une trame de 20 ms. de signal. Chaque mot est modélisé par un HMM dont le nombre d'états, donné dans le tableau 5.2, est fonction du nombre de phonèmes composants le mot. Par la suite, nous utiliserons toujours le même nombre d'états par bande que pour le système de référence. Etant donné la simplicité de la tâche et le peu de données d'apprentissage, des densités mono-gaussiennes à matrice de covariance diagonale sont associées aux états pour modéliser l'attache aux données.

Le décodage est guidé par l'utilisation d'un réseau syntaxique dans lequel les 10 mots à reconnaître sont en parallèle et plusieurs algorithmes d'apprentissage ont été testés. L'apprentissage par Viterbi et l'application directe de l'algorithme de Baum-Welch donne un taux de reconnaissance de 99.8% et l'utilisation de la réestimation par Baum-Welch après une initialisation par Viterbi permet de corriger la seule erreur pour obtenir un taux de reconnaissance de 100%.

annulation (16)	*casino* (12)	*cinéma* (12)	*concert* (10)	*corso* (10)
guide (6)	*message* (10)	*musée* (8)	*quitter* (8)	*suivant* (10)

TAB. 5.2 – *Liste des mots de l'application. Le chiffre entre parenthèse correspond au nombre d'état du modèle.*

Au vu des résultats du système de référence, il est évident que l'objectif des expériences présentées n'est pas d'améliorer ce dernier. Nous nous proposons en fait d'étudier la modélisation de la parole par champs de Markov sur cette application simple afin de mieux comprendre les différences entre les HMM et les RFM.

5.2 Application à une analyse par banc de filtre

5.2.1 Motivations

Dans un premier temps, nous nous proposons d'appliquer le modèle RFM-*sync*1 sur des paramètres acoustiques directement issus d'une analyse par banc de filtres. En effet, rappelons que la motivation principale pour l'étude d'un modèle bi-dimensionel était de pouvoir modéliser la parole directement dans le plan temps/fréquence, ce type d'analyse ne reposant sur aucune hypothèse de production du signal. Par ailleurs, le modèle proposé étant capable de modéliser des interactions entre les bandes de fréquences, il est nécessaire de le tester en utilisant une analyse acoustique qui laisse place à la modélisation de ces interactions. De plus, la modélisation par HMM ne donne pas d'aussi bons résultats en utilisant la sortie du banc de filtre comme analyse acoustique que le système de référence HMM_{cep}. Lorsque le système de référence est appliqué directement à la sortie du banc de 24 filtres, le taux de reconnaissance est de alors 94.6%. Notons que les corrélations entre les canaux fréquentiels ne sont pas modélisées dans ce système, les matrices de covariance des gaussiennes associées aux états étant toujours diagonales. Nous pouvons donc espérer qu'un modèle plus complexe permette une meilleure modélisation en utilisant une représentation par banc de filtre de la parole. Soulignons que le modèle proposé ne modélise pas non plus la corrélation entre les observations dans les différents canaux fréquentiels mais prend plutôt en compte des dépendances entre canaux au niveau du processus caché. Nous espérons seulement que la modélisation de ces dépendances permet de compenser en partie l'absence de modélisation des corrélations.

5.2.2 Modélisation et décodage

Comparaison des stratégies de décodage

Les résultats présentés correspondent aux différentes stratégies de décodage appliquées aux modèles HMM_{cep} et HMM_{fbk} et à une modélisation par champ de Markov de la sortie du banc de filtre. Le modèle RFM-*sync*1 est défini comme un modèle à 24 bandes, chaque bande correspondant à un filtre, et l'heuristique d'apprentissage définie à la section 3.3.4 est utilisée pour l'estimation des paramètres. Rappelons que cette heuristique utilise un hyper-paramètre, γ, permettant de contrôler l'importance des potentiels verticaux par rapport aux potentiels horizontaux. Le tableau 5.3 présente les résultats obtenus en fonction de γ pour le modèle RFM-*sync*1 et pour les deux modèles HMM_{cep} et HMM_{fbk}, ce dernier correspondant au modèle de référence appliqué à la sortie du banc de filtre. La première ligne du tableau (ICM uniforme), correspond à une segmentation par l'algorithme ICM initialisé par une segmentation uniforme dans le calcul du champ caché x^* (3.23). La deuxième ligne correspond à une segmentation ICM initialisée par un décodage de Viterbi indépendant dans chaque bande. Enfin, la dernière ligne du tableau correspond à une segmentation basée sur un algorithme de recuit simulé. Rappelons que la formulation sous forme de distribution de Gibbs d'une chaîne de Markov permet l'utilisation des techniques de décodage développées pour les champs de Markov pour les deux modèles HMM_{cep} et HMM_{fbk}.

	HMM		RFM (γ)			
	cep	fbk	0.0	0.005	0.02	0.05
ICM uniforme	84.4	59.6	43.2	43.8	43.6	47.2
ICM Viterbi	99.8	94.6	69.0	69.6	70.0	69.2
Recuit simulé	99.6	91.8	78.4	–	–	–

TAB. 5.3 – *Taux de reconnaissance (en %) en fonction de l'hyper-paramètre γ et de l'algorithme de reconnaissance utilisé.*

La première conclusion à tirer de ces résultats est que, dans tous les cas, la modélisation par HMM donne de meilleurs résultats que les champs de Markov, même lorsqu'on utilise la sortie du banc de filtre comme représentation de la parole. L'écart de performance entre les deux approches HMM rappelle, si il est nécessaire, que la représentation cepstrale du signal est mieux adaptée aux HMM à matrices de covariances diagonales que le banc de filtre. Par contre, si on introduit dans le HMM une modélisation explicite de la corrélation entre les différentes bandes en utilisant des gaussiennes avec des matrices de covariance pleines, le taux de reconnaissance est alors de 100%. La modélisation de ces corrélations est donc indispensable à la modélisation de la parole dans le plan temps/fréquence.

Indépendemment de la qualité de la modélisation qui sera discutée par la suite, ces résultats permettent de comparer les stratégies de décodage proposées. La comparaison des deux premières lignes du tableau met en évidence la sensibilité de l'algorithme ICM aux conditions initiales. L'initialisation basée sur l'utilisation de l'algorithme de Viterbi indépendemment dans chaque bande s'avère efficace mais n'est pas très utilisable en pratique. En effet, en raison de l'apprentissage heuristique, on dispose dans cette expérience des HMM pour chacune des bandes considérées comme indépendantes ce qui rend possible l'initialisation de l'ICM par Viterbi. Lorsque les paramètres des RFM sont estimés directement à l'aide des algorithmes proposés précédemment, cette stratégie d'initialisation d'une solution ne sera plus possible à moins de disposer d'un HMM pour chaque bande, en plus du RFM. Bien que coûteuse en temps de calcul, l'utilisation d'un algorithme de recuit simulé semble une bonne solution. En effet, on remarque que, si l'algorithme de Viterbi donne de meilleurs résultats pour les HMM que le recuit simulé, les performances obtenues avec un recuit simulé sont, pour le modèle $RFM_{0.0}$, nettement meilleures par rapport à l'ICM.

Commentaires

Quelque soit la stratégie de décodage utilisée, les résultats obtenus en modélisant la sortie d'un banc de filtre par le modèle RFM-*sync*1 sont décevants. Cependant, plusieurs points méritent d'être étudiés dans ces résultats afin d'analyser et de comprendre les problèmes liés à cette approche.

D'une part, les taux de reconnaissance pour le modèle $RFM_{0.0}$, c'est à dire pour une modélisation indépendante asynchrone des sous-bandes, sont significativement moins bons que ceux obtenus avec une modélisation synchrone des bandes avec le modèle HMM_{cep}. Une explication possible de cette différence est que la représentation par banc de filtre du signal est très variable, en particulier pour une bande isolée puisqu'on travaille alors sur un sous-ensemble de l'espace acoustique. Pour une bande considérée indépendemment des autres, cela signifie que les estimations des paramètres des gaussiennes associées aux états sont mauvaises. Lorsqu'on travaille dans l'espace acoustique complet, en particulier lorsque toutes les bandes sont considérées simultanément, la représentation est alors un peu moins variable, ce qui explique les meilleures performances de l'approche HMM_{fbk}. Cette hypothèse est confirmée par la comparaison des variances obtenues pour les modèles HMM_{fbk} et $\mathrm{RFM}_{0.0}$. La comparaison bande par bande montre que, pour des moyennes du même ordre de grandeur dans les deux modèles, les variances obtenues avec le RFM sont soit plus petites soit beaucoup plus beaucoup plus grandes que pour le HMM. En effet, lorsqu'on réalise l'apprentissage sur une bande isolée, la séparation des données en états est difficile. Les paramètres de certains états sont alors estimés à partir de très peu ou de trop de données d'où les variances soit très faibles, soit très élevées.

D'autre part, la modélisation de la synchronie ne semble pas permettre de prendre en compenser efficacement l'absence de modélisation des corrélations entre les bandes. En effet, lorsqu'on accorde plus d'importance à la synchronie entre les bandes en augmentant la valeur de l'hyper-paramètre γ utilisé dans l'heuristique d'apprentissage, aucune différence significative des taux de reconnaissance ne sont'est observée. L'absence d'influence de γ pour un décodage basé sur l'ICM initialisé par Viterbi peut s'expliquer par le fait que la segmentation initiale pour l'ICM est optimale dans le cas où les bandes sont réellement indépendantes. L'algorithme ICM convergeant vers un optimum local, il est possible que la solution trouvée soit en fait proche de la solution initiale, quelque soit la valeur des poids de synchronisation. En revanche, cette analyse n'est plus vraie pour l'algorithme ICM appliqué à une segmentation initiale uniforme. En regardant les ordres de grandeurs des énergies mises en jeu dans l'algorithme ICM nous nous sommes rendus compte que l'énergie de synchronisation (i.e. l'énergie associée aux cliques verticales) est très faible, en particulier par rapport à l'énergie d'attache aux données ce qui explique en partie le problème rencontré.

Bien que la modélisation de la synchronie entre sous-bandes ne soit pas directement liée aux ocrrélations entre sous-bandes, il est cependant intéressant de regarder, pour un mot donné, la structure de la matrice des poids de transition et de la comparer aux matrices de covariance obtenues pour chaque état d'un HMM à matrice de covariance pleine. La figure 5.1 montre deux exemples de matrices de synchronisation obtenues pour $\gamma = 0.02$. Les matrices de poids de transitions montrent une structure présentant des pics sur la deuxième dia-

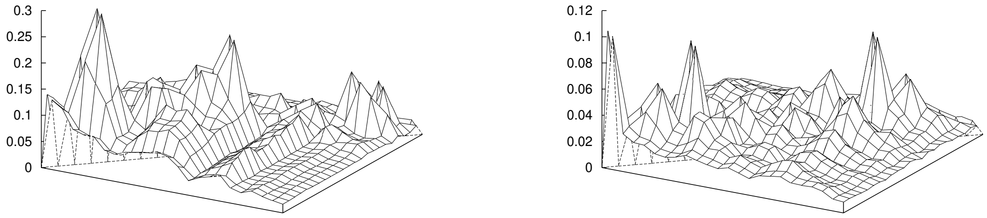


FIG. 5.1 – *Matrice des poids de synchronisation pour les mots "guide" et "cinema" avec* $\gamma = 0.02$.

gonale, indiquant que certaines bandes voisines sont plutôt synchronisées. Il est cependant à noter que les bandes voisines prises deux à deux ne sont pas systématiquement synchrones. Les matrices de covariance associées à chaque états du modèle HMM_{fbk} présentent une structure quasi diagonale pour certains états alors que pour d'autres, des "zones" de corrélation apparaissent clairement. Une synchronisation "globale" stationnaire, ne dépendant pas du temps ou plutôt des états, ne peut donc remplacer la modélisation des corrélations dépendantes des états. Nous donnons en annexe D quelques exemples de matrices de corrélations.

5.2.3 Hyper-paramètre de régularisation

Appliqué aux distributions de Gibbs

L'expérience précédente a montré que lorsque γ augmente, le nombre de changements d'états effectués par l'algorithme ICM appliqué après une segmentation par Viterbi augmente également mais que les variations d'énergies liées aux processus caché X restent faible par rapport aux énergies d'attache aux données. Cette constatation rejoint le problème de la mauvaise modélisation des durées dans les HMM, principalement due à la faible participation à la vraisemblance finale des probabilités de transitions.

Dans les problèmes de régularisation, il est courant d'utiliser un hyper-paramètre permettant d'accorder plus ou moins d'importance aux connaissances a priori dont on dispose (voir par exemple [97]). Dans le cadre de la modélisation de la parole par distribution de Gibbs, il est également possible d'ajouter à la définition du modèle un hyper-paramètre permettant d'accorder plus d'importance aux énergies *a priori* par rapport à l'attache aux données, en multipliant l'énergie *a priori* par un facteur β dans l'expression de l'énergie *a posteriori* . L'énergie *a posteriori* d'un champ $X = x$ connaissant l'observation $Y = y$ est alors donnée par

$$U(x|y) = \beta U(x) + U(y|x) \ , \tag{5.1}$$

où, dans le cas qui nous intéresse, les énergies *a priori* et *a posteriori* sont données respectivement par les équations (3.4) et (3.7).

La figure 5.2 montre les résultats obtenus pour différents paramètres de régularisation avec les modèles $\mathrm{RFM}_{0.0}$ et $\mathrm{RFM}_{0.02}$ pour un décodage basé sur l'algorithme ICM initialisé par Viterbi. Ces résultats montrent clairement que lorsque la variabilité de l'observation est grande, la régularisation de la segmentation par un *a priori* permet d'augmenter les taux de reconnaissance. En effet, lorsque les bandes sont asynchrones, le taux de reconnaissance passe de 69 % pour $\beta = 0$ à 81.6 % pour $\beta = 16$. Lorsqu'on ajoute une modélisation de la synchronie entre les bandes, l'augmentation des performances est moins importante mais demeure significative.

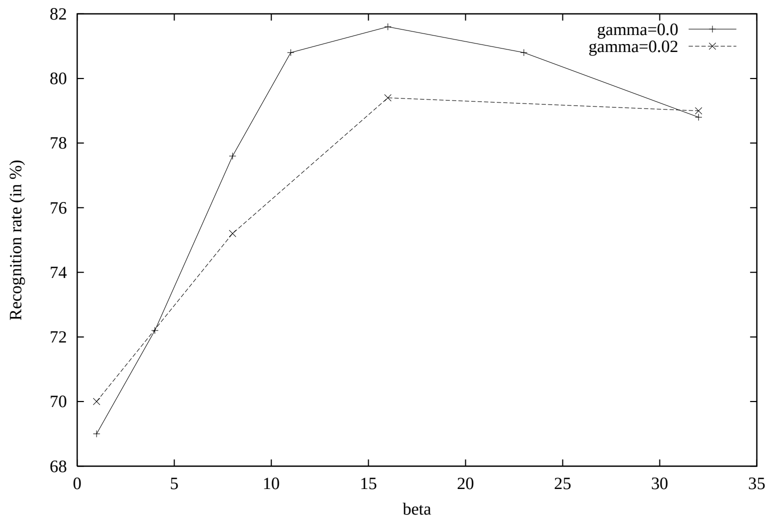


FIG. 5.2 – *Taux de reconnaissance en fonction de β pour $\gamma = 0$ (trait continu) et pour $\gamma = 0.02$ (trait pointillé).*

Appliqué aux chaînes de Markov cachés

Dans le cadre de la modélisation par modèle de Markov caché, l'hyper-paramètre de régularisation peut-être vu comme l'équivalent acoustique du facteur multiplicatif, ou *fudge factor*, généralement appliqué au modèle de langage [33]. Ce facteur permet d'équilibrer les scores acoustique et linguistique dans les algorithmes de recherche et est rendu nécessaire par le mélange de probabilités et de vraisemblances intervenant dans ce score. On retrouve l'utilisation conjointe des probabilités et des vraisemblances dans le calcul du score acoustique et nous nous proposons d'introduire un facteur de régularisation au niveau des probabilités de transition. L'équation de décodage (1.7) devient alors

$$p(y|W=w) = \max_{x \in \Omega} P[X=x|W=w]^{\beta}\, p(Y=y|X=x, W=w) \ . \qquad (5.2)$$

Nous avons testé la régularisation appliquée aux HMM en reconnaissance de la parole téléphonique continue, sur une application de reconnaissance de noms propres prononcés et épelés. Cette application utilise un décodage acoustico-phonétique pour la reconnaissance de la prononciation, tandis que la reconnaissance de l'épélation est basée sur des modèles de lettre. La suite de phonèmes et de lettre reconnue est ensuite utilisée pour une recherche lexicale permettant de trouver le nom le plus proche. Une description plus complète de cette application peut-être trouvée dans [43, 23] et nous nous contenterons de regarder les performances de la phase de décodage sans réaliser de recherche lexicale. Notons quand même que les tests portent sur environ 22 000 occurrences de phones et 25 000 de lettres, les intervalles de confiance définis en introduction de ce chapitre ne s'appliquant donc pas ici.

Le tableau 5.4 donne les taux de reconnaissance et la précision[1] pour les phonèmes et les lettres pour plusieurs valeurs du paramètre de régularisation. Pour comparaison, nous indiquons dans la dernière colonne les résultats obtenus sans régularisation mais en ajoutant une pénalité fixe de transition entre modèle. En effet, la précision sur les phonèmes obtenus sans régularisation montre que le taux d'insertions est très élevé et l'utilisation d'une pénalité fixe de transition est une solution généralement utilisée pour diminuer le nombre d'insertions.

β	1.0	8.0	12.0	p=-17.5
Phones	62.8/4.57	57.3/32.3	52.0/32.0	55.4/34.4
Lettres	82.3/73.78	83.2/81.6	81.1/79.6	82.8/78.4

TAB. 5.4 – *Taux de reconnaissance/précision en reconnaissance des phones et des lettres en fonction de β dans une modélisation par HMM.*

Les résultats montrent clairement que le paramètre de régularisation permet également de diminuer le nombre d'insertion dans le décodage acoustico-phonétique. La même remarque s'applique également aux lettres, même si le nombre d'insertions sans régularisation est déjà faible en raison de l'utilisation d'un silence court entre les mots [89]. Les performances obtenues pour $\beta = 8$ sont légèrement meilleures que celles obtenues en utilisant la pénalité fixe. La réduction du nombre d'insertions obtenue grâce à la régularisation laisse penser qu'en accordant plus d'importance à la chaîne de Markov en reconnaissance de la parole continue, on obtient une meilleure modélisation de la durée des évènements acoustiques.

1. Nous rappelons que la précision correspond au taux de reconnaissance moins le taux d'insertion.

5.2.4 Discussion

Plusieurs enseignements sont à tirer de l'ensemble de ces résultats. La comparaison des stratégies de décodage montre que la méthode basée sur le recuit simulé donne de bonnes performances, comparables avec l'algorithme de Viterbi dans le cas des HMM, et est parfaitement adapté aux approches multi-bandes.

Par ailleurs, ces expériences ont mis en évidence la faible importance de la synchronisation. La comparaison avec les matrices de covariance d'un HMM montre aussi qu'un modèle stationnaire de synchronisation n'est pas réaliste. Même si on est en droit d'attendre de meilleurs taux de reconnaissance, en utilisant, par exemple, un décodage à base de recuit simulé en conjonction avec un paramètre de régularisation, ces expériences montrent la difficulté de la modélisation directe de la parole dans le plan temps/fréquence. Cette modélisation s'appliquant à une observation d'une grande variabilité, il est important d'utiliser un fort *a priori* lors du décodage. L'apprentissage heuristique ne permettant pas d'équilibrer directement la contribution de l'*a priori* par rapport à l'observation, il est nécessaire d'utiliser alors un paramètre explicite de régularisation. En revanche, en utilisant une estimation directe des paramètres des potentiels *a priori* et *a posteriori* , on peut espérer que les contributions respectives des deux énergies s'équilibrent. De plus, dans le formalisme des champs de Markov, toutes les quantités participant au calcul du score acoustique sont des probabilités, réduisant ainsi l'intérêt d'une régularisation explicite.

5.3 Application du formalisme des champs aux chaînes

Nous nous proposons maintenant d'étudier le formalisme champ Markovien défini dans les chapitres précédents à la modélisation par chaînes de Markov cachés. En particulier, après l'étude "théorique" présentée au chapitre 4, nous nous intéressons aux performances des différents algorithmes d'initialisation et d'estimation des paramètres dans le cas de données réelles. Afin de comparer les méthodes proposées avec les techniques HMM standards appliquées au modèle de référence HMM_{cep}, cette étude est menée dans le cas mono-bande avec une représentation cepstrale de la parole.

Le tableau 5.5 donne les taux de reconnaissance obtenus pour plusieurs procédures d'estimation des paramètres des modèles et pour plusieurs stratégies de décodage.

Toutes les approches étudiées donnent des résultats similaires. Du fait de la simplicité de la tâche de reconnaissance étudiée, les trois algorithmes d'initialisation utilisés seuls (Viterbi, ICM et recuit simulé (RS)) donnent de bonnes performances et les algorithmes de réestimation des paramètres ne sont pas vraiment nécessaires. On peut cependant vérifier que l'algorithme EM généralisé appliqué sans initialisation donne un taux de reconnaissance comparable, bien qu'un peu plus faible, aux autres méthodes. En effet, le taux de reconnaissance est alors de 81.8% avec un décodeur ICM et de 99.6% en utilisant un décodage basé sur le recuit simulé. L'utilisation directe de l'algorithme de

Apprentissage	ICM	RS	Viterbi
Viterbi	84.8	96.8	99.8
Viterbi + BW	85.6	99.6	100.0
ICM	87.2	99.6	–
ICM + GEM	88.6	–	–
RS	88.0	99.2	–
RS + GEM	89.0	99.0	–

TAB. 5.5 – *Taux de reconnaissance en fonction des algorithmes d'apprentissage et de décodage.*

Baum-Welch donne un taux de reconnaissance comparable puisque l'on obtient 80.2% en utilisant un décodage à base d'ICM.

5.4 Application à la modélisation multi-bande

Finalement, nous appliquons le formalisme proposé à la modélisation multi-bandes en utilisant, contrairement aux expériences présentées dans la section 5.2, une représentation cepstrale de la parole dans chaque bande. La bande passante totale, $[0, 4000]$ Hz, est divisée en sous-bandes régulièrement réparties sur une échelle MEL. Les coefficients cepstraux dans chaque bande sont calculés en appliquant une transformée de Fourier discrète inverse des coefficients spectraux, calculés par transformée de Fourier discrète, correspondant à la bande considérée. Cette analyse cepstrale est légèrement différente de la technique classique qui consiste à utiliser la transformée en cosinus de la sortie du banc de filtre. Le détail du calcul de cette représentation cepstrale est donné en annexe E.

5.4.1 Cas de la parole propre

Plusieurs divisions en sous-bandes de la bande passante totale ont été expérimentées et les résultats obtenus pour différents modèles sont donnés au tableau 5.6. La première ligne correspond à un RFM dont les paramètres sont appris en utilisant l'heuristique, les paramètres de synchronisation entre bandes n'étant pas estimés (*i.e.* $\gamma = 0$). Dans ce cas, le décodage se fait par l'algorithme ICM initialisé par une segmentation obtenue avec Viterbi. La deuxième ligne donne les résultats pour un modèle dont les paramètres sont initialisés par l'algorithme ICM. La troisième ligne correspond aux modèles précédents après réestimation des paramètres par l'algorithme EM. Dans les deux dernières expériences, les résultats sont donnés pour un décodage par ICM. Dans les noms des expériences, le chiffre suivant la lettre b indique le nombre de bandes tandis que celui suivant la lettre c correspond au nombre de coefficients cepstraux utilisés pour représenter le signal dans chaque bande.

Les résultats présentés dans le tableau 5.6 sont à considérer avec précaution dans la mesure où les différences entre les meilleurs et les moins bons taux de reconnaissance, pour un type d'apprentissage donné, ne sont pas forcément

	b1c12	b3c5	b5c3	b7c2
heuristique	99.8	99.4	97.6	95.0
ICM	87.2	84.6	78.8	78.2
ICM-GEM	88.6	80.6	75.4	76.0

TAB. 5.6 – *Taux de reconnaissance (en %) pour différentes architectures de sous-bandes.*

significatives. Cependant, on observe que quelque soit la méthode d'apprentissage utilisée, les performances diminuent lorsque le nombre de bande augmente. Cette dégradation des performances n'est pas uniquement due au fait que chaque bande est moins bien représentée puisque dans la configuration b7c2, lorsqu'on augmente le nombre de coefficients cepstraux dans chacune des 7 sous-bandes, le taux de reconnaissance augmente mais de manière marginale. Ainsi pour l'apprentissage heuristique, le taux de reconnaissance passe de 95% avec deux coefficients cepstraux par bande à 96% pour trois coefficients et 96.4% pour cinq. La même remarque s'applique aux deux autres méthodes d'apprentissage testées. Pour un décodage basé sur le recuit simulé, cette tendance est encore plus nette puisque le taux de reconnaissance passe de 97.8% avec 3 bandes à 92.6% avec 5 bandes dans le cas d'un apprentissage ICM.

Dans le cas d'un apprentissage heuristique, l'ajout d'interactions entre les bandes pour différentes valeurs de γ détériore les performances. En revanche, bien que les chiffres ne soient pas significatifs, il semblerait que pour des paramètres estimés par l'algorithme EM généralisé, la modélisation de la synchronisation entre les bandes ait une certaine influence. En effet, les résultats sont moins bons pour ICM-GEM par rapport à ICM seul mais il semblerait que l'écart de performance se réduise lorsque le nombre de bande augmente.

Enfin, la régularisation a été testée avec l'apprentissage heuristique mais aucune amélioration significative du taux de reconnaissance n'a été relevée. Dans le cas b5c3, ce dernier est de 98.2% pour $\beta = 8$. Des résultats similaires ont été obtenus pour les deux autres algorithmes d'apprentissage. Dans le cas d'une estimation ICM-GEM, l'ajout d'un facteur de régularisation dégrade les performances tandis que pour l'apprentissage ICM, on observe une augmentation marginale. Cette constatation montre que la synchronisation n'est pas un *a priori* pertinent pour la modélisation multi-bandes.

Bien que n'étant pas comparables, ces résultats sont à rapprocher de ceux donnés dans le tableau 1.1 où des performances meilleures que celles de la bande totale sont obtenues pour deux bandes. Dans l'expérience reportée dans le tableau 1.1, les scores des deux sous-bandes sont recombinés à l'aide d'un perceptron multi-couche, alors que aucune recombinaison de scores partiels n'est effectuée dans l'approche champs de Markov. Cette remarque montre que l'étape de recombinaison des scores est capitale dans les HMM multi-bandes.

RSB (dB)	∞	30	20	10
bande #1	96.4	9.0	9.8	10.0
bande #2	94.4	17.4	17.4	14.2
bande #3	92.2	80.4	59.8	28.4
moyenne	99.6	28.0	13.2	10.4
icm-gem/sa	93.6	49.4	37.8	24.8

TAB. 5.7 – *Taux de reconnaissance (en %) pour différents rapports signal/bruit dans une architecture à trois sous-bandes.*

5.4.2 Cas de la parole bruitée

L'architecture en 3 sous-bandes a été utilisée pour mener une première étude de la robustesse du modèle $RFM - sync1$ par rapport à un bruit additif. Dans ce but, un bruit additif coloré a été ajouté aux données de test avec différrents rapports signal/bruit (RSB). La coloration utilisée pour le bruit concentre principalement l'énergie dans les basses fréquences en filtrant un bruit blanc par un filre dont la fonction de transfert est donnée figure 5.3.

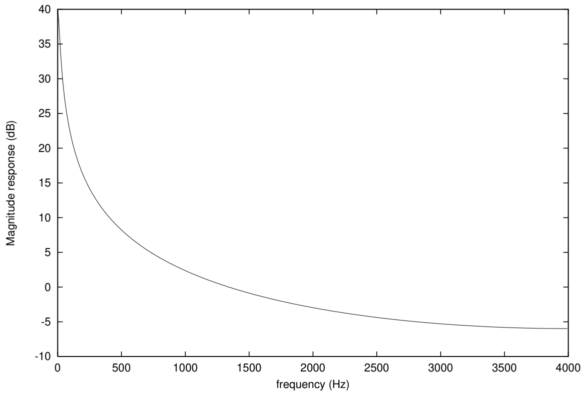


FIG. 5.3 – *Coloration spectrale du bruit additif ajouté aux données de test.*

Pour chacune des sous-bandes, les paramètres d'un HMM ont été estimés par l'algorithme de Baum-Welch et les trois premières lignes du tableau 5.7 montrent l'évolution du taux de reconnaissance en fonction du RSB dans chacune des bandes. On note une baisse rapide dans les deux bandes basses fréquences tandis que la bande haute fréquence montre une décroissance plus lente du taux de reconnaissance, cette dernière bande étant moins affectée par le bruit que les deux autres. La quatrième ligne montre les résultats obtenus avec un modèle multi-bande classique en utilisant la moyenne comme règle de fusion des scores partiels tandis que la dernière ligne montre les résultats obtenus avec une modélisation par champs de Markov.

Ces résultats montre que la modélisation de la synchronie entre les bandes permet d'augmenter la robustesse aux bruits additifs par rapport à une règle simple de fusion des scores partiels dans une approche multi-bande conventionnelle. Cependant, le seul décodage dans la bande faiblement bruité donne toujours de meilleures performances ce qui laisse penser qu'une meilleure stratégie de fusion des scores partiels devrait permettre d'améliorer les performances par rapport à la moyenne. Malgrè cela, le comportement observé du modèle $RFM - sync1$ en présence de bruit additif est encourageant.

5.5 Discussion

Dans ce chapitre, nous avons appliqué le modèle RFM-*sync*1 sur une tâche très simple de reconnaissance mono-locuteur de mots isolés. Les résultats ne permettent donc pas de conclure sur les avantages de la modélisation par champs de Markov par rapport aux approches traditionnelles utilisant les HMM et il sera nécessaire d'étudier le modèle proposé sur une tâche plus complexe, comme par exemple la reconnaissance en mode indépendant du locuteur et/ou en environnement bruité.

Cependant, les expériences permettent de situer l'approche champs de Markov et de valider en situation réelle les algorithmes associés que nous avons proposés. Nous avons montré que pour une approche mono-bande basée sur la représentation cepstrale du signal, les algorithmes d'estimation des paramètres donnent des résultats similaires au système de référence lorsqu'on utilise un décodage basé sur le recuit simulé. L'algorithme ICM souffre du problème de la segmentation initiale et ne donne des performances acceptables que si l'on est capable de donner une bonne approximation de la segmentation optimale. Dans ce cas, l'utilisation d'une segmentation basée sur l'algorithme ICM donne de bonnes performances pour des temps de calcul largement inférieurs à ceux du recuit simulé. De plus, le recuit simulé présente l'inconvénient d'être fortement dépendant du choix de la température initiale utilisée.

L'approche multi-bande a montré que plus le nombre de bandes est élevé, plus les taux de reconnaissance sont faibles, malgré la modélisation des interactions inter-bande. Le modèle de synchronie permet cependant d'accroître la robustesse au bruit additif par rapport à une approche multi-bande classique dans le domaine cepstral. Les résultats obtenus en modélisant directement la sortie d'un banc de 24 filtres sont également décevants. Cette dernière expérience a montré l'intérêt d'une régularisation de la solution lorsque les observations présentent une grande variabilité. La régularisation suppose que l'on dispose d'un bon modèle *a priori* mais les dernières expériences présentées montrent que l'*a priori* ne permet de compenser la variabilité introduite par la division de la bande totale en sous-bandes. Une des faiblesses du modèle *a priori* utilisé dans cette approche réside dans la stationnarité des poids de synchronisation. En effet, les matrices de covariances obtenues avec le modèle HMM_{fbk} montrent clairement que les corrélations inter-bandes ne sont pas stationnaires. Sur des segments longs, comme ceux utilisés dans ces expériences, il est évident que l'hypothèse de stationnarité de la synchronisation n'est pas vérifiée. Enfin, on est en droit de se demander si la synchronisation des bandes est un choix pertinent. En effet, les travaux de Greenberg [47] montrent que l'oreille humaine est relativement insensible à l'asynchronie spectrale entre canaux fréquentiels. Cependant, nos modèles sont loins de refléter le comportement de l'oreille humaine et nous pensons que la notion de synchronie peut s'avérer utile pour de la parole bruitée ou encore en présence de réverbérations.

En résumé, ces expériences montrent que les algorithmes d'estimation des paramètres et de décodage étudiés permettent la modélisation de la parole à l'aide de distribution de Gibbs. L'intérêt d'un bon modèle *a priori* du processus caché a également été mis en évidence. Le modèle $RFM - sync1$ proposé présente de nombreux défauts qui limitent ses performances et demande à être développé plus en détail

Chapitre 6

Conclusions et perspectives

Dans ce document, nous avons proposé un nouveau cadre théorique pour la modélisation de la parole dans le plan temps/fréquence à l'aide de champs de Markov. Nous proposons une généralisation de l'algorithme EM pour l'estimation au sens du maximum de vraisemblance des paramètres d'une distribution de Gibbs et une stratégie de décodage basée sur l'algorithme de recuit simulé, ces deux algorithmes étant appliqués avec succès aux chaînes de Markov cachées. Nous illustrons cette nouvelle approche en appliquant à la reconnaissance mono-locuteur de mots isolés un modèle multi-bande, dans lequel un terme de synchronisation est introduit au niveau du processus caché.

Si l'on considère le problème de la reconnaissance de la parole comme un problème de segmentation et de classification simultanées, les expériences effectuées avec le modèle de champs de Markov proposé montrent que plus l'observation est variable, plus il est important de disposer d'un bon modèle *a priori* de la segmentation. En particulier, les expériences menées sur la modélisation de la parole représentée par la sortie d'un banc de filtres montrent qu'en accordant plus d'importance au processus caché (qui joue alors le rôle d'*a priori*), les taux de reconnaissance augmentent. A l'inverse, la modélisation par modèles de Markov cachés repose sur un *a priori* très faible et il est alors nécessaire d'utiliser une représentation de la parole la moins variable possible.

L'amélioration des systèmes de reconnaissance de la parole passe donc par la recherche d'un bon compromis entre une représentation la moins variable possible et une bonne modélisation *a priori* du processus. La recherche d'une représentation stable, idéalement invariante, de la parole, présente le défaut de ne pas utiliser une partie de l'information présente dans le signal. En effet, les indices acoustiques trop variables sont rejetés pour obtenir la représentation la plus stable possible. Ainsi, les analyses acoustiques reposent souvent sur une hypothèse source-filtre de production de la parole et les paramètres liés à la source ne sont pas considérés car trop variables. Par ailleurs, le problème de la définition d'un bon *a priori* sur le processus modélisé n'est pas évident et nécessite de disposer, d'une idée des indices pertinent pour la modélisation et d'un bon modèle stochastique utilisant ces indices, ainsi que des algorithmes d'estimation des paramètres et de reconnaissance de la parole liés à ce modèle.

Les champs de Markov appliqués à la reconnaissance de la parole offrent une alternative intéressante pour une meilleure modélisation du processus. Le modèle de champs aléatoires que nous avons étudié dans cette thèse va évidemment dans le sens d'une meilleure modélisation *a priori* . Cependant, tous les problèmes liés à cette approche sont loin d'être résolus et de nombreuses études restent encore à faire avant de disposer d'un modèle de champs de Markov efficace en modélisation de la parole. Parmi les problèmes à traiter, il est possible de distinguer trois grands axes de recherche distincts.

Le premier problème à traiter consiste à améliorer la modélisation des interactions entre les sous-bandes. En effet, on a vu qu'un des points faibles du modèle RFM-*sync*1 réside dans la modélisation stationnaire de la synchronisation des bandes. Une première amélioration possible du modèle consiste à rendre les paramètres de synchronisation non-stationnaire en introduisant, par exemple, des paramètres de synchronisation dépendant des états dans chacune des bandes. On peut alors imaginer de remplacer les coefficients $f_{k,l}$ par $f_{k,l}(i,j)$, ces derniers coefficients contrôlant la fréquence des configurations pour lesquelles les états i et j sont observés respectivement dans les bandes k et l au même instant. Il est aussi possible d'envisager de modéliser un autre type d'interactions que la synchronisation ou bien de modéliser la synchronisation d'une autre manière. En particulier, il nous semble intéressant de supprimer la contrainte du nombre d'états similaire dans les différentes bandes, les bandes haute fréquences nécessitant *a priori* plus d'état que les bandes basse fréquence. Une autre approche possible consiste à utiliser les distributions de Gibbs dans les modèles de segments, en fixant dans un premier temps la durée du segment. Cette dernière approche présente également l'avantage de simplifier la forme du processus caché rendant alors plus simple la modélisation *a priori* de ce dernier.

La représentation temps/fréquence du signal nous parait être le deuxième point important à étudier. Nous avons travaillé sur une représentation cepstrale par sous-bande et sur la modélisation directe de la sortie d'un banc de filtre. La première approche suppose une division relativement grossière de la bande totale tandis que la deuxième présente une trop grande variabilité. Dans l'optique de modéliser la parole vue comme une image, le spectre de modulation [46] semble être une bonne solution dans la mesure où cette représentation contient beaucoup d'information et est relativement peu variable. La modélisation directe dans le plan temps/fréquence reste évidemment possible mais nécessite l'étude de représentations plus normalisées que la sortie du banc de filtre. On pourrait envisager, par exemple, de travailler sur le re-échantillonnage de diverses représentation temps/fréquence (ou temps/échelle) et sur la normalisation de la représentation obtenue avant modélisation de l'image à l'aide de distributions de Gibbs.

Enfin, les algorithmes de décodage proposés, et plus particulièrement le recuit simulé, souffrent d'un problème de temps de calcul élevé et il est indispensable de travailler sur l'optimisation et l'amélioration de ces temps de calcul. Au niveau du processus caché, on utilise traditionnellement en parole un modèle gauche-droite afin de modéliser l'évolution temporelle du signal. Lorsqu'on a recours à des énergies barrière dans l'approche champs de Markov, les algorithmes travaillent sur l'espace d'état complet même si la plupart des configurations de cette espace sont inadmissibles comme solution. Dès lors, il s'avère particulièrement intéressant de travailler dans un espace d'état contraint aux configurations admissibles afin d'optimiser le nombre de calcul en guidant la recherche d'une solution.

Finalement, nous avons présenté une modélisation par champs Markoviens de la parole appliquée à la reconnaissance de mots isolés. Afin de pouvoir être utilisées dans des situations pratiques, ces techniques devront être étendues à la reconnaissance de mots connectés et de parole continue. Le problème de segmentation devient alors plus complexe que dans le cas de mots isolés. Dans l'état actuel des travaux, il est possible d'utiliser les champs de Markov pour le *rescoring* des N meilleures hypothèses fournies par un système classique de reconnaissance de la parole. En revanche, pour l'application directe de la modélisation par champs de Markov, deux problèmes sont à résoudre. D'une part, il est à craindre que si l'on part d'une segmentation trop éloignée de la solution optimale, le recuit simulé soit long à converger et qu'il soit difficile de trouver une température initiale adaptée au problème à résoudre. D'autre part, il est nécessaire d'assurer une segmentation cohérente dans chacune des bandes ou de redéfinir la notion de frontière de segment.

Bibliographie

[1] Dario Albesano, Franco Mana, and Roberto Gemello. Continuous speech recognition with neural networks: an application to railway timetables enquires. In A. Esposito G. Chollet, M Di Benedetto and M. Marinaro, editors, *Speech Processing, Recognition and Artificial Neural Networks*, pages 216–220. Springer Verlag, 1998.

[2] J. B. Allen. How do humans process and recognize speech? *IEEE Trans. on Speech and Audio Processing*, 2(4):567–576, October 1994.

[3] R. Auckenthaler and J. S. Mason. Score normalisation in a multi-band speaker verification system. In *Speaker Recognition and its Commercial and Forensic Applications (RLA2C)*, pages 102–105, 1998.

[4] L. R. Bahl et al. The IBM large vocabulary continuous speech recognition system for the ARPA NAB News task. In *ARPA SLT Workshop*, pages 121–126, 1995.

[5] L. R. Bahl and F. Jelinek. Decoding for channels with insertions, deletions and subtsitutions with applications to speech recognition. *IEEE Trans. on Information Theory*, 21:404–411, July 1975.

[6] J. K. Baker. *Stochastic modeling for automatic speech understanding.* R. Reddy ed., New York Academic Press, 1975.

[7] R. Bakis. Continuous speech recognition via centisecond acoustic states. In *Proc. ASA Meeting*, April 1976.

[8] L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process. *Inequalities*, 3:1–8, 1972.

[9] B. Benmiloud and W. Pieczynski. Estimation de paramètres dans les chînes de markov cachées et segmentation d'images. *Traitement du signal*, 12(5):433–454, 1995.

[10] Laurent Besacier. *Un modèle parallèle pour la reconnaissance automatique du locuteur.* PhD thesis, Université d'Avignon et du pays de Vaucluse, 1998.

[11] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Royal Statistical Soc.*, B-48:192–236, 1974.

[12] J. Besag. On the statistical analysis of dirty pictures. *J. Royal Statistical Soc.*, 48(3):259–302, 1986.

[13] H. Bourlard. Reconnaissance de la parole: modélisation ou description? In *XXIes Journée d'Etude sur la Parole*, pages 263–272, 1996.

[14] H. Bourlard, S. Dupont, and C. Ris. Multi-stream speech recognition. Research Report RR 96-07, IDIAP, Dec. 1996.

[15] H. Bourlard et al. Towards subband-based speech recognition. In *EUSIPCO*, 1996.

[16] H. Bourlard and C. J. Wellekens. Connected speech recognition by phonemic semi-Markov chains for state occupancy modelling. In I. T. Younf et al., editors, *Signal Processing III: Theories and Applications, EUSIPCO*, pages 511–514. Elsevier Science Publishers, 1986.

[17] P. Cardin et al. Crim's spontaneous speech recognition system for the ATIS task. In *Intl. Conf. Speech and Language Processing*, 1992.

[18] G. Celeux and J. Diebolt. L'algorithme SEM: un algorithme d'apprentissage probabiliste pour la reconnaissance de mélanges de densités. *Revue de Statistique Appliquée*, 34(2), 1986.

[19] C. Cerisara, J.-P. Haton, J.-F. Mari, and D. Fohr. Multi-band continuous speech recognition. In *Eurospeech*, 1997.

[20] Christophe Cerisara. Dealing with loss of synchronism in multi-band continuous speech recognition systems. In K. Ponting, editor, *NATO ASI Computational Models for Speech Pattern Processing*, pages 90–95. Springer Verlag, 1997.

[21] Bernard Chalmond. An iterative Gibbsian technique for reconstruction of m-ary images. *Pattern Recognition*, 22(6):747–761, 1989.

[22] D. Chandler. *Introduction to Modern Statistical Mechanics*. Oxford University Press, 1987.

[23] G. Chollet, J. Černocký, G. Gravier, J. Hennebert, D. Petrovska-Delacrétaz, and F. Yvon. Towards fully automatic speech processing techniques for interactive voice servers. In G. Chollet, M. Di Benedetto, A. Esposito, and M. Marinaro, editors, *Speech Processing, Recognition and Artificial Neural Networks. Proceedings of the 3rd International School on Neural Nets "Eduardo R. Caianiello"*. Springer-Verlag, 1998. ISBN 1-85233-094-5.

[24] Gérard Chollet, Jean-Luc Cochard, Andrei Constantinescu, Cédric Jaboulet, and Philippe Langlais. Swiss French PolyPhone and PolyVar:

telephone speech databases to model inter- and intra-speaker variability. Research Report RR 96-01, IDIAP, April 1996.

[25] A. P. Dempster, N. M. Laid, and D. B. Durbin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Soc.*, 39(39):1–38, 1977.

[26] L. Deng. A stochastic model of speech incorporating hierarchical nonstationarity. *IEEE Trans. on Speech and Audio Processing*, 1(4):471–474, 1993.

[27] L. Deng, M. Aksmanovic, D. Sun, and J. Wu. Speech recognition using hidden Markov models with polynomial regression functions as nonstationary states. *IEEE Trans. on Speech and Audio Processing*, 2(4):507–520, 1994.

[28] Xavier Descombes. *Champs Markoviens en analyse d'images*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Décembre 1993.

[29] Xavier Descombes et al. Estimation of Markov random field prior parameters using Markov chain Monte-Carlo maximum likelihood. Technical report, INRIA RR-3015, Oct. 1996.

[30] V. Digalakis, M. Ostendorf, and J. R. Rohlicek. Improvements in the stochastic segment model for phoneme recognition. In *DARPA Workshop on Speech and Natural Language*, 1989.

[31] V. Digalakis, J. R. Rohlicek, and M. Ostendorf. A dynamical system approach to continuous speech recognition. In *DARPA Workshop on Speech and Natural Language Processing*, February 1991.

[32] J. A. du Preez. Efficient training of high-order Markov models using first-order representations. *Computer Speech and Language*, 12:23–39, 1998.

[33] M. Federico and R. De Mori. Language models. In *Spoken Dialogues with Computers*, chapter 7. Academic Press, 1998.

[34] S. Furui. Speaker independent isolated word recognizer using dynamic features of speech spectrum. *IEEE Trans. on Acoust., Speech, Signal Processing*, 34(1):52–59, Feb. 1986.

[35] Sadaoki Furui. An overview of speaker recognition technology. In *Automatic Speech and Speaker Recognition: Advanced Topics*, chapter 2. Kluwer Academic Publishers, 1996.

[36] M. J. F. Gales and S. J. Young. Hmm recognition in noise using Parallel Model Combination. In *Proc. Eurospeech*, pages 837–840, 1993.

[37] Jean-Luc Gauvain and Chi-Hui Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans. on Speech and Audio Processing*, 2(2), April 1994.

[38] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE trans. on PAMI*, 6(6):721–741, 1984.

[39] D. Genoud, G. Gravier, F. Bimbot, and G. Chollet. Combinig methods to improve speaker verification decision. In *Intl. Conf. on Speech and Language Processing*, volume 3, pages 1756–1759, 1996.

[40] Hans-Otto Georgii. *Gibbs measures and phase transitions*, volume 9 of *Studies in Mathematics*. de Gruyter, 1988.

[41] Zoubin Ghahramani and Michael I. Jordan. Factorial hidden Markov models. Technical report, Computational Cognitive Science Technical Report 9502, July 1996.

[42] O. Ghitza and M. M. Sondhi. Hidden Markov models with templates as non-stationary states: an application to speech recognition. *Computer Speech and Language*, 2:101–119, 1993.

[43] G. Gravier, G. Etorre, F. Yvon, and G. Chollet. Directory name retrieval using HMM modeling and robust lexical access. In *Workshop on Automatic Speech Recognition and Understanding*, 1997.

[44] G. Gravier, M. Sigelle, and G. Chollet. Markov random field modeling for speech recognition. *Australian Journal for Intelligent Information Processing Systems*, 5(4), 1998.

[45] G. Gravier, M. Sigelle, and G. Chollet. Toward Markov random field modeling of speech. In *Intl. Conf. on Spoken Language Processing*, December 1998.

[46] S. Greenberg and B. Kingsbury. The modulation spectrogram: In pursuit of an invariant representation of speech. In *IEEE Intl. Conf. on ASSP*, pages 1647– 1650, 1997.

[47] Steven Greenberg and Takayuki Arai. Speech intelligibility is highly tolerant of cross-channel spectral asynchrony. In *Joint proceedings of the Acoustical Society of America and the International Congress on Acoustics*, Seattle, 1998.

[48] Jean-Paul Haton. Neural networks for automatic speech recognition: a review. In A. Esposito G. Chollet, M Di Benedetto and M. Marinaro, editors, *Speech Processing, Recognition and Artificial Neural Networks*, pages 259–280. Springer Verlag, 1998.

[49] H. Hermansky, M. Pavel, and S. Tibrewala. Towards ASR using partially corrupted speech. In *Int. Conf. on Spoken Language Processing*, pages 458–461, Oct. 1996.

[50] M. Homayounpour and G. Chollet. A comparison of some relevant parametric representations for speaker verification. In *ESCA Workshop on Speaker Recognition, Identification and Verification*, pages 185–188, 1994.

[51] Qiang Huo and Chorkin Chan. Contextual vector quantization for speech recognition with discrete Hidden Markov Model. *Pattern recognition*, 28:513–517, 1995.

[52] Qiang Huo and Chorkin Chan. On the use of bi-directional contextual dependence in acsoutic modeling for speech recognition. In *ICASSP*, 1995.

[53] Qiang Huo and Chorkin Chan. A study on the use of bi-directional contextual dependence in Markov random field-based acoustic modelling for speech recognition. *Computer Speech and Language*, 10:95–105, 1996.

[54] E. Ising. Beitrag zur theorie des ferromagnetisms. *Zeitschrift fur Physik*, 31:253–258, 1925.

[55] F. Jelinek. Continuous speech recognition by statistical methods. *IEEE Proceedings*, 64:532–556, April 1976.

[56] Pierre Jourlin. *Approche bimodale du traitement automatique de la parole : Application à la reconnaissance du message et du locuteur.* PhD thesis, Université d'Avignon et du pays de Vaucluse, 1998.

[57] Denis Jouvet. *Reconnaissance de mots connectés indépendamment du locuteur par des méthodes statistiques.* PhD thesis, Ecole Nationale Superieure des Télécommunications, 1988.

[58] B.-H. Juang, W. Chou, and C.-H. Lee. Statistical and discriminative methods for speech recognition. In *Automatic Speech and Speaker Recognition - Advanced Topics*, chapter 5, pages 109–132. Kluwer Academic Publishers, 1996.

[59] B.-H. Juang and L. R. Rabiner. Mixture autoregressive hidden Markov models for speech signals. *IEEE Trans. on Acoust., Speech, Signal Processing*, 23(6):1404–1413, December 1985.

[60] A. Kannan and M. Ostendorf. A comparison of trajectory and mixture modeling in segment-based word recognition. In *Proc. IEEE Intl. Conf. Acoust., Speech, Signal Processing*, volume 2, pages 327–330, April 1993.

[61] P. Kenny, M. Lennig, and P. Mermelstein. A linear predictive HMM for vector-valued observations with applications to speech recognition. *IEEE Trans. on Acoust., Speech, Signal Processing*, 38(2):220–225, 1990.

[62] M. Khoumri. Estimation d'hyper-paramètres pour la déconvolution d'images satellitaires. Rapport de stage dea, INRIA Sophia, 1997.

[63] Nam Soo Kim and Chong Kwan Un. Frame-correlated Hidden Markov Model based on extended logarithmic pool. *IEEE Trans. on Speech and Audio Processing*, 5(2):149–160, March 1997.

[64] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[65] S. Lakshmanan and H. Derin. Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(8):799–813, 1989.

[66] Kenneth Lange. A gradient algorithm locally equivalent to the EM algorithm. *J. Royal Statistical Soc.*, 57(2):425–437, 1993.

[67] C.J. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9(2):171–185, 1995.

[68] A. Lippman. *A maximum entropy method for expert systems*. PhD thesis, Brown University, 1986.

[69] H. Lucke. Improved acoustic modeling for speech recognition using 2D Markov random fields. In *Int. Conf. on ASSP*, 1995.

[70] J.-F. Mari, J.-P. Haton, and A. Kriouile. Automatic word recognition based on second-order hidden Markov models. *IEEE Trans. on Speech and Audio Processing*, 5(1), January 1997.

[71] A. A. Markov. An exemple of statistical investigation in the test of 'Eugene Onyegin' illustrating coupling of tests in chains. *Proc. Acad. Sci. St. Petersburg*, 7:153–162, 1913.

[72] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *Journal of the American Statistical Association*, 82(397):76–89, March 1987.

[73] N. Metropolis, A. W. Rosenbluth, A. H. Teller, M. R. Rosenbluth, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 13(5):1087–1091, 1953.

[74] Chafic Mokbel. *Reconnaissance de la parole dans le bruit: Bruitage / Débruitage*. PhD thesis, Ecole Nationale Superieure des Télécommunications, 1992.

[75] Todd K. Moon. The Expectation-Maximization algorithm. *IEEE Signal Processing Magazine*, 1996.

[76] R. De Mori and M. Federico. *Computational Models of Speech Pattern Processing*, volume 169 of *NATO ASI Series*, series f: computer and systems sciences Language Model Adaptation. Springer Verlag, 1998.

[77] Renato De Mori. *Spoken Dialogues with Computers*. Academic Press, 24-28 Oval Road, London NW1 7DX, UK, 1998.

[78] J. Moussouris. Gibbs and Markov random systems with constraints. *J. of Statistical Physics*, 10:11–31, 1974.

[79] M. Newman, L. Gillick, D. Paul, and B. Peskin. Speaker identification through LVCSR. In *NIST Speaker Recognition Workshop*, 1997.

[80] H. Noda et al. A MRF-based parallel processing algorithm for speech recognition using linear predictive HMM. In *ICASSP*, volume 1, pages 597–600, 1994.

[81] H. Noda, M. N. Shirazi, and B. Zhang. A parallel processing algorithm for speech recognition using Markov random fields. In *Trans. IEICE*, April 1993. (in Japanese).

[82] Stefan Ortmanns and Hermann Ney. A word graph algorithm for large vocabulary continuous speech recognition. *Computer Speech and Language*, 11:43–72, 1997.

[83] M. Ostendorf and S. Roukos. A stochastic segment model for phoneme-based continuous speech recognition. *IEEE Trans. on Acoust., Speech, Signal Processing*, 37(12):1857–1869, 1989.

[84] Mari Ostendorf. From HMMs to Segment Models. In *Automatic Speech and Speaker Recognition - Advanced Topics*, chapter 8. Kluwer Academic Publishers, 1996.

[85] K. K. Paliwal. Use of temporal correlation between successive frames in a hidden Markov model based speech recognizer. In *Proc. IEEE Intl. Conf. Acoust., Speech, Signal Processing*, 1993.

[86] Wojciech Pieczynski. Champs de Markov cachés et estimation itérative. *Traitement du Signal*, 11(2):141–153, 1994.

[87] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random field. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[88] A. B. Poritz. Linear predictive hidden Markov models and the speech signal. In *Proc. IEEE Intl. Conf. Acoust., Speech, Signal Processing*, pages 1291–1294, 1982.

[89] David Pye. Automatic recognition of continuous spelled Swiss-German letters. Technical report, IDIAP, 1994.

[90] L. Rabiner and B.-H. Juang. *Fundamentals of speech recognition.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[91] L. R. Rabiner, J. G. Wilpon, and B.-H. Juang. A segmental k-means training procedure for connected word recognition. *AT&T Technical Journal*, 65:21–31, 1986.

[92] D. Reynolds. Experimental evaluation of features for robust speaker identification. In *IEEE Transactions on Speech and Audio Processing*, volume 2, pages 639–643, 1994.

[93] Douglas A. Reynolds. Speaker identification and verification using gaussian mixture speaker models. In *ESCA Workshop on Speaker Recognition, Identification and Verification*, pages 27–30, 1994.

[94] M. J. Russel and R. K. Moore. Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition. In *Proc. IEEE Intl. Conf. Acoust., Speech, Signal Processing*, pages 5–8, 1985.

[95] Gilbert Saporta. *Probabilités, analyse des données et statistique*. Technip Paris, 1990.

[96] Ben M. Shahshahani. A Markov random field approach to Bayesian speaker adaptation. *IEEE Trans. on Speech and audio processing*, 5(2):183–191, 1997.

[97] S. Sibsi. Regularization and inverse problems. In *Maximum Entropy and Bayesian Methods*, pages 389–396. Kluwer Academics Publishers, 1989.

[98] Marc Sigelle. Simultaneous image restoration and hyperparameter estimation for incomplete data by cumulant analysis. Technical report, INRIA RR-3249, Sept. 1997.

[99] Marc Sigelle and Florence Tupin. Champs de Markov en traitement d'image. Cours de l'Ecole Nationale Superieure des Télécommunications, Module C3M, 1999.

[100] Martin A. Tanner. *Tools for Statistical Inference*. Springer-Verlag, 1993.

[101] D. M. Titterington. Recursive parameter estimation using incomplete data. *J. R. Statist. Soc.*, B, 46:257–267, 1984.

[102] M. J. Tomlinson, M. J. Russel, R. K. Moore, A. P. Buckland, and M. A. Fawley. Modelling asynchrony in speech using elementary single-signal decomposition. In *Intl. Conference on ASSP*, volume 2, pages 1247–1250, 1997.

[103] A. P. Varga and R. K. Moore. Hidden Markov model decomposition of speech and noise. In *Proc. Intl. Conf. on ASSP*, pages 845–848, 1990.

[104] J. Černocký, D. Petrovska-Delacrétaz, S. Pigeon, P. Verlinde, and G. Chollet. A segmental approach to text-independent speaker verification. In *Eurospeech*. Budapest, Sept. 1999.

[105] A. J. Viterbi. Error bounds for convolutional codes and asymptotically optimal decoding algorithm. *IEEE Trans. on Information Theory*, 13:260–269, April 1967.

[106] C. J. Wellekens. Explicit time correlation in hidden Markov models for speech recognition. In *Proc. IEEE Intl. Conf. Acoust., Speech, Signal Processing*, pages 284–287, 1987.

[107] Linde Y, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Trans. on Communication*, 28:84–95, 1980.

[108] Laurent Younes. Estimation and annealing for Gibbsian fields. *Ann. Inst. Henri Poincaré*, 24(2):269–294, 1988.

[109] Laurent Younes. Parametric inference for imperfectly observed Gibbsian fields. *Probability Theory and Related Fields*, 82:625–645, 1989.

[110] Laurent Younes. Parameter estimation for imperfectly observed Gibbs fields and some comments on Chalmond's EM Gibbsian algorithm. In P. Barone and A. Frigessi, editors, *Proceedings of stochastic models, statistical methods and algorithms in image analysis*. Springer, 1991.

[111] Steve Young. *HTK 1.4 Reference Manual*. Cambridge University Engineering Department - Speech Group, 1992.

[112] J. Zerubia and L. Blanc-Féraud. Hyperparameter estimation of a variational model using a stochastic gradient model. In *Proc. of SPIE Bayesian Inference for Inverse Problems*, 1998.

[113] J. Zerubia and R. Chellapa. Mean field approximation using compound Gauss-Markov radnom field for edge detection and image restauration. In *Intl. Conf. on ASSP*, pages 2193–2196, 1990.

[114] Jun Zhang. The mean field theory in EM procedure for Markov random fields. *IEEE Trans. on Signal Processing*, 40(10), October 1992.

[115] Yunxin Zhao, Lee A. Atlas, and Xinhua Zhuang. Application of the Gibbs ditribution to hidden Markov modeling in speaker independent isolated word recognition. *IEEE Trans. on Signal Processing*, 39(6):1291–1298, 1991.

Annexe A

Equations de maximisation pour les champs de Markov cachés

Nous reprenons dans cette annexe les notations du chapitre 2 pour donner le détail des équations de maximisation utilisées dans les approches gradient stochastique et EM de l'estimation des paramètres d'un champ de Markov caché.

A.1 Maximisation de la vraisemblance

A.1.1 Dérivée première

La log-vraisemblance par rapport aux paramètres de la loi *a priori* θ est donnée, d'après l'équation (2.9), par

$$l(\theta) = \ln \sum_{x \in \Omega} \exp(-U_\theta(x) - U_\lambda(y|x)) - \ln \sum_{x \in \Omega} \exp(-U_\theta(x)) \ . \tag{A.1}$$

Quelque soit la fonction énergie $\varphi(x)$, la dérivée du logarithme de la fonction de partition associée est donnée par

$$\frac{\partial \ln \sum_{x \in \Omega} \exp -\varphi(x)}{\partial \theta} = -\frac{\sum_{x \in \Omega} \varphi'(x) \exp -\varphi(x)}{\sum_{x \in \Omega} \exp -\varphi(x)} \ , \tag{A.2}$$

$\varphi'(x)$ étant la dérivée de la fonction $\varphi(x)$ par rapport à θ. En posant $\varphi(x) = U_\theta(x)$ et en remarquant que dans l'équation (A.2), le dénominateur est la fonction de partition *a priori* Z_θ, on en déduit que

$$\frac{\partial \ln \sum_{x \in \Omega} \exp -U_\theta(x)}{\partial \theta} = -E[U'_\theta(x)]$$

où l'espérance est donnée pour la loi *a priori* avec les paramètres θ. En suivant le même raisonnement pour $\varphi(x) = U_\theta(x) + U_\lambda(y|x)$, on montre alors qu'en dérivant (A.1) par rapport à θ on obtient

$$\frac{\partial l(\theta)}{\partial \theta} = E[U'_\theta(x)] - E[U'_\theta(x)|Y = y] \ . \tag{A.3}$$

A.1.2 Dérivée seconde

En reprenant la fonction générique $\varphi(x)$, on cherche maintenant à dériver par rapport à θ la fonction

$$E[\varphi'(x)] = \frac{\sum_{x\in\Omega} \varphi'(x) \exp -\varphi(x)}{\sum_{x\in\Omega} \exp -\varphi(x)} \ .$$

On montre aisément que la dérivée du numérateur est donnée par

$$\frac{\partial \sum_{x\in\Omega} \varphi'(x) \exp -\varphi(x)}{\partial \theta} = \left(\sum_{x\in\Omega} \varphi''(x) \exp -\varphi(x)\right) - \left(\sum_{x\in\Omega} (\varphi'(x))^2 \exp -\varphi(x)\right) \ , \tag{A.4}$$

$\varphi''(x)$ étant la dérivée seconde de $\varphi(x)$ par rapport à θ, tandis que l'on a pour le dénominateur

$$\frac{\partial \sum_{x\in\Omega} \exp -\varphi(x)}{\partial \theta} = \sum_{x\in\Omega} \varphi'(x) \exp -\varphi(x) \ . \tag{A.5}$$

A partir des équations (A.4) et (A.5), on établit alors le résultat suivant:

$$\begin{aligned} \frac{\partial E[\varphi'(x)]}{\partial \theta} &= \frac{1}{Z}\left(\sum_{x\in\Omega} \varphi''(x) \exp -\varphi(x) - \sum_{x\in\Omega} (\varphi'(x))^2 \exp -\varphi(x)\right) + \\ &\quad \frac{1}{Z^2}\left(\sum_{x\in\Omega} \varphi'(x) \exp(-\varphi(x))\right)^2 \\ &= E[\varphi''(x)] - var(\varphi'(x)) \ . \end{aligned} \tag{A.6}$$

Dans l'équation (A.6), l'opérateur $var(f(x))$ désigne la variance de la fonction $f(x)$ sous la distribution de Gibbs définie par le potentiel $\varphi(x)$. En prenant successivement $\varphi(x) = U_\theta(x)$ et $\varphi(x) = U_\theta(x) + U_\lambda(y|x)$, comme pour le calcul de la dérivée première, on obtient alors l'équation (2.13), soit

$$\frac{\partial^2 l(\theta)}{\partial \theta^2} = \big(E[U''_\theta(x)] - E[U''_\theta(x)|Y = y]\big) - \big(var(U'_\theta(x)) - var(U'_\theta(x)|Y = y)\big) \ . \tag{A.7}$$

A.2 Maximisation de la fonction auxiliaire

On s'intéresse ici à la fonction auxiliaire $Q(\theta, \lambda; \theta^{(n)}, \lambda^{(n)})$ donnée au chapitre 2 par l'équation (2.18) et à sa maximisation par rapport à θ. Dans la mesure où seul le paramètre θ nous intéresse dans cette annexe nous noterons la fonction auxiliaire $Q(\theta; \theta^{(n)})$.

Rappelons que la log-vraisemblance du processus conjoint (x, y) est donnée par

$$p_\theta(x, y) = -U_\theta(x) - U_\lambda(y|x) - \ln(Z_\theta)$$

ce qui nous donne alors pour la fonction auxiliaire

$$Q(\theta; \theta^{(n)}) = -E_{\theta^{(n)}}[U_\theta(x)|Y = y] - E_{\theta^{(n)}}[U_\lambda(y|x)|Y = y] - \ln Z_\theta$$

puisque Z_θ est une constante.

Cherchons maintenant à dériver cette équation. Les espérances étant considérées pour la valeur courante $\theta^{(n)}$ du paramètre et non pas par rapport à la variable θ, on a alors

$$\frac{\partial E_{\theta^{(n)}}[\varphi(x)|Y = y]}{\partial \theta} = E_{\theta^{(n)}}[\varphi'(x)|Y = y]$$

pour une fonction quelconque $\varphi(x)$, qui peut dépendre ou ne pas dépendre de θ. En utilisant la dérivée du logarithme de la fonction de partition donnée par l'équation (A.1.1) et en remarquant que la fonction $U_\lambda(y|x)$ ne dépend pas de θ, on obtient finalement

$$\frac{\partial Q(\theta; \theta^{(n)})}{\partial \theta} = E_\theta[U'_\theta(x)] - E_{\theta^{(n)}}[U'_\theta(x)|Y = y] \ .$$

A.3 Cas d'un potentiel linéaire par rapport à un paramètre

Lorsque le potentiel U_θ est linéaire par rapport au paramètre θ les équations établies précédemment prennent une forme plus simple. Posons $U_\theta(x) = \theta\varphi(x)$ où $\varphi(x)$ est une fonction de x ne dépendant pas de θ. En effet on a dans ce cas $U'_\theta(x) = \varphi(x)$ et la dérivée seconde est nulle.

La dérivée de la log-vraisemblance donnée par (A.3) devient alors

$$\frac{\partial l(\theta)}{\partial \theta} = E[\varphi(x)] - E[\varphi(x)|Y = y]$$

tandis que la dérivée seconde (A.7) est alors donnée par

$$\frac{\partial^2 l(\theta)}{\partial \theta^2} = var(\varphi(x)|Y = y) - var(\varphi(x)) \ .$$

Similairement, la dérivée de la fonction auxiliaire de l'algorithme EM devient alors

$$\frac{\partial Q(\theta; \theta^{(n)})}{\partial \theta} = E_\theta[\varphi(x)] - E_{\theta^{(n)}}[\varphi(x)|Y = y] \ .$$

Annexe B

Algorithme EM généralisé appliqué au modèle RFM-*sync*1

B.1 Rappel du problème

Nous présentons ici les formules exactes pour l'algorithme EM généralisé introduit en 3.3. Dans la première présentation, nous avons donné le principe de l'algorithme ainsi qu'une partie des formules de réestimation dans le cas d'une observation unique et pour des densités mono-gaussiennes. Cette annexe récapitule le fonctionnement de l'algorithme en donnant les formules étendues au cas général pour lequel on dispose de plusieurs observations appartenant à $\mathbb{R}^d$ et de densités multi-gaussiennes, pour un modèle

$$\Theta_{N,K} = \{A^{(k)}\ k \in [1,K],\ F,\ b_{ik}(.)\ i \in [1,N] \text{ et } k \in [1,K]\}\ ,$$

avec

$$b_{ik}(y_{t,k}) = \sum_{l=1}^{L} w_{ikl}\, \mathcal{N}(y_{t,k}; \mu_{ikl}, \Sigma_{ikl})\ . \tag{B.1}$$

Dans cette équation, la fonction $\mathcal{N}$ désigne une densité gaussienne en dimension d donnée par

$$\mathcal{N}(y_{t,k}; \mu_{ikl}, \Sigma_{ikl}) = \frac{1}{\sqrt[d]{2\pi}|\Sigma_{ikl}|^{1/2}} \exp\left(-\frac{1}{2}(y_{t,k} - \mu_{ikl})'\Sigma_{ikl}^{-1}(y_{t,k} - \mu_{ikl})\right)\ , \tag{B.2}$$

x' étant la transposée du vecteur x.

Nous supposerons pour la suite que l'on dispose de M segments, $y^{(1)}, \dots, y^{(M)}$, de longueur respective $T_1, \dots, T_M$, pour l'estimation des paramètres. On détaille maintenant les étapes *Expectation* (sec. B.3) et *Maximization* (sec. B.2) permettant d'obtenir une nouvelle estimée $\theta^{(n+1)}$ des paramètres connaissant l'estimée courante $\theta^{(n)}$.

B.2 Formules de maximisation (M-Step)

Rappellons tout d'abord que l'expression de la fonction auxiliaire (3.8) dans le cas général est donnée par [1]

$$\begin{aligned}
Q(\theta,\theta^{(n)}) &= -\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{i=0}^{N}\sum_{j=1}^{N+1} a_{i,j}^{(k)} E_{\theta^{(n)}}[\varphi_{i,j}^{(k)}(x)|Y=y^{(m)}] \\
& -\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{l=k+1}^{K} f_{k,l} E_{\theta^{(n)}}[\psi_{k,l}(x)|Y=y^{(m)}] \\
& -M \ln Z_\theta \\
& -\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{i=1}^{N} \ln(b_{ik}(y_{t,k})) E_{\theta^{(n)}}[\delta(x_{t,k}=i)|Y=y^{(m)}] \ .
\end{aligned}$$

B.2.1 Paramètres de la loi *a priori*

On donne dans un premier temps les équations de maximisation des paramètres de la loi *a priori* .

Poids de transition

Nous avons vu dans la section 3.3.1 le principe de la maximisation de la fonction auxiliaire en appliquant un pas de gradient stochastique pour chacun des vecteurs de paramètres $a_{i,(k)}$. En reprenant ici les mêmes notations, nous devons résoudre

$$a_{i,(k)}^{(n+1)} = a_{i,(k)}^{(n)} - J_{i,(k)}^{-1}(a_{i,(k)}^{(n)}) \;\; h_{i,(k)}(a_{i,(k)}^{(n)}) \tag{B.3}$$

pour $i = 0,\ldots,N$ et pour $k = 1,\ldots,K$. Dans le cas d'observations multiples, le j-*ème* élément du vecteur $h_{i,(k)}(a_{i,(k)}^{(n)})$ est donné par

$$\begin{aligned}
h_{i,j}^{(k)}(a_{i,(k)}^{(n)}) &= \frac{\partial Q(\theta,\theta^{(n)})}{\partial a_{i,j}^{(k)}}(a_{i,(k)}^{(n)}) \\
&\simeq M\ E_{\theta^{(n)}}[\varphi_{i,j}^{(k)}(x)] - \sum_{m=1}^{M} E_{\theta^{(n)}}[\varphi_{i,j}^{(k)}(x)|Y=y^{(m)}] \ ,
\end{aligned} \tag{B.4}$$

et que l'élément (j,m) de la matrice de Jacobi $J_{i,(k)}(a_{i,(k)})$ est donné par

$$\begin{aligned}
\frac{\partial h_{i,j}^{(k)}(a_{i,(k)}^{n})}{\partial a_{i,m}^{(k)}} &= \frac{\partial^2 Q(\theta,\theta^{(n)})}{\partial a_{i,j}^{(k)}\partial a_{i,m}^{(k)}}(a_{i,(k)}^{n}) \\
&\simeq -M\ cov_{\theta^{(n)}}(\varphi_{i,j}^{(k)}(x),\varphi_{i,m}^{(k)}(x)) \ .
\end{aligned} \tag{B.5}$$

1. Les états émetteurs d'une chaîne sont numérotés de 1 à N, les états 0 et $N+1$ étant utilisés pour décrire les états artificiels initiaux et finaux.

En pratique, la matrice de Jacobi pouvant avoir des structures très particulières dans le cas de chaînes contraintes, comme les chaînes gauche-droite, on utilisera la pseudo-inverse obtenu après décomposition en valeurs singulière de la matrice plutôt que l'inverse traditionnelle.

Poids de synchronisation

On montre aisément que l'équation de maximisation (3.16) devient dans le cas général

$$f_{k,l}^{(n+1)} = f_{k,l}^{(n)} + \frac{M\ E_{\theta^{(n)}}[\psi_{k,l}(x)] - \sum_{m=1}^{M} E_{\theta^{(n)}}[\psi_{k,l}|Y = y^{(m)}]}{M\ var_{\theta^{(n)}}(\psi_{k,l}(x))}\ , \tag{B.6}$$

pour $k = 1, \ldots, K$ et $l = 1, \ldots, K$.

B.2.2 Paramètres de la loi *a posteriori*

Pour chaque densité i, k donnée par l'équation (B.1), il est nécessaire d'estimer l'ensemble des poids w_{ikl}, des vecteurs de moyenne μ_{ikl} et des matrices de covariance Σ_{ikl}, pour $i = 1, \ldots, N$, $k = 1, \ldots, K$ et $l = 1, \ldots, K$. Les formules de maximisation pour ce type de loi sont bien connues et on montre aisément (cf. [90], section 6.7) les formules suivantes,

$$w_{ikl}^{(n+1)} = \frac{\sum_{m=1}^{M}\sum_{t=1}^{T_m} \gamma_t(i,k,l)}{\sum_{m=1}^{M}\sum_{t=1}^{T_m}\sum_{q=1}^{L} \gamma_t^{(m)}(i,k,q)} \tag{B.7}$$

$$\mu_{ikl}^{(n+1)} = \frac{\sum_{m=1}^{M}\sum_{t=1}^{T_m} \gamma_t^{(m)}(i,k,l)\ y_{t,k}^{(m)}}{\sum_{m=1}^{M}\sum_{t=1}^{T_m} \gamma_t^{(m)}(i,k,l)} \tag{B.8}$$

$$\Sigma_{ikl}^{(n+1)} = \frac{\sum_{m=1}^{M}\sum_{t=1}^{T_m} \gamma_t^{(m)}(i,k,l)\ (y_{t,k}^{(m)} - \mu_{ikl}^{(n+1)})(y_{t,k}^{(m)} - \mu_{ikl}^{(n+1)})'}{\sum_{m=1}^{M}\sum_{t=1}^{T_m} \gamma_t^{(m)}(i,k,l)}\ , \tag{B.9}$$

où $\gamma_t^{(m)}(i,k,l)$ est, pour le segment m, la probabilité *a posteriori* d'être dans l'état i de la bande k à l'instant t et d'associer l'observation $y_{t,k}^{(m)}$ à la composante l du mélange. On peut montrer que cette quantité est donnée par la formule

$$\gamma_t^{(m)}(i,k,l) = \frac{w_{ikl}\,\mathcal{N}(y_{t,k}^{(m)};\mu_{ikl},\Sigma_{ikl})}{\sum_{q=1}^{L} w_{ikq}\,\mathcal{N}(y_{t,k}^{(m)};\mu_{ikq},\Sigma_{ikq})} E_{\theta(n)}[X_{t,k}=i|Y=y^{(m)}] \ . \tag{B.10}$$

B.3 Estimation des espérances (E-Step)

L'ensemble des équations (B.3) à (B.10) repose sur le calcul d'espérances selon les lois *a priori* et *a posteriori* de fonctions du champ caché. Ces espérances ne sont bien évidemment pas calculables analytiquement et on a alors recours à des approximations stochastiques. Les espérances étant, dans le cadre de l'approximation stochastique du gradient, données pour la valeur courante $\theta^{(n)}$ des paramètres, nous pouvons associer à chaque segment d'apprentissage $y^{(m)}$, P échantillons selon les lois *a priori* et *a posteriori* . On dispose donc des échantillons $x_{\text{prior}}^{(m,p)}$ et $x_{\text{post}}^{(m,p)}$ pour $m=1,\ldots,M$ et $p=1,\ldots,P$ afin d'approximer les espérances par une moyenne empirique.

Pour les paramètres de transitions, nous utiliserons les approximations stochastiques suivantes,

$$E_{\theta(n)}[\varphi_{i,j}^{(k)}(x)] \simeq \frac{1}{MP}\sum_{m=1}^{M}\sum_{p=1}^{P}\varphi_{i,j}^{(k)}(x_{\text{prior}}^{(m,p)}) \tag{B.11}$$

$$E_{\theta(n)}[\varphi_{i,j}^{(k)}(x)|Y=y^{(m)}] \simeq \frac{1}{P}\sum_{p=1}^{P}\varphi_{i,j}^{(k)}(x_{\text{post}}^{(m,p)}) \tag{B.12}$$

$$E_{\theta(n)}[\varphi_{i,j}^{(k)}(x)\varphi_{i,n}^{(k)}(x)] \simeq \frac{1}{MP}\sum_{m=1}^{M}\sum_{p=1}^{P}\varphi_{i,j}^{(k)}(x_{\text{prior}}^{(m,p)})\varphi_{i,n}^{(k)}(x_{\text{prior}}^{(m,p)}) \ , \tag{B.13}$$

pour évaluer les équations (B.4) et (B.5). Des équations similaires peuvent être utilisée pour l'approximation des espérances de (B.6), soit

$$E_{\theta(n)}[\psi_{k,l}(x)] \simeq \frac{1}{MP}\sum_{m=1}^{M}\sum_{p=1}^{P}\psi_{k,l}(x_{\text{prior}}^{(m,p)}) \tag{B.14}$$

$$E_{\theta(n)}[\psi_{k,l}^2(x)] \simeq \frac{1}{MP}\sum_{m=1}^{M}\sum_{p=1}^{P}\psi_{k,l}^2(x_{\text{prior}}^{(m,p)}) \ . \tag{B.15}$$

Finalement, l'estimation des paramètres des mélanges de Gaussiennes associés aux états s'appuie sur l'approximation

$$E_{\theta(n)}[X_{t,k}=i|Y=y^{(m)}] \simeq \frac{1}{P}\sum_{p=1}^{P}\delta(x_{\text{prior}}^{(m,p)}(t,k)=i) \ . \tag{B.16}$$

B.4 En résumé ...

Pour résumer, la figure B.1 illustre la procédure EM d'estimation des paramètres du modèle RFM-*sync*1 dans le cas général en faisant référence aux équations présentées ci-dessus. Notons que les étapes E et M de l'algorithme peuvent être effectuées simultanément en remplacant directement les équations d'approximations (B.11) à (B.16) dans les équations de maximisation.

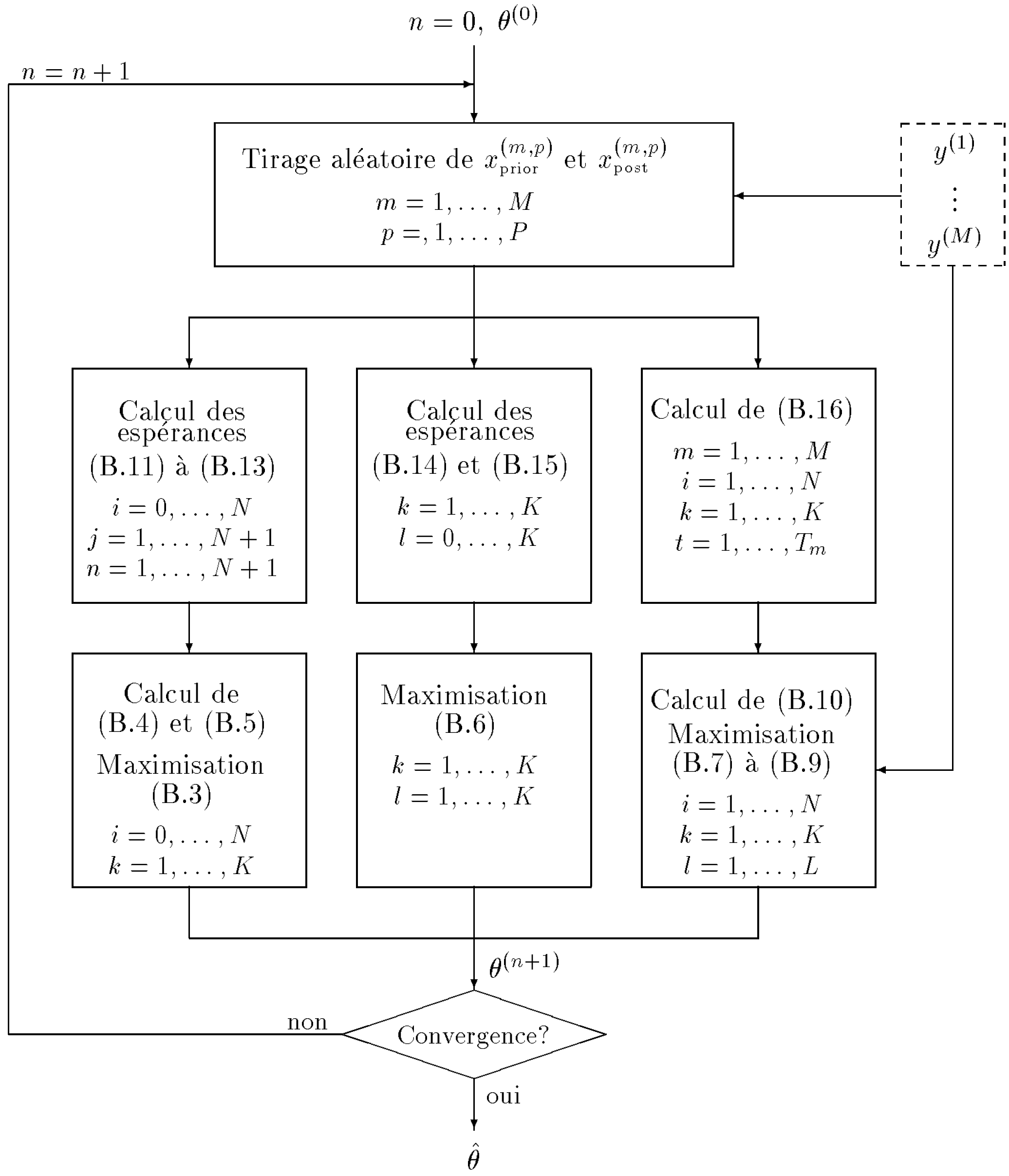


FIG. B.1 – *Schéma récapitulatif de la procédure EM.*

Annexe C

Estimation des paramètres d'un champ de Markov sur des données simulées.

Nous avons donné au chapitre 4 des figures illustrant la convergence des algorithmes d'intialisation et d'estimation des paramètres d'un champ de Markov pour les trois modèles *n2*, *n3* et *k2*. En complément des figures présentés dans le corps du document, nous donnons dans cette annexe quelques figures supplémentaires sur lesquels l'analyse des résultats présentés au chapitre 4 s'appuie.

C.1 Initialisation des paramètres

C.1.1 Données difficilement séparables

Dans les modèles mono-bande *n2* et *n3*, les moyennes entre deux états consécutifs sont séparées de 2 pour un variance de 1 ce qui rend les données relativement séparables. Afin d'étudier le comportement de nos algorithmes dans le cas de données moins séparables, nous avons repris les simulations du chapitre 4 pour des écarts de moyenne de 1, les variances restant fixées à 1. Pour le modèle *n2*, les moyennes sont respectivement de -0.5 et 0.5 et les résultats sont donnés figure C.1 pour l'initialisation par algorithme ICM et par recuit simulé.

Pour le modèle *n3*, les moyennes sont repectivement de -1, 0 et 1 et les résultats sont donnés figure C.2.

C.1.2 Modèle multi-bandes

Nous illustrons figure C.3 l'initialisation des moyennes et des variances des densités de probabilités par ICM et par recuit simulé avec et sans sychronisation. Les figures de gauche correspondent à la première bande et celle de droite à la deuxième.

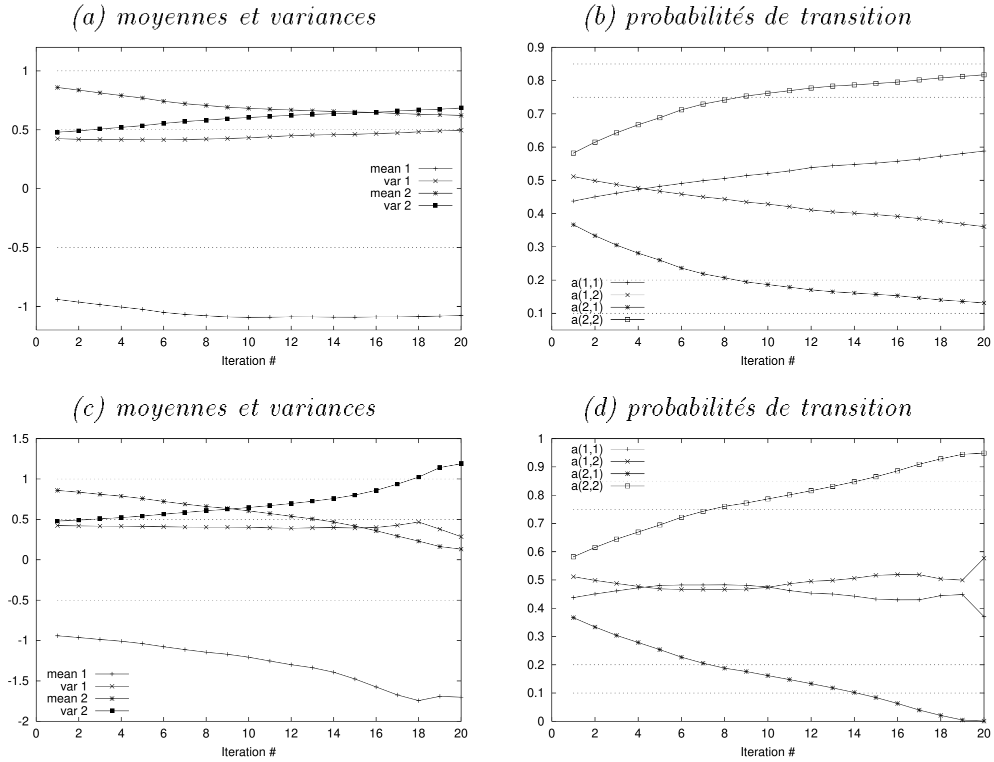


FIG. C.1 – *Initialisation des paramètres du modèle n2 dans le cas de données difficilement séparables. Les figures (a) et (b) illustrent l'algorithme ICM tandis que les figures (c) et (d) correspondent au recuit simulé.*

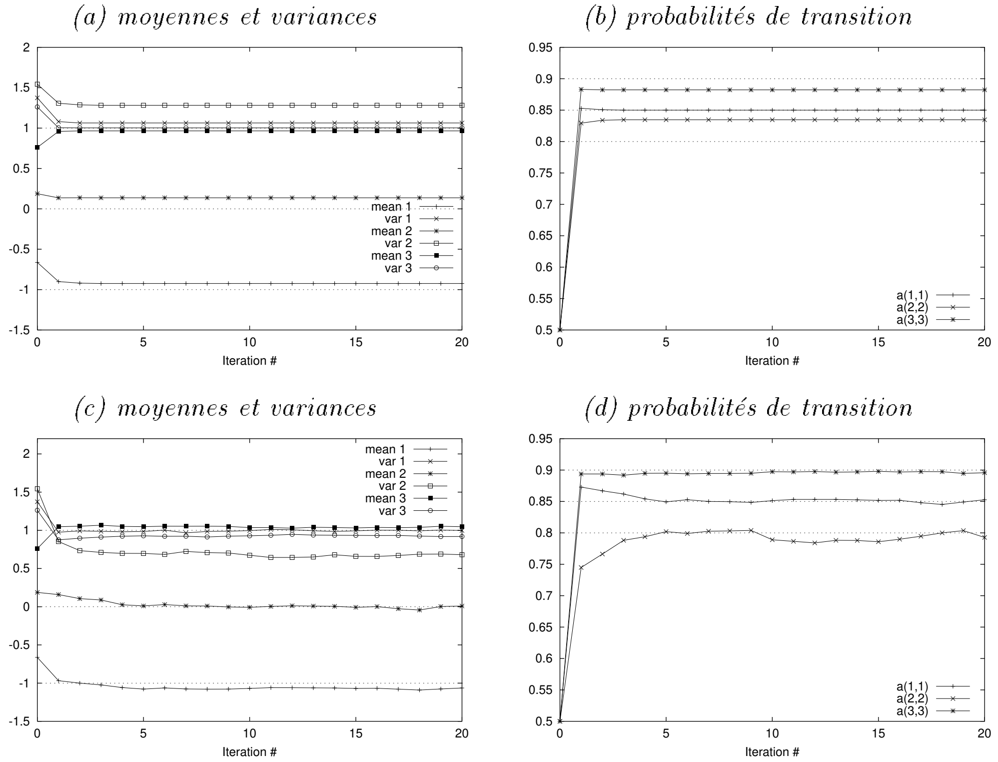


FIG. C.2 – *Initialisation des paramètres du modèle n3 dans le cas de données difficielement séparables. Les figures (a) et (b) illustrent l'algorithme ICM tandis que les figures (c) et (d) correspondent au recuit simulé.*

C.2 Algorithme EM généralisé

C.2.1 Variantes d'estimation pour le modèle n2

Plusieurs variantes de l'algorithme EM généralisé ont été utilisées. En particulier, il est possible d'approximer les espérances par une seule réalisation ($M = 1$), et l'approche EM devient alors équivalente à un algorithme de gradient stochastique. Les résultats obtenus avec cette approche sont illustrés figure C.4. Nous avons également testé une approche où les poids de transition sont considérés comme indépendants. Les résultats pour cette dernière approche sont donnés figure C.5.

C.2.2 Modèle multi-bande

La figure C.6 illustre la convergence des paramètres en appliquant directement l'algorithme EM au modèle multi-bandes *k2* pour deux valeurs du paramètres de synchronisation: $f = 0$ et $f = 1$. Comme dans la section C.1.2, les figures de la colonne de gauche correspondent à la première bande tandis que celle de la colonne de droite correspondent à la deuxième bande.

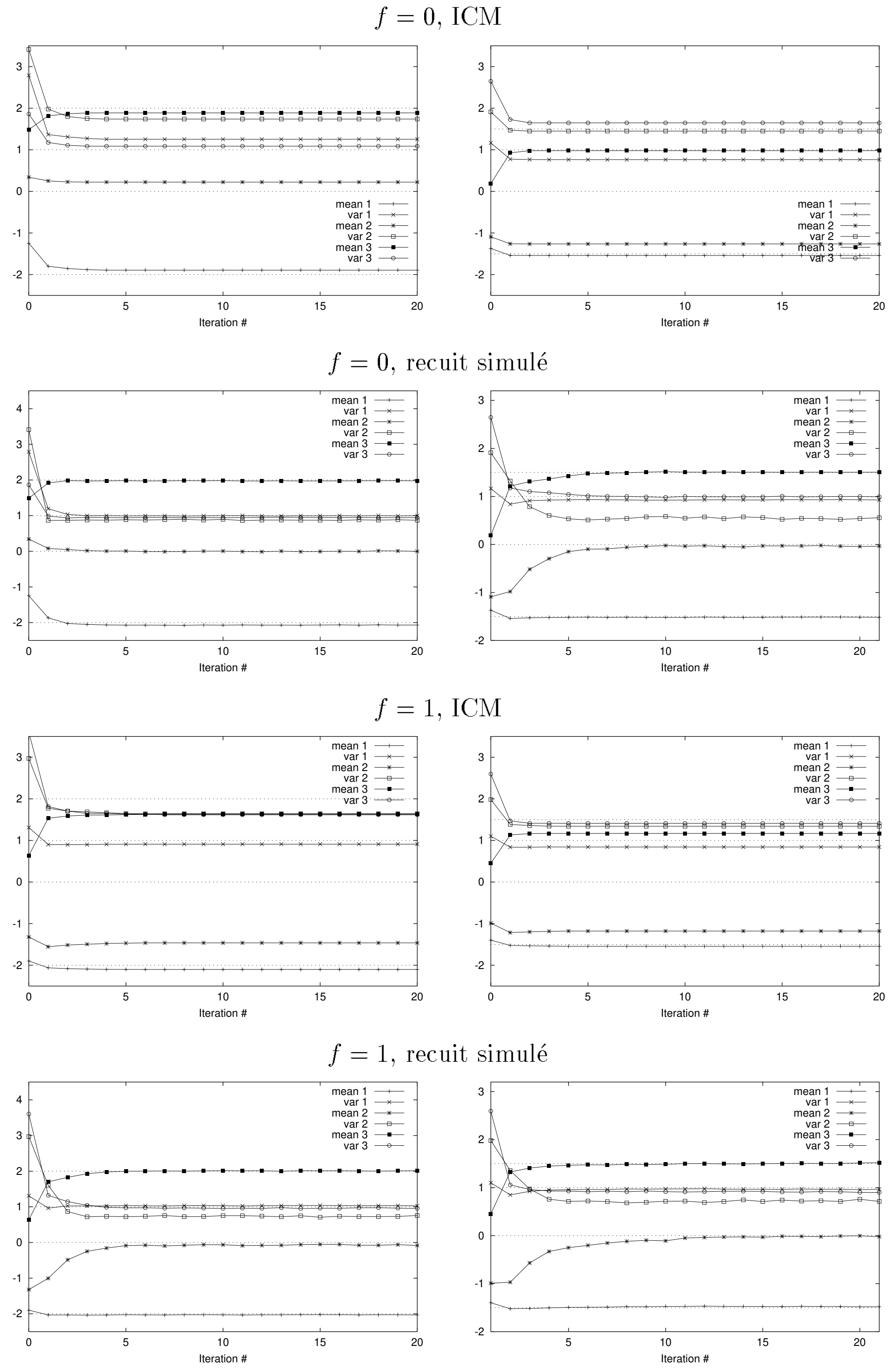


Fig. C.3 – *Initialisation des paramètres des densités du modèle k2 pour $f = 0$ et $f = 1$.*

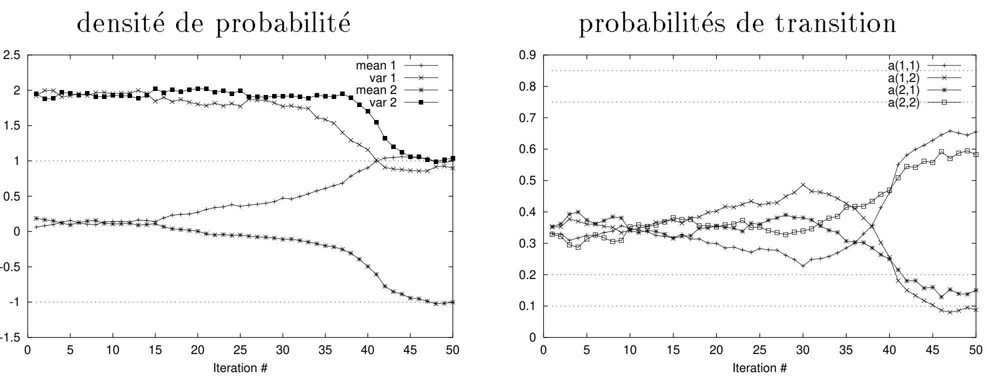


FIG. C.4 – *Algorithme EM avec $M = 1$ appliqué au modèle n2.*

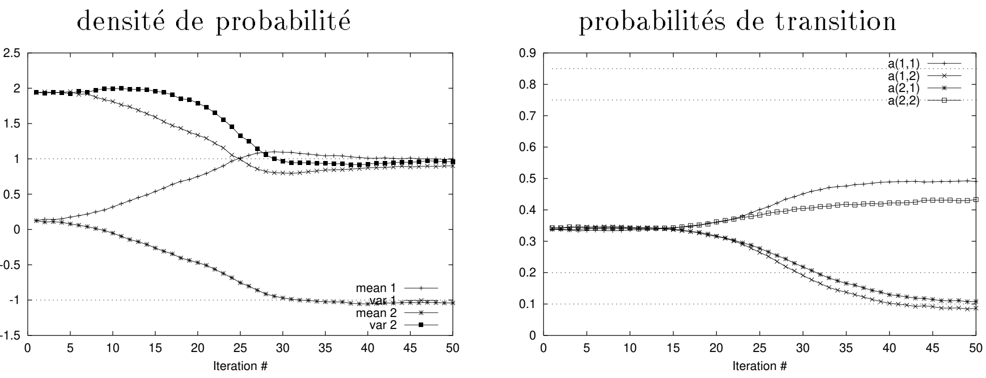
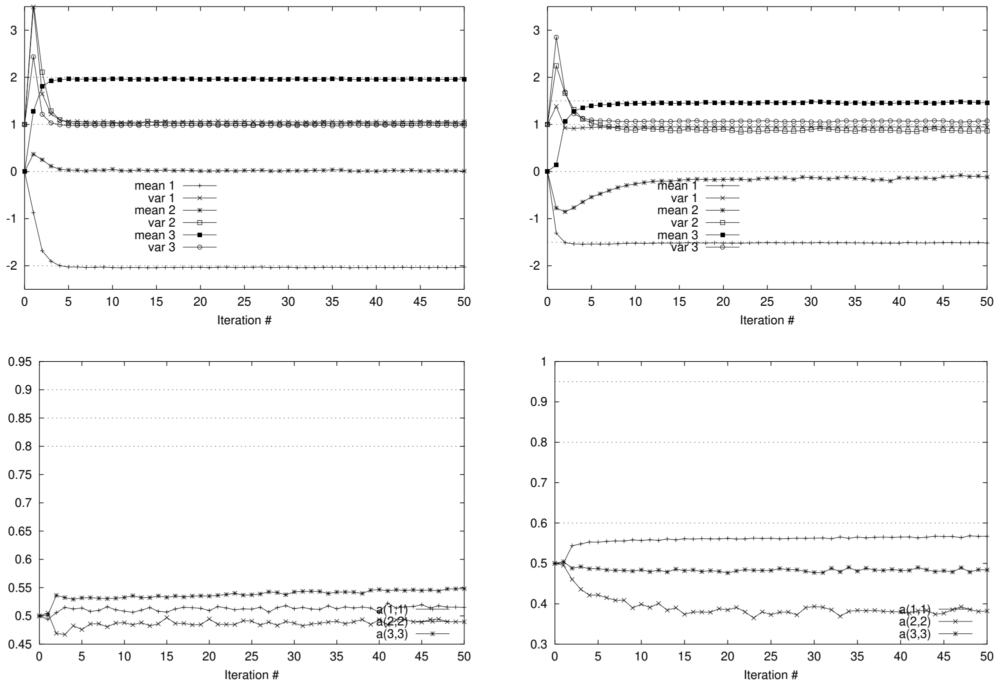


FIG. C.5 – *Algorithme EM appliqué au modèle n2 en considérant les poids de transition comme indépendant.*

$f = 0$



$f = 1$

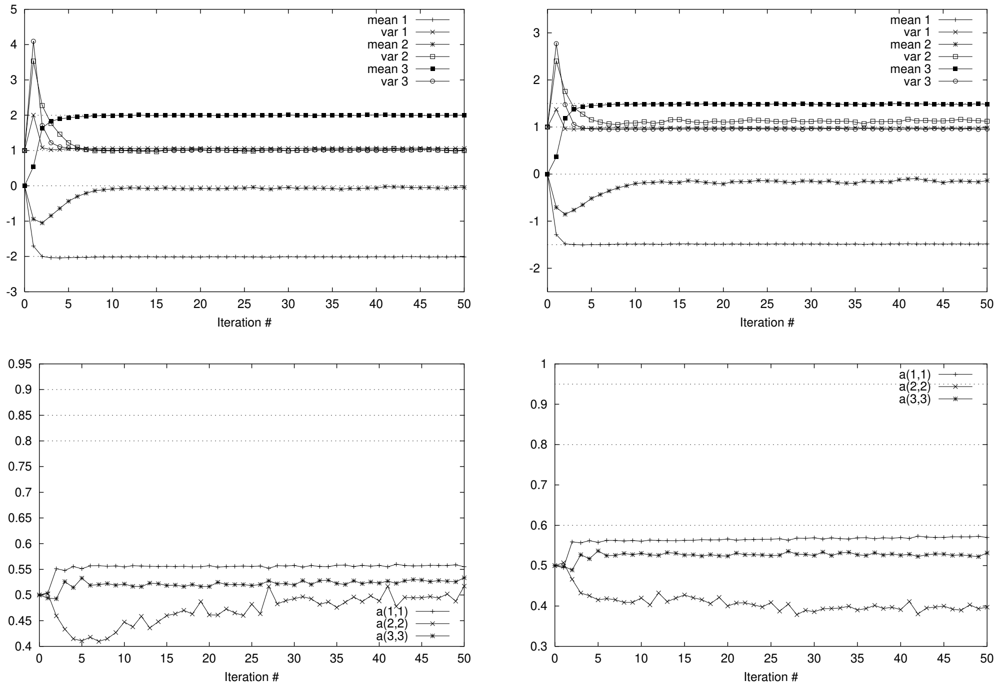


FIG. C.6 – *Estimation des paramètres des densités du modèle k2 pour $f = 0$ et $f = 1$.*

Annexe D

Exemple de matrice de covariances associées à une représentation par banc de filtres

Nous donnons dans cette annexe les matrices de covariances obtenues pour chaque état du mot “guide” avec un HMM dont les gaussiennes associées aux états sont à matrices de covariances pleines. La parole est représentée dans ce cas par la sortie d'un banc de 24 filtres.

Pour certains états la matrice de covariance est quasi-diagonale tandis que des “zones” de corrélations apparaissent dans d'autres états. La corrélation entre bandes n'est donc pas stationnaire.

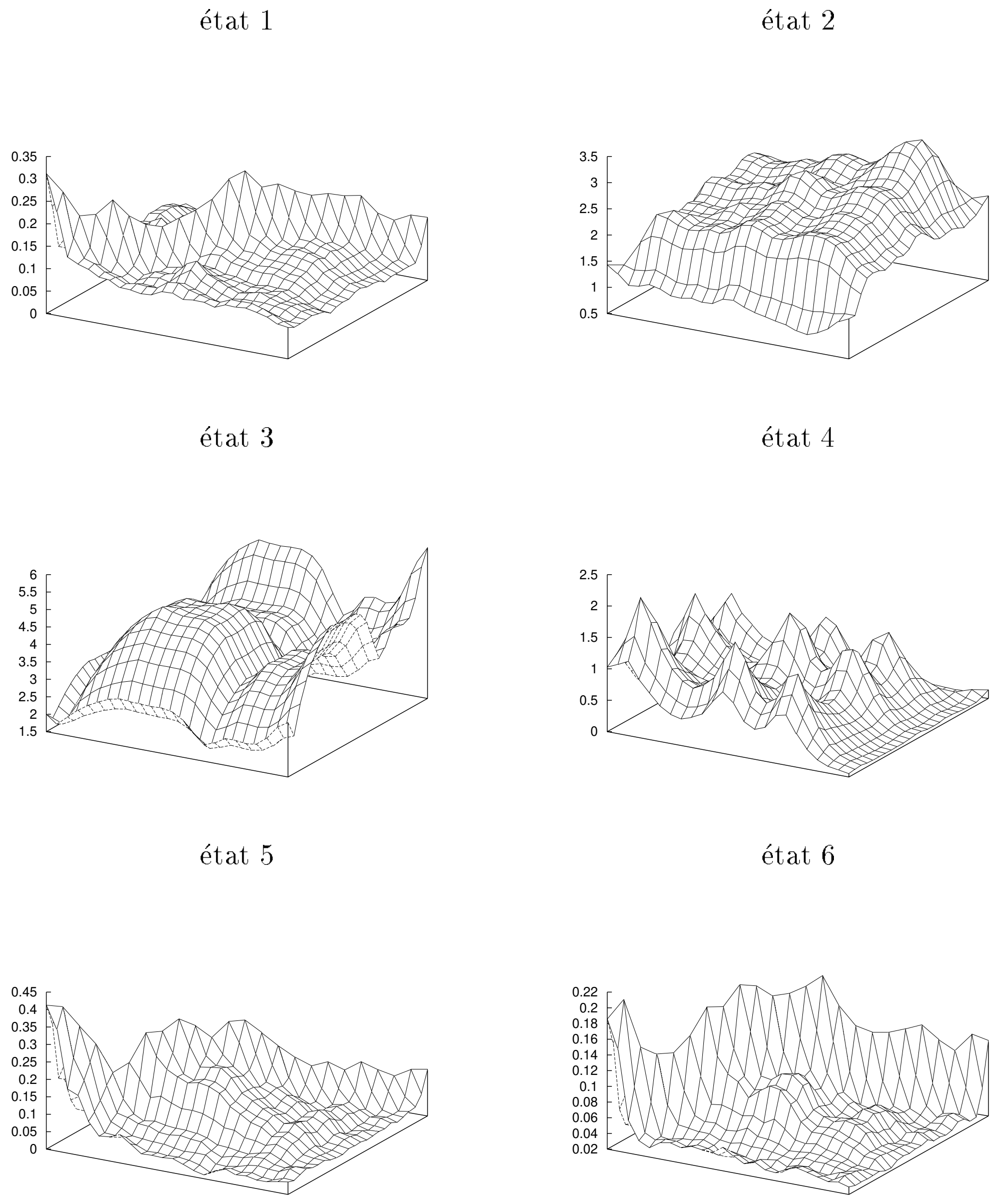


FIG. D.1 – *Matrices de covariance pour chaque état du mot "guide".*

Annexe E

Calcul des cesptres dans l'approche multi-bande

Dans cette annexe, nous détaillons les formules utilisées pour le calcul de P coefficients cepstraux dans chacune des bandes d'un signal échantillonné à la fréquence F_e et divisé en K bandes régulièrement répartis sur une échelle Mel.

Soit $x = \{x_1, \ldots, x_T\}$, T échantillons d'un signal correspondant à une trame de parole après pondération par une fenêtre de Hamming symétrique. On note $X = \{X_1, \ldots, X_N\}$ la transformée de Fourier discrète du signal x évalué sur N points. La première étape du calcul consiste à déterminer les fréquences de coupure de chacune des bandes $k = 1, \ldots, K$, données par

$$\begin{aligned} f_{\text{inf}}(k) &= \text{Mel}^{-1}\left(\frac{(k-1)}{K}\,\text{Mel}\left(\frac{F_e}{2}\right)\right) \\ f_{\text{sup}}(k) &= \text{Mel}^{-1}\left(\frac{k}{K}\,\text{Mel}\left(\frac{F_e}{2}\right)\right) \quad , \end{aligned}$$

où $f_{\text{inf}}(k)$ et $f_{\text{sup}}(k)$ sont les fréquences de coupure, respectivement inférieure et supérieure, pour la bande k, la fonction Mel() étant donnée par

$$\text{Mel}(f) = 2595\,\log\left(1 + \frac{f}{700}\right) \quad .$$

La seconde partie du calcul consiste, pour chaque bande, à extraire les X_i correspondant à la bande passante considérée, à symétriser la séquence ainsi obtenue et à appliquer une transformée de Fourier discrète (inverse) au log du module de la séquence symétrisée. L'indice correspondant à une fréquence de coupure f est donné par

$$i = N\frac{f}{F_e}$$

et on notera $I(k)$ et $S(k)$ les indices correspondants aux fréquences de coupure inférieure et supérieure pour la bande k[1]. Pour une bande k, la séquence sy-

1. Les fréquences étant réelles, il est nécessaire d'arrondir la valeur obtenue à l'indice le plus proche.

métrisée $X^{(k)}$ correspondante, de longueur $L_k = 2(S(k) - I(k) + 1)$, est donnée par

$$X^{(k)} = \left\{X_{I(k)}, X_{I(k)+1}, \ldots, X_{S(k)}, 0, X_{S(k)}, X_{S(k)-1}, X_{I(k)+1}\right\}$$

et les P coefficients cepstraux sont alors donnés par

$$c_i = \sum_{j=1}^{L_k} \ln(|X_j^{(k)}|) \, \cos\left(\frac{2\pi ij}{L_k}\right) \qquad i = 1, \ldots, P.$$