École Doctorale
d'Informatique,
Télécommunications
et Électronique de Paris

Thèse

présentée pour obtenir le grade de docteur
de l'École Nationale Supérieure des Télécommunications

Spécialité : **Signal et Images**

Leïla ZOUARI-BELTAÏFA

Vers le Temps Réel en Transcription Automatique de la Parole Grand Vocabulaire

Soutenue le 22 mars 2007 devant le jury composé de

Régine ANDRE-OBRECHT	Président
Denis JOUVET	Rapporteurs
Jean-Luc ZARADER	
Paul DELEGLISE	Examinateurs
Chafic MOKBEL	
Dijana PETROVSKA	
Gérard CHOLLET	Directeur de thèse

Résumé

Transcrire automatiquement la parole contenue dans un flux audio ne relève plus aujourd'hui de l'utopie scientifique : les systèmes actuels, basés généralement sur les modèles Markoviens, sont très performants et leur utilisation dans des contextes applicatifs exigeants (indexation automatique,...) est désormais envisageable. Pour autant, si une utilisation off-line s'avère possible, ces systèmes sont généralement beaucoup trop lents pour être utilisés dans des contextes applicatifs "temps-réels" tels que le sous-titrage ou la traduction automatiques, le dialogue homme-machine ... Le travail effectué au cours de cette thèse s'attache alors à proposer des méthodes de réduction du temps de calcul des systèmes de transcription en vue de permettre leur utilisation dans de tels contextes. Nous nous sommes particulièrement concentrés sur le calcul des probabilités, tâche occupant à elle seule souvent plus de la moitié du temps global de traitement.

Pour évaluer les approches développées, un système de reconnaissance de référence doit être implémenté. Nous avons ainsi construit et amélioré un système de transcription grand vocabulaire et ceci en s'appuyant sur le corpus radiophonique distribué à l'occasion de la campagne d'évaluation ESTER.

Les distributions des modèles acoustiques utilisés par les systèmes sont généralement représentées par des mélanges à composantes gaussiennes et le calcul des probabilités d'émission est particulièrement lié au nombre de gaussiennes considérées dans ces mélanges. Etant donné que seulement certaines de ces gaussiennes ont un réel impact sur le décodage, notre travail s'est porté sur l'évaluation de méthodes de sélection de gaussiennes. En pratique, ces méthodes sont basées sur la classification. Lorsque les gaussiennes de chaque mélange sont regroupées dans une structure arborescente, un parcours de l'arbre depuis sa racine permet de retrouver la feuille la plus proche des données de test. Les distributions gaussiennes situées à ce niveau sont sélectionnées. Cette approche n'étant pas optimale, nous avons proposé un partitionnement hiérarchique basé sur la similarité entre les distributions. La coupure de l'arbre à des hauteurs différentes permet de définir plusieurs niveaux de classification correspondant chacun à une sélection de gaussiennes. Les distributions choisies sont à l'intersection de toutes les sélections.

Dans le cas où toutes les distributions de tous les mélanges sont regroupées au moyen de l'algorithme k-moyenne, on obtient k classes représentées par leurs centroïdes respectifs. Les performances de la transcription sont évaluées selon le centroïde le plus proche des données de test, la sélection des gaussiennes relevant alors du choix du centroïde. Malheureusement, ces méthodes de sélection ne prennent pas en compte ces différents contextes, puisque toutes les distributions sont mélangées avant d'être regroupées en classes, perdant ainsi aux centroïdes correspondants leur modélisation contextuelle. Nous avons développé une nouvelle méthode de sélection permettant de conserver ces informations, en effectuant la sélection pour chacun des contextes considérés. Pour augmenter la représentativité des centroïdes, l'algorithme de partitionnement hiérarchique sus-mentionné est appliqué à chaque mélange. Un ensemble compact de centroïdes par contexte est fourni.

Ces dernières années, les méthodes de sous-quantification vectorielle sont apparues comme une alternative aux approches basées sur la sélection des gaussiennes. Il s'agit d'une quantification vectorielle par ensembles de dimensions. Ces méthodes sont également basées sur la classification pour former le dictionnaire des références ou *codebook*. Cette classification est opérée par regroupement de toutes les distributions de tous les mélanges, perdant de ce fait toute information contextuelle. Par conséquent, nous avons proposé une sous-quantification vectorielle contextuelle. L'amélioration de la représentativité et la réduction de dimension du *codebook* sont réalisées par partitionnement hiérarchique.

En conclusion, nous avons étudié et testé des méthodes existantes de sélection de gaussiennes et de sous-quantification vectorielle pour réduire le temps de calcul des vraisemblances. Puis nous avons proposé et évalué, dans les mêmes conditions, des méthodes originales de sélection de gaussiennes et de sous-quantification vectorielle pour remédier à certaines défaillances et/ou améliorer les méthodes existantes. Les résultats obtenus sont intéressants et dépassent certaines méthodes existantes.

A mes parents

à Samir et Nessim

Remerciements

Cette thèse a été préparée au laboratoire CNRS LTCI-TSI de l'Ecole Nationale Supérieure des Télécommunications, sous la direction de M. Gérard Chollet.
Elle n'a pu être ce qu'elle est aujourd'hui que grâce à son aide continue et à ses précieuses recommandations. Ses conseils judicieux m'ont permis de progresser dans le domaine du Traitement de la Parole et de parvenir à ces résultats. Pour tout cela je dirai : "MERCI M. Gérard Chollet".

Je tiens à remercier aussi M. Marc Sigelle et M. Chafic Mokbel pour les précieuses réflexions émises lors des discussions passionnantes que nous avons partagées et qui ont beaucoup apporté à ce travail.

Ma gratitude revient aussi aux membres du jury Mme Régine André-Obrecht, M. D. Jouvet, J-L. Zarader, M. P. Deleglise et Mme. D. Petrovska, qui m'ont fait l'honneur d'accepter d'évaluer ma thèse.

Un merci particulier à Abdelkader Oukaci, Zahir Hamroune et Youssef Boukhabrine pour leur aide et à tous les membres du laboratoire avec qui les échanges ont été fructueux.

Enfin je remercie l'ensemble de ma famille et en particulier mon mari Samir, mon fils Nessim pour leur patience.

Table des matières

Table des figures

Liste des tableaux

Liste des Abbréviations

BBI	Bucket-Box-Intersection
BN	Broadcast News
CI	Context Independent
CD	Context Dependent
CTS	Conversational Telephone Speech
GMM	Gaussian Mixture Model
GS	Sélection des Gaussiennes
GSM	Global System for Mobile
HMM	Hidden Markov Model
KPPV	K Plus Proches Voisins
LFCC	Linear Frequency Cesptral Coefficients
MAP	Maximum A Posteriori
MFCC	Mel Frequency Cesptral Coefficients
MLLR	Maximum Likelihood Linear Regression
PDA	Personal Digital Assistant
QV	Quantification Vectorielle
SRP	Système de Reconnaissance de la Parole
SVM	Support Vector Machines
SVQ	Sous-Quantification Vectorielle
WER	Word Error Rate
ZCR	Zero-Crossing Rate

Introduction générale

La parole est certainement le moyen le plus naturel de communication. Pour autant il est tout aussi certain qu'il est plus facile, d'un point de vue sémantique, de manipuler sa transcription que le signal de parole en propre. Cette transcription, résultat de la tâche de reconnaissance, peut ainsi être mise en oeuvre dans le cadre d'applications off-line (recherche d'informations audio par exemple) ou on-line, telles que la dictée vocale, les centres d'appels, ...

La reconnaissance de la parole mobilise depuis plus de trente ans la communauté scientifique de traitement de la parole. Ce large effort de recherche a permis d'améliorer continuement les systèmes de transcription dont les performances actuelles permettant d'envisager une utilisation "grand public". Pour autant, les défis de la transcription ne sont pas tous résolus puisque la complexité des systèmes actuels rend difficile leur utilisation dans certains contextes applicatifs tels que le dialogue "homme-machine" ou dans le cadre de plateformes embarquées, évidemment restreintes en termes de ressources informatiques. En effet, ces applications requièrent, en plus de la précision, une réponse en temps réel c'est-à-dire la reconnaissance ne dure pas plus que l'énoncé de parole. Ainsi, la reconnaissance automatique de la parole en temps réel constitue un des enjeux majeurs de la recherche actuelle. L'importance de la vitesse en reconnaissance automatique peut être illustrée par les nombreux travaux de recherche de laboratoires ou d'industriels tels que IBM [64], Dragon [69], ATT [5], l'université de Cambridge [24], Microsoft [3], ...

Pour certains, la reconnaissance se limite à un problème de précision. L'argument avancé étant que les machines sont de plus en plus performantes et rapides. Pour autant, cette augmentation des ressources va de pair avec une augmentation de la complexité des systèmes rendant ainsi non valide cette argumentation. Les avancées de la recherche sont elles-mêmes tributaires de ces temps de calculs élevés dédiés à la reconnaissance : comment progresser rapidement lorsque les expérimentations nécessaires peuvent prendre jusqu'à plusieurs semaines ? Quant aux systèmes industriels, l'exigence au niveau des ressources est encore plus forte puisque le surcroît des ressources entraine une augmentation du coût du produit. De plus, un système de reconnaissance s'intégrera généralement dans un produit non dédié (commande de GPS, PDA doté d'une reconnaissance vocale, ...).

Bien entendu, l'accélération de la transcription ne doit pas entraîner de diminution significative des performances des systèmes de transcription. Les dernières évaluations américaines portant sur la transcription de la parole en temps réel [56] ont pourtant montré des baisses notables de performances des systèmes lorsque le fonctionnement de ces derniers est contraint par le temps. Ces résultats valident ainsi le besoin de techniques efficaces qui réduisent le temps de reconnaissance sans engendrer de pertes de performance.

Les systèmes de reconnaissance de la parole actuels se basent sur les modèles de Markov cachés pour représenter les différentes unités acoustiques et par la suite les mots. Dans ce contexte, le processus de reconnaissance consiste à trouver la séquence de mots la plus probable ou encore le chemin optimal dans un graphe de mots. Cette recherche est dominée par deux tâches : le parcours du graphe et le calcul des vraisemblances. Cette dernière tâche occupe souvent à elle seule plus que la moitié du temps global de reconnaissance. Cette thèse aura pour objectif d'explorer les méthodes visant à accélerer le calcul de ces vraisemblances et ceci en s'intéressant en particulier à la réduction du nombre de densités calculées, et ce, sans perte significative de précision.

Cette thèse est organisée en six chapitres. Les chapitres 1 et 2 sont consacrés à la description des techniques existantes de transcription automatique des émissions radio et de segmentation en événements sonores. Ces deux chapitres détaillent en même temps les systèmes de transcription de référence développés dans le cadre de la campagne d'évaluation ESTER, en précisant les améliorations apportées en aval de la campagne.
Le chapitre 3 présente l'état de l'art concernant la réduction du temps de décodage. Ce chapitre s'attarde sur les techniques de calcul rapide des vraisemblances par restriction du nombre de densités calculées. Trois philosophies concernant ces techniques, souvent basées sur la classification, seront abordées. Nous distinguerons ainsi les méthodes qui organisent les distributions gaussiennes dans une structure hiérarchique, celles qui les regroupement au moyen d'algorithmes de type k-moyennes avant d'en sélectionner certaines, et celles qui opèrent une sous-quantification vectorielle pour limiter leur nombre. Dans la suite de ce document, chacune de ces approches a été mise en oeuvre, expérimentée puis améliorée.

Les chapitres 4, 5 et 6 détaillent alors les études relatives à chacune d'entre elles. Dans le quatrième chapitre, nous proposons une nouvelle méthode de partitionnement hiérarchique en vue d'une meilleure sélection des distributions gaussiennes. Le partitionnement des distributions est réalisé en utilisant la distance *Kullback-Leibler Pondérée*. La sélection est opérée à plusieurs niveaux de la

structure hiérarchique. Cette opération permet d'éliminer progressivement les hypothèses de distributions peu vraisemblables et de ne garder que celles qui le sont à tous les niveaux.

Le chapitre 5 détaille les méthodes basées sur un regroupement des distributions (par k-moyennes) dans le but de réaliser la sélection des gaussiennes. Les améliorations étudiées relativement à cette méthode relèvent d'une part de l'introduction de l'information contextuelle dans le processus de classification et d'autre part de l'utilisation du partitionnement hiérarchique en vue d'améliorer la classification contextuelle.

Selon la même organisation, le chapitre 6 précise finalement les techniques de sous-quantification vectorielle expérimentées ainsi que les améliorations développées au cours de la thèse : sous-quantification vectorielle contextuelle en prenant en compte le contexte pendant le regroupement (réalisé avec la distance de Kullback-Leibler) ; combinaison de cette dernière méthode avec celle du partitionnement hiérarchique exposée dans le chapitre 4.

Le dernier chapitre (chapitre 7) sera alors l'occasion de revenir sur l'ensemble du travail accompli au cours de cette thèse et de présenter quelques perspectives sur les thématiques de la transcription en temps réel.

Chapitre 1

Transcription Automatique des Emissions Radio

1.1 Introduction

Dans ce chapitre, on se propose de décrire brièvement les concepts fondamentaux de la reconnaissance automatique de la parole ainsi que certaines techniques à l'état de l'art. Puis nous rapportons nos travaux de développement de systèmes de transcription des émissions radiophoniques qui sont réalisés dans le cadre de la campagne d'évaluation ESTER.
Nos contributions concernent en particulier la modélisation acoustique. Par conséquent, nous détaillons les étapes de construction et d'amélioration de ces modèles.
Les systèmes obtenus sont considérés comme des systèmes de référence et seront utilisés pour valider des approches de réduction de temps dans les chapitres suivants.

1.2 Reconnaissance de la parole

1.2.1 Bref historique

Les fondements de la technologie récente de la reconnaissance de la parole ont été élaborés par F. Jelinek et son équipe à IBM dans les années 70 [34].
Les premiers travaux (années 80) se sont intéressés aux mots, et ce, pour des applications de vocabulaire réduit. Au début des années 90, les systèmes de reconnaissance continus et indépendants du locuteur ont vu le jour. En commençant par une tâche de 100 mots (Resource Management), la technologie s'est développée rapidement et déjà vers le milieu des années 90, une précision raisonnable est atteinte pour une tâche de dictée non restreinte [67].
La plupart de ce développement est réalisé dans le cadre du programme d'évaluation de la DARPA (Defense Advanced Research Projects Agency). Ces campagnes ne cessent d'augmenter les difficultés des tâches à chaque fois que la technologie est maîtrisée pour une tâche donnée (taux d'erreur <

10%) comme le montre la figure 1.1.

Les dernières évaluations concernent les journaux radio ou télé diffusés ("Broadcast News" ou BN) et les conversations téléphoniques spontanées (CTS).

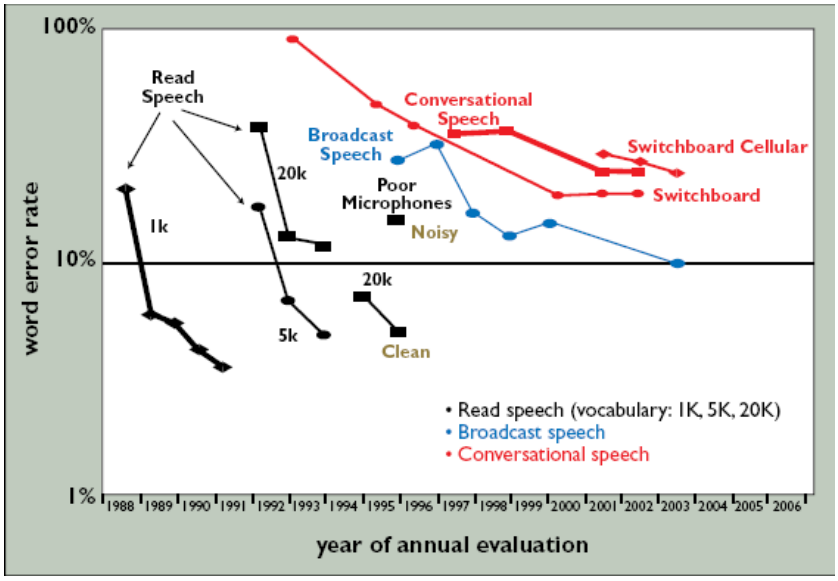


FIG. 1.1 – Historique de l'évolution du taux d'erreur en fonction de la complexité des données ([16])

1.2.2 Approche statistique

Les premiers travaux de reconnaissance de la parole ont essayé d'appliquer des connaissances d'experts en production et en perception mais les recherches ont montré que de telles connaissances sont inadéquates pour capter la complexité du signal continu. De nos jours, les techniques de modélisation statistique apportent les meilleures performances [63].

Formulation du problème

La formulation statistique du problème de reconnaissance suppose que la parole est représentée par une séquence de vecteurs acoustiques $O = o_1..o_T$ et que cette séquence encode la suite de mots : $M = m_1..m_K$.

La transcription orthographique de la parole se ramène alors à un problème de décodage où on cherche à trouver la séquence de mots $M^{'}$ tel que :

$$M^{'} = argmax\, p(M/O) = argmax\, p(O/M)\, p(M) \tag{1.1}$$

$p(O/M)$ est déterminé par un modèle acoustique et $p(M)$ par un modèle de langage. D'où l'architecture globale d'un Système de Reconnaissance de la Parole (SRP) suivante (figure 1.2) :

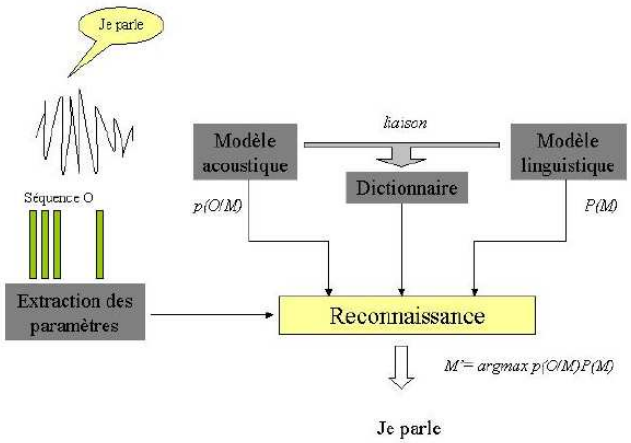


FIG. 1.2 – Architecture globale d'un Système de Reconnaissance

Vecteurs acoustiques

Le signal de parole est connu pour sa forte redondance. Ceci est dû aux contraintes physiques des articulateurs qui produisent la parole (lèvres, langue, ..) qui les empêchent de bouger rapidement. Ce signal peut être compressé en retenant seulement les informations pertinentes. En reconnaissance de la parole, il s'agit d'extraire des séquences de vecteurs de paramètres qui contiennent des informations sur le contenu spectral local du signal. Ces vecteurs sont appelés observations, et représentent ce que le système observe.

Typiquement le système extrait un vecteur de 39 paramètres toutes les 10 millisecondes. Souvent, il s'agit de 12 coefficients cepstraux et de l'énergie complétés par leurs dérivées première et seconde.

Modèles acoustiques

Étant donné que le vocabulaire des mots possibles peut être très grand, chaque mot est décomposé en une séquence d'unités de sons élémentaires appelés *phones*. Un phone n'est autre que la réalisation acoustique d'un phonème.

Vue la nature stochastique du signal de parole, les locuteurs ne prononcent pas souvent un mot de la même manière. Cette variation de prononciation se manifeste par la durée du son émis et par son contenu spectral (observations). En plus, les phonèmes, quant ils sont prononcés dans des contextes différents peuvent produire des variations du contenu spectral. Ce phénomène est désigné par la co-articulation [63].

a) *Modèles de Markov cachés* : Les modèles de Markov cachés ou HMM sont des machines à états finis qui permettent de modéliser les aspects stochastiques du signal de parole. Le modèle est défini par un ensemble d'états et de transitions permises entre ces derniers. Chaque transition s'accom-

pagne par l'émission d'une observation.

Souvent, la probabilité d'émission est représentée par une densité de probabilité multi-gaussienne :

$$b_j(o) = \sum_{k=1}^{G} \omega_k \mathcal{N}(o, \mu_k, \Sigma_k) \tag{1.2}$$

ω : poids, $\mathcal{N}$: loi normale de moyenne μ et de matrice de covariance Σ supposée diagonale.

G : nombre de gaussiennes.

$$N(o, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(o-\mu)^t \Sigma^{-1}(o-\mu)) \tag{1.3}$$

d est la dimension des vecteurs de paramètres.

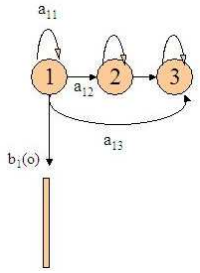


FIG. 1.3 – Modèle acoustique

b) *Estimation des paramètres* : avant d'utiliser un HMM pour calculer la vraisemblance d'une séquence d'observations, ses paramètres (probabilités d'émissions et de transition) doivent être estimés avec des données d'apprentissage. Pour ce faire, les modèles de suites de mots sont construits par concatenation de modèles de mots lesquels sont formés par concatenation de modèles de phones. Etant donnée une suite d'observations, les paramètres du modèle sont modifiés de façon à rendre plus probable l'émission de ces observations par ce modèle. Dans ce cas, on parle d'estimation basée sur le critère du maximum de vraisemblance (MV). D'autres critères d'apprentissage existent. On peut citer l'apprentissage discriminant basé sur la maximisation de l'information mutuelle (MMIE) [26] ou la minimisation de l'erreur de phone (MPE)[23] ou encore l'apprentissage adaptatif (SAT), qui vise à réduire l'impact des variations inter-locuteurs
[4].

c) *Co-articulation* : le système modélise la co-articulation en supposant que les densités des observations dépendent aussi bien du phonème en question que de ceux qui l'entourent. On distingue les biphones (un contexte droit ou gauche), les triphones (un contexte droit et un contexte gauche), les quinphones, ..

Modèles de langage

La modélisation acoustique permet de réaliser la transcription phonétique d'une phrase. Or, en absence de contraintes linguistiques, il est possible que cette suite soit très différente de la chaîne attendue. Par conséquent, il est nécessaire d'introduire des connaissances sur les niveaux supérieurs du langage. Dans le cadre des SRP Markoviens, l'utilisation de modèle de langage probabiliste est naturelle. La probabilité d'une séquence de mots $M = m_1..m_K$ est :

$$p(M) = \prod p(m_k/m_{k-1}, m_{k-2}, .., m_1) \tag{1.4}$$

En général, on se limite à un historique de $n-1$ mots précédents et on parle de modèle N-grammes :

$$p(M) = \prod p(m_k/m_{k-1}, m_{k-2}, .., m_{k-n+1}) \tag{1.5}$$

Souvent $n = 2$ ou 3 ou 4.
Les modèles de langage sont estimés sur de grandes bases de données textuelles contenant des centaines de millions de mots.

Dictionnaire

On suppose que le système connaît le vocabulaire du (des) locuteur(s) et la recherche est restreinte aux séquences de mots présents dans le dictionnaire. Ce dernier liste les mots et leurs prononciations sous forme de suites de phones. Le performances du système de reconnaissance sont directement liées au taux de mots hors vocabulaire.

Décodage

L'objectif du décodage est de trouver la séquence de mots la plus probable sachant le dictionnaire et les modèles acoustiques et de langage. En pratique, il s'agit de trouver la suite d'états la plus probable dans un trellis de mots (espace de recherche) où chaque noeud représente un état de phone donné à un temps t [27]. Pour ce faire, deux algorithmes sont fréquemment utilisés : l'algorithme synchrone au temps de Viterbi et l'algorithme A* qui est asynchrone.
Vue la taille de l'espace de recherche, la détermination du meilleur chemin peut devenir compliquée. Une approche multi-passes peut être utilisée pour réduire la complexité du décodage. Par exemple pour la première passe on peut utiliser un bigramme et des modèles acoustiques simples et dans la seconde un trigramme et des modèles acoustiques plus compliqués.
L'information entre les passes est transmise via un treillis de mots ou les N meilleures hypothèses. Le treillis est un graphe où les noeuds correspondent à des instants et les arcs correspondent aux hypothèses de mots. Les N meilleures hypothèses est une liste des meilleures séquences de mots et de leurs scores respectifs.

Mesure de performance

Les systèmes de reconnaissance de la parole sont évalués en terme de taux de mots erronés (ou WER pour Word Error Rate).

$$WER = 100 * \frac{substitutions + omissions + insertions}{nombre \;\; de \;\; mots \;\; prononces} \tag{1.6}$$

Suivant l'application, le WER peut varier considérablement. Ceci dépend de :

- Taille du vocabulaire : les applications à vocabulaire réduit ont un WER plus faible.
- Perplexité du modèle de langage : plus faible est la perplexité (mesure d'adéquation entre le modèle de langage et un document donné), plus réduit le WER.
- Bruit : moins les conditions d'enregistrement sont bruitées, plus faible est le WER.
- Prononciation : La parole spontanée est plus difficile à reconnaître que la parole lue.
- Données d'apprentissage : une quantité plus importante de données d'apprentissage permet d'obtenir de meilleures performances.

Le WER est une estimation des performances d'un système de reconnaissance et sa fiabilité dépend du nombre de tests réalisés (donc du nombre d'unités acoustiques à reconnaître qui est dans notre cas le mot). La mesure de l'intervalle de confiance est introduite afin de mesurer la précision de notre taux d'erreur (ou de reconnaissance). Dans [11], les réussites sont modélisées par une distribution binomiale. Si N est le nombre de tests effectués (mots) et P le taux de reconnaisance alors l'intervalle de confiance $[P-, P+]$ à $x\%$ est :

$$P\pm = \frac{P + \frac{z_x^2}{N} \pm z_x \sqrt{\frac{P(1-P)}{N} + \frac{z_x^2}{4N^2}}}{1 + \frac{z_x^2}{N}} \tag{1.7}$$

avec $z_{95\%} = 1.96$. Il y a $x\%$ de chance que le taux de trouve dans cet intervalle.

1.2.3 Techniques avancées

Pour réduire le taux d'erreur des systèmes de reconnaissance de la parole, de nouvelles méthodes ont été récemment développées et évaluées. Parmi ces dernières :

Paramétrisation : des techniques discriminantes d'extraction des paramètres acoustiques. En particulier, l'analyse linéaire discriminante et ses variantes permettent une réduction du taux d'erreur de l'ordre de 2% [63, 88, 23]. Cette approche réduit la dimension des vecteurs de paramètres par projection sur des directions qui maximisent la séparation entre les classes de phonèmes. Elle opère sur des vecteurs de paramètres obtenus par concatenation d'observations successives ou de grande dimensions.

Apprentissage : La prise en compte du contexte permet d'obtenir des modèles beaucoup plus précis mais aussi plus nombreux que les modèles indépendants du contexte. Ainsi, pour optimiser les performances de reconnaissance, il est nécessaire de trouver un compromis entre le degré de précision des modèles et la possibilité de bien estimer leurs paramètres. Comme, en pratique certains contextes apparaissent très peu dans les données d'apprentissage, des méthodes d'amélioration de l'apprentissage ont été développées [57].

- Repli ou *Backing - Off* : un modèle disposant de peu de données est remplacé par un autre modèle moins précis mais plus fréquent dans la base d'apprentissage. Par exemple des modèles de type triphone peuvent être remplacés par des biphones ou même par des monophones (càd les phones indépendants du contexte) [26].

- Lissage des modèles ou *Smoothing* : Il s'agit de combiner des paramètres de modèles spécifiques avec ceux d'autres modèles plus génériques donc disposant de suffisamment de données d'apprentissage pour être bien estimés. La technique du *Maximum a Posteriori* permet de lisser les paramètres des modèles acoustiques indépendants du locuteur (ou de l'environnement) pour une meilleure adaptation au locuteur (ou à l'environnement). La méthode "*Deleted interpolation*" est souvent utilisée dans les systèmes discrets ou semi-continus dépendant du contexte qui disposent d'un nombre important de poids à estimer. Ces poids peuvent être lissés à partir ce ceux des modèles indépendants du contexte [89, 82].
- Partage des paramètres ou *Sharing* : Pour faire face à une faible quantité de données d'apprentissage devant le nombre important de paramètres à estimer et augmenter la robustesse des systèmes, différents niveaux de partage des paramètres ont été proposés dans la littérature. Dans [85], on distingue le partage des modèles, des états, des mélanges et des vecteurs.

Adaptation : En reconnaissance automatique de la parole, il est courant que la base d'apprentissage et de test soient très différentes. Ceci est généralement dû aux variations de locuteurs, du canal, du bruit, etc.

Au niveau acoustique, deux techniques d'augmentation de la robustesse des systèmes existent : les techniques de traitement du signal qui tentent de corriger le signal à décoder et les techniques d'adaptation de modèles qui modifient les paramètres des modèles de façon à représenter au mieux le signal observé.

Pour réduire l'effet du canal, une façon simple de faire consiste à soustraire la moyenne ceps-

trale et normaliser la variance à un. Par ailleurs, la normalization de la longueur du conduit vocal peut être appliquée aux vecteurs de paramètres pour réduire la variabilité inter-locuteurs. L'adaptation des modèles requiert la transcription des données. Lorsque la transcription correcte est disponible, on parle d'adaptation supervisée, sinon elle est non supervisée. Manifestement, l'adaptation non supervisée améliore les performances de la reconnaissance même pour des taux d'erreur initiaux élevés. De ce fait, elle est souvent utilisée dans un contexte de décodage multi-passes.

Les deux approches d'adaptation les plus couramment utilisées :

- Considérer les paramètres du modèle comme des variables aléatoires et les estimer avec le critère du *Maximum a posteriori MAP*. L'inconvénient de cette méthode est que seulement les paramètres pour lesquels il existe des données d'adaptation sont re-estimés.
- Estimer des transformations des paramètres des modèles. La technique *Maximum Likelihood Linear Regression MLLR* trouve une transformation qui maximise la vraisemblance des données d'adaptation.

  Cette méthode est assez appliquée à l'adaptation non supervisée car les transformations peuvent disposer d'un nombre réduit de paramètres (partagés par plusieurs unités phonétiques) et sont de ce fait assez robuste aux erreurs de reconnaissance [63].

Combinaison des sorties : Le paradigme le plus utilisé en reconnaissance de la parole est basé sur la règle du maximum a posteriori. Le décodage par consensus est une alternative à la procédure de "scoring" des hypothèses de recherche. Les réseaux de consensus sont obtenus en fusionnant les noeuds du treillis et en dupliquant les arcs afin d'obtenir un graphe linéaire (voir figure 1.4) . La transcription est obtenue en prenant à chaque fois le mot le plus probable et de concatener les mots retenus.

Bien que plusieurs systèmes de reconnaissance peuvent avoir des taux d'erreurs comparables, leurs sorties présentent des différences notables [63, 88]. Dans le cas où un ou plusieurs systèmes trouvent le mot correct, il est intéressant d'en tenir compte. Pour ce faire la combinaison des sorties par vote est proposée [21]. Cette technique permet de combiner les sorties des différents systèmes en une unique sortie dont le taux d'erreur est inférieur à ceux de tous les systèmes initiaux. Ainsi, pour chaque phrase, les sorties sont alignées de façon à former un *réseau de d'hypothèses de mots* (voir figure 1.5). La décision finale est prise suite à un vote majoritaire et/ou une mesure de confiance.

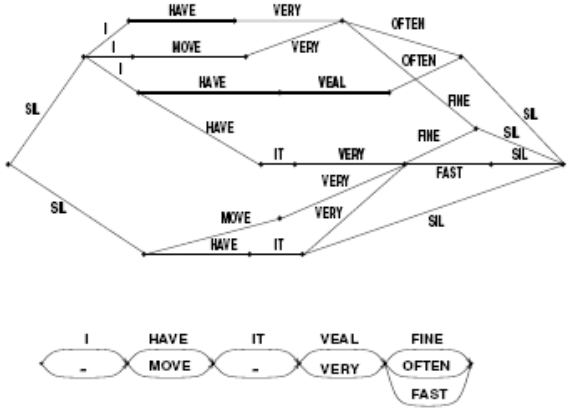


FIG. 1.4 – Décodage par consensus (extrait de [48])

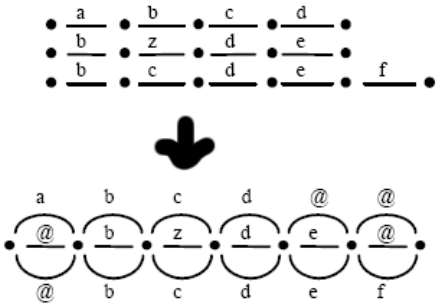


FIG. 1.5 – Combinaison des systèmes (extrait de [21])

1.3 Systèmes de base

Dans ce paragraphe, nous présentons nos systèmes de transcription automatique des émissions radiophoniques qui rentrent dans le cadre général du projet ESTER. Ces systèmes, qui sont entraînés et testés sur l'ensemble des données selon le protocole d'évaluation défini dans le projet, constituent nos systèmes de référence pour les chapitres à venir. Le premier système nommé HTK-Sirocco fait appel à la boite à outils HTK [89] pour l'apprentissage et au décodeur Sirocco [30] pour le décodage. Le second système est entièrement basée sur les outils Sphinx à savoir SphinxTrain [82] pour l'estimation des paramètres et Sphinx3-0.5 [9] pour l'évaluation.

1.3.1 La campagne Ester

Aux USA, une longue tradition de campagnes d'évaluation dans le domaine des technologies vocales et du langage naturel [66] a permis de mettre à la disposition de la communauté scientifique des corpus de grande taille et des protocoles d'évaluation fiables [13, 14]. Une première campagne d'évaluation en France des systèmes de dictée vocale a été lancée dans les années 1990 [19]. La campagne ESTER s'inscrit dans la continuité de cette logique d'évaluation et de développement de corpus et de protocoles. Cette campagne est dédiée à la transcription enrichie et à l'indexation de journaux radiophoniques de langue française. Elle défini un nombre de tâches d'évaluation (transcription, segmentation, extraction) et des données de test et d'apprentissage (audio et textuelles).

Tâches

La campagne ESTER d'Evaluation des Systèmes d'indexation d'Emissions Radiophoniques [29, 25] s'organise autour de trois tâches : la transcription (T), la segmentation (S) et l'extraction d'informations (E). Les tâches T et S constituent le noyau de la campagne tandis que la tâche E regroupe des thèmes prospectifs. Bien que ces tâches ne soient pas indépendantes, les tâches sont évaluées séparément avec une métrique propre.

La tâche de transcription (T) consiste à produire une transcription à partir du signal audio. Les transcriptions sont évaluées en termes de mots erronés (Word Error Rate) calculés à partir des transcriptions de référence après normalisation (même notation des chiffres, majuscule ou minuscule, abbréviations, ..).

La tâche transcription en temps réel (TTR) vise à évaluer les systèmes de transcription en temps réel sur un monoprocesseur standard.

La tâche de segmentation (S) consiste à détecter, suivre et grouper des événements sonores et se divise en trois sous-tâches :

- Suivi d'événements sonores (SES) : détecter dans un document les plages correspondant à un événement sonore (parole et musique).
- Segmentation et regroupement des locuteurs (SRL) : découper le flux audio en tours de parole et regrouper les plages associées à un même locuteur.
- Segmentation et vérification du locuteur (SVL) : détecter les zones du document où un locuteur donné connu à l'avance est présent.

La tâche d'extraction (E) permet de juger l'enrichissement des transcriptions de plus haut niveau comme les entités nommées (EN), une segmentation et structuration du document en thèmes cohérents, et le suivi de ces thèmes (comme SVL pour un thème et non un locuteur).

Ressources

Trois types de ressources sont fournies dans le cadre du projet ESTER : un corpus de parole transcrite, des ressources textuelles (journaux et transcriptions) et des ressources audio non-transcrites. Les ressources audio sont présentées dans le tableau 1.1.

Source	Train/Dev	Test	non transcits
France Inter	32h20/2h	2h	300h
France info	8h/2h	2h	1000h
RFI	23h/2h	2h	500h
RTM	18h/2h	2h	100h
Surprise	-	2h	-
Total	90h	10h	2000h
Période	1998-2000/2003	2004	2004

TAB. 1.1 – Ressources acoustiques

Ainsi, la campagne se base sur l'utilisation de 100 heures d'émissions radiophoniques, dont 82 heures d'apprentissage, 8 heures de développement et 10 heures de test. La source d'une chaîne radio "surprise" reste inconnue des participants pour mesurer si la connaissance préalable des sources influence les performances. Il s'agit de Radio Classique.
Les données d'apprentissage, qui sont transcrites manuellement (en utilisant le logiciel Transcriber [49]), incluent des tours de parole avec l'identité des locuteurs, les conditions acoustiques et divers éléments tels que la présence de bruit.
Les ressources textuelles pour la campagne correspondent aux années 1987 à 2003 du journal *le monde*, augmentés de transcriptions des débats du Conseil Européen, soit au total à peu près 400 millions de mots.

1.3.2 Système HTK-Sirocco

Ce système, initialement conçu pour participer à la campagne Ester, sera utilisé dans les prochains chapitres pour évaluer des techniques de réduction du temps de reconnaissance. Les étapes de développement d'un tel système sont :

Segmentation audio

Les données sonores sont fournies sous forme d'enregistrements de durées variant de 15 à 60 minutes. Une première segmentation "en groupes de souffle" est nécessaire. Cette segmentation permet d'éviter les problèmes de discontinuité linguistique causés par le changement de locuteur. Chaque enregistrement est découpé en segments de parole séparés par des silences courts et les segments de silence long sont supprimés.
Les fichiers sonores sont accompagnés par leur transcription en mots. Pour produire la transcription

phonétique, un alignement forcé est réalisé. Les modèles acoustiques utilisés pour l'alignement sont de type monophone à 64 gaussiennes par état.

Apprentissage des modèles acoustiques

Les paramètres acoustiques sont formés par des coefficients cepstraux MFCC (12MFCC+E) et leurs dérivées première et seconde.La paramétrisation est réalisée avec le logiciel Spro4.0 [28].

Deux types de modèles acoustiques sont construits et évalués sur la base de test Ester : 40 modèles indépendants du contexte et 13000 modèles dépendants du contexte. Ces derniers modèles se partagent 6207 états liés et contiennent 32 gaussiennes par état. Le logiciel Sirocco ne prend pas en compte les co-articulations entre les mots par conséquent les modèles contextuels sont intra-mots. L'estimation de leurs paramètres est effectuée au moyen la boite à outils HTK [89].

Décodage

Le moteur de décodage est Sirocco [30]. Ce décodeur, développé à l'ENST, est basé sur l'algorithme de décodage synchrone au temps de type Viterbi. Le modèle de langage est un trigramme et contient a peu près 4 millions de trigrammes et 4 millions de bigrammes. Le dictionnaire est formé par 65k mots distincts (dont 118000 prononciations différentes). Les WER obtenus avec des monophones et les triphones sur toute la base de test (10 heures) sont :

Modèles	WER (%)
128	45.6
256	43.3

TAB. 1.2 – Résultats des monophones à 128 et 256 gaussiennes par état

Radio	WER (%)
Classique	32.8
Culture	43.9
Inter1	35.3
Inter2	41.7
RFI	41.0
RTM	49.9
Total	41.5

TAB. 1.3 – Résultats des triphones à 32 gaussiennes par état

Nous remarquons que les performances des triphones sont peu différentes de celles de monophones et que le taux d'erreur est dans les deux cas assez élevé. Ceci peut s'expliquer par l'absence d'une normalisation des paramètres vu la variabilité des conditions d'enregistrements (plusieurs microphones, enregistrement en studio ou à l'extérieur ou au téléphone, ..) et des locuteurs.

Signalons aussi que le temps de décodage des monophones est de l'ordre de 10 fois le temps réel, celui des triphones est nettement plus important (supérieur à 60 fois le temps réel), ce qui les rend beaucoup moins pratiques. Ainsi, nous avons choisi d'utiliser ce système avec des modèles indépendants du contexte pour évaluer certaines méthodes de réduction du temps. Lesquelles méthodes sont basées sur l'utilisation de modèles disposant d'un nombre important de distributions par état. Ces méthodes sont développées dans le chapitre 4.

1.3.3 Système Sphinx

Ce système a été expérimenté postérieurement à la campagne ESTER. Il utilise les outils logiciels libres SphinxTrain [82] et Sphinx3-0.5 [9] développés l'université de Carnegie Mellon pour la construction des modèles acoustiques et le décodage. Le choix de ce système est basé essentiellement sur la rapidité relative du décodeur Sphinx3-0.5 comparé à Sirocco ainsi qu'à sa capacité de prendre en considération les contextes entre les mots.

Extraction des paramètres

Les paramètres acoustiques sont de type MFCC (12MFCC+E) et leurs dérivées première et seconde. Ces vecteurs sont normalisés par rapport à la moyenne et la variance sur une phrase. La normalisation par rapport à la variance permet de diminuer la variabilité par rapport au locuteur, ce qui est intéressant vu le nombre important de locuteurs intervenant dans la base. L'extraction des paramètres est réalisée avec l'outil Wave2feat [82].

Modèles acoustiques

Nous avons construit 36 modèles indépendants du contexte et 80000 modèles dépendants du contexte à 6108 états liés. Ces derniers prennent en compte les contextes inter-mots et disposent chacun de 32 gaussiennes par état.
Les monophones sont évalués sur 1 heure de parole issue du canal Radio Classique de la base de test d'Ester. Etant beaucoup plus performants sur radio Classique, les triphones sont testés sur toute la base (10 heures) et seront retenues pour les expérimentations avec le système Sphinx. Les résultats sont rapportés dans les tableaux 1.4 et 1.5.

Modèles	WER (%)
32	37.7
64	36.7
128	35.6
256	35.2
512	35.3

TAB. 1.4 – Résultats des monophones sur une heure de test

Radio	WER1(%)	WER2(%)	WER3(%)
Classique	28.7	26.3	25.9
Culture	40.2	39.1	38.9
Info	31.5	29.1	28.9
Inter	35.6	33.6	33.5
RFI	36.4	32.8	32.5
RTM	43.8	39.1	38.5
Total	36.2	33.4	32.9

TAB. 1.5 – Résultats des triphones sur 10 heures de test

$WER1$ est le taux d'erreur lorsqu'on utilise 4 millions de bigrammes et 4 millions de trigrammes, $WER2$ correspond à 17 millions de bigrammes et 16 millions de trigrammes et $WER3$ est obtenu lorsque les données de développement sont utilisées pendant l'apprentissage des modèles acoustiques. Le système initial (WER1) tourne à 12 fois le temps réel. Ce système sera considéré comme système de base dans les expériences des chapitres 3, 5 et 6.
Etant donné, le grand nombre global de distributions (6108 * 32), pour optimiser le temps de décodage de ce système nous avons opté pour des techniques adéquates dont la sélection des gaussiennes basées sur la classification de type k-moyenne et la sous-quantification vectorielle. Ces méthodes feront l'objet des chapitres 5 et 6.

Modèles Spécifiques

La base d'apprentissage est transcrite manuellement. Nous avons construit des modèles contextuels par genre de locuteur (homme/femme). Leur évaluation nous a permis d'obtenir un gain de taux de mots erronées absolu de 0.1%. Les expérimentations avec des modèles dépendants de la bande passante ne sont pas concluantes.

Répartition des Gmms

La réalisation acoustique de certains phonèmes, notamment les occlusives, est effectué en 5 phases : un silence correspondant à l'occlusion, une explosion due au relâchement de l'air comprimé, une aspiration, une éventuelle vibration des cordes vocales, et les transitions vers le son vocalique suivant. Par conséquent, pour améliorer le WER des modèles contextuels, nous avons développés des modèles contextuels à 5 états. Leurs performances sur une heure de test (radio classique) sont comme suit :

Modèles	WER 3 états (%)	WER 5 états (%)
16	31.0	27.6
32	28.7	27.1

TAB. 1.6 – Résultats des triphones à 3 et à 5 états

En se basant sur l'hypothèse que l'état du milieu est souvent le moins dépendant du contexte (pour tous les phonèmes), nous avons essayé de limiter le nombre d'états liés au milieu des triphones et augmenter celui des autres états. Les résultats pour les triphones à 16 gaussiennes par état en maintenant à chaque fois un nombre global de 6000 états liés sont :

états liés du milieu	WER (%)
2	26.9
4	27.1
8	27.2
-	27.5

TAB. 1.7 – Résultats de la limitation du nombre de GMMs

On peut constater, pour l'état du milieu, que seulement deux états liés suffisent, ce qui semble assez logique puisque ce dernier est moins dépendant du contexte que les autres états et donc moins variable. Par conséquent, il est plus intéressant d'attribuer des états liés aux autres états afin de modéliser leurs variabilité qui est plus importante.

1.4 Conclusion

Dans ce chapitre, entièrement dédié à la transcription automatique de la parole dans les émissions radio, nous avons présenté quelques concepts généraux avant de passer à des techniques avancées qui relèvent de l'état de l'art du domaine. Puis nous avons décrit les deux systèmes de reconnaissance grand vocabulaire que nous avons expérimenté. Disposant des mêmes ressources linguistiques, nous avons souligné les différences entre ces deux systèmes qui se situent au niveau de la paramétrisation et la prise en compte du contexte entre les mots. En outre, pour le second système, nous avons constaté une réduction du taux d'erreur lorsque la quantité de données et/ou les ressources linguistiques sont plus abondantes.

Chapitre 2

Segmentation audio

2.1 Introduction

Souvent dans les émissions radio, plusieurs types de signaux sont présents : parole, musique, parole + musique, jingles, silence, etc. Suivant le type d'application, différentes segmentations sont envisageables :

- segmentation musique/non-musique pour le traitement de la musique (classification par genre ou par instrument, etc),

- séparation parole/fond musical des segments de parole + musique pour le mixage audio ou la séparation des sources, etc,

- segmentation parole/non-parole pour la transcription orthographique et éventuellement la recherche d'information.

Ce travail s'insère dans le cadre du développement d'un système de transcription automatique des émissions radio. De ce fait, on se propose dans un premier temps de réaliser une segmentation du flux audio dans le but d'extraire les parties contenant de la parole ou de la parole mélangée avec la musique. La construction d'un tel système suscite le choix d'une paramétrisation adéquate aussi bien pour la parole que pour la parole et la musique.

Les coefficients MFCC (MEL Frequency Cepstral Coeficients) ont prouvé leur efficacité en traitement automatique de la parole. Leur succès provient, entre autres, de l'utilisation de l'échelle MEL qui favorise les basses fréquences. Dernièrement, [44] a montré que les coefficients MFCC peuvent aussi représenter la musique sans pour autant se prononcer sur leur optimalité. A priori, l'échelle MEL n'est pas la plus appropriée pour la musique puisqu'il peut y avoir autant d'information en basses fréquences qu'en hautes fréquences.

Par conséquent, on se propose de trouver la meilleure paramétrisation de la musique entre MFCC et LFCC (Linear Frequency Cepstral Coeficients). Les paramètres retenus seront combinés avec les coefficients MFCC pour mieux discriminer la parole et les mélanges de parole et de musique.

2.2 État de l'art

Les méthodes de segmentation temporelle du signal audio peuvent être subdivisées en trois catégories. Celles qui détectent le silence en se basant sur l'énergie du signal, celles qui calculent les distances entre les trames successives pour détecter les frontières, et celles basées sur les modèles. Ces dernières réalisent la segmentation et la classification en même temps. Pour toutes ces méthodes, deux étapes de traitement sont nécessaires : l'extraction des paramètres et la classification.

2.2.1 Extraction des paramètres

Plusieurs systèmes de segmentation ont été rapportés dans le passé [79] [44] [68]. Ils se basent sur différents types de paramètres extraits à partir du signal audio. Ces paramètres peuvent être divisés en trois catégories principales. Les paramètres temporels, les paramètres fréquentiels et les paramètres issus de ces deux domaines.

- **Les paramètres temporels** incluent principalement l'énergie et le taux de passage à zéro (ZCR) [79, 80, 68]. Ces paramètres ont été initialement employés en reconnaissance de la parole. Les variations brusques du ZCR renseignent sur les transitions entre sons voisés et sons non voisés et donc la présence du signal de parole. L'énergie du signal, souvent utilisée pour détecter le silence, permet de discriminer la parole de la musique. En effet, l'énergie à court terme varie beaucoup plus pour la parole que pour la musique.

Comme l'énergie et le ZCR sont complémentaires, leur association permet de mieux distinguer les sons voisés des non voisés ; un ZCR faible (important) et une énergie importante (faible) signifient un son (non) voisé [70].

- **Les paramètres fréquentiels** sont souvent issus de la densité spectrale de puissance [70]. On distingue :

  – le centroïde spectral : c'est le centre de gravité de la densité spectrale de puissance. Il est plus élevé pour la musique car les hauteurs des sons sont réparties fréquentiellement sur une zone plus importante pour la musique que celle pour la parole.

  – le flux spectral mesure la variation de l'amplitude du spectre. Cette variation est plus importante pour la musique.

  – le *spectral Roll off Point*. Ce point correspond à 95% de la puissance de la densité spectrale de puissance. Cette mesure caractérise les transitions voisé-non voisé en parole. Pour les sons non voisés une partie importante de l'énergie est localisée en hautes fréquences. Pour les sons voisés l'énergie est concentrée en basses fréquences.

- Parmi les paramètres issus de **paramétrisation temporelle et fréquentielle**, on peut citer l'énergie à quatre hertz. En parole, les changements syllabiques ont souvent lieu autour de cette fréquence.
La musique possède une variation d'énergie à 4hz plus faible que la parole [80].

Les paramètres MFCC (Mel Frequency Cepstral Coefficients), souvent utilisés en traitement de la parole ont montré leur capacité à modéliser la musique [44]. Une étude comparative d'un ensemble de paramètres (énergie, taux de passage par zéro, pitch et MFCC) dans [7] a montré la supériorité des coefficients cepstraux pour la discrimination parole/musique.

2.2.2 Classification

Les systèmes de segmentation de l'état de l'art font souvent appel à des techniques tels que les modèles de Markov cachés (HMM) [75], les Modèles de Mélanges Gaussiens (GMM) [81, 22, 71], les K Plus Proches Voisins (KPPV)
[80], les réseaux de neurones , plus récemment les machines à vecteurs de support ("Support Vector Machines") (SVM) [31] ainsi que de leur combinaison [2].
Dans les approches basées sur les modèles (HMM ou GMM), ces derniers sont estimés sur chaque classe et la classification est opérée sur des fenêtres glissantes. Dans [81] plusieurs configurations de GMMs sont évaluées. Les meilleurs résultats sont obtenus avec un mélange gaussien à matrices de covariance diagonales. Dans [75] on utilise des HMMs pour segmenter les émissions radio en trois classes : parole, musique et parole sur fond musical. Des performances de bonne classification de 80.2% sont obtenus pour la musique et de 96.3% pour la parole.
Dans l'approche KPPV, le vecteur d'entrée est affecté au label le plus fréquent parmi ses k plus proches voisins dans le dictionnaire. La distance Euclidienne est souvent employée pour la mesure du voisinage. Dans [20], un taux de segmentation de 95% est reportée.
[45] propose un schéma à deux étapes pour segmenter le flux audio en parole, musique, bruit d'environnement et silence. En premier lieu, le signal est décomposé en parole/non parole au moyen de la technique des KPPV et la quantification vectorielle. En second lieu, les segments de non parole sont répertoriés en musique, silence ou bruit d'environnement en appliquant un ensemble de règles. La précision de ce système est de 96%.

2.3 Système de segmentation

Classiquement, la segmentation du flux audio est réalisée en deux temps. Dans un premier temps, on extrait du signal les paramètres jugés pertinents. Ces derniers doivent caractériser au

mieux les classes à discriminer. Puis un processus de segmentation/classification permet d'affecter chaque partie du signal à une classe.

Dans le cadre du pôle de compétitivité Cap-Digital et du réseau d'excellence KSpace, un système de segmentation audio a été réalisé. L'objectif d'un tel système est d'extraire la parole et la parole mélangée avec la musique [93] .

2.3.1 Base de données

Nous avons utilisé une base de données de l'émission de variété télé *Le Grand Échiquier*. L'enregistrement dure trois heures et demi. Il contient de la parole, de la musique, la combinaison des deux (chants, jingles, ..) et des sons divers tels que les rires, les applaudissements, les effets spéciaux. Après avoir étiqueté manuellement cette base (au moyen de l'outil transcriber [49]), nous l'avons découpée comme suit : 2h30 pour l'apprentissage et le reste pour l'évaluation. Le contenu de ces deux parties est explicité sur la Figure 2.1.

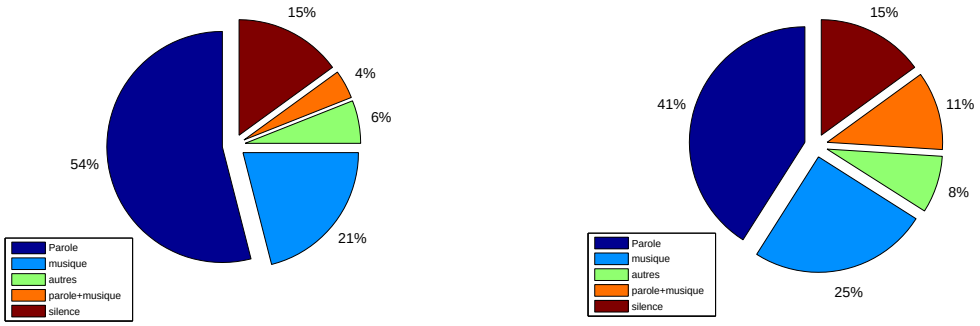


FIG. 2.1 – Contenu des bases d'apprentissage (à gauche) et de test (à droite)

2.3.2 Paramétrisation

Notre paramétrisation est basée sur les coefficients cepstraux. Le signal audio est extrait de la séquence vidéo, échantillonné à 16khz, puis les coefficients MFCC (MEL Frequency Cepstral Coefficients) et LFCC (Linear Frequency Cepstral Coefficients) sont calculés à partir d'un banc de 24 filtres. Ces filtres, de type MEL (échelle logarithmique) ou linéaires, sont appliqués toutes les 10 millisecondes sur une fenêtre glissante de durée 20 millisecondes. Aux coefficients statiques (12 MFCC ou LFCC + énergie) nous rajoutons les dérivées première et seconde, ce qui permet d'obtenir des vecteurs de paramètres de dimension 39.

2.3.3 Classification

Comme ces travaux sont menés dans un objectif de reconnaissance de la parole, une approche de type GMM est adoptée. Ainsi, chaque classe (parole, musique, parole + musique et autres) est

modélisée par un Modèle de Mélange Gaussien.
La classification, réalisée toutes les 10 millisecondes, est basée sur la règle suivante : soient N classes $C_1, C_2, .., C_N$ et le vecteur de test O. Le vecteur O est affecté à la classe la plus vraisemblable càd celle pour laquelle la vraisemblance $P(O/C_i)$ est maximale.
Un nombre de composants de 256 gaussiennes par GMM est choisi empiriquement.

2.3.4 Mesure de performance

L'évaluation est réalisée trame par trame. Les performances sont mesurées sur la base du rappel (R) et de la précision (P).

$$R = \frac{\sum_c T(c|c)}{\sum_c T(c)}; \quad P = \frac{\sum_c T(c|c)}{\sum_c T(c) + T(c|nc)} \tag{2.1}$$

où $T(c|nc)$ (ou $T(c|c)$) est la durée des segments où l'événement c a été détecté à tort (ou à raison), $T(nc|c)$ la durée où c n'a pas été détecté à tort, $T(c)$ la durée où c est présent et $T(nc)$ le temps où c n'est pas présent.
Les événements de base sont la parole, la musique et la parole+musique. Suivant les tests, d'autres événements sont évalués tels que la non parole ou non musique ou encore non parole + non musique (appluadissements, rires, ..). Les performances des systèmes seront comparées sur la base de la F-mesure définie par :

$$F - mesure = \frac{2 * R * P}{R + P} \tag{2.2}$$

Les temps seront mesurés en secondes. Dans les expériences qui suivent, on notera les valeurs de $F - mesure$ pour différentes $marges$, où $marge$ correspond à l'écart toléré (des limites) entre la segmentation automatique et la segmentation manuelle (en millisecondes).

2.4 Paramétrisation de la parole

Un système de segmentation parole/non parole a été développé. Il est basé sur une mise en compétition de deux modèles GMMs de parole et de non parole à 256 gaussiennes chacun. La figure 2.2 illustre les valeurs de F-mesure en fonction de la marge pour les deux paramétrisations MFCC et LFCC.

D'après la courbe de segmentation parole/non parole, les paramètres MFCC sont plus performants que les LFCC pour la segmentation en deux classes. Pour vérifier qu'ils le sont en détection de la parole, nous avons tracé les histogrammes de détection de la parole (marge = 700ms). On en déduit que les MFCC sont meilleurs en détection de la parole. Les performances de ces deux paramétrisations pour la classification en non parole sont comparables.

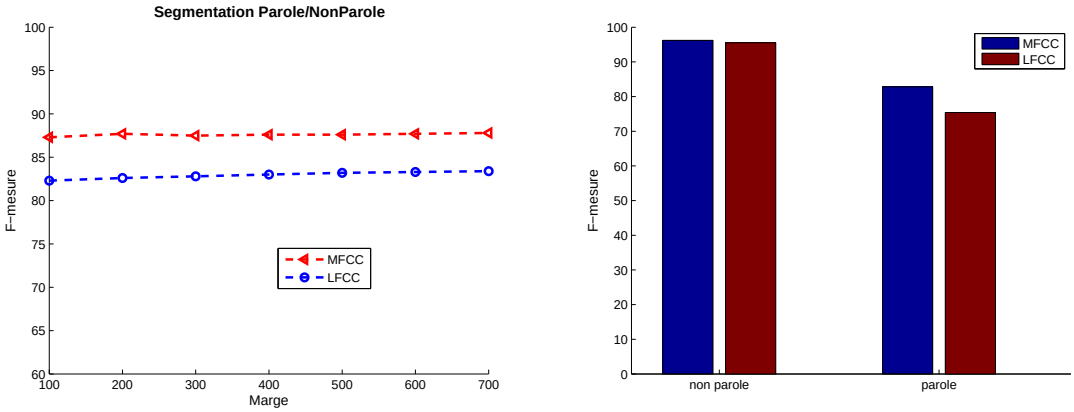
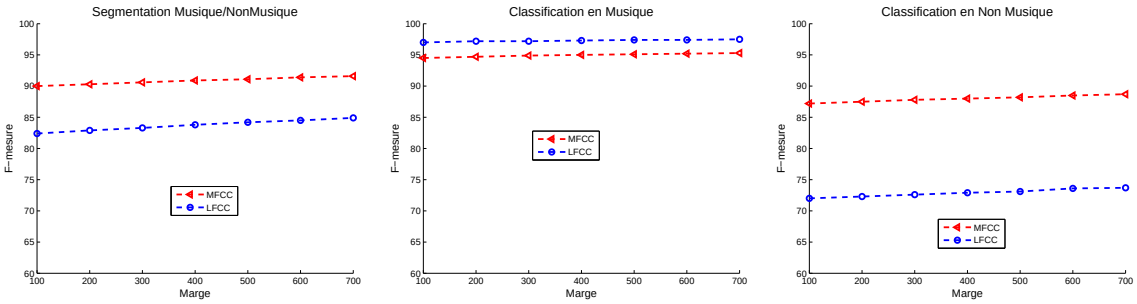


FIG. 2.2 – Segmentation parole/non parole

2.5 Paramétrisation de la musique

Le système de segmentation musique/non musique est également basé sur des GMM à 256 composantes gaussiennes.



(a) (b) (c)

FIG. 2.3 – Segmentation musique/non musique

D'après la figure 2.3 :

- Le système de segmentation musique/non musique est performant lorsque la paramétrisation est basée sur les MFCC (Fig. 2.3a). Ceci peut s'expliquer par la bonne classification des segments de non musique (Fig. 2.3c) formés majoritairement par la parole.
- Les segments de musique sont mieux classés dans le cas où des paramètres LFCC sont employés (Fig. 2.3b). Cependant, lorsque les coefficients MFCC sont utilisés ce taux est très peu dégradé ($> 94\%$).

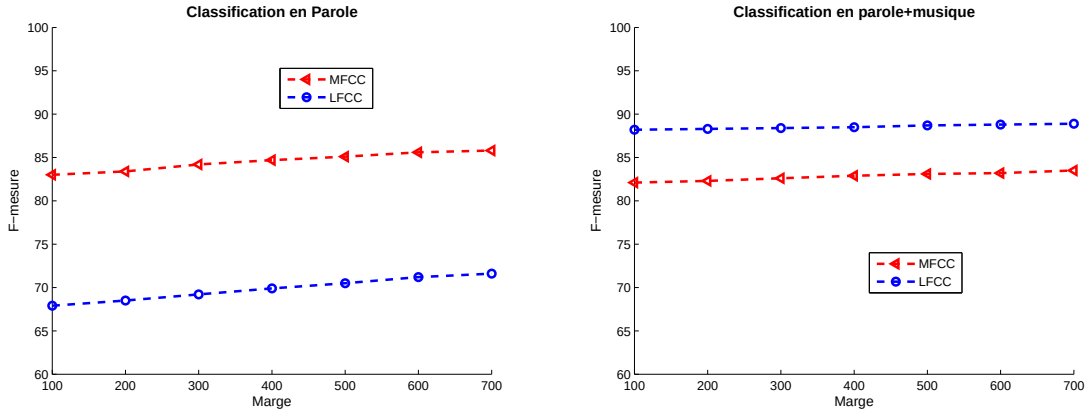
2.6 Paramétrisation de la parole et parole+musique

Comme nous l'avons déjà précisé, l'objectif du système de segmentation est de trouver une paramétrisation adéquate pour la parole et pour la parole mélangée avec la musique. Les expériences

ci-dessus nous permettent de constater que les coefficients MFCC sont plus adéquats que les coefficients LFCC pour la détection de la parole et le contraire pour la musique. On se demande alors laquelle de ces deux paramétrisations est meilleure pour un mélange de parole et de musique, ou encore faut-il les combiner pour mieux détecter la parole et la parole + la musique.

2.6.1 Systèmes de MFCC/LFCC

Nous avons développé deux systèmes de segmentation parole/ musique/ parole+musique/ autres en utilisant les paramétrisations MFCC et LFCC. Sur la figure 2.4 sont reportées les performances du système en classification en parole et en parole + musique.



(a) (b)

FIG. 2.4 – Performances des paramétrisations MFCC et LFCC

On remarque un écart important entre la classification de la parole en utilisant les MFCC et celle en utilisant les LFCC (figure 2.4a), ce qui confirme les résultats du paragraphe 2.4.
Pour la classification en parole + musique, les LFCC sont plus performants que les MFCC (figure 2.4b). Néanmoins, la différence n'est pas très importante.

Bien qu'intéressantes, ces constatations ne nous permettent pas de trancher entre MFCC et LFCC car dans les données de test, on ne connaît pas a priori les proportions de segments de parole et de parole + musique. D'où l'intérêt de faire appel à des techniques de fusion dans l'espoir de trouver une meilleure combinaison qui permet à la fois de détecter aussi bien la parole pure que la parole mélangée avec du bruit.

2.6.2 Combinaison des paramètres

La fusion des paramètres est réalisée par une simple concatenation des paramètres MFCC et LFCC. On obtient des vecteurs de dimension 78 (39*2). La segmentation en 4 classes est reconduite

en utilisant ces nouveaux vecteurs de paramètres. Aussi, nous avons relevé les performances en classification de la parole et de la parole + musique.

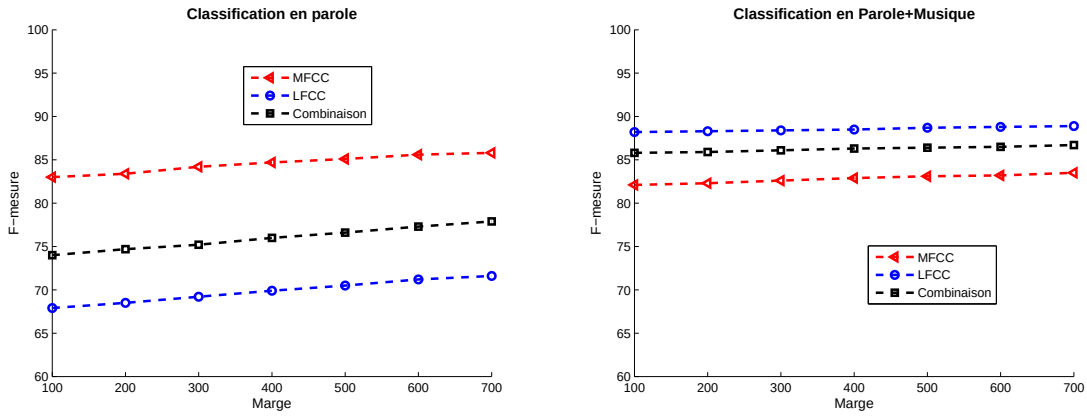


FIG. 2.5 – Classification parole/musique/parole+musique/autres

La figure Fig. 2.5 montre que les performances du système issu de la de combinaison des paramètres sont entre celles du système MFCC et celles du système LFCC.

2.6.3 Combinaison des scores

Disposant des systèmes MFCC pour la parole et LFCC pour la parole et la musique, la combinaison des scores est réalisée en leur affectant des poids différents afin de privilégier l'une ou l'autre des paramétrisations. A chaque instant t, si $P(O_{MFCC};t)$ et $P(O_{LFCC};t)$ sont les vraisemblances d'une observation O calculées avec les systèmes MFCC et LFCC, alors son score de fusion peut s'exprimer par : $P(O_{fusion};t) = \lambda P(O_{MFCC};t) \times (1-\lambda)P(O_{LFCC};t)$
Le poids λ permet de donner plus d'importance à une modalité ou à l'autre.

Nous avons fait varier le poids λ entre 0 et 1. D'après la figure Fig.2.6, on peut constater que les performances de détection de la parole se dégradent lorsque λ augmente et le contraire pour la parole + la musique. Ce qui est en concordance avec les résultats précédents. Mais, vue la forte dégradation des performances de détection de la parole, on est tenté par l'utilisation de valeurs faibles de λ.
Le poids λ correspondant au meilleur compromis LFCC/MFCC est dans notre cas $\lambda = 0.1$. Dans le cas général, ce paramètre peut être déduit suite à des tests sur une base de développement dont on suppose que les proportions en parole et en parole+musique sont à peu près les mêmes que ceux de la base de test.

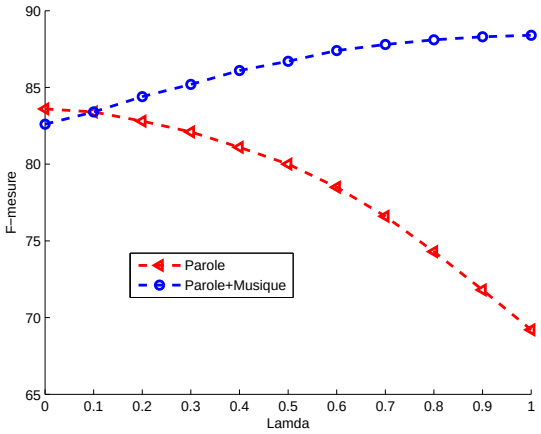


FIG. 2.6 – Performances de la classification de la parole et de la parole + la musique en fonction du poids λ

2.6.4 Combinaison des décisions

L'idée consiste à fusionner les meilleurs systèmes de segmentation parole/non parole (P/NP) et parole+musique/non parole+musique (MP/NMP) pour en déduire une meilleure segmentation en 4 classes : parole (P), parole+musique (MP), musique (M) et autres (A).

Les systèmes de segmentation P/NP et MP/NMP sont déduits des systèmes de classification en 4 classes, basés respectivement sur les MFCC et les LFCC. Ainsi par exemple pour le système basé sur les coefficients MFCC les segments de parole + musique (MP) sont labéllisés parole et les segments de musique (M) ou de non parole et non musique (A) sont labéllisés non parole (NP). Les règles de fusion, appliquées à chaque trame (10ms) sont explicitées dans le tableau Tab. 2.1.

TAB. 2.1 – Règles de combinaison des décisions

Système MFCC (P/NP)	Fusion (P/M/MP/A)	Système LFCC (MP/NMP)
P $\rightarrow$ P	P	NMP $\leftarrow$ P
MP $\rightarrow$ P	MP	MP $\leftarrow$ MP
A $\rightarrow$ NP	A	NMP $\leftarrow$ A
M $\rightarrow$ NP	M	MP $\leftarrow$ M

Les performances du système issu de la fusion (marge = 300ms) sont représentées sur la figure Fig. 2.7.

On remarque que la combinaison des systèmes apporte une amélioration par rapport aux coefficients LFCC pour la détection de la parole sans dépasser les performances avec les MFCC ce qui confirme leur supériorité en représentation de la parole. Pour la classification de la parole + musique les

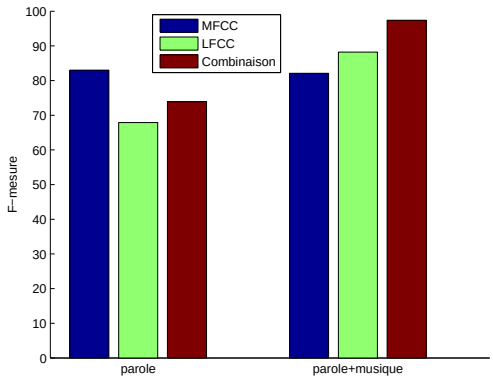


FIG. 2.7 – Histogrammes de la fusion des décisions

performances de la fusion dépassent celles des deux systèmes de base. Par conséquent, la fusion des décisions constitue le meilleur compromis dans le cas où la nature des données de test est inconnue.

2.7 Conclusion

Dans le cadre d'une transcription automatique des émissions radio diffusées, une segmentation audio est réalisée. L'objectif d'une telle segmentation est de caractériser au mieux les segments de parole et de parole+musique qui seront par la suite transcrits. Les paramètres testés sont les coefficients MFCC, LFCC et leurs combinaisons. Les résultats de nos expériences montrent que les coefficients MFCC sont plus adéquats pour la discrimination de la parole, les coefficients LFCC le sont pour la musique+parole et leur combinaison l'est pour un flux audio contenant à la fois la parole et la parole mélangée avec de la musique.

Chapitre 3

Techniques de reconnaissance rapide

3.1 Introduction

La reconnaissance de la parole grand vocabulaire est une tâche qui consomme beaucoup de ressources. L'évaluation de la répartition des ressources et du temps CPU pendant le décodage montre qu'environ 75% de la mémoire est utilisée pour stocker les hypothèses de recherche [83] et qu'entre 30% et 70% [24] du temps CPU est occupé par le calcul des vraisemblances.

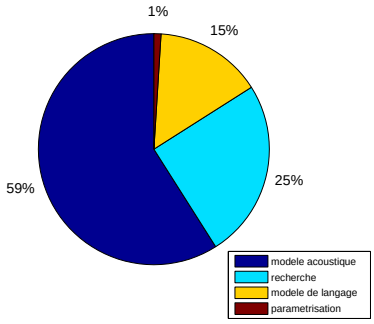


FIG. 3.1 – Consommation des ressources en temps CPU (d'après [83])

Ainsi, pour réduire le temps de reconnaissance, les méthodes proposées dans la littérature se sont focalisées sur :

1. la réduction de l'espace de recherche. En effet, l'élimination des hypothèses peu prometteuses permet de diminuer le temps de propagation et de réduire l'espace mémoire qu'elles occupent. Pour ce faire, les techniques d'élagage et de prédiction sont souvent employées.
2. l'optimisation du temps de calcul des vraisemblances. Deux possibilités existent : la première sélectionne seulement les composantes dont l'apport au calcul des vraisemblances est significatif, la seconde applique des méthodes rapides de calcul de toutes les composantes des mélanges.

Ce chapitre passe en revue la littérature des méthodes existantes de réduction du temps du décodage. Certaines approches qui sont déjà développées dans le système Sphnix sont évaluées. Il est organisé comme suit : D'abord, les techniques de réduction de l'espace de recherche et d'introduction anticipée du modèle de langage seront décrites. Ensuite les méthodes d'accélération du calcul des vraisemblances seront présentées. On s'intéressera en particulier aux techniques de sélection de gaussiennes (GS) par classification hiérarchique et k-moyennes. Enfin quelques approches de sous-quantification vectorielles seront détaillées.

3.2 Recherche lexicale

Dans les Systèmes de Reconnaissance de la Parole (SRP) grand vocabulaire, la taille du lexique, et par la suite celle de l'espace de recherche, sont souvent très importantes. De ce fait, des méthodes de suppression des hypothèses peu probables sont nécessaires. Dans la littérature, les techniques d'élagage et de prédiction d'hypothèses sont les plus utilisées.

3.2.1 Élagage de l'espace de recherche

Lors du décodage, les scores de certaines hypothèses partielles peuvent être faibles. Ces hypothèses ne constituent pas des candidats potentiels et peuvent, par conséquent, être élaguées [55]. L'élagage est généralement basé sur le score ou un nombre maximal d'hypothèses. En revanche, il peut être appliqué à différents niveaux : mot, phone ou état (voir figure 3.2).
Dans certains SRP, on trouve d'autres types d'élagage tels que le nombre maximal de mots dans l'historique [72] ou encore la suppression d'hypothèses dont la fin ne sera jamais atteinte [30](quand le nombre d'observations restantes est faible).

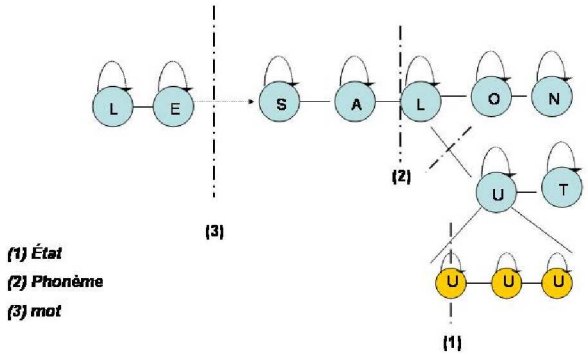


FIG. 3.2 – Niveaux d'élagage : mots, phones et états

Recherche en faisceau

Les hypothèses dont les probabilités ou scores sont faibles par rapport au meilleur score sont supprimées.

Soit $Q(t,s)$ le score acoustique de l'état actif s à l'instant t. $Q(t,s\prime) = \max_{s} Q(t,s)$ est le meilleur score obtenu à l'instant t par l'état actif $s\prime$. A l'instant t, les hypothèses dont les scores $Q(t,s)$ vérifient : $Q(t,s) < \alpha * Q(t,s\prime)$ sont élaguées. α est le coefficient d'élagage.

Généralement, lorsque les modèles acoustiques et linguistiques représentent bien le signal [87], la différence entre le meilleur score et les autres scores est importante. Par conséquent, maintes hypothèses sont élaguées et seulement quelques noeuds restent actifs.

Dans le cas contraire, quand par exemple les conditions d'apprentissage et de test sont différentes, tous les scores sont faibles. Ainsi, de nombreuses hypothèses restent actives, ce qui peut engendrer une augmentation du temps de décodage.

N meilleures hypothèses

Cette méthode, qui est une variante de la première, consiste à retenir à chaque instant un nombre maximal N de meilleures hypothèses (états, phones ou mots). Le nombre N est fixé a priori. La limitation du nombre de mots simultanés permet de contrôler la taille mémoire utilisée à chaque instant [87]. En effet, la taille mémoire nécessaire au stockage des hypothèses est directement liée au nombre de mots retenus.

Cet élagage est particulièrement utile dans le cas où le nombre d'hypothèses actives demeure important malgré la recherche en faisceau. Cette situation peut être rencontrée lorsque les données sont bruitées ou quand les conditions d'apprentissage et de test sont différentes.

Quelques expériences

Les expériences sont réalisées avec le système Sphinx (1.3.3) et portent sur une heure de parole de la base de test ESTER. Dans la première expérience le seuil d'élagage (de mots) est faible et le nombre de mots simultanés (Nmax) est élevé, dans la deuxième c'est le contraire. Le tableau 3.1 rapporte les temps cpu de décodage (TG), de calcul des vraisemblances (TV) et de recherche (TR), ainsi que le taux de mots erronés (WER). Les deux expériences sont réalisées sur une même machine de caractéristiques : processeur 3.6 Ghz, 6 Go de RAM, 50Go de disque dur et 512 Kb de mémoire cache.

(seuil,Nmax)	TG	TV	TR	WER (%)
(1e-55,100)	14.42	3.38	10.93	28.7
(1e-35,20)	3.95	2.59	1.38	28.7

TAB. 3.1 – Elagage des hypothèses de mots

L'intervalle de confiance à 95% est de [28.35 ; 29.11].

On peut constater que pour un même taux de mots erronés, la différence de temps de recherche entre les deux expériences est considérable. En effet, plus la largeur du seuil et le nombre d'hypothèses simultanées sont importants, plus le nombre de chemins à parcourir est conséquent ainsi que le temps de recherche. Les meilleures valeurs d'élagage seront retenues pour la suite des expériences.

3.2.2 Prédiction des phones

Lorsque la fin d'un phone est atteinte, plusieurs hypothèses de phones suivants sont envisageables. L'objectif de cette méthode dit aussi "*Phonème look-ahead*" est de prédire les hypothèses de phones valables [55].

Pour ce faire, un score approximatif de chaque phone suivant est calculé au moyen de quelques échantillons. Puis un élagage à seuil est appliqué à tous les scores approximatifs. Cet élagage permet de supprimer les transitions correspondantes aux hypothèses de phones les moins prometteuses [59, 60]. Formellement, soit :

- $\tilde{\alpha}$ le phone actuel dans l'arbre lexical,
- α le phone suivant $\tilde{\alpha}$,
- $\tilde{q}(\alpha, t, \Delta t)$ la probabilité que le phone α produise la suite de vecteurs $x_{t+1}, .., x_{t+\Delta t}$. La quantité Δt doit être inférieure à la durée d'un phone c'est-à-dire 60-70 ms.

Soit $\tilde{Q}(t, \alpha)$ le score approximatif de α à l'instant t. Il peut s'exprimer sous la forme du produit du score de prédiction de α $\tilde{q}(\alpha, t, \Delta t)$ et celui de $\tilde{\alpha}$ à savoir $Q(t, S_{\tilde{\alpha}})$ comme suit :

$$\tilde{Q}(t, \alpha) = \tilde{q}(\alpha, t, \Delta t) * Q(t, S_{\tilde{\alpha}}) \tag{3.1}$$

avec $S_{\tilde{\alpha}}$ est le dernier état de $\tilde{\alpha}$.

Si $Q(t) = \max_{\alpha} \tilde{Q}(t, \alpha)$ est le meilleur score approximatif de toutes les hypothèses de phones à l'instant t, alors la transition $(\tilde{\alpha}, \alpha)$ est élaguée si :

$$\tilde{Q}(t, \alpha) < coef * Q(t) \tag{3.2}$$

$coef$ est le seuil d'élagage (*Look-Ahead*) de phone. Ce seuil est fixé a priori.

Pour réduire l'effort de calcul des scores de prédiction $\tilde{q}$:

1. des modèles indépendants du contexte sont utilisés. Ces modèles peuvent également disposer d'un nombre réduit de composants par état [60].
2. lors du calcul du score de prédiction d'une phone dépendant du contexte, le monophone correspondant est déterminé.
3. l'alignement linéaire du modèle monophone avec les quelques fenêtres, permet de calculer le score acoustique de ce monophone ou encore le score de prédiction $\tilde{q}$.

3.2.3 Quelques expériences

Pour étudier l'apport du nombre de fenêtres de prédiction (*nbr fenêtres*) sur le temps de décodage, plusieurs valeurs du seuil d'élagage des phones sont testées (de 0 à 1). Pour une beam de $1e-3$, les résultats sont comme suit :

TAB. 3.2 – Apport de la technique de prédiction de phone sur le temps de décodage

nbr fenêtres	TG	TV	TR	WER (%)
rien	3.95	2.59	1.38	28,7
2	4.58	3.00	1.58	28.7
4	4.71	3.08	1.62	28.7
5	4.71	3.05	1.65	28.7

D'après le tableau 3.2, on peut remarquer que cette méthode n'apporte pas d'amélioration au temps de décodage. Pour les valeurs extrêmes du beam (0 et 1) les résultats ne sont pas très différents. Donc cette méthode ne sera pas retenue pour la suite.

3.3 Modèle de langage

Avec une représentation des mots du lexique en arbre, les fins de mots sont associées aux feuilles de ce dernier. En conséquence, l'application d'un modèle de langage bigramme n'est pas possible lors de la transition entre deux mots puisqu'on ne connaît pas le mot suivant avant d'avoir atteint une feuille. Pour appliquer correctement le modèle de langage, deux méthodes sont classiquement utilisées :

1- Mémorisation du mot précédent. Ce qui revient à considérer pour espace de recherche un graphe dans lequel il existe une copie de l'arbre lexical pour chaque mot prédécesseur [55] [30]. Les copies d'arbre sont souvent très coûteuses en temps et en espace mémoire.

2- Report du score du bigramme à la fin du mot suivant. Dans ce cas, on parle de bigramme retardé. La factorisation du modèle de langage est une alternative à ces méthodes. Elle semble plus prometteuse [54].

3.3.1 Bigrammes retardés

L'utilisation des bigrammes retardés ou *delayed bigrams*, présente les avantages suivants [87] :

- rapidité du système par rapport aux copies d'arbre.
- plusieurs hypothèses peuvent être élaguées avant d'arriver à la fin du mot suivant.
- la probabilité du bigramme n'est ajoutée qu'aux mots suivants retenus après l'élagage et non à tous les mots suivants.

L'inconvénient de cette approche est que des hypothèses potentielles peuvent être supprimées alors qu'elles auraient survécu à l'élagage si la probabilité du bigramme avait été ajoutée. Dans [86], une augmentation du taux de mots erronés de 12% est reportée.
Pour retenir ces hypothèses en utilisant un bigramme retardé, un choix judicieux du seuil d'élagage s'impose. Une autre possibilité consiste à prendre un faible seuil, puis appliquer une deuxième passe de décodage pour rescorer le treillis. Par ailleurs, on pourra également faire appel aux techniques de prédiction (*look-ahead*) décrites ci-après.

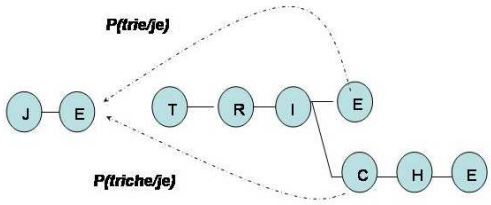


FIG. 3.3 – Bigrammes retardés

3.3.2 Factorisation du modèle de langage

Lorsque le lexique est organisé en arbre, le retard dans l'application de la probabilité bigramme est partiellement compensé par l'application de la technique de factorisation linguistique (en anglais *langage model look-ahead*) [54]. Cette méthode consiste selon Ortmann [61] à utiliser lors de la transition du mot_i à tous les mots suivants mot_j, la probabilité du bigramme $p(mot_k/mot_i)$ la plus importante. k est un mot suivant i. Puis, lorsque l'identité du mot suivant est connue, la probabilité introduite est remplacée par celle du bigramme effectif. Une alternative [87] consiste à considérer au moment de la transition la somme des probabilités des unigrammes des mots suivants comme le montre l'exemple de la figure3.4.

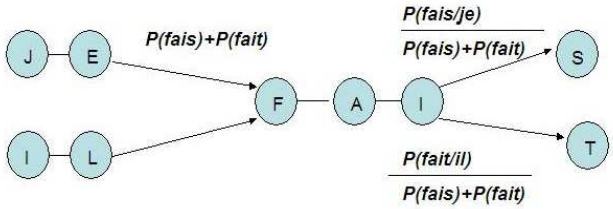


FIG. 3.4 – Factorisation du modèle de language

3.4 Calcul des vraisemblances

Les SRP actuels utilisent des HMMs renfermant des dizaines de milliers de distributions gaussiennes pour obtenir de bonnes performances. Ceci engendre un temps de calcul des vraisemblances pouvant atteindre 70% [24] du temps global du décodage. Des méthodes de calcul rapide des vraisemblances ont été proposées. On distingue les méthodes basées sur la réduction du nombre de paramètres, et les autres techniques qui accélèrent le temps de calcul au moyen d'astuces.

3.4.1 Sélection des paramètres

En suivant [85, 8], on peut organiser les méthodes de sélection des paramètres en quatre niveaux : fenêtre, mélange à distributions gaussiennes (GMM), distribution gaussienne et composant.

Fenêtre

Ce niveau concerne les techniques qui décident de calculer ou non la vraisemblance d'une trame. La motivation de cette idée est que le signal vocal varie lentement et donc une fenêtre est peu différente de sa précédente. La plus simple méthode opère comme suit [40] :
Étant donnée une suite de N fenêtres :

- la vraisemblance de la première fenêtre S_1 est calculée,
- les $N-1$ fenêtres suivantes se verront attribuer le même score S_1 [86] ou elles seront supprimées, ce qui revient en quelque sorte à un sous-échantillonnage.

Les résultats obtenus, lors de l'application de cette méthode en faisant varier le paramètre N sont :

N	TG	TV	TR	WER (%)	Gau/Fr
rien	3.95	2.59	1.38	28.7	130 481
2	2.88	1.52	1.35	30.6	67 915
3	2.48	1.14	1.33	32.7	47 079
4	2.16	0.92	1.23	34.5	36 080

TAB. 3.3 – WER en fonction du sous-échantillonnage

Gau/Fr est le nombre moyen de gaussiennes évaluées par observation. D'après ce tableau, une augemtation notable du taux de mots erronés est constatée. Pour améliorer de performances de cette méthode, une extension basée sur la quantification vectorielle a été proposée [10]. Elle consiste à :

1. construire un dictionnaire des vecteurs moyennes à partir des données d'apprentissage,
2. lors du décodage, l'observation est assignée à une entrée dans le dictionnaire,
3. la vraisemblance d'une fenêtre est calculée seulement si son entrée dans le dictionnaire est différente de celle de l'observation précédente. Autrement, elle est copiée sur la vraisemblance de la fenêtre précédente [10].

GMM

Ici on trouve les méthodes qui ignorent certains mélanges lors du calcul de la vraisemblance d'une observation [10]. Lee [38] propose de calculer le score d'un GMM dépendant du contexte (CD GMM) seulement lorsque le score du GMM indépendant du contexte (CI GMM) correspondant est assez élevé. Pour ce faire, la procédure à suivre pour chaque CD GMM est :

1. Trouver le CI GMM correspondant et calculer son score.
2. Supprimer les CI GMM dont le score est faible. Les deux possibilités d'élagage à seuil [10] ou k meilleures hypothèses (figure 3.5) peuvent être employées.
3. Pour chaque CD GMM, si le CI GMM correspondant est retenu après l'élagage alors son score est re-calculé sinon c'est le score du CI GMM qui lui sera attribué.

Dans un système de reconnaissance avec des HMMs phonétiquement lié (PTM) [39], seulement 14% des GMM sont utilisées sans dégradation des résultats.

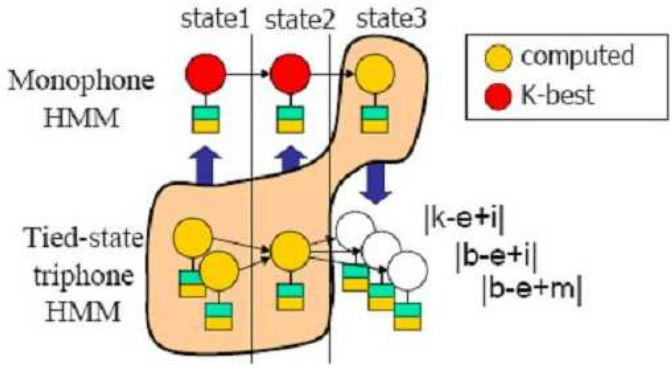


FIG. 3.5 – Calcul du score des CD GMMs (d'après [38])

Cette méthode a été appliquée en faisant varier le seuil d'élagage $Ci - beam$ dans le but de trouver un compromis entre le temps global de décodage et le WER.

$Ci - beam$	TG	TV	TR	WER (%)	Gau/Fr
0 (très large)	4.09	2.67	1.42	28.7	130 481
1e-80	3.95	2.59	1.38	28.7	130 481
1e-7	3.68	2.27	1.40	28.8	130 243
1e-5	2.85	1.56	1.36	28.9	81 148
1e-4	2.62	1.22	1.39	29.3	59 395
1e-3	2.93	1.03	1.89	29.9	36 920
1 (très étroit)	3.64	0.50	3.14	37.5	1 683

TAB. 3.4 – Seuil d'élagage des GMMs en fonction du WER

D'après le tableau 3.4, on peut constater qu'à partir d'une certaine valeur (1e-3) le taux d'erreur commence à augmenter ainsi que le temps de décodage. Dans l'autre sens, à partir de 1e-80, il n'est

plus intéressant d'augmenter la largeur du *beam*. Par conséquent, cette meilleure valeur est retenue pour la suite des expériences.

Gaussienne

Dans un GMM, seulement quelques gaussiennes dominent le calcul de la vraisemblance d'une observation [5]. Par conséquent plusieurs techniques ont été proposées dans la littérature pour les détecter [24, 5, 65, 53]. Souvent des méthodes basées sur un regroupement k-moyenne ou le partitionnement hiérarchique des distributions sont utilisées pour la sélection des gaussiennes.
Le principe de la sélection des gaussiennes par regroupement k-moyennes est comme suit :

- réaliser un découpage de l'espace des distributions et affecter les distributions aux régions.
- en déduire un centroïde par région (classe),
- lors du décodage, les scores des centroïdes sont calculés. Les distributions dont le centroïde admet le score le plus important sont utilisées.

La partitionnement hiérarchique vise à organiser les distributions dans une structure hiérarchique. Un parcours de l'arbre obtenu permet de localiser la feuille dont les gaussiennes sont intéressantes pour une observation donnée.
Ces techniques font l'objet d'une étude plus détaillée dans les prochains chapitres.

Composant

La dimension de l'espace acoustique est découpée en parties, souvent appelées flux et qui sont eux mêmes modélisées par des GMMs. La vraisemblance totale est obtenue par sommation des vraisemblances des flux. Dans ce cas aussi quelques gaussiennes dominent la vraisemblance d'un sous-espace.

3.4.2 Calcul rapide des scores

Parmi les méthodes de calcul rapide des vraisemblances, on peut citer l'utilisation des systèmes hybrides Réseaux de neurones/modèles de Markov cachés (RN/HMMs) et de la virgule fixe.
Plusieurs études se sont intéressées à l'utilisation d'un réseau de neurones comme frontal d'un HMM. Il a été démontré qu'un perceptron multicouches est équivalent à un estimateur de probabilité *a posteriori* d'appartenance à une classe [32].
Diverses expériences ont démontré que de tels systèmes hybrides améliorent les performances des HMMs, indépendants du contexte [12] ou à contexte réduit, tout en réduisant le temps de calcul. C'est notamment le cas du système Abbot de l'université de Cambridge (CU) basé sur des réseaux de neurones récurrents [76] et ou encore celui de l'institut ICSI de type MLP/HMMs [91].
Dans [52], il a été constaté que le taux d'erreur est d'autant plus faible que la taille du réseau de

neurones et par la suite celle de la base d'apprentissage est importante. Toutefois, lorsque la taille de la base d'apprentissage est importante, les systèmes hybrides ne sont plus pratiques à cause de leur durée excessive d'apprentissage. En effet, le nombre de paramètres à estimer augmente avec la taille de la base et par conséquent le temps d'apprentissage. A titre indicatif, [52] rapporte que l'apprentissage de 142 heures avec un MLP à 8000 noeuds cachés sur un matériel spécialement conçu dure 21 jours, ce qui n'est pas intéressant pendant les compagnes d'évaluation telles que Nist [56] où les participants seront amenés à faire tourner leurs systèmes une seule fois. En revanche, lors de la campagne de transcription enrichie Nist 04, un corpus Fisher de 2000 heures est fournie. Le temps d'apprentissage d'un MLP avec cette base a été estimé à un an [91]. Ce qui justifie l'intérêt de certaines études pour l'optimisation du temps d'apprentissage. Hormis, ce problème d'apprentissage, les méthodes hybrides demeurent néanmoins prometteuses.

3.5 Partitionnement hiérarchique

L'espace acoustique est découpé par un algorithme de partitionnement hiérarchique. On obtient une structure de type arbre binaire dont les feuilles sont associées aux composantes gaussiennes. Pendant le décodage, un parcours de l'arbre permet d'assigner l'observation à une feuille. Ensuite la vraisemblance de cette observation est calculée en utilisant seulement les gaussiennes de cette feuille.

3.5.1 Kdtree

Kdtree est une structure qui représente la subdivision de l'espace des distributions au moyen d'hyper-plans perpendiculaires aux axes des coordonnées [83]. La construction d'un tel arbre et l'organisation des densités dans cette structure sont réalisés comme suit [62] :

- une distribution gaussienne (ou son vecteur moyenne) est choisie comme racine de l'arbre
- les distributions situées à droite de l'hyper-plan passant par la racine sont placées au niveau de noeud fils de droite, celles situées à gauche de cet hyper-plan sont placées sur le noeud de gauche.
- ces deux étapes sont réitérées jusqu'à l'affectation de toutes les distributions.

On obtient, pour un Kdtree de profondeur d, un partitionnement de l'espace de dimension k en 2^d régions disjointes appelées seaux ("buckets") de l'arbre. La figure3.6 montre un exemple de Kdtree et la répartition de l'espace par dix hyper-plans en dimension 2.

Lors du calcul de la vraisemblance d'un vecteur pendant le décodage, ses coordonnées sont utilisées pour parcourir l'arbre et définir la région (bucket) à laquelle il appartient. Puis seulement les distributions de cette région qui sont utilisées pour le calcul de sa vraisemblance.

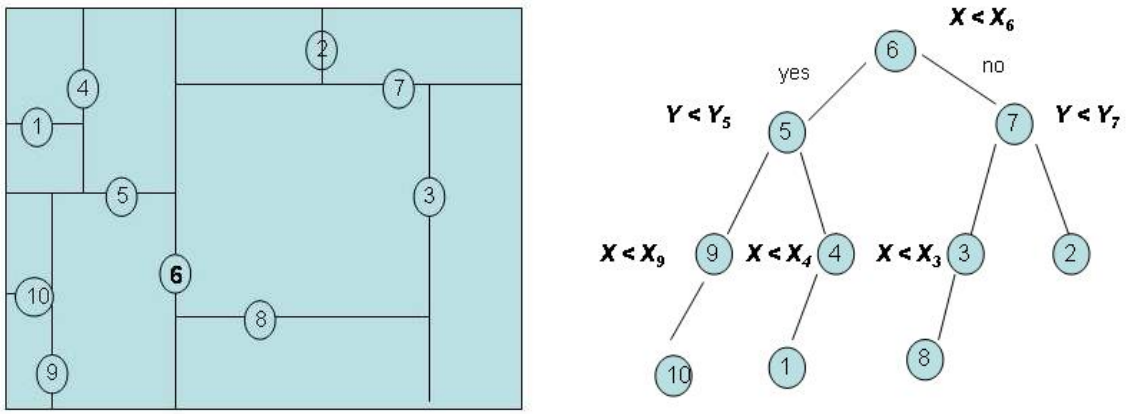


FIG. 3.6 – Partitionnement de l'espace par Kdtree (d'après [62])

3.5.2 *Bucket-Box-Intersection*

La méthode Bucket-Box-Intersection (BBI) [35] est une extension de l'algorithme Bucket-Voronoi-Intersection [36] [86] qui fait appel à l'organisation Kdtree pour partitioner l'espace. Cet algorithme (BBI) permet d'évaluer rapidement les vraisemblances moyennant une erreur d'approximation prédéfinie. L'idée initiale provient du fait que les valeurs d'une distribution gaussienne sont importantes seulement dans une région réduite. Cette région est limitée à une hyper-ellipsoïde dont les axes sont parallèles aux axes des coordonnées (figure 3.7).

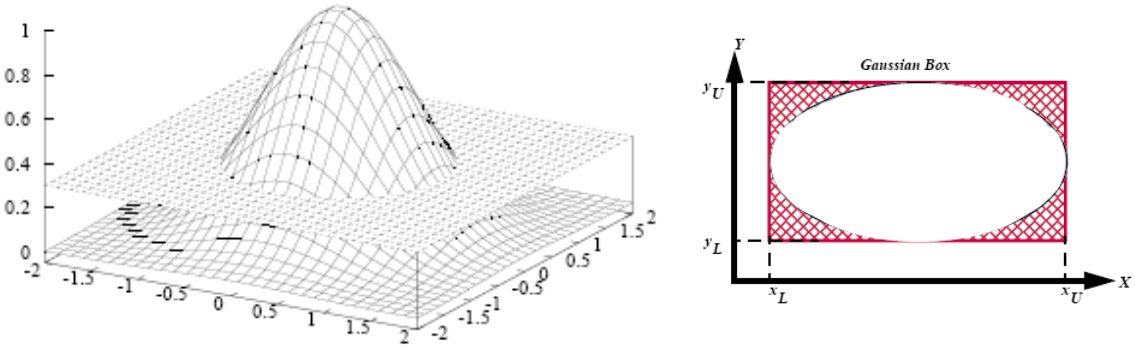


FIG. 3.7 – Détermination de l'hyper-ellipsoide (d'après [83])

Ainsi, pour une distribution gaussienne de moyenne μ et de matrice de covariance σ, si on fixe un seuil minimum de densité T, on peut calculer une boite délimitée par des hyper-plans perpendiculaires aux axes qui incluent l'ellipsoïde.

$$[y_U, y_L] = \mu_j \pm \sqrt{-2\sigma_j^2 \left[T + 0.5 \ \log(2\pi^k) \prod_{j=1}^{k} \sigma_j^2\right]} \tag{3.3}$$

Après classification de toutes les densités (figure3.8) avec l'algorithme Kdtree, un parcours de l'arbre permet de localiser l'observation au niveau des feuilles. Les gaussiennes issues de l'intersection de la région sélectionnée (feuille de l'arbre) et des boites de gaussiennes seront retenues pour le calcul de la vraisemblance.

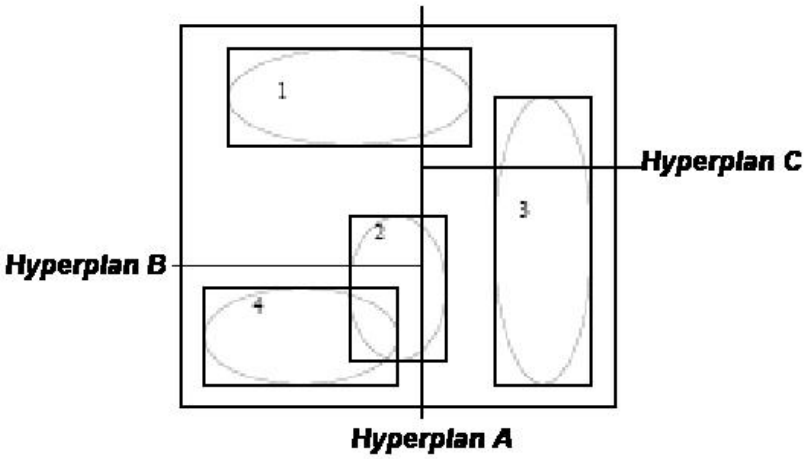


FIG. 3.8 – Classification et delimitations BBI (d'après [86])

Cette méthode est plus intéressante dans les systèmes ayant un nombre important de densités par état [35], et notamment les systèmes semi-continus. Son application à la reconnaissance de la parole en utilisant le système *Janus* permet d'accélérer la vitesse de décodage de 20% [86].

3.5.3 Arbres de décision

Dans [65], une alternative de construction de l'arbre hiérarchique des distributions est proposée. Il s'agit d'un arbre de classification de tous les composants qui est construit en trois étapes.

1. Les données d'apprentissage sont alignées à des modèles d'allophones.
2. Un classification hiérarchique de ces données, basée sur la perte minimale d'entropie, permet d'obtenir un arbre binaire. Chaque noeud de l'arbre est caractérisé par un vecteur et un seuil.
3. Au niveau des feuilles, chaque vecteur de paramètres est assigné à la gaussienne la plus représentative (parmi toutes les densités de l'allophone correspondant).

Au moment du décodage [64], l'arbre est parcouru de la racine aux feuilles. A chaque noeud, le vecteur observation est multiplié par le vecteur caractéristique du noeud. Suivant que ce produit

est supérieur ou pas au seuil du noeud, l'observation est affectée au noeud fils gauche ou droite. Ce processus est réitéré jusqu'à atteindre une feuille. Les gaussiennes de cette feuille sont utilisées pour le calcul du score de l'observation.

3.6 Regroupement k-moyennes

Dans ce paragraphe nous rapportons les méthodes de sélection des gaussiennes par regroupement k-moyennes existantes. Des variantes de ces méthodes portant en particulier sur l'amélioration de la classification et l'affectation des gaussiennes par état sont également décrites. L'expérimentation de certaines de ces méthodes sont décrites dans le chapitre 5.

3.6.1 Sélection des gaussiennes

La méthode classique de sélection des gaussiennes par regroupement k-moyenne, initialement proposée par Bocchierri [5] (et détaillée dans le paragraphe 5.2.1), consiste à :

- découper l'espace des distributions gaussiennes en classes au moyen de l'algorithme k-moyennes. Une classe est supposée représenter, pour une distribution donnée, son voisinage (voir 3.9).
- générer un représentant ou *codeword* de chaque classe.
- former pour chaque classe un *shortlist* càd une liste des gaussiennes appartenant à cette classe. Certaines classes se recouvrent, par conséquent plusieurs distributions appartiennent à plusieurs classes.

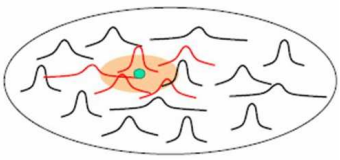


FIG. 3.9 – Classification et génération de classes

Lors du décodage, la distance de l'observation à toutes les classes est calculée et la classe las plus proche est sélectionnée. Les distributions du *shortlist* correspondant sont utilisées pour le calcul effectif de la vraisemblance.
L'application de cette méthode en reconnaissance des chiffres a permis d'obtenir une réduction du temps de calcul des vraisemblances d'un facteur neuf sans dégradation des performances [5].

L'utilisation de plusieurs *shortlists* et la limitation du nombre de distributions gaussiennes par *shortlist* ont été expérimentés dans [43] [58]. Un gain de précision est constaté. En revanche, la vitesse de calcul des émissions a triplé [43].

Dans [40] on s'intéresse à la réduction de la mémoire occupée par les tables de classification pour une application de reconnaissance sur un dispositif embarqué. D'où l'emploi de la méthode de sélection de gaussiennes en utilisant des classes disjointes pour éviter les affectations multiples des distributions aux classes. Pendant le décodage, plusieurs *shortists* peuvent être choisis. De ce fait, le temps de calcul des vraisemblances est réduit de 66% pour une dégradation relative des résultats de l'ordre de 4%.

Une variation de la sélection des gaussiennes [17] consiste à effectuer une quantification vectorielle de l'espace des données au lieu de celui des distributions. Chaque *shortlist* contient les gaussiennes les plus vraisemblables. En reconnaissance grand vocabulaire [53] et en faisant varier la taille des *shortlists*, un gain de la durée de calcul des vraisemblances d'un facteur de trois est constaté lorsque les modèles sont phonétiquement liés sinon cinq.

Enfin, la combinaison de la sélection des gaussiennes et de la réduction de l'espace de recherche [84] produit un gain en temps plus important, meilleur que celui apporté par chacune des deux approches séparément.

3.6.2 Affectation des gaussiennes par état

La sélection des gaussiennes en reconnaissance des chiffres apporte un gain de temps de calcul des vraisemblances d'un facteur de neuf [5] mais lorsqu'il s'agit de grand vocabulaire, ce facteur se réduit manifestement à trois [37]. La non prise en compte de l'appartenance des gaussiennes aux états pour la formation des classes en est une explication. En effet, les états actifs ne disposant pas de distributions gaussiennes dans le *shortlist* choisi sont approximés [37]. Suite au processus d'élagage, les chemins correspondants à ces états dont les scores sont faibles peuvent être élagués, augmentant de ce fait le taux d'erreur.
Pour remédier à un tel inconvénient, il a été proposé [24] d'affecter à chaque classe un nombre minimum de distributions gaussiennes issues d'un même état [37]. En particulier, une approche de double anneau ("*dual ring*" en anglais) est proposée : lors du décodage, une classe est choisie puis

- toutes les gaussiennes de cette classe dont la distance au vecteur observation est inférieure à un seuil T_1 sont retenues.
- s'il n' y a pas de distributions dans cette région, alors seulement la (ou les) gaussienne(s) dont la distance au vecteur observation est minimale et inférieure un seuil T_2 ($T_2 > T_1$) est (sont) retenue(s).

– sinon l'état est remplacé par un seuil fixé a priori (voir figure3.10).

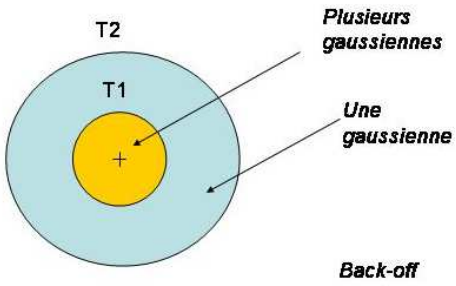


FIG. 3.10 – Approche double anneau

Cette méthode a permis d'obtenir un gain de temps de calcul des vraisemblances d'un facteur de cinq.

Dans [24], on met en cause la représentation du centroïde d'une classe de toutes les distributions de cette classe ; étant formé par le regroupement de distributions gaussiennes issues de différents contextes, ce dernier ne pourra pas représenter tous ces contextes. De ce fait, il propose de retenir pour une classe et un état donnés, les gaussiennes qui maximisent la vraisemblance des données d'apprentissage, et non les gaussiennes les plus proches du centroïde de classe comme le préconise la procédure classique de sélection des gaussiennes. Suite à l'application de cette procédure un gain d'un facteur de sept est relevé.

3.7 Sous-Quantification vectorielle

La sous-quantification vectorielle (SQV) est une quantification vectorielle appliquée à des sous-espaces de données.
Dans ce contexte, Mak [47, 6, 46] propose de subdiviser la dimension de l'espace acoustique en plusieurs flux, et construire un *codebook* par flux. Les étapes de ce processus sont :

– construire des modèles continus,
– définir les flux et leurs dimensions,
– regrouper les composants de chaque flux et en déduire les représentants par flux,
– remplacer les modèles continus de chaque flux par les représentants correspondants.

Aiyer [1] reprend cette méthode et essaye de repérer dans un même flux, les parties qui se répètent fréquemment afin d'éviter la redondance de calcul. Une réduction du temps de calcul de 2 fois est

relevée.

Dans [42], la SQV est utilisée pour éviter le calcul de la vraisemblance d'une observation lorsque le score d'un flux est très faible. Comme on suppose que les flux sont indépendants (matrice de covariance diagonale), la vraisemblance s'exprime sous la forme d'un produit des vraisemblances des sous-espaces. Si la vraisemblance est très faible pour un sous-espace, elle l'est pour tout l'espace, par conséquent elle ne sera pas calculée pour le reste des sous-espaces. En utilisant cette méthode, une augmentation de la vitesse du calcul des vraisemblances de 1.2 à 1.8% est relevée [41].

Contrairement aux CD HMMs, qui sont utilisés par la plupart des systèmes de reconnaissance de l'état de l'art, les modèles discrets sont rarement impliqués à cause de leur taux d'erreur important (de 1.5 à 2 fois celui des CD HMMs [18]). Une méthode de SQV appliquée à des modèles discrets a été dernièrement proposée [18]. Chaque vecteur de donnée est subdivisé en plusieurs sous-vecteurs. Puis chaque sous-vecteur (flux) est affecté à un symbole défini dans un dictionnaire de références. Pour un nombre important de flux de 15 à 24, le temps de décodage est réduit de 1 à 2 fois le temps réel sans perte de performance.

Finalement, dans [73], la SQV est appliquée aux vecteurs moyennes et covariances des gaussiennes en supposant que les matrices de covariance sont diagonales. Pour chaque flux, l'algorithme k-moyennes est utilisé pour classifier les vecteurs moyennes et covariances et un certain nombre de représentants par flux est déduit. Lors du décodage, la vraisemblance est calculée en utilisant ces représentants [74].

3.8 Autres méthodes

Projection Search Algorithm (PSA), est une méthode de partitionnement dynamique de l'espace [62]. Elle vise à determiner les distributions dont les prototypes sont situés dans un hypercube centré par l'observation. Pour chaque dimension i de l'espace, les prototypes candidats sont ceux dont les coordonnées d'indice i se trouvent entre les deux plans parallèles $H_i^{-\epsilon}$ et $H_i^{+\epsilon}$. Les plans $H_i^{-\epsilon}$ et $H_i^{+\epsilon}$ sont parallèles à l'axe des coordonnées d'indice i et situés à une distance de ϵ de l'observation.

Les prototypes retenus sont issus de l'intersection de tous les espaces. Ils sont situés dans une boite centrée sur l'observation.

Dans un système de reconnaissance phonétiquement lié (PTM), un nombre important de gaussiennes est partagé par les états des allophones issus d'un même phone [78]. Afin de réduire ce nombre (pour un meilleur apprentissage), une classification hiérarchique des états basée sur l'entropie est effectuée. Le regroupement se termine lorsque la distance dépasse un seuil fixé à priori [77].

3.9 Conclusion

Dans ce chapitre, nous avons passé en revue la littérature sur l'accélération du temps de décodage en reconnaissance automatique de la parole. Comme on s'intéresse particulièrement aux méthodes de calcul rapide des vraisemblances, nous avons détaillé les méthodes correspondantes. Nous avons noté deux grandes approches basées respectivement sur le regroupement des distributions et la sous-quantification vectorielle. Dans la première catégorie, on distingue la classification par partitionnement hiérarchique et la classification k-moyenne.

La comparaison des méthodes existantes n'est pas triviale pour maintes raisons :

- les conditions expérimentales des systèmes sont souvent différentes.
- les mesures ne sont pas similaires. On en distingue globalement le nombre de gaussiennes (par observation) calculés ou le temps de décodage. Mais ce dernier dépend fortement du système.
- les méthodes basées sur le partitionnement hiérarchique sont plus adaptées aux systèmes disposant de nombre élevé de composants par état. Les approches qui s'appuient sur la classification k-moyenne sont intéressantes lorsque le nombre global de composants est important.

L'évaluation de certaines techniques et nos contributions font l'objet des trois prochains chapitres qui porteront respectivement sur le partitionnement hiérarchique et le regroupement k-moyennes pour la sélection des gaussiennes et sur la sous-quantification vectorielle.

Chapitre 4

Partitionnement Hiérarchique Multi-niveaux

4.1 Introduction

Dans ce chapitre, on propose une nouvelle méthode de sélection des gaussiennes basée sur le partitionnement hiérarchique des distributions. Cette proposition se décline en trois contributions : d'abord la distance de *Kullback Leibler Pondérée* qui mesure la similarité entre distributions gaussiennes est définie et comparée aux distances existantes. Ensuite, pour trouver la meilleure classification, nous avons fixé trois critères de choix du nombre final de classes.
Cette approche de classification sera enfin exploitée dans le contexte de sélection des gaussiennes, et ce, à plusieurs niveaux. La particularité de notre proposition par rapport aux méthodes existantes réside dans le fait de ne retenir une gaussienne que si toutes ses classes (noeuds aïeux) sont vraisemblables. Ce critère permet d'éviter au mieux les erreurs de classification.

4.2 Regroupement hiérarchique des distributions

Les systèmes Markoviens de Reconnaissance Automatique de la Parole (RAP) utilisent un nombre très important de distributions gaussiennes pour améliorer la précision de leurs modèles acoustiques. Un des inconvénients majeurs de cette pratique est l'augmentation de la complexité des systèmes, nécessitant de ce fait de très importantes bases de données pour apprendre les paramètres de leurs modèles. Dans la littérature, différents critères sont utilisés pour déterminer le nombre optimal de distributions. Souvent, il s'agit de trouver un compromis entre la précision des modèles (nombre de distributions gaussiennes) et la possibilité de bien estimer leurs paramètres.

4.2.1 Approches existantes

La procédure classique de construction des modèles acoustiques consiste à commencer par des modèles à une seule distribution gaussienne par état [57], puis augmenter progressivement le nombre de distributions par dédoublement de la gaussienne de poids ([57]) ou de variance la plus importante. Les moyennes des deux distributions issues du dédoublement sont perturbées d'un facteur σ (souvent $\sigma = \pm 0.2$) de la variance puis elles sont ré-estimés. Ce processus de dédoublement/perturbation est réitéré jusqu'à ce que :

* le nombre de gaussiennes fixé a priori est atteint.
* la quantité de données d'apprentissage devient insuffisante. Dans ce cas, un seuil minimum du nombre d'observations nécessaires à l'estimation des distributions est prédéfini.
* l'augmentation de la vraisemblance devient non significative.
* le gain du Critère d'Information Bayésien (BIC) est négatif ou inférieur à un seuil. Ce critère contrôle la complexité du système en pénalisant la vraisemblance par le nombre de paramètres.

Récemment, Messina et al. [50] ont proposé d'augmenter le nombre de distributions seulement lorsqu'aucune d'elles ne peut représenter des données d'apprentissage. Ainsi, les distances entre chaque observation et les distributions gaussiennes sont calculées. La distribution correspondant à la distance minimale est relevée. Suivant que cette valeur est inférieure ou non à un certain seuil, la distribution est mise à jour au moyen de l'observation ou un nouveau composant est crée.

Dans le but de réduire le nombre de composants d'un système à états phonétiquement liés, Digalakis [17] classifie les distributions gaussiennes et en déduit un représentant par classe qu'il ré-estime. La métrique utilisée pour la classification est basée sur l'augmentation de l'entropie engendrée par la fusion des distributions. Les expériences ont montré une réduction du nombre de gaussiennes à moins de 40% sans dégradation significative des performances. Aussi, a-t-on relevé que le système ainsi obtenu est plus précis (+0.8% de précision) qu'un système classique disposant du même nombre de composantes gaussiennes.

Les modèles dépendants du contexte sont souvent construits par copie des modèles indépendants du contexte à une seule composante gaussienne par état. Puis, leur nombre de gaussiennes par état est augmenté par dédoublement/perturbation/ré-estimation [90]. Néanmoins, dans le système Julius [38], il a été empiriquement prouvé que des modèles dépendants du contexte qui sont initialisés par copie de modèles indépendants du contexte à plusieurs composants par mélange puis ré-estimés sont plus performants que les modèles classiques.

4.2.2 Méthode proposée

On se propose de déterminer le nombre optimal de composants d'un mélange Gaussien en utilisant le même principe d'augmentation/réduction que Digalakis [17]. L'idée de base consiste à explorer un espace très large des composantes construits de façon classique càd par dédoublement/perturbation/réestimation [57]. Puis la dimension de cet espace est réduite par regroupement des distributions similaires considérées comme étant redondantes. Pour ce faire :

- les distributions de chaque mélange sont organisées dans une structure de type arbre binaire à l'aide d'un algorithme de classification hiérarchique ascendant.
- à chaque niveau de l'arbre correspond une possibilité de classification. Afin d'en déduire la meilleure, plusieurs critères de coupure de l'arbre sont proposés et évalués.

En revanche, pour améliorer la classification, nous proposons une distance basée sur la similarité [92] entre les distributions et la comparons à la distance utilisée par Digalakis [17, 33] basée sur la perte d'entropie.

4.2.3 Organisation des gaussiennes

La procédure de construction de l'arbre hiérarchique ascendant est appliquée à chaque mélange de distributions gaussiennes. Elle repose sur les étapes suivantes :

1. Calcul des distances entre toutes les paires de distributions.
2. Fusion des deux distributions les plus proches [51]. Soit $g_1(n_1, \mu_1, \Sigma_1)$ et $g_2(n_2, \mu_2, \Sigma_2)$ deux distributions gaussiennes auxquelles n_1 et n_2 observations sont affectées pendant l'apprentissage et dont les matrices de covariance sont supposées diagonales. Si g_1 et g_2 sont regroupées en $g_3(n_3, \mu_3, \Sigma_3)$ alors :

$$\begin{aligned} n_3 &= n_1 + n_2 \\ \mu_3 &= \frac{n_1}{n_1+n_2}\mu_1 + \frac{n_2}{n_1+n_2}\mu_2 \\ \Sigma_3 &= \frac{n_1}{n_1+n_2}\Sigma_1 + \frac{n_2}{n_1+n_2}\Sigma_2 + \frac{n_1 n_2}{(n_1+n_2)^2}(\mu_1-\mu_2)(\mu_1-\mu_2)^T \end{aligned}$$

   g_3 remplace (g_1, g_2) et l'ensemble total des distributions est réduit de 1.
3. Si le nombre global de distributions gaussiennes est supérieur à 1, revenir à la première étape.

A la fin de ce processus, on obtient un arbre binaire (figure 4.1) dont les noeuds de chaque niveau correspondent à des distributions mono-gaussiennes.
Dans la mesure où dans les systèmes continus chaque état dispose de ses propres distributions gaussiennes, le processus de regroupement hiérarchique est appliqué à chaque état, et on obtient autant d'arbres binaires que d'états.

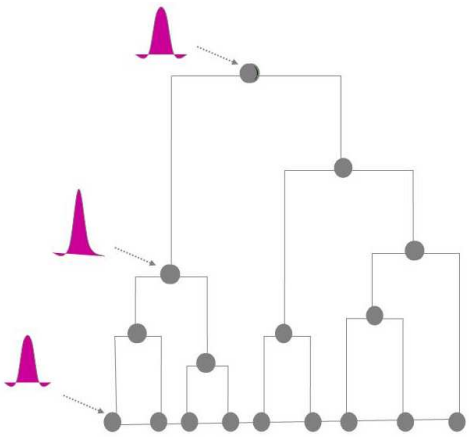


FIG. 4.1 – Arbre binaire de classification

4.2.4 Métriques pour le regroupement

Deux distances ont été choisies. La première, basée sur les données, est la perte en vraisemblance engendrée par la fusion de distributions. La seconde que nous avons baptisée Distance de Kullback Pondérée KLP repose sur la dissimilarité entre distributions.

1. Perte de vraisemblance (PV) : lorsque deux distributions gaussiennes $g_1(n_1, \mu_1, \Sigma_1)$ et $g_2(n_2, \mu_2, \Sigma_2)$ sont regroupées en $g_3(n_3, \mu_3, \Sigma_3)$ la perte en vraisemblance résultante est définie par comme étant la différence entre la somme des vraisemblances de g_1 et g_2 et la vraisemblance de g_3, et ce, sur les données d'apprentissage.

$$PV(g_1, g_2, g_3) = \log \frac{\|\Sigma_3\|^{(n_1+n_2)/2}}{\|\Sigma_1\|^{n_1/2}\|\Sigma_2\|^{n_2/2}} \tag{4.1}$$

   Cette distance est similaire à la distance de perte d'entropie proposée par Hwang [33] et utilisée dans [17]. La distance PV a été expérimentée avec succès pour l'adaptation des modèles [51].

2. Divergence de Kullback-Leibler Pondérée (KLP) : C'est la distance de Kullback-Leibler symétrisée appliquée à deux fonctions de densité de probabilité pondérées par leurs données d'apprentissage.

$$KLP(g_1; g_2) = \frac{1}{2}tr(n_1\Sigma_1\Sigma_2^{-1} + n_2\Sigma_2\Sigma_1^{-1}) + \frac{1}{2}(\mu_1 - \mu_2)^T(n_1\Sigma_1^{-1} + n_2\Sigma_2^{-1})(\mu_1 - \mu_2) - (n_1 + n_2)d \tag{4.2}$$

   d est la dimension des vecteurs de paramètres.

   L'utilisation des informations apportées par la quantité de données d'apprentissage est particulièrement avantageuse dans le cas où celles-ci et les données de test ont les mêmes proportions.

Le détail de calcul de ces distances est décrit dans l'annexe A.

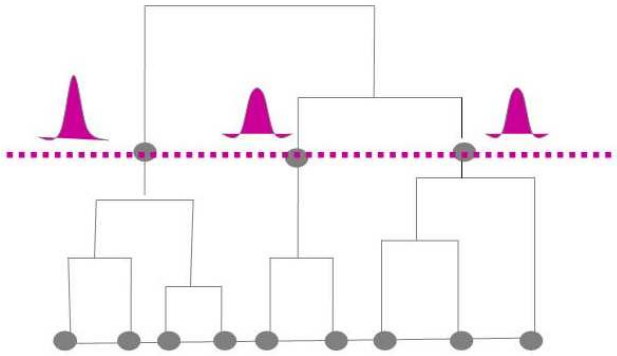
4.2.5 Critères de coupure de l'arbre

De la racine aux feuilles, plusieurs niveaux de coupure de l'arbre et donc de classification sont envisageables. Pour trouver le niveau qui engendre la meilleure classification, trois critères de coupure sont proposés.

a) **Nombre de classes fixé *a priori*** : l'arbre est parcouru des feuilles vers la racine. On s'arrête lorsque le nombre de noeuds à un niveau donné atteint le nombre de classes prédéfini.

b) **Quantité de données suffisante** : le nombre de classes dépend de la quantité de données d'apprentissage utilisée pour estimer les distributions de chaque classe. L'arbre est parcouru à partir de la racine. On s'arrête lorsque la quantité de données d'apprentissage devient faible (inférieure à un seuil prédéfini). La coupure de l'arbre aura lieu au niveau du noeud parent.

c) **Groupement de distributions similaires** : La coupure de l'arbre est réalisée lorsque la distance entre deux niveaux atteint une valeur maximale. Ce qui correspond à un groupement de distributions dissimilaires.



Rappelons que les noeuds situés au niveau de la coupure de l'arbre sont des distributions mono-gaussiennes qui proviennent du regroupement des feuilles. De ce fait, on les appellera centroïdes ou classes.

Disposant de modèles acoustiques continus, la coupure des arbres de tous les états donne lieu à de nouveaux modèles que nous appellerons dans la suite *modèles réduits*. Les états des *modèles réduits* sont alors modélisés par les distributions classes ou centroïdes.

4.2.6 Expériences de validation

Dans les expériences de ce chapitre nous avons utilisé le système HTK/Sirocco décrit dans la paragraphe 1.3.2 du chapitre 1. Les tests sont réalisés sur une heure de parole issue de la radio classique. Les modèles acoustiques initiaux sont indépendants du contexte et à 256 gaussiennes par état. L'intervalle de confiance à 95% du système initial correspondant est de l'odre de 1%.
Pour chaque état,

- les gaussiennes sont classifiées à l'aide de l'algorithme de classification hiérarchique ascendant.
- pour cela, la distance de perte en vraisemblance (PV) et la distance de Kullback-Leibler pondérée (KLP) sont utilisées pour le regroupement.
- les classes de distributions sont obtenues après coupure de l'arbre en suivant les critères : nombre de classes fixé, nombre de classes variable en fonction des données ou de la distance.

Le système dont les modèles réduits sont obtenus après regroupement avec la distance KLP sera appelé système KLP. Celui utilisant la distance PV est dit système PV.

Pour évaluer l'apport du regroupement, un système initial de référence (ref) est défini. Pour ce faire, des modèles acoustiques indépendants du contexte à 32, 64, 80, 128, 180, 256, 220 et 512 gaussiennes par état sont construits par la méthode de dédoublement / perturbation / estimation classique. Ces modèles sont évalués dans les mêmes conditions que les systèmes PV et KLP.

a) Nombre de classes fixé a priori

En suivant la procédure décrite ci-dessus en partant de modèles à 256 gaussiennes par état et en coupant l'arbre de façon à obtenir respectivement 32, 64, 80, 128, 180 classes, les modèles ainsi obtenus sont ré-estimés. Deux itérations Baum-Welch ont été suffisantes pour l'apprentissage de ces modèles.

La figure 4.2 montre que les systèmes PV et KLP sont plus performants que le système de base avec un léger avantage en faveur du système KLP. En particulier, lorsque le nombre de gaussiennes par état est réduit à 32 ou 64, le taux d'erreur est diminué de 3% absolu. Cette diminution dépasse l'intervalle de confiance. Ce qui confirme l'hypothèse initiale de non optimalité de la méthode de construction des modèles acoustique par dédoublement/perturbation.
Dans le cas où le nombre de classes est important, on remarque que la différence entre les différents systèmes est faible.

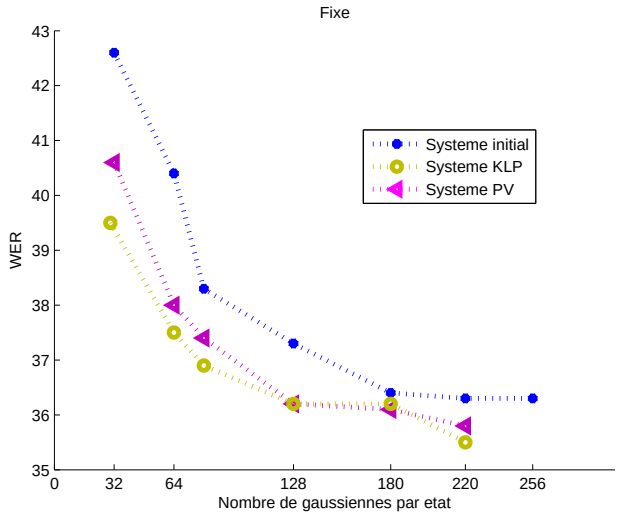


FIG. 4.2 – Taux de mots erronés (WER) en fonction du nombre de gaussiennes par état pour les trois systèmes

b) Quantité de données suffisante

Rappelons que ce critère implique que chaque classe dispose d'une quantité suffisante de données pour son estimation. Ceci se traduit par une coupure de l'arbre lorsque la pondération d'une distribution est inférieur à un seuil prédéfini empiriquement.

En utilisant ce critère, on obtient des modèles réduits dont le nombre de distributions par mélange est variable. Par conséquent, dans les expériences qui suivent, une valeur moyenne est considérée. Les résultats sont indiqués sur la figure 4.3.

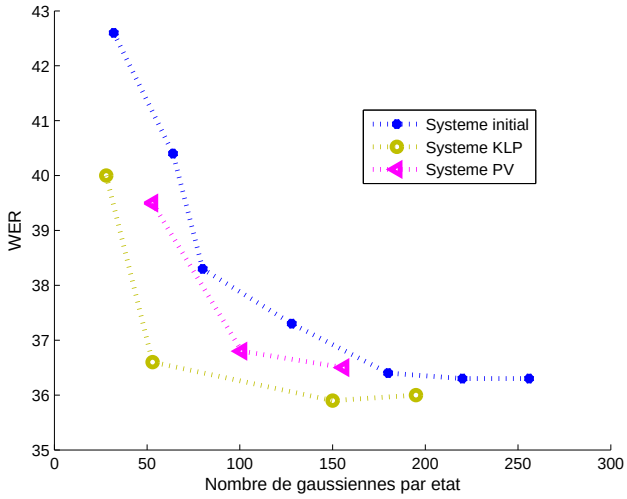


FIG. 4.3 – Coupure de l'arbre basée sur les données

On remarque, encore une fois que le système KLP est plus performant que le système PV ou encore le système de base. En particulier, au moyen de seulement 53 gaussiennes par état, ses performances sont similaires à celles du système initial à 256 gaussiennes par état. Ceci correspond à une réduction absolue du taux d'erreur de 4.8% par rapport au système de base de même nombre de gaussiennes par état. Une explication de ces performances est la suivante : en se basant seulement sur la similarité entre les distributions, le regroupement KLP peut donner lieu, à chaque niveau, à des distributions disposant de peu de données d'apprentissage (peu représentatives des données, moins bien estimées, ..) ce qui est compensé par le critère de coupure de l'arbre conditionné par les données.

c) Nombre de classes fonction de la distance

Ce critère permet d'éviter de regrouper deux distributions gaussiennes distantes c'est-à-dire :

– elles sont dissimilaires dans le cas de la distance KLP.

– leur fusion provoque une perte de vraisemblance importante dans le cas de PV.

En considérant plusieurs seuils de distance de coupure minimale, les résultats sont reportés sur la figure 4.4.

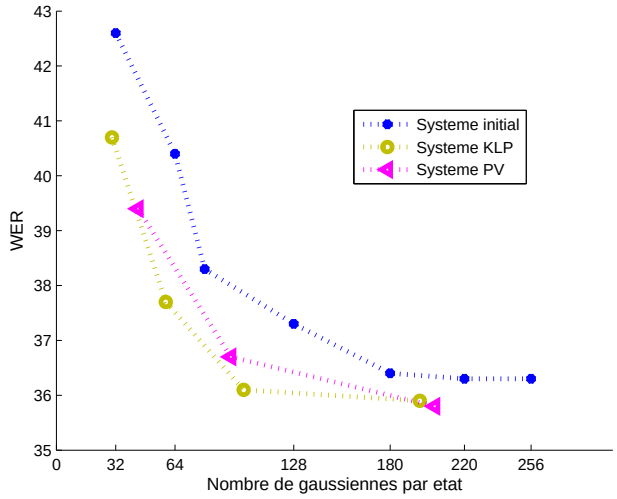


FIG. 4.4 – Coupure de l'arbre en fonction de la distance

On peut constater que les systèmes PV et KLP (ici de performances comparables), sont meilleurs que le système de base ce qui est en concordance avec les résultats précédents. En appliquant les regroupements KLP et PV, on obtient globalement les mêmes performances que le système de référence mais en utilisant seulement 40% du nombre global de gaussiennes. Ces résultats sont intéressants mais demeurent moins performants que les expériences précédentes (avec 53 gaus-

siennes) où le nombre est réduit à 20%.

Afin d'expliquer d'avantage ces résultats, nous avons réalisé les expériences de comparaison des critères ci-dessous.

d) Comparaison des critères

Pour mieux visualiser l'apport de chaque critère de coupure de l'arbre aux distances KLP et PV, nous avons reporté les courbes correspondantes sur la figure 4.5.

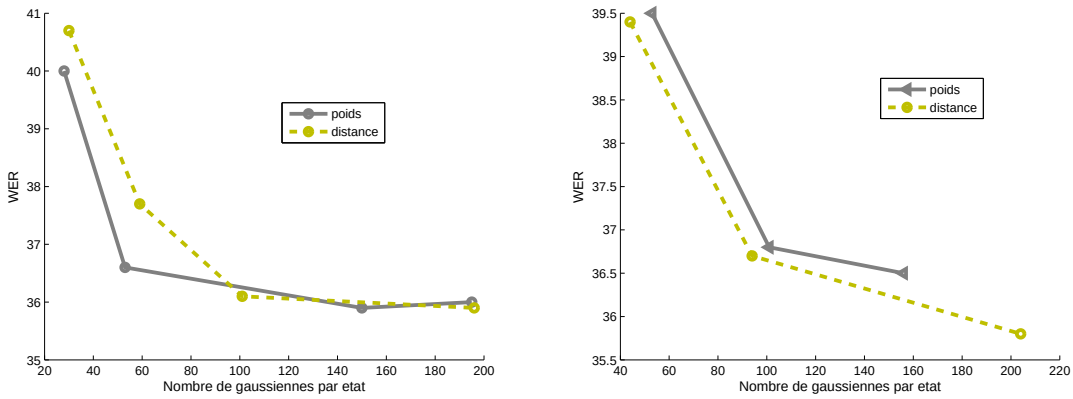


FIG. 4.5 – Taux d'erreur pour les critères de coupure de l'arbre basés sur les données et la distance pour les distances KLP (à gauche) et PV (à droite)

Pour le système KLP, la coupure basée sur les données semble plus intéressante que celle fonction de la distance. Dans le cas d'un regroupement PV, c'est plutôt le contraire sans pour autant dépasser l'intervalle de confiance. Ceci peut s'expliquer par :

- Dans le système KLP, le regroupement, à chaque niveau, concerne les distributions similaires. Par conséquent, il peut arriver qu'à un niveau donné, certaines distributions ne disposent pas d'assez de données d'apprentissage. Ce qui peut être corrigé par une coupure basée sur les données.

- En assurant une perte de vraisemblance minimale à l'issue de chaque regroupement PV, les clusters obtenus sont assez représentatifs des données d'apprentissage mais certains d'entre eux peuvent se ressembler. Dans ce cas la coupure basée sur la distance permet de supprimer cette redondance.

4.3 Sélection des distributions gaussiennes

4.3.1 Motivation

Comme déjà mentionné au début de ce chapitre, les systèmes de reconnaissance actuels font appel à des modèles de Markov cachés renfermant des dizaines de milliers de distributions gaussiennes pour assurer une bonne précision. Or pour une trame de test donnée, seulement quelques gaussiennes sont intéressantes [5, 24]. Pour le confirmer, nous avons réalisé quelques tests sur une partie de Radio Classique avec le système HTK/Sirocco (paragraphe 1.3.2).

Pour un état donné (mélange de 128 gaussiennes), nous avons relevé les 10 meilleures et les 10 plus faibles vraisemblances calculées à l'aide des 10 meilleures et moins bonnes composantes, et ce, pour deux séquences de trames. Les résultats sont reportés sur la figure 4.6.

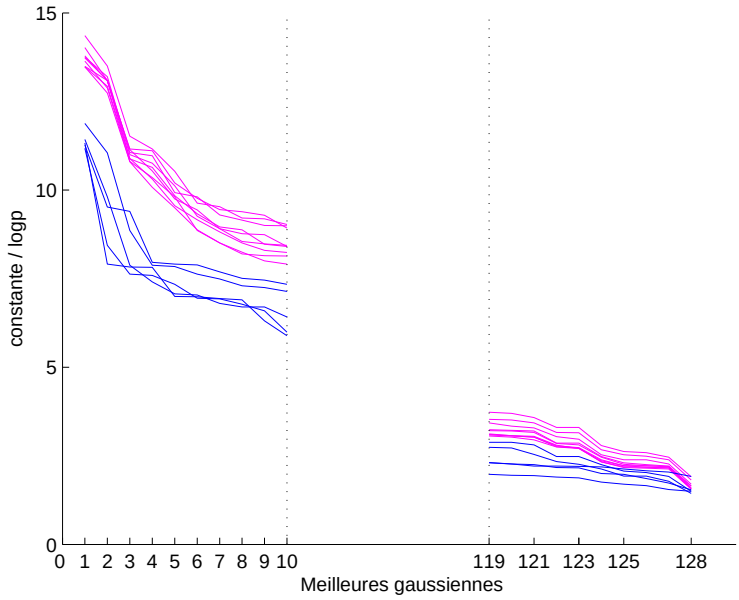


FIG. 4.6 – Vraisemblances des meilleures et des pires gaussiennes

On remarque un écart important entre les meilleurs et les plus faibles valeurs de la vraisemblance.On en déduit que l'apport de certains composants au calcul de la vraisemblance globale est faible voir même parfois négligeable.

Pour réduire le temps de décodage, on se propose, dans la suite de ce chapitre, de détecter les composantes ayant les plus fortes vraisemblances. Le calcul de la vraisemblance globale ne tiendra compte que de ces composantes.

4.3.2 Méthode proposée

L'objectif de la sélection des gaussiennes est d'utiliser pendant le décodage seulement les distributions dont l'apport en vraisemblance est important. Pour ce faire, une classification hiérarchique des distributions de chaque mélange est réalisée (comme décrit plus haut). A chaque état, la coupure de l'arbre binaire donne lieu à un ensemble de distributions gaussiennes appelées centroïdes ou *(codewords)*. Rappelons qu'un centroïde ou *codeword* est formé par une distribution gaussienne. Les distributions situées au niveau des feuilles des centroïdes sont dites (*shortlists*) et seront également stockées.

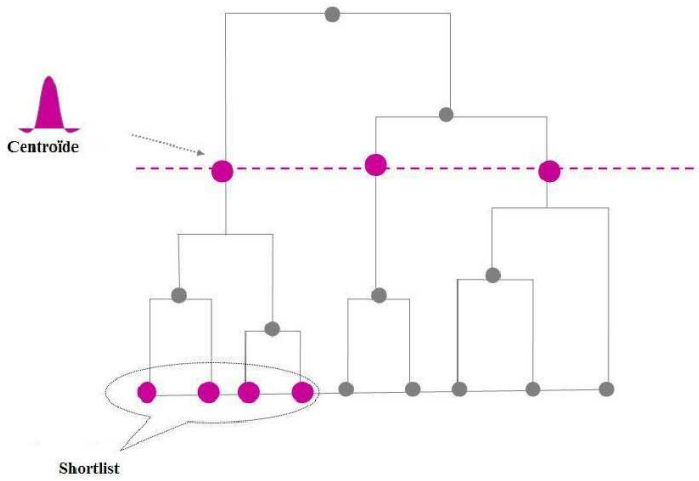


FIG. 4.7 – Sélection des gaussiennes

Pendant le décodage, le choix des distributions à utiliser est réalisé comme suit :

i. Les densités des *codewords* sont calculées et triées.

ii. La vraisemblance de l'observation O est calculée en utilisant :

1. les *codewords*. Ce qui revient en quelque sorte à une quantification vectorielle par état.
2. les distributions des *shortlists* dont les vraisemblances des *codewords* sont importantes.
3. les distributions des *shortlists* retenues à l'étape 2 et dont la vraisemblance est importante. Pour ce faire, toutes les densités des *shortlists* retenues sont calculées et triées. Puis, seulement les distributions les plus vraisemblables d'entre elles seront utilisées pour le calcul de la vraisemblance. Le but de ce choix est de vérifier si le nombre de gaussiennes issues des propositions précédentes est optimal ou pas.
4. les distributions des *shortlists* retenues dont la pondération est importante (supérieure à un seuil). En effet, les distributions dont le poids est important sont les plus présentes et représentatives des données.

Cette méthode présente certains avantages par rapport au partitionnement hiérarchique par Kdtree (paragraphe 3.5.1). Les premières expériences de ce chapitre montrent que seulement 25 % des gaussiennes peuvent être utilisées sans perte de performance. Ce taux pourrait éventuellement être réduit par l'ajout d'une étape de sélection des gaussiennes. Dans [86] le temps de calcul des émissions par BBI (basée sur un partitionnement de type Kdtree) est réduit de seulement 20%.

4.3.3 Conditions expérimentales

Dans la suite des expériences de ce chapitre, nous utilisons le système HTK/Sirocco et des modèles acoustiques indépendants du contexte à 512 gaussiennes par état. Le regroupement est réalisé avec la distance KLP puisqu'elle a permis d'obtenir les meilleures performances dans les expériences précédentes.
En s'inspirant de [5], les performances des différents systèmes proposés seront mesurées en termes de taux de reconnaissance (ou mots erronés) et de réduction de nombre de densités calculées. La fraction de densités calculées est définie par :

$$C = \frac{(G_{select} + G_{qv})}{G_{total}} \tag{4.3}$$

avec G_{select} et G_{total} sont respectivement le nombre distributions utilisées et totales. G_{qv} les la taille du *codebook*.

4.3.4 Sélection mono-niveau

Les méthodes développées dans ce paragraphe utilisent la classification hiérarchique des distributions et une coupure de l'arbre binaire à un seul niveau pour réduire le nombre global de densités calculées sans perte significative des performances. Ces méthodes sont principalement la quantification vectorielle et la sélection des gaussiennes par état.

Quantification vectorielle

On se propose d'évaluer les modèles réduits et d'explorer leurs limites de performance. Nous procédons en deux étapes :

- Nous calculons la vraisemblance avec toutes les distributions centroïdes.
- Nous nous limitons à un nombre maximal de distributions en commençant par les plus vraisemblables.

Les résultats respectifs sont reportés dans le tableau 4.1 et la figure 4.8.

Modèles réduits	WER (%)
40	39.2
120	36.4
512	35.6

TAB. 4.1 – Performances de modèles réduits

D'après le tableau 4.1, les performances des modèles réduits à 120 gaussiennes (c'est à dire 23% du nombre global des distributions) sont légèrement inférieures à celles du système initial (+0.8% WER absolu). Par contre, les résultats du système réduit à 40 gaussiennes ne sont pas intéressants ; a priori le nombre de classes n'est pas suffisant.

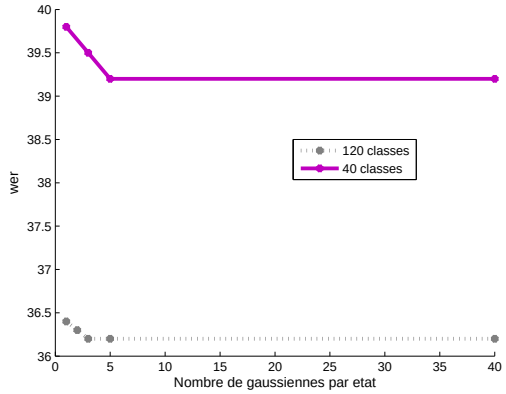


FIG. 4.8 – Calcul de la vraisemblance en utilisant les meilleurs *codewords* des modèles réduits 60 et 120

On peut déduire de la figure 4.8 que pour le système à 40 gaussiennes par état, les 5 meilleures gaussiennes sont suffisantes pour obtenir les mêmes performances que 40. Pour les modèles réduits à 120 gaussiennes par état, seulement les trois premières sont intéressantes. Ceci n'est pas surprenant puisque les modèles 120 sont plus précis que ceux avec 40 gaussiennes.

Sélection des gaussiennes par état

Les vraisemblances des *codewords* sont calculées et triées dans l'ordre décroissant. Puis les *shortlists* des premiers *codewords* sont utilisées pour le calcul de la vraisemblance globale.
Comme le nombre de distributions retenues à chaque état est variable, une valeur moyenne est considérée. Cette valeur est définie par le rapport entre le nombre de vraisemblances calculées et le nombre de noeuds correspondants dans le graphe de décodage. Les tests portant sur les modèles réduits 40 et 120 gaussiennes par état, en faisant varier le nombre de *shortlists* retenues, ont permis d'obtenir les résultats de la figure 4.9.

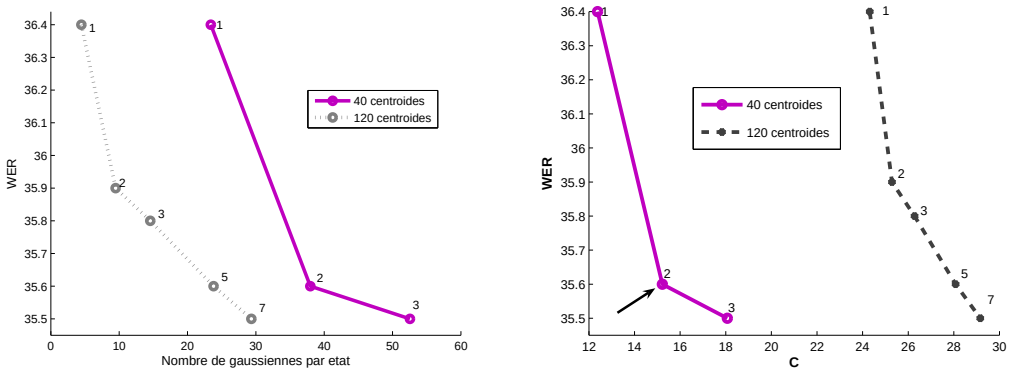


FIG. 4.9 – Calcul des vraisemblances en utilisant les meilleures *shortlists*

Plusieurs remarques peuvent être formulées :

- en comparant ces résultats à ceux du paragraphe précédent, on peut constater que le taux d'erreur est plus faible lorsqu'on utilise les *shortlists* retenues plutôt que les *codewords* correspondants. Ce qui est logique et attendu.
- la figure 4.9 (à gauche) montre que pour le même taux d'erreur, le nombre de gaussiennes utilisées est plus faible pour les modèles à 120 centroïdes que ceux à 40. En particulier, au moyen de seulement une vingtaine de de gaussiennes, on obtient pratiquement les mêmes performances que le système initial à 512 gaussiennes.
- le taux de calcul des vraisemblances C est globalement plus intéressant pour les modèles réduits 40 que 120 (figure 4.9 (à droite)). En effet, ce facteur est très vite augmenté dans le dernier cas par la taille du *codebook* à savoir 120.
- le meilleur compromis entre un facteur C faible et une dégradation non significative du taux d'erreur est atteint pour système réduits 40 en retenant les deux meilleures *shortlists*. Ce taux correspond à une réduction du temps de calcul des vraisemblances d'un facteur de *sept*.

Sélection des gaussiennes des *shortlists*

La sélection des gaussiennes par état a permis de confirmer la présence de quelques *shortlists* déterminants dans le calcul de la vraisemblance (paragraphe 4.3.4). Ainsi, on se propose de repousser les limites de cette méthode en cherchant à vérifier si toutes les distributions gaussiennes des *shortlists* retenues sont intéressantes ou pas. Pour ce faire, une fois les meilleures *shortlists* choisies, les vraisemblances de leurs distributions gaussiennes sont calculées et triés. Puis seulement les plus vraisemblables d'entre elles sont utilisées pour le calcul de la vraisemblance globale. Les expériences ont été menées avec des modèles réduits à 40 centroïdes. Nous avons retenu les deux meilleures

shortlists ce qui correspond à un nombre moyen de 38 gaussiennes par état. En faisant varier le nombre de gaussiennes retenues on obtient les résultats de la figure 4.10.

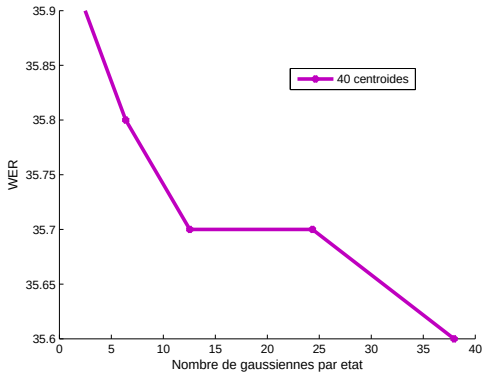


FIG. 4.10 – Sélection des gaussiennes des *shortlists* retenues

On remarque que seulement 5 gaussiennes suffisent pour avoir à peu près le même taux d'erreur que celui des *shortlists* retenus. Ce taux d'erreur n'est pas différent de celui des modèles initiaux à 512 gaussiennes par état (dans l'intervalle de confiance). Par conséquent, l'objectif des expériences qui suivent est de détecter ces distributions.

Sélection en fonction du poids

Les distributions des *shortlists* retenues sont triées par poids. Les distributions gaussiennes de poids important sont retenues. Les expériences ont été menées avec les modèles réduits à 40 gaussiennes par état en utilisant les distributions des deux meilleures *shortlists*. On faisant varier le seuil du nombre de densités calculées, on obtient les résultats de la figure 4.11.

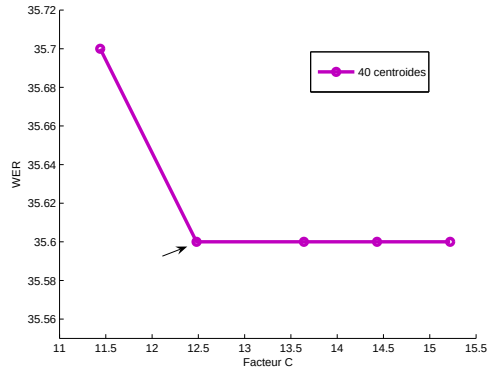


FIG. 4.11 – Sélection des gaussiennes des *shortlists* retenues par poids

Le facteur C est réduit à 12 % avec une dégradation négligeable des performances. Ceci correspond à une réduction du temps de calcul des vraisemblances d'un facteur de neuf.

4.3.5 Sélection multi-niveaux

Dans les expériences précédentes, un seul niveau de coupure de l'arbre est considéré et on a pu constater que le système est d'autant plus précis que la taille de son *codebook* est importante. C'est le cas par exemple des modèles réduits 120 gaussiennes par état dont la quantification vectorielle est plus performante que les modèles réduits 40. Or, ce cas a été vite écarté, en sélection des gaussiennes, à cause de la taille du *codebook* qui augmente de façon importante le temps de calcul des vraisemblances (paragraphe 4.3.4).

Aussi proposons-nous une approche multi-niveaux afin de tirer profit de la précision des modèles réduits à large *codebook* et de la rapidité de ceux à *codebook* limité.

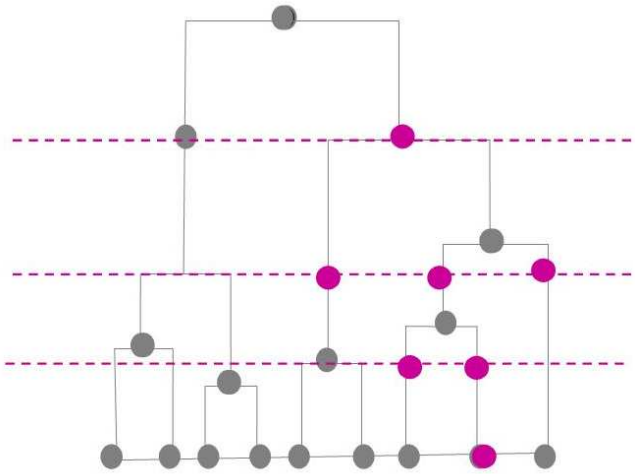


FIG. 4.12 – Coupure multi-niveaux de l'arbre binaire

Suite à la coupure de l'arbre en plusieurs niveaux, différents calculs de la vraisemblance sont envisageables. Dans le cas de deux niveaux, le calcul de la vraisemblance à un état peut être déduit comme suit :

1. calculer les vraisemblances des *codewords* du niveau supérieur, les trier et en sélectionner les plus vraisemblables.
2. trouver les *codewords* correspondants du niveau inférieur.
3. calculer la vraisemblance en utilisant :
   a. les *codewords* du niveau inférieur, ce qui revient à une quantification vectorielle.
   b. trier les *codewords* du niveau inférieur et utiliser les *shortlists* correspondants aux *codewords* les plus vraisemblables. C'est la sélection des gaussiennes par état appliquée au niveau inférieur.

a) Quantification vectorielle

Deux niveaux sont considérés : 40 et 120 *codewords*. Les vraisemblances des 40 *codewords* sont calculées et triées. Puis, on fait varier le nombre de *codewords* retenus (en commençant par les meilleurs), et on relève les *codewords* correspondants au niveau 120. La vraisemblance de l'état est calculée au moyen des *codewords* 120 relevés.

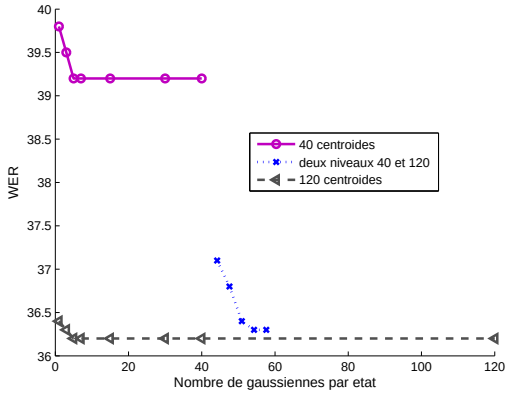


FIG. 4.13 – Taux d'erreur des *codewords* mono niveau 40, 120 et multi niveaux 40-120

D'après la figure 4.13, on peut remarquer que le WER est largement inférieur à celui obtenu en considérant seulement le niveau 40 et légèrement supérieur à celui issu d'un regroupement mono niveau 120.
Par ailleurs, le nombre de densités calculées est assez faible par rapport au taux d'erreur engendré. En effet, avec seulement 50 gaussiennes par état, la perte en taux d'erreur par rapport au système 512 gaussiennes par état initial est de 0.8%. Ce même résultat est obtenu par une quantification vectorielle mono-niveau avec 120 *codewords* (paragraphe 4.3.4).

b) Sélection des gaussiennes à deux niveaux

Ici, on a suivi la même procédure que ci-dessus, seulement pour le calcul de la vraisemblance ce sont les *shortlist*s du niveau inférieur qui sont utilisées et non les *codewords* correspondants. Deux expériences sont réalisées en considérant les deux bi-niveaux : 40-60 et 40-120.
Les résultats respectifs sont indiqués sur la figure 4.14.
Globalement, les résultats sont assez intéressants. Pour le regroupement bi-niveau 40-60, on remarque le même taux d'erreur que le système initial à 512 gaussiennes par état est atteint avec à peu près seulement 80 densités calculées (C=15.78%).
Pour le regroupement bi-niveau 40-120, ce taux pourrait être atteint avec seulement 67 calculs de densité (C=13.21%).

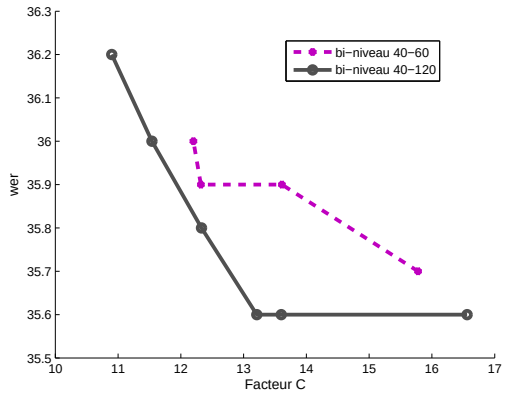


FIG. 4.14 – Calcul de la vraisemblance au moyen des shortlists du niveau inférieur pour une coupure bi-niveau 40-120 et 40-60.

Ces résultats sont plus performants que ceux obtenus avec une sélection des gaussiennes à un seul niveau (paragraphe 4.3.4).

Sélection des gaussiennes par poids

Cette sélection est appliquée après la sélection à deux niveaux sus-mentionnée où on fait varier le nombre de gaussiennes récupérées au niveau des feuilles. On sélectionnera celles de poids importants. Dans ces expériences, nous avons deux niveaux : 40 et 60. On a utilisé les deux meilleures classes du niveau 40 et nous avons fait varier le seuil de poids. Le meilleur compromis entre WER et le facteur

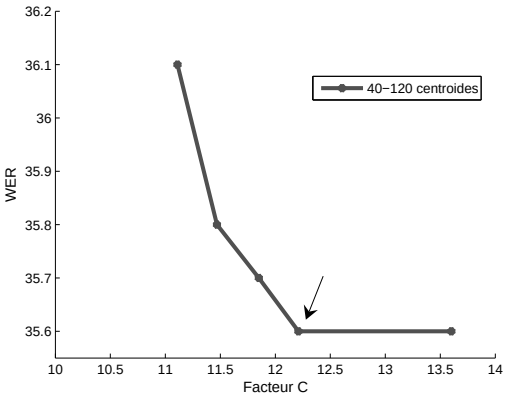


FIG. 4.15 – Sélection des gaussiennes par poids

C est 35.7% et 11.85%. Ce qui correspond à une réduction du calcul des vraisemblances de 8.34 fois sans pratiquement de perte de performance. La sélection des gaussiennes à deux niveaux s'est alors améliorée.

Sélection des gaussiennes à trois niveaux

En multi-niveaux l'arbre est parcouru de la racine aux feuilles. La sélection est opérée en cascade et à chaque niveau on ne retient que les distributions les plus vraisemblables avant de passer au niveau inférieur.

La sélection des gaussiennes appliquée à trois niveaux se résume ainsi :

1. on calcule toutes les densités du niveau supérieur, on les trie et pour les meilleures d'entre elles on cherche les distributions du niveau inférieur correspondantes.
2. les densités de ces distributions sont également calculées, triées et pour les meilleures elles on cherche les distributions du niveau inférieur correspondantes.
3. même procédure que l'étape (2).
4. toutes les gaussiennes situées au niveau des feuilles correspondantes sont utilisées pour le calcul de la vraisemblance.

Les expériences ont été menées en considérant trois niveaux correspondant à 40, 60 et 120 centroïdes. Pour le niveau 1 (40 centroïdes) nous avons retenu les trois meilleurs centroïdes. Puis nous avons fait varier le pourcentage de densités retenues au niveau 2 (en utilisant 80% ou 90%). Pour le dernier niveau (120 centroïdes), 90% des densités sont retenues. Les résultats correspondants sont reportés sur la figure 4.16.

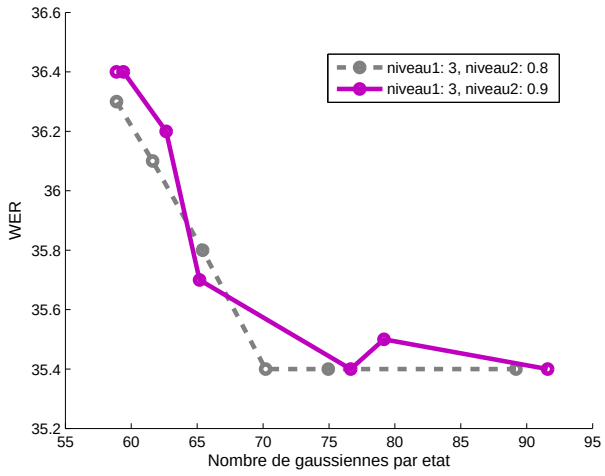


FIG. 4.16 – Sélection des gaussiennes à trois niveaux

D'après ces courbes 4.16, deux niveaux suffisent.

4.4 Conclusion

Dans ce chapitre, nous avons proposé une méthode de classification basée sur le partitionnement hiérarchique. Pour ce faire nous avons défini la distance de *Kullback Leibler Pondérée* et démontré son efficacité. Les tests de validation sont réalisés pour trois critères de classification des distributions dont les plus intéressants sont basés respectivement sur la similarité et la perte en vraisemblance. Dans une seconde étape, nous avons étudié l'impact de cette classification sur la sélection des gaussiennes multi-niveaux. L'idée consiste à retenir seulement distributions dont tous les noeuds supérieurs sont vraisemblables. Ceci permet d'éliminer progressivement les feuilles (distributions) dont le score des parents sont faibles tout en approchant au mieux les distributions vraisemblables. D'autres critères de sélection basés essentiellement sur la quantité de données d'apprentissage sont rajoutés pour affiner d'avantage la sélection. L'application de cette méthode a permis de réduire le nombre de densités calculées à 12% sans perte significative de performance.

Chapitre 5

Sélection Contextuelle des Gaussiennes

5.1 Introduction

Dans le chapitre précédent, nous avons décrit des approches de sélection des gaussiennes basées sur le partitionnement hiérarchique. Dans ce chapitre, on s'intéresse plutôt à des méthodes s'appuyant sur un regroupement des distributions de type k-moyennes dans le but de réaliser la sélection des gaussiennes. Il se subdivise en deux grandes parties :

1. dans la première partie certaines méthodes représentatives de l'état de l'art sont rappelées puis appliquées. Il s'agit plus précisément de la méthode classique de sélection des gaussiennes (paragraphe 3.6.1) et de certaines de ses variantes. Ces méthodes seront évaluées en utilisant notre système de base et nos ressources associées.
2. dans la seconde partie, quelques contributions sont présentées et expérimentées dans les mêmes conditions. Ces contributions concernent :
   - la sélection des gaussiennes basée sur le contexte.
   - la sélection des gaussiennes à l'issue d'un partitionnement hiérarchique. C'est une extension de la méthode de partitionnement hiérarchique proposée au chapitre 4, qui est basée sur le contexte.
   - la sélection contextuelle des trames.

Toutes ces méthodes sont comparées entre elles et par rapport au système de base. Pour ce faire, on se place dans les mêmes conditions (décrites dans le paragraphe 1.3.3). Les mêmes outils de mesure de performance sont également adoptés.

5.2 Sélection classique des gaussiennes

5.2.1 Principe

Rappelons le principe de la technique de sélection classique des gaussiennes (GS) décrite dans le paragraphe 3.6.1. La probabilité d'émission d'une observation o par un état i :

$$b_i(o) = \sum_{m \in M_i} \omega_m \mathcal{N}(o, \mu_m, \Sigma_m) \quad avec \sum_{m \in M_i} \omega_m = 1 \tag{5.1}$$

avec : M_i le nombre de composants du mélange de l'état i, ω_m est le poids d'une gaussienne m, $\mathcal{N}$ est une loi normale de moyenne μ et de matrice de covariance Σ supposée diagonale.
Les modèles acoustiques sont d'autant plus précis que les vecteurs de paramètres sont proches des moyennes des distributions gaussiennes. Lorsqu'une observation se trouve au niveau de la "queue" d'une distribution gaussienne on parle de "outlier" et la contribution de la gaussienne correspondante au calcul de sa vraisemblance est faible. L'objectif de GS classique est de détecter les gaussiennes correspondantes aux "outliers" pour ne pas les utiliser [5, 37].
Pour ce faire, une classification de toutes les distributions au moyen de la distance Euclidienne pondérée (δ) entre leurs moyennes est réalisée.

$$\delta(\mu_i, \mu_j) = \frac{1}{d} \sum_{k=1}^{d} \{\Omega(k)(\mu_i(k) - \mu_j(k))\}^2 \tag{5.2}$$

d est la dimension des observations, $\mu_i(k)$ est le k^{eme} composant du vecteur μ_i et $\Omega(k)$ est l'inverse de la racine carré du k^{eme} élément de la moyenne des covariances de toutes les gaussiennes. A chaque itération on obtient des classes (clusters) χ_ϕ représentées chacune par un centroïde ou *codeword* c_ϕ tel que :

$$c_\phi = \frac{1}{card(\chi_\phi)} \sum_{m \in \chi_\phi} \mu_m \quad \phi = 1..\Phi \tag{5.3}$$

Le processus est répété en optimisant la distorsion moyenne (δ_{avg}) :

$$\delta_{avg} = \frac{1}{M} \sum_{m=1}^{M} \left\{ \min_{\phi=1}^{\Phi} \delta(\mu_m, c_\phi) \right\} \tag{5.4}$$

M est le nombre total de distributions.
Les clusters ainsi produits sont disjoints. Leur utilisation en reconnaissance de la parole peut engendrer des erreurs. Par conséquent, pour construire des classes qui se partagent certaines distributions, une nouvelle définition du voisinage v_ϕ du codeword c_ϕ est introduite comme suit :

$$\mathcal{N}(., \mu_m, \sigma_m) \in v_\phi \quad si \quad \frac{1}{d} \sum_{i=1}^{d} \frac{(c_\phi(i) - \mu_m(i))^2}{\sigma_m^2(i)} \leq \theta \tag{5.5}$$

θ est un seuil fixé a priori.
Les distributions gaussiennes d'un même cluster forment une *shortlist*.

Comme l'estimation de la variance est souvent bruitée, on a choisi le critère suivant :

$$\mathcal{N}(.,\mu_m,\sigma_m) \in v_\phi \quad si \quad \frac{1}{d}\sum_{i=1}^{d}\frac{(c_\phi(i)-\mu_m(i))^2}{\sigma^2_{avg}(i)} \leq \theta \tag{5.6}$$

σ_{avg} est la moyenne des matrices de covariances.

Pendant le décodage, les distances de l'observation à tous les centroïdes sont calculées. Le centroïde le plus proche est retenu. Puis, seules les distributions gaussiennes dont la distance à ce centroïde est inférieure à θ (déterminée empiriquement) sont retenues pour le calcul de la vraisemblance. Les autres sont remplacées par une valeur constante.

5.2.2 Système de base

Les expériences qui suivent sont réalisées avec le système Sphinx sur une heure de parole (Radio Classique) de la base de test Ester (paragraphe 1.3.3). Suite aux optimisations de temps faites dans le chapitre 3 (élagage d'hypothèses de mots), le système de base tourne à 4 fois le temps réel sur la machine de caractéristiques décrites dans le paragraphe 3.2.1. Rappelons que les modèles acoustiques de base sont de type triphone (CD), à 32 gaussiennes par état et 6108 états liés. A ces modèles, nous rajoutons des modèles indépendants du contexte (CI) à 32, 64 et 128 gaussiennes par état que l'on utilisera dans certaines expériences faisant intervenir le contexte.

Pour l'évaluation de ces modèles, nous avons relevé les temps cpu de décodage TG, de calcul des vraisemblances TV et de recherche TR, ainsi que le taux de mots erronés WER (%) et le nombre de densités calculées par observation GAUS/FR. TG= TV+TR

Type	Gauss/état	TG	TV	TR	WER	GAUS/FR
CD	32	3.92	2.53	1.38	28.7	130481
CI	32	3.05	0.37	2.67	37.7	3456
CI	64	3.21	0.44	2.76	36.5	6912
CI	128	3.38	0.68	2.65	35.2	13747

TAB. 5.1 – Performances des modèles dépendants et indépendants du contexte

Type désigne le type du modèle. Rappelons aussi que l'intervalle de confiance du système de base (de type CD) à 95% est [28.35 ;29.11].

On constate que les modèles dépendants du contexte sont nettement plus précis que ceux indépendants du contexte. Le nombre de densités calculées (GAUS/FR) et par la suite le temps de calcul des vraisemblances (TV) sont également beaucoup plus élevés. Ce qui est attendu vu le nombre de paramètres mis en jeu.

Pour les modèles indépendants du contexte (CI), on remarque que le temps de recherche est important. En effet, dans ce cas les scores des états sont proches et beaucoup de chemins sont à parcourir avant de trouver le chemin optimal.

5.2.3 Quelques expériences

Pour évaluer l'influence de la méthode de sélection classique des gaussiennes sur notre système de base, nous avons réalisé deux types d'expérience : en premier lieu nous avons fait varier la taille du *codebook*, en second lieu nous avons effectué des tests avec d'autres données (plus compliquées).

Influence du *codebook*

Pour une classification de toutes les gaussiennes (de tous les états) en 512, 1024 et 3456 classes, on fait varier le seuil θ et on mesure le taux de mots erronés (WER) et le nombre de densités calculées par observation (GAUS/FR) dont on déduit la fraction de calcul des vraisemblances C. Les états qui ne disposent pas de composantes dans la classe sélectionnée sont approximés par la vraisemblance de l'état du modèle indépendant du contexte correspondant. Cette valeur, étant préalablement calculée (voir paragraphe 3.4.1) n'entraîne pas de surcroît de calcul.

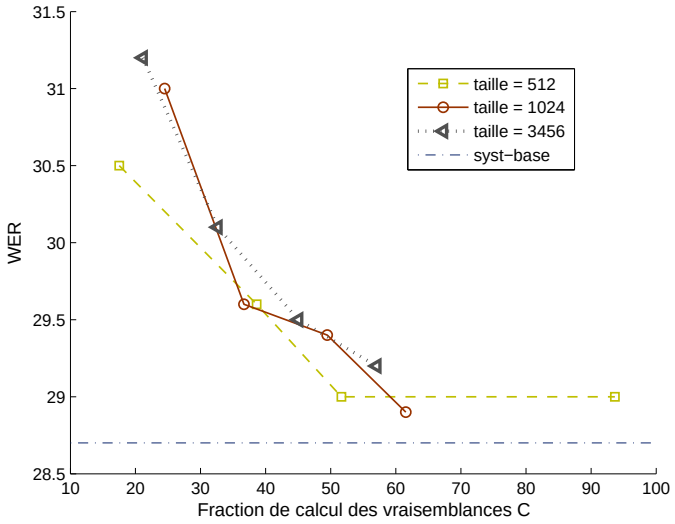


FIG. 5.1 – Sélection classique des gaussiennes avec taille du codebook variable

D'après la figure 5.1, on peut constater que :

- les performances du système (WER,C) ne sont pas proportionnelles à la taille du *codebook*. Une taille du *codebook* de 512 donne des résultats légèrement meilleurs que ceux des *codebooks* de tailles 1024 et 3456. Mais, vu la largeur de intervalle de confiance, on peut considérer que les courbes correspondantes sont similaires.
- le meilleur compromis entre C et WER correspond au couple (WER,C)=(29.0%,51.64%). Ce qui veut dire qu'à peut près la moitié des densités sont calculées pour une dégradation absolue du taux d'erreur de 0.3%.

– le facteur C est toujours plus élevé que celui des travaux dans [24] (à savoir 35%) pour une même augmentation de WER. Ces derniers sont réalisés en utilisant la base ARPA 1994 H1 (dictée de journaux dans de bonnes conditions d'enregistrement).

Ceci pourrait expliquer la différence de résultats. En effet, la base Ester que nous utilisons est assez bruitée (plusieurs microphones, des canaux téléphoniques, parfois avec un fond musical). Par conséquent, la variabilité des modèles acoustiques est plus importante et nécessite un nombre assez important de composantes.

Pour confirmer cette hypothèse nous avons réalisé les expériences sur l'influence des données ci-après.

Influence des données

Nous avons appliqué la méthode de sélection classique des gaussiennes aux données de test ESTER de la Radio Marocaine (RTM) qui est plus bruitée que la Radio Classique.

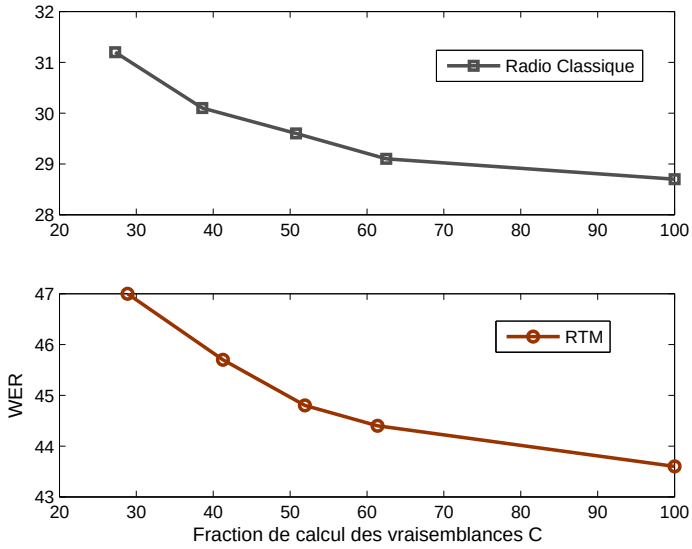


FIG. 5.2 – Impact de la quantité de données sur la sélection des gaussiennes

La courbe relative aux données RTM montre une dégradation plus rapide du WER en fonction de C que Radio Classique. Par exemple, pour 62% de densités calculées la WER de Radio Classique augmente de 0.4% absolu alors que pour RTM cette augmentation est de 0.8%. Ce qui confirme le fait que la sélection classique soit moins performante pour des données bruitées. En effet, dans ce dernier cas, la variabilité acoutique est plus importante et requièrt par conséquent plus de distributions.

5.3 Sélection classique contrainte

5.3.1 Alternatives au repli

Pour éviter l'élagage de certains états actifs qui ne disposent pas de distributions dans la classe choisie et comme alternatives au repli pratiqué dans les expériences ci-dessus, nous avons évalué :

1. l'utilisation de la distribution la plus proche de la classe retenue, ainsi pour chaque état actif, au moins une gaussienne est utilisée.
2. le calcul de toutes les densités de l'état.

Pour 3456 classes, les résultats sont rapportés sur la figure 5.3 et les tableaux 5.2, 5.3, 5.4.

θ	TG	TV	TR	WER	GAUS/FR	C (%)
syst-base	3.95	2.54	1.39	28.5	130 481	100
1.3	3.08	1.85	1.19	33.0	15287	11.71
1.6	3.18	1.95	1.19	31.2	27576	21.13
1.9	3.22	1.95	1.27	30.1	42578	32.63
2.2	3.35	2.07	1.26	29.5	58774	45.04
2.5	3.45	2.16	1.29	29.2	74461	57.04

TAB. 5.2 – Sélection classique des gaussiennes avec repli

θ	TG	TV	TR	WER	GAUS/FR	C (%)
1.3	2.24	1.14	1.09	33.1	20185	18.11
1.6	2.34	1.27	1.06	31.2	32110	27.25
1.9	2.56	1.48	1.08	30.1	46873	38.57
2.2	3.20	1.97	1.24	29.6	62803	50.78
2.5	3.48	2.24	1.23	29.1	78116	62.51
2.9	3.72	2.39	1.33	29.0	95482	75.82

TAB. 5.3 – Sélection classique avec au moins une gaussienne par état

θ	TG	TV	TR	WER	GAUS/FR	C (%)
1.3	2.25	1.14	1.10	33.1	20185	18.11
1.6	3.19	1.96	1.22	30.9	66660	53.73
1.9	3.39	2.08	1.29	29.7	66698	58.76
2.2	3.51	2.19	1.32	29.5	73265	58.89
2.5	3.65	2.31	1.33	29.0	83250	66.45

TAB. 5.4 – Sélection classique avec une ou toutes les gaussienne par état

On peut remarquer que :

– ce n'est pas seulement TV qui est proportionnel au nombre de densités calculées mais aussi TR et TG. Toutefois, les variations de TV sont plus importantes que celles de TR car il dépend plus du nombre de densités calculées.

– l'utilisation de toutes les distributions d'un état dont aucune composante n'appartient à la classe choisie ne fait que détériorer les résultats.

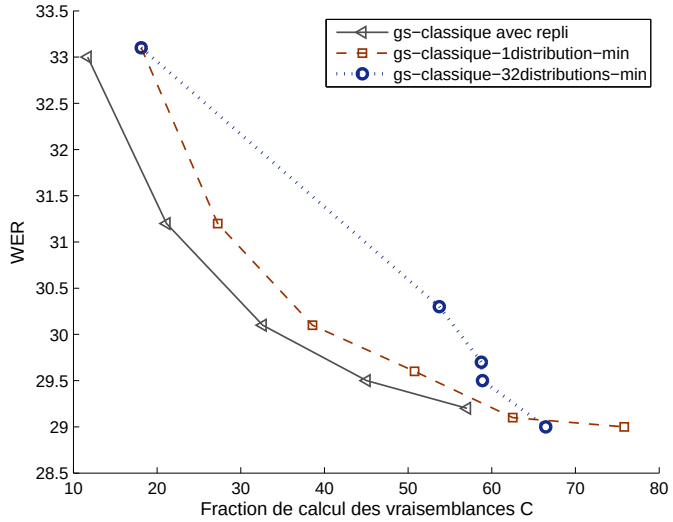


FIG. 5.3 – Taux d'erreur en fonction de la fraction C

- le meilleur compromis entre WER et C correspond aux valeurs respectives 29.0% et $\sim$ 66 % donc une réduction du temps de décodage de l'ordre de 11%.
- vue la largeur de l'intervalle de confiance du système de base, on peut considérer que les résultats du repli et ceux du calcul de la vraisemblance avec une seule gaussienne sont similaires. La deuxième méthode sera utilisée par la suite.

5.3.2 Normalisation de la distance

Pour la formation des *codeword*s, une alternative proposée par [24] consiste à considérer la condition :

$$\mathcal{N}(.;\mu_m,\sigma_m) \in v_\phi \quad si \quad \frac{1}{d}\sum_{i=1}^{d}\frac{(c_\phi(i)-\mu_m(i))^2}{\sqrt{\sigma^2_{avg}(i)\sigma^2_m(i)}} \leq \theta \tag{5.7}$$

Ce qui encourage l'affectation des distributions dont la variance est grande à cette classe. L'expérimentation de cette mesure permet d'obtenir les résultats ci-dessous.

θ	TG	TV	TR	WER	GAUS/FR	C(%)
1.3	2.45	1.22	1.22	32.0	20 093	18.04
1.6	2.52	1.36	1.16	31.0	31 074	26.46
1.9	3.02	1.73	1.28	30.0	44 086	36.43
2.2	3.33	1.99	1.33	29.7	57 739	46.89
2.5	3.32	2.04	1.27	29.5	70 737	56.86

TAB. 5.5 – Sélection des gaussiennes avec normalisation de la distance

La normalisation de la distance améliore légèrement les performances du système initial. Ceci est en accord avec les résultats décrits dans [37].

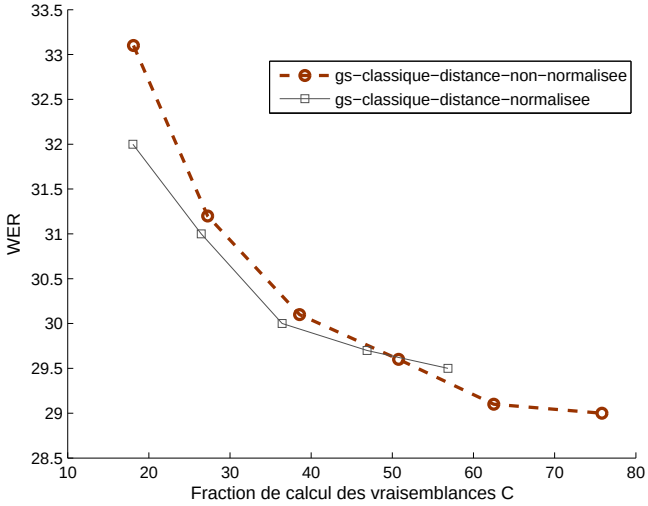


FIG. 5.4 – Sélection des gaussiennes avec et sans normalisation de la distance

5.3.3 Limitation du nombre de gaussiennes par état

En sélection classique des gaussiennes, toutes les gaussiennes appartenant à la classe choisie sont utilisées pour le calcul de la vraisemblance, et ce, indépendamment des états auxquels elles appartiennent. Autrement dit, la probabilité d'émission d'un état actif, peut être calculée aussi bien avec une gaussienne que par toutes les composantes du mélange. Pour réduire d'avantage le nombre de densités calculées, Knill et al. [37] ont proposé de limiter le nombre de gaussiennes utilisées par chaque état actif. Pour tester cette approche, nous avons choisi un seuil θ assez large de 2.5 et nous avons limité le nombre maximal de gaussiennes retenues par état dans les clusters à *maxcount*.

maxcount	TG	TV	TR	WER	GAUS/FR
1	2.14	0.80	1.33	66.5	6618
2	2.35	0.99	1.35	49.3	8593
3	2.49	1.16	1.32	43.5	12689
4	2.62	1.31	1.30	40.3	16840
5	2.77	1.46	1.29	37.9	21032
6	2.91	1.61	1.29	36.7	25207
32	3.53	2.24	1.33	29.1	78116

TAB. 5.6 – Sélection classique avec un nombre maximum de gaussiennes par état et par classe

D'après le tableau 5.6, le WER est assez élevé. On peut expliquer ce résultat par le fait que le choix des gaussiennes pour un état donné est arbitraire. Ceci étant en effet indépendant de leur apport en vraisemblance.

5.4 Sélection contextuelle

Dans la méthode classique de sélection des gaussiennes (paragraphe 5.2.1) aucune information sur le contexte n'est prise en compte pour la formation des centroïdes. En plus, le regroupement de toutes les distributions représentant tous les contextes fait perdre l'information sur le contexte. De ce fait, certains contextes peuvent ne pas être représentés par les centroïdes, et les distributions correspondantes sont assignées à des centroïdes différents. Pour remédier à un tel inconvénient, nous proposons une méthode de sélection des gaussiennes basée sur le contexte.

5.4.1 Principe

La sélection contextuelle des gaussiennes consiste à considérer comme centroïdes ou *codebooks* l'ensemble des distributions de tous les états des modèles indépendants du contexte. Les distributions dépendantes du contexte sont associées à ces dernières au moyen d'une distance Euclidienne pondérée.

On obtient autant de classes que de distributions indépendantes du contexte. Les *shortlists* sont composées des distributions dépendantes du contexte assignées à chaque classe. Lors du décodage

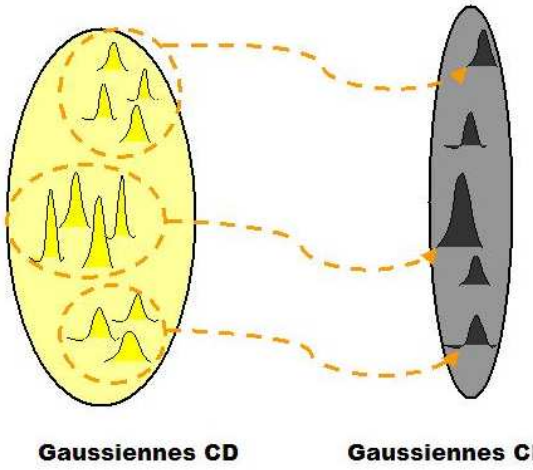


FIG. 5.5 – Asssociation entre distributions dépendantes et indépendantes du contexte

la vraisemblance est calculée de la même manière que la méthode classique de sélection des gaussiennes, en utilisant en premier lieu les centroïdes. Puis, la classe dont le score du centroïde est le plus important est choisie, et les distributions de sa *shortlist* sont utilisées pour le calcul effectif de la vraisemblance.

5.4.2 Expériences

Pour valider cette approche, des expériences sont réalisées en utilisant le même système de base que les expériences précédentes de ce chapitre. La taille du *codebook* de 3456 provient de l'utilisation de 36*3*32 distributions indépendantes du contexte issues de 36 modèles à 3 états chacun et 32 gaussiennes par état. En faisant varier la valeur du paramètre θ (voir pargraphe 5.2.1) on obtient les résultats du tableau 5.7 et de la figure 5.6.

θ	TG	TV	TR	WER	GAUS/FR	C (%)
1.3	2.21	1.11	1.10	33.2	16590	15.36
1.6	2.55	1.37	1.17	31.5	27967	24.08
1.9	2.81	1.60	1.20	30.2	42251	35.02
2.2	3.09	1.85	1.23	29.3	57902	47.02
2.5	3.54	2.19	1.34	29.3	73239	58.77
2.9	3.76	2.40	1.36	29.2	91191	72.53

TAB. 5.7 – Sélection contextuelle

D'après les tableaux 5.7 et la figure 5.6, cette méthode apporte peu d'amélioration par rapport à la méthode classique de sélection des gaussiennes.
Comparée au système de base, on relève un point intéressant (WER, C)=(29.3%,47.02%) qui correspond à une perte de WER de 0.6% pour à peu près la moitié des densités calculées. Ceci correspond également à un gain d'une fois le temps CPU puisque la vitesse de décodage passe de 4 à 3 fois le temps CPU.

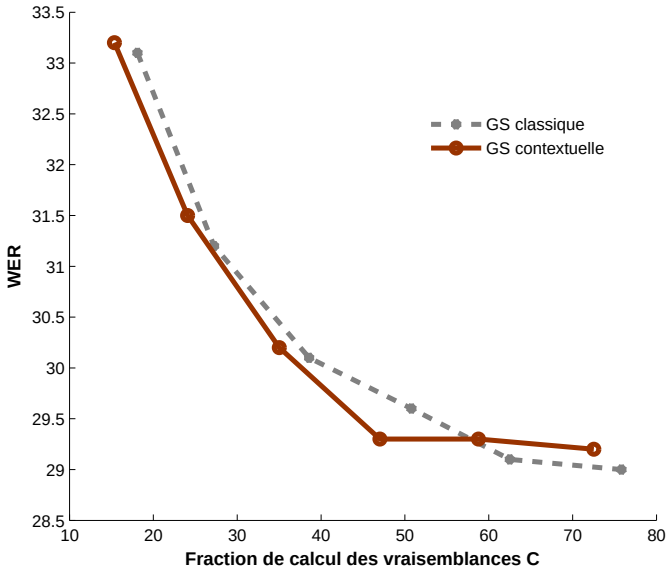


FIG. 5.6 – Taux d'erreur en fonction de la fraction C

5.5 Sélection contextuelle et partitionnement hiérarchique

Dans le chapitre 4, une méthode de partitionnement hiérarchique a été présentée. Elle permet d'améliorer la représentativité des modèles CI. Ainsi, pour améliorer la représentativité des *centroïde*s qui sont des distributions indépendantes du contexte, on s'est proposé d'appliquer la méthode de partitionnement hiérarchique. Le partitionnement est opéré pour chaque état. Le *codebook* final est construit en utilisant toutes les distributions de tous les états obtenues après partitionnement.

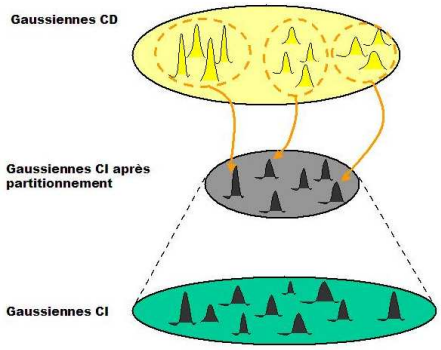


FIG. 5.7 – Correspondance entre distributions CD et celles CI après partitionnement

5.5.1 Influence du partitionnement

Maintenant, les modèles CI à 32 gaussiennes par état proviennent du partitionnement hiérarchique des modèles CI à 64 et 128 gaussiennes par état et coupure de l'arbre obtenu au niveau de 32 noeuds. Nous avons reconduit les mêmes expériences de sélection contextuelle en utilisant les nouveaux modèles CI. Les résultats sont reportés sur la figure 5.8.

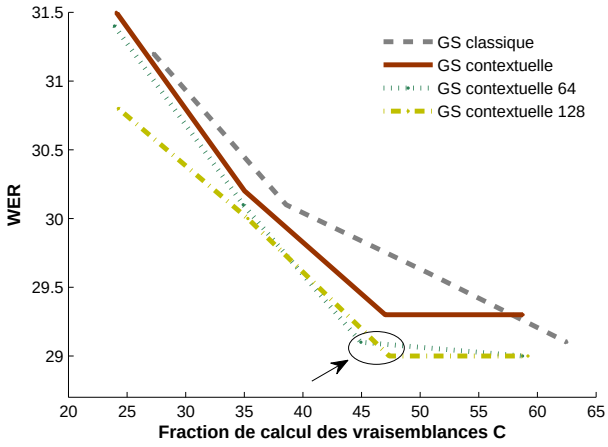


FIG. 5.8 – Correspondance entre distributions dépendantes du contexte et celles dépendantes du contexte après partitionnement

On constate que les courbes de sélection contextuelle avec partitionnement hiérarchique et des modèles initiaux 64 et 128 sont au desoous de celles de GS classique et GS contextuelle avec les modèles CI à 32 gaussiennes par état.
En particulier, pour à peu près 47% des densités, les systèmes avec partitionnement ont un taux d'erreur de moins 0.8% (ce qui est en dehors de l'intervalle de confiance). D'où l'intérêt du partitionnement.

5.5.2 Influence de la taille du *codebook*

En partant de modèles indépendants du contexte renfermant 32 gaussiennes par état (soit au total 32*108 = 3456 distributions gaussiennes), le partitionnement hiérarchique est réalisé pour chaque état pour passer de 32 à 5, 10 et 16 distributions gaussiennes par état. Ceci correspond à des *codebooks* de tailles 540 (108*5), 864 (108*10) et 1728 (108*16).

θ	TG	TV	TR	WER	GAUS/FR	C (%)
1.6	2.96	1.56	1.29	30.9	36354	28.27
1.9	3.19	1.85	1.33	30.0	52575	40.70
2.2	3.52	2.13	1.38	29.1	69093	53.36
2.5	3.65	2.26	1.36	29.2	84242	64.91

TAB. 5.8 – Sélection contextuelle hiérarchique : *codebook* de taille 540

θ	TG	TV	TR	WER	GAUS/FR	C (%)
1.6	2.73	1.47	1.26	30.9	32371	25.47
1.9	3.09	1.76	1.30	29.8	47850	37.33
2.2	3.34	2.01	1.33	29.1	64176	49.84
2.5	3.54	2.19	1.34	29.1	79588	61.65

TAB. 5.9 – Sélection contextuelle hiérarchique : *codebook* de taille 864

θ	TG	TV	TR	WER	GAUS/FR	C (%)
1.6	2.70	1.45	1.25	31.0	30246	24.50
1.9	3.08	1.76	1.31	30.0	45199	35.96
2.2	3.26	1.95	1.30	29.0	61239	48.25
2.5	3.48	2.15	1.32	29.1	76661	60.07

TAB. 5.10 – Sélection contextuelle hiérarchique : *codebook* de taille 1728

D'après la figure 5.9, la sélection contextuelle hiérarchique est d'autant plus intéressante que la taille du *codebook* est importante. Mais comme les courbes des *codebooks* 864 et 1728 sont pratiquement identiques, on peut parler de saturation ou limite de performance.

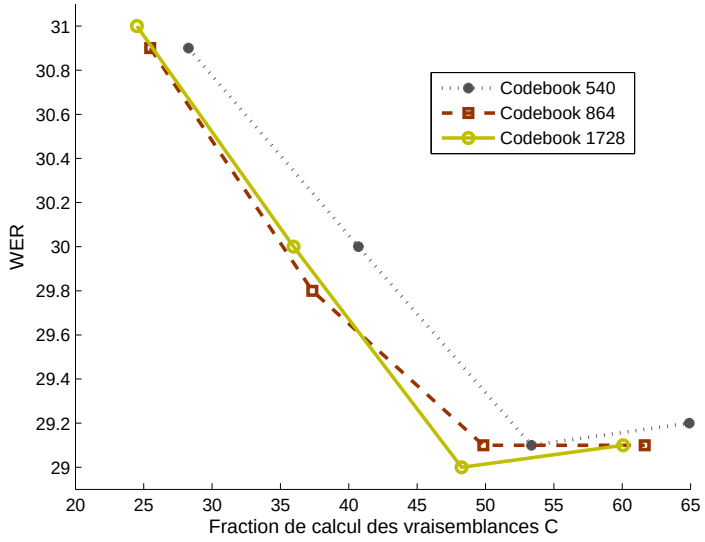


FIG. 5.9 – Influence de la taille du *codebook* sur la sélection hiérarchique contextuelle

5.6 Sélection des trames

5.6.1 Motivation

Sachant que le signal de parole est redondant, nous avons réalisé quelques expériences pour vérifier si c'est toujours les mêmes gaussiennes qui apportent le plus au calcul de la vraisemblance.

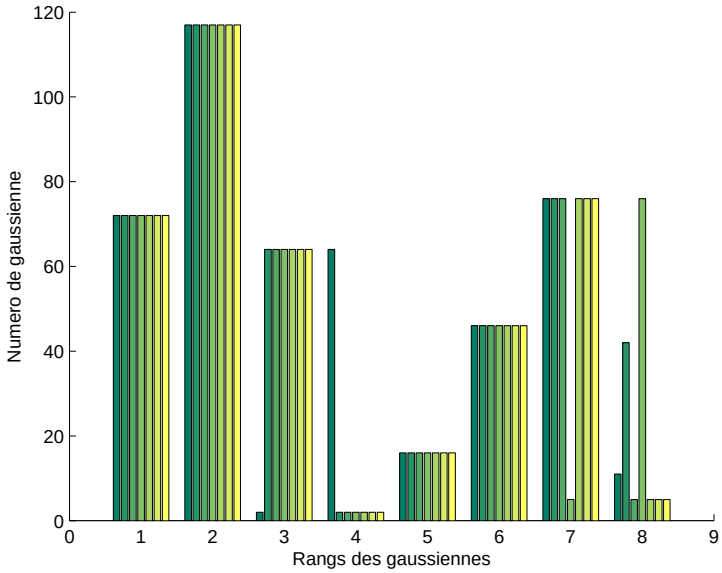


FIG. 5.10 – Classification de gaussiennes

D'après la figure 5.10, on voit que pour 7 échantillons successifs (de du son "a"), on a pratiquement les mêmes gaussiennes qui disposent des meilleurs scores. Ce fait est constaté sur d'autres échantillons, ce qui nous a encouragé à réaliser les manipulations de sélection de trames ci-après.

5.6.2 Sélection contextuelle des trames

La sélection des trames se déroule en deux temps :

1- Classification

- regrouper les distributions en k classes
- calculer un centroïde par classe (vecteur moyen des moyennes)

Deux types de regroupement ont été utilisés : le regroupement k-moyennes et un regroupement contextuel c'est-à-dire en supposant que les centroïdes sont des distributions des modèles CI.

2- Décodage : lors du calcul de la vraisemblance d'une trame i

- $trame^i$ assignée au centroïde j le plus proche
- si $trame^{i-1}$ est assignée au même centroïde j, alors la vraisemblance de la trame i est la même que celle de $i-1$ sinon elle est re-calculée.

Nous avons apliqué cette méthode pour différents nombres de classes (k = 540, 864 et 1728). Nous avons également relevé le taux d'erreur et le facteur de calcul des vraisemblances C.

k	WER	GAUS/FR	C (%)
1728	29.5	61380	47.04
864	29.2	64336	49.30
540	29.2	66506	50.96

TAB. 5.11 – Sélection des trames

k	WER	GAUS/FR	C (%)
1728	29.0	61239	46.93
864	29.1	64176	49.18
540	29.1	69093	52.95

TAB. 5.12 – Sélection contextuelle des trames

D'après les tableaux 5.12, 5.11 et la figure 5.11 on peut remarquer que :

- le WER obtenu en utilisant la sélection par regroupement contextuel est plus faible,
- pour la sélection contextuelle le temps est proportionnel au nombre de classes, ce qui n'est pas le cas pour le regroupement k-moyennes. D'où la robustesse de la première méthode,
- le meilleur compromis entre WER et C est WER = 29% et C = 47%. Ce qui coïncide avec le meilleur résultat de sélection contextuelle hiérarchique.

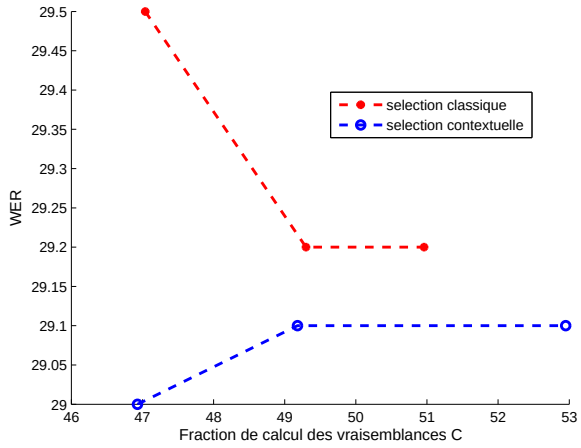


FIG. 5.11 – Taux d'erreur en fonction de la fraction C

5.7 Conclusion

Afin de réduire le temps de calcul des vraisemblances, nous avons employé des méthodes de sélection classique des gaussiennes et proposé des approches basées sur la sélection contextuelle. Nous avons constaté un gain de temps lors de l'introduction de l'information contextuelle par rapport à la sélection classique. Ce gain est encore plus important lorsque cette méthode est étendue à la sélection des trames. Ainsi seulement 47% des densités sont calculées sans pratiquement de perte de performance.

La combinaison de la sélection contextuelle des distributions gaussiennes et du partitionnement hiérarchique introduit au chapitre 4 de ces dernières donne des résultats comparables à ceux issus de la sélection contextuelle des trames.

Chapitre 6

Sous-Quantification Vectorielle Contextuelle

6.1 Introduction

Alors que les méthodes de sélection des gaussiennes décrites dans les chapitres précédents permettent de réduire le temps de calcul, la quantification vectorielle, qui diminue la complexité des modèles acoustiques, présente l'avantage d'utiliser à la fois moins de mémoire et de temps cpu.
Dans ce chapitre, nous proposons et nous décrivons des techniques de quantification et de sous-quantification vectorielles basées sur le contexte et le partitionnement hiérarchique.
Pour ce faire, nous évaluerons d'abord l'influence de l'information contextuelle sur les performances de la quantification vectorielle, et ce, à plusieurs niveaux : état, phone et tous les modèles.
Ensuite, nous nous intéresserons à la sous-quantification vectorielle qui n'est autre qu'une quantification par flux de données. Les résultats de cette dernière approche seront comparés à ceux de la quantification vectorielle et à ceux de la sous-quantification avec prise en compte du contexte (sous-quantification contextuelle).
Pour améliorer d'avantage les performances de la sous-quantification contextuelle, nous allons enfin réaliser une combinaison de cette méthode avec la technique de partitionnement hiérarchique décrite dans le chapitre 4.
Toutes les expériences de ce chapitre seront réalisées sur une même machine en utilisant les mêmes ressources et matériel que dans le chapitre précédent (voir paragraphe 5.2.2).
Les mêmes outils de mesure de performance seront également adoptés.

6.2 Quantification Vectorielle

Une quantification permet de discrétiser un espace continu et/ou de réduire la taille d'une base de données en minimisant la perte d'information.

Considérons un ensemble de P vecteurs $x_i, 1 \leq i \leq P$ dans un espace à D dimensions. Une quantification vectorielle (QV) de cet ensemble consiste a trouver Q vecteurs g_j $(1 \leq j \leq Q, \ \ avec \ \ Q < P)$ appartenant au même espace et qui minimisent un critère d'erreur lié à la perte d'information. Ces vecteurs g_j sont appelés les *centroïdes* ou *codewords*, et l'ensemble des *centroïdes* forme le *codebook*. Le critère d'erreur est généralement défini par une notion de distance entre les vecteurs de données et les *centroïdes*. Habituellement, l'erreur considérée est de la moyenne des carrés des distances Euclidiennes :

$$E = \frac{1}{P} \sum_{i=1}^{P} ||x_i - g_{k_i}||^2 \tag{6.1}$$

où k_i est l'indice du centroïde le plus proche du vecteur x_i.

6.2.1 Quantification par regroupement

Les méthodes de quantification de l'espace des distributions existantes sont souvent basées sur le regroupement [47, 73, 1].

La méthode proposée par [73] consiste à regrouper les moyennes et les variances des modèles initiaux au moyen de l'algorithme *k-moyenne*. Chaque classe (cluster) obtenue est représentée par un centroïde ou *codeword* qui n'est autre que la moyenne des vecteurs (moyenne ou variance) de la classe.

FIG. 6.1 – Quantification vectorielle

Une table de correspondance entre vecteurs moyennes (ou variances) et *centroïdes* correspondants est construite. Lors du décodage, les moyennes et variances des distributions sont remplacées par celles des *centroïdes* auxquels elles sont affectées pour le calcul des vraisemblances.

6.2.2 Quantification contextuelle

Les méthodes de QV existantes, souvent basées sur le regroupement des distributions (paragraphe 3.7), ne prennent pas en considération le contexte. Nous proposons de rajouter cette information de la même manière que pour la sélection des gaussiennes (chapitre 5).
L'idée est d'utiliser comme *codebook* les distributions des modèles indépendants du contexte (CI). Ces dernières sont supposées modéliser tous les contextes que l'on retrouve dans les modèles dépendants du contexte (CD).
De ce fait, les distributions CI forment les *centroïdes* de l'espace acoustique et les distributions CD sont affectées à ces derniers au sens d'une distance de Kullback Leibler symmétrisée. Ce choix de distance est motivé par le fait que c'est une mesure de dissimilarité entre distributions qui prend en considération la moyennne et la variance.
La QV contextuelle est réalisée par état, par phone ou pour tout l'espace de toutes les distributions.

- Lorsque la QV est opérée par état, les distributions de chaque état du modèle CI représentent le *codebook* de tous les états des allophones correspondants qui sont situés au même emplacement topologique. Cette idée est basée sur l'hypothèse que chaque état de modèle CI est une généralisation de certains états CD à plusieurs contextes.

- Pour la QV par phone, il s'agit d'une extension de la quantification contextuelle par état aux trois états d'un même phone ou allophone. Cette approche est motivée par le fait que les données utilisées pour l'estimation d'un phone sont partagées entre ses differents allophones.

- La QV de tout l'espace admet comme *codebook* toutes les distributions de tous les états CI.

6.2.3 Optimisation du temps

Nous considérons comme système de base le système Sphinx (paragraphe 5.2.2), dont les modèles acoustiques sont des triphones à trois états, 32 gaussiennes par état et 6108 états liés. Les monophones développés sont à 32, 64 et 128 gaussiennes par état et ont 108 états. Ils correspondent respectivement à un nombre global de 3456, 6912 et 13824 distributions.

1- **Système de base** : comme la vraisemblance d'une observation o est souvent calculée dans le domaine logarithmique, en dimension 1 elle est proportionnelle à :
$-log(\sigma_i) - 0.5 * \frac{(o_i - \mu_i)^2}{\sigma_i^2}$.
μ_i : moyenne de la distribution et σ_i sa variance.

Pour calculer cette quantité, nous avons besoin de 3 accès mémoire et 4 opérations arithmétiques par dimension et par gaussienne. Ainsi, si on désigne par N le nombre total de gaussiennes et D la dimension de l'espace des paramètres, alors le nombre d'opérations est $7ND$.
Dans notre cas : N= 6108 *32 et D = 39 d'où 7ND est de l'ordre de 53 millions opérations.

2- **Système quantifié** : désignons par M la taille du *codebook*. On distingue trois opérations :
- calcul des vraisemblances des *codewords* : 7MD
- stockage des scores des M *codewords* : M
- chargement mémoire de l'index du *codeword* et de la vraisemblance et sommation : 3N

Soit au total $7MD + M + 3N$ opérations.

Lorsque la QV est effectuée par état et que tous les états (108) sont actifs alors :

M =32/état → 64 millions d'opérations

M =64/état → 65 millions d'opérations

M =128/état→ 67 millions d'opérations

On remarque que la taille du codebook a un impact peu important sur le temps de calcul des vraisemblances. Comme il est presque impossible que tous les états soient simultanément actifs, on peut considérer un nombre d'opérations moyen de 20 millions.

Quantification par phone : si tous les 36 phones sont actifs

M =32*3/phone → 64 millions d'opérations

M =64*3/phone → 65 millions d'opérations

M =128*3/phone → 67 millions d'opérations

Si la moitié des phones sont actifs alors le nombre d'opérations est de l'ordre de 30 millions.

Quantification de tout l'espace :

Pour N = 3456 → 1.5 millions d'opérations

pour N = 6912 → 2.4 millions d'opérations

pour N = 13824 → 4.3 millions d'opérations

Le nombre d'opérations est proportionnel à la taille du codebook. D'où il faut trouver un compromis entre taille *codeboo*k optimale et un temps de calcul réduit.

6.2.4 Réduction de la mémoire

a- **Système initial** : essentiellement N moyennes et N variances de dimension D chacune. Ce qui fait 2 ND * 4 octets (si réel) ou encore 60 MB (mégabytes).

b- **Système après quantification** :

stockage (moyenne + variance) : 8 MB

table de correspondance (entiers) : 2N

score des *codewords* : 4M

Soit au total 8MD + 2N +4M ou encore → 1 MB , d'où un gain important en taille mémoire.

6.2.5 Expériences et résultats

Nous avons évalué pour les mêmes tailles de *codebook* les méthodes QV et QV contextuelle par état, par phone et de tout l'espace.

Pour la QV contextuelle par état, chacune des 32 distributions de chaque état CD est associée à la distribution la plus proche parmi 32, 64 ou 128 distributions (CI) au sens d'une distance de Kullback Leibler symetrisée.

Pour la QV contextuelle par phone, chaque *codebook* est formé des distributions de tous les états d'un même phone. Disposant de 3 états par phone, chacune des 32 distributions de chaque état CD est associée à la distribution la plus proche parmi 32*3, 64*3 ou 128*3 distributions (CI).

Les résultats de ces expériences sont reportés sur les tableaux 6.1, 6.2, 6.3, 6.4 et la figure 6.2

M	TG	TV	TR	WER	GAUS/FR	C(%)
3456	2.61	1.30	1.31	30.0	15 708	12.03
6912	2.77	1.43	1.34	29.5	15 101	11.57
13824	2.88	1.55	1.32	29.8	14 494	11.10

TAB. 6.1 – QV par regroupement : Taux d'erreur en fonction de la taille du *codebook*

M	TG	TV	TR	WER	GAUS/FR	C (%)
3456	2.91	1.63	1.27	30.9	29665	22.73
6912	2.87	1.60	1.27	30.8	23482	17.99
13824	3.01	1.73	1.27	30.8	20023	15.34

TAB. 6.2 – QV contextuelle par état : Taux d'erreur en fonction de la taille du *codebook*

M	TG	TV	TR	WER	GAUS/FR	C (%)
3456	2.61	1.39	1.22	30.5	22256	17.05
6912	2.62	1.37	1.20	30.4	18442	14.13
13824	2.67	1.46	1.20	30.7	16166	12.38

TAB. 6.3 – QV contextuelle par phone : Taux d'erreur en fonction de la taille du *codebook*

M	TG	TV	TR	WER	GAUS/FR	C (%)
3456	2.56	1.28	1.27	29.7	16301	12.49
6912	2.74	1.40	1.33	29.5	14627	11.21
13824	2.73	1.27	1.45	29.5	13433	10.29

TAB. 6.4 – QV contextuelle par pour tout l'espace : Taux d'erreur en fonction de la taille du *codebook*

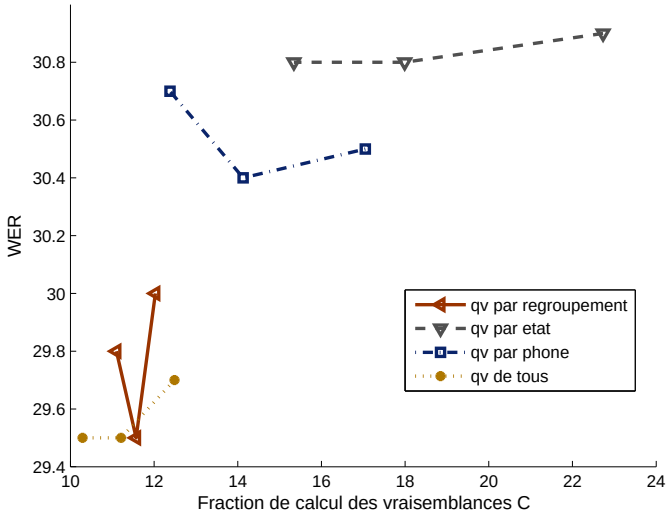


FIG. 6.2 – Quantification vectorielle et quantification vectorielle contextuelle par état, phone pour tous les états

Plusieurs constatations :

– Pour la QV : le WER n'est pas inversement proportionnel à la taille du *codebook*. Ceci, précédemment mentionné dans [73], est dû à la non robustesse de la méthode de classification. Par contre, le temps de calcul des vraisemblances est proportionnel à la taille du *codebook*, ce qui en concordance avec l'énoncé du paragraphe 6.2.3.

– Pour la QV contextuelle le taux d'erreur est inversement proportionnel à la taille du dictionnaire des références. d'où la robustesse de cette méthode.

– Pour la QV contextuelle, quelle que soit la taille du *codebook* total, la nombre de gaussiennes par observation est plus important pour la quantification par état que par phone ou de tout l'espace. En effet, dans le premier cas, deux états situés à deux emplacements typologiques différents ne disposent pas du même *codebook*. Donc les gaussiennes correspondantes ne peuvent pas avoir des représentants en communs et le phénomène de *cache* n'est par conséquent pas utilisé. En revanche, comme le nombre de représentants par état est rela-

tivement faible (32/64/128 < 3456/6912/13824), la quantification vectorielle est dans ce cas moins précise d'où le taux d'erreur plus élevé.

– la plus faible dégradation du taux d'erreur est de 0.7% absolu et correspond à C=10.29% de densités calculées. Ce résultat est légèrement meilleur que les performances de la QV par regroupement.

– quelques fois, le nombre de gaussiennes par observation n'est pas proportionnel au temps de calcul des vraisemblances. Ceci peut être dû au phénomène de cache qui est dans ce cas peu utilisé (lorsque beaucoup de représentants sont différents).

6.3 Sous-quantification vectorielle

Dans le paragraphe précédent, nous avons vu qu'avec la QV contextuelle l'augmentation de la taille du *codebook* peut permettre de réduire le taux d'erreur. Mais cette manipulation reste limitée et des erreurs de quantification persistent. Afin de réduire ces erreurs, nous nous proposons de procéder à une sous-quantification vectorielle.

La sous-quantification vectorielle (SQV) consiste à découper le vecteur d'observation en groupes de dimensions (flux) et à modéliser les flux comme des observations indépendantes. Ainsi, pour K flux, la probabilité d'émission d'une observation o par un état j est : $b_j(o) = \prod_{k=1}^{K} b_j^k(o^k)$.

Ce qui revient à un quantification vectorielle par flux.

Si N le nombre de distributions initiales en dimension D et M la taille du *codebook* alors le nombre d'opérations nécessaires au calcul de la vraisemblance est : $7MD + KM + 3KN$.

Ce qui signifie que le temps de calcul des vraisemblances est d'autant plus important que le nombre de flux l'est.

6.3.1 Sous-quantification des distributions

Dans [73], le vecteur de paramètres de dimension D est subdivisé en K parties : $d_1, d_2, ..d_K$ telles que $d_1 + d_2 + .. + d_K = D$. Pour chaque sous partie d_i :

– la moyenne et la variance de chaque gaussienne sont concaténées dans un même vecteur.

– les N vecteurs issus de la concaténation sont classifiés en M classes et le centroïde de chacune est calculé.

– un tableau d'appartenance des moyennes et variances des gaussiennes aux classes est construit.

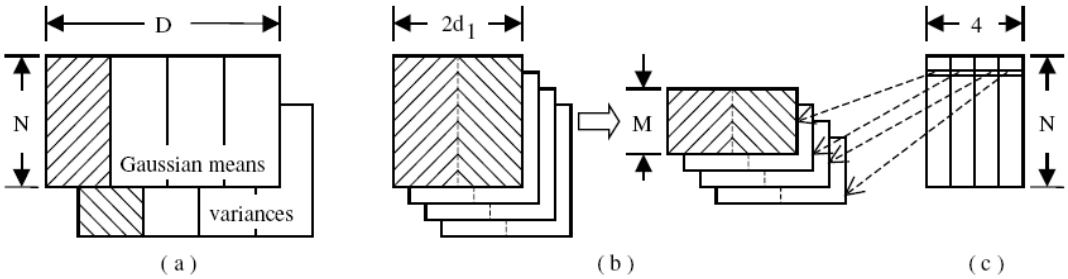


FIG. 6.3 – Sous-quantification vectorielle : a) *codebook* initial b) concaténation et regroupement par flux (en 4 classes) c) table de correspondance (D'après [73])

Dans notre cas, le vecteur de paramètres est de dimension 39 et composé de : l'énergie E + 12 coefficients cepstraux (mfcc) et leurs dérivées première et seconde. Trois subdivisions K de l'espace sont considérées :

- K = 1 : ce qui revient à une quantification vectorielle
- K = 3 : (E , 12mfcc) + δ (E , 12mfcc) + δ δ (E , 12mfcc)
- K = 4 : (E , δ E , δ δ E) + 12mfcc + δ 12mfcc + δ δ 12mfcc

Pour une taille du *codebook* de 3456, les résultats de la sous-quantification vectorielle sont :

K	TG	TV	TR	WER	$GAUS/FR$	$C(\%)$
1	2.61	1.30	1.31	30.0	15708	13.03
3	2.68	1.33	1.34	29.6	13949	10.69
4	2.81	1.35	1.35	29.7	15910	12.19

TAB. 6.5 – Taux d'erreur en fonction du nombre de flux

On constate que la sous-quantification est plus intéressante que la quantification ($K > 1$). Toutefois, l'augmentation absolue du taux d'erreur est de l'ordre de 1% ce qui assez important.

6.3.2 Sous-quantification contextuelle

Il s'agit de la même procédure de quantification vectorielle contextuelle appliquée à des ensembles de dimensions. Ainsi, les distributions des modèles indépendants du contexte (CI) sont découpées en sous-dimensions. Les distributions des modèles dépendants du contexte (CD), également découpées en sous-dimensions, sont associées aux distributions CI du même sous-espace.

Pour comparer la sous-quantification vectorielle SQV et la SQV contextuelle, les mêmes flux que le paragraphe 6.3.1 sont utilisés. La taille du *codebook* est de 3456.
Les résultats de la sous-quantification vectorielle par état, par phone ou de tout l'espace sont reportés sur les tableaux 6.6, 6.7, 6.8 et la figure 6.4.

K	TG	TV	TR	WER	GAUS/FR	C (%)
1	2.91	1.63	1.27	30.9	29665	22.73
3	2.76	1.56	1.26	30.0	21876	16.76
4	2.66	1.50	1.16	29.4	26238	20.01

TAB. 6.6 – SQV contextuelle par état : taux d'erreur en fonction du nombre de flux

K	TG	TV	TR	WER	GAUS/FR	C (%)
1	2.61	1.31	1.22	30.5	22235	17.04
3	2.75	1.43	1.31	30.0	17996	13.79
4	2.87	1.56	1.31	29.9	21512	16.48

TAB. 6.7 – SQV contextuelle par phone : taux d'erreur en fonction du nombre de flux

K	TG	TV	TR	WER	GAUS/FR	C (%)
1	2.56	1.28	1.27	29.7	16301	12.48
3	2.63	1.32	1.31	29.4	15148	11.60
4	2.78	1.46	1.31	29.0	17875	13.69

TAB. 6.8 – SQV contextuelle pour tous : taux d'erreur en fonction du nombre de flux

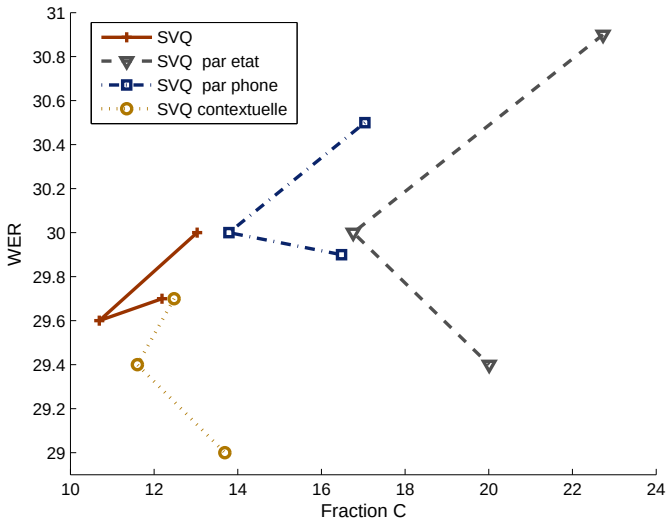


FIG. 6.4 – Résultats de la SQV et SQV contextuelle

Les conclusions que nous pouvons tirer de ces expériences de sous-quantification vectorielle contextuelle sont :

- globalement, les résultats obtenus après SQV sont meilleurs que ceux obtenues par QV. Ceci étant prévisible puisque l'erreur de quantification est plus faible.

- la sous quantification vectorielle contextuelle de tout l'espace est plus performante que celle par regroupement. Ce qui justifie l'importance de l'information contextuelle lors de la classification.

– la variabilité des paramètres (WER et C) et donc l'effet de la SQV est plus important pour une quantification par état que par phone ou encore de tout l'espace.

– dans pratiquement tous les cas, le nombre de flux est proportionnel à la fraction de calcul des vraisemblances. Ce qui en concordance avec ce qui est énoncé au début de ce paragraphe.

– le meilleur gain obtenu avec 4 flux (en appliquant la SQV contextuelle) correspond à 13.69% de densités calculées pour seulement une augmentation de +0.3% du wer. Ce qui est assez intéressant. Cette augmentation rentre dans l'intervalle de confiance [28.35 ;29.11] du système de base. Par conséquent on peut dire qu'il n' y a pas eu de dégradation.

6.4 Partitionnement hiérarchique contextuel

La qualité de la QV contextuelle est fortement liée à celle des modèles CI, et en particulier à leurs nombre de paramètres (taille du *codebook*) et de leur précision. Pour réduire la taille de ces derniers (donc le temps de calcul) tout en conservant leur précision, nous proposons de leur appliquer un partitionnement hiérarchique comme décrit dans le chapitre 4.

Rappelons le principe de cette méthode introduite dans le paragraphe 4.2.3 :

– partir d'un *codebook* de N distributions par état.
– les regouper au moyen d'un algorithme de regroupement hiérarchique basée sur la distance de Kullback Leibler Pondérée.
– les classer en M classes par coupure de l'arbre au niveau comprenant exactement M noeuds.
– les M noeuds forment les *centroïdes* du nouveau *codebook*.

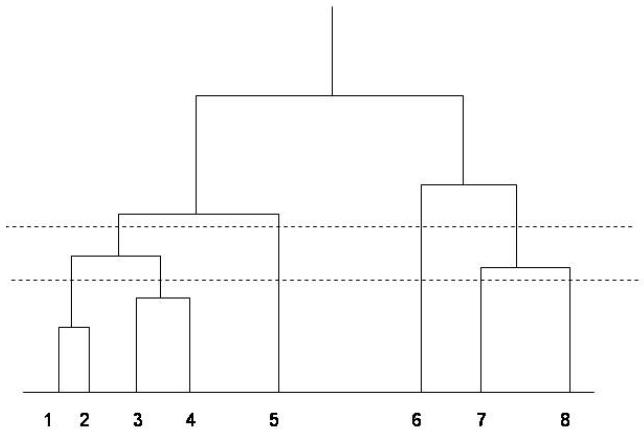


FIG. 6.5 – Regroupement hiérarchique des distributions

Afin d'évaluer l'apport du partitionnement hiérarchique à la SQV contextuelle, il est appliqué aux états des monophones à 32 gaussiennes par état càd une taille initiale de codebook de $M = 3456$. Il permet de réduire le nombre de gaussiennes par état de 32 à 5 ou 10 ou 16 ce qui correspond à un nombre global de *codebooks* de 540 ou 864 ou 1728 ou encore $M/6$, $M/4$ et $M/2$. D'ou la réduction du temps de calcul des vraisemblances.

Bien entendu, pour la SQV, le partitionnement hiérarchique est réalisé par flux.

L'expérimentation de la SQV contextuelle de tout l'espace après partitionnement hiérarchique permet d'obtenir les résultats suivants :

K	TG	TV	TR	WER	GAUS/FR	C (%)
1	2.85	1.52	1.32	29.6	25975	20.32
3	2.75	1.40	1.34	29.5	19576	15.41
4	3.00	1.61	1.39	28.8	22814	17.89

TAB. 6.9 – Taux d'erreur en fonction du nombre de flux pour un codebook de taille 540

K	TG	TV	TR	WER	GAUS/FR	C (%)
1	2.74	1.41	1.32	29.8	21143	17.07
3	2.70	1.35	1.34	29.8	17330	13.94
4	2.97	1.54	1.39	29.4	20150	15.85

TAB. 6.10 – Taux d'erreur en fonction du nombre de flux pour un codebook de taille 864

K	TG	TV	TR	WER	GAUS/FR	C (%)
1	2.72	1.37	1.34	29.7	18919	15.82
3	2.76	1.37	1.38	29.4	16074	13.64
4	2.91	1.52	1.38	29.4	18817	15.74

TAB. 6.11 – Taux d'erreur en fonction du nombre de flux pour un codebook de taille 1728

On peut constater que :

– encore une fois le WER est inversement proportionnel au nombre du flux.

– la variabilité de WER et C est d'autant plus importante que la taille du codebook est faible et que nombre de flux est important. En effet, pour les *codebook* réduits, la compensation de l'erreur de quantification introduite par la taille est plus efficace.

– Pour un codebook de taille 540 et quelque soit le nombre de flux on a toujours le facteur C plus grand que celui de 864 et 1728 et un WER inférieur. Comme on accorde plus d'importance au WER, on peut considérer ces modèles comme étant les meilleurs.

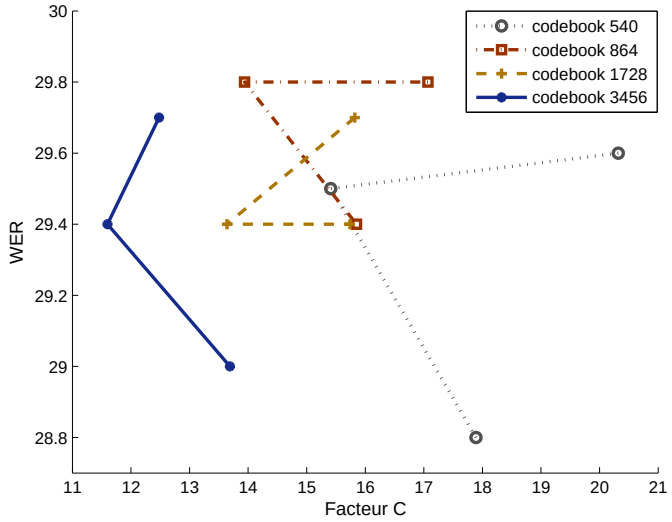


FIG. 6.6 – Résultats de la SQV contextuelle hiérarchique pour plusieurs tailles de *codebook*

– Un point intéressant est $(WER, C) = (28.8\%, 17.89\%)$. Pour pratiquement pas de perte en WER seulement $\sim$ 18 % de densités sont utilisées. C'est le meilleur compromis entre WER et C obtenu dans ce chapitre. Il correspond à un gain de temps de calcul des vraisemblance de 37% sans perte de performance.

6.5 Conclusion

Des méthodes de quantification vectorielle (QV) et de sous-quantification (SQV) contextuelle ont été développées, évaluées et comparées à l'existant. Nous pouvons en tirer les conclusions suivantes. D'abord la QV contextuelle est plus performante que QV par regroupement. Un *codebook* formé en tenant compte du contexte est meilleur que celui obtenu par un simple regroupement.
Ensuite, la sous-quantification vectorielle donne des résultats meilleurs que la quantification, ce qui est attendu puisque la distorsion engendrée est plus faible. En revanche, la sous-quantification vectorielle est d'autant plus performante que le nombre de flux est important. Encore faut-il faire un compromis entre nombre de flux et temps de calcul.
Enfin, comme la SQV contextuelle dépend des modèles indépendants du contexte, le partitionnement hiérarchique de ces derniers réduit d'avantage l'erreur introduite par la sous-quantification ainsi que le temps de calcul des vraisemblances, permettant de ce fait d'obtenir les meilleurs résultats de ce chapitre.

Conclusions et Perspectives

Le principal frein à l'utilisation des systèmes de reconnaissance automatique de la parole relève aujourd'hui principalement de leur relative lenteur d'exécution, lenteur rendant par exemple complexe leur portage dans des environnements pauvres en ressources calculatoires (PDA,...). L'étude réalisée au cours de cette thèse a ainsi porté sur la recherche de méthodes efficaces d'accélération de la reconnaissance et ceci en se concentrant sur la phase la plus coûteuse en temps de calcul, celle du calcul des vraisemblances.

Une telle étude nécessite naturellement d'avoir à sa disposition des systèmes de reconnaissance dits de référence sur lesquels l'influence des méthodes d'accélération peut être évaluée. Une première partie de notre travail a ainsi consisté à construire nos propres systèmes de référence en nous appuyant sur les techniques et méthodes existantes de l'état de l'art. Les performances de ces systèmes ont été par la suite améliorées tant au niveau acoustique que linguistique d'une part en augmentant la précision des modèles et d'autre part en prenant en compte des ensembles d'apprentissage plus larges.

Une seconde amélioration de ces systèmes a été mise en oeuvre par la définition d'une méthode *ad hoc* de segmentation audio nous permettant d'extraire les segments de parole et ceux de *parole+musique*. L'étude menée nous a alors permis de déterminer les caractéristiques les plus à même de distinguer ces deux classes : si les paramètres MFCC (*Mel Frequency Cepstral Coefficients*) sont plus adéquats pour la parole, les paramètres LFCC (*Linear Frequency Cepstral Coefficients*) le sont quant à eux relativement aux segments *parole+musique*. La combinaison des systèmes MFCC/LFCC permet alors de caractériser au mieux ces deux classes.

Concernant le coeur de notre travail, une première phase d'étude des méthodes d'accélération du décodage et en particulier de celles liées à la limitation du nombre de densités, nous a permis de regrouper celles-ci en trois catégories distinctes selon qu'elles se basent sur le partitionnement hiérarchique, sur la classification de type k-moyennes ou sur la sous-quantification vectorielle. Notre travail a consisté à expérimenter et à améliorer les méthodes issues de chacune de ces catégories.

Dans un premier temps, nous nous sommes intéressés au partitionnement hiérarchique des distributions gaussiennes de chaque mélange. La sélection de ces dernières est réalisée par état. Dans ce contexte, nous avons proposé une méthode de partitionnement hiérarchique basée sur l'utilisation

de la distance de *Kullback Leibler Pondérée* et la classification des distributions de chaque mélange à plusieurs niveaux pour une meilleure sélection. L'avantage de l'utilisation de la métrique proposée est lié au fait que cette dernière est basée sur la similarité entre les distributions, d'où l'optimalité de la classification. La sélection des gaussiennes à plusieurs niveaux permet au mieux d'éviter les erreurs de classification. En effet, les résultats d'une telle sélection lorsqu'elle est appliquée à des modèles acoustiques indépendants du contexte montrent une réduction de 88% du nombre de densités calculées en maintenant un égal niveau de performance relativement au taux de mots erronés (+0.1% absolu).

Les modèles acoustiques dépendants du contexte qui sont beaucoup plus précis que ceux indépendants du contexte, renferment un nombre relativement élevé d'états modélisés chacun par peu de distributions gaussiennes. Ainsi, dans une seconde étape, nous avons choisi de sélectionner les gaussiennes issues de ces modèles en opérant une classification de toutes les distributions de tous les états. Les méthodes existantes sont souvent basées sur un regroupement de type k-moyennes et ne prennent pas en compte le contexte lors du processus de classification. Cette perte d'information peut engendrer une limitation de la représentativité de certaines classes. Notre contribution se situe alors au niveau de la sélection contextuelle des gaussiennes ; sélection que nous avons combinée avec le partitionnement hiérarchique des distributions dont nous avons montré qu'il permet d'améliorer les modèles indépendants du contexte. Les approches proposées permettent de réduire le nombre de densités considérées de la même manière que les méthodes existantes tout en maintenant un niveau de performance plus élevé (augmentation du taux de mots erronés de seulement +0.3% absolu).

La restitution de l'information contextuelle a également été explorée en sous-quantification vectorielle ; sous-quantification souvent basée sur le regroupement des distributions afin de réduire leur nombre et par conséquent le temps de calcul des vraisemblances. Les expérimentation de cette proposition ont montré un gain important en termes de réduction du nombre de densités calculées puisque cette réduction se mesure autour de 87% (pour une augmentation du taux de mots erronés de seulement +0.1% absolu). Une réduction du temps de calcul est d'ailleurs mesurée aussi bien comparativement au système de base que relativement au même système bénéficiant d'une réduction par sous-quantitification vectorielle *standard*.

Cette thèse présente plusieurs contributions permettant de réduire le temps de calcul des vraisemblances et ceci relativement à chacune des trois catégories de méthode utilisées par la communauté. Chacune des méthodes améliorées définies au cours de cette étude a été évaluée indépendamment

des autres. Une perspective à court terme de notre travail relève donc de la mise en oeuvre d'une combinaison de ces méthodes.

Les dernières expérimentations montrent un temps de calcul des vraisemblances similaire sinon inférieur à celui de la recherche. Il devient alors nécessaire de s'intéresser à optimiser ce temps de recherche. Cette optimisation peut alors être obtenue en utilisant des techniques de prédiction du modèle de langage.

Enfin, étant donné que le temps de reconnaissance d'un système est étroitement lié à ses performances (en termes de taux d'erreurs), une amélioration des ressources acoustiques et linguistiques mises en jeu permet également de réduire la durée du décodage. Cette amélioration peut être réalisée au moyen de modèles acoustiques plus précis (quinphones, modèles par canal/locuteur, ..) ou linguistiques plus grand (de types quadrigrammes par exemple). La complexité de calcul engendrée par l'augmentation de la taille des ressources peut être diminuée par un décodage en plusieurs passes.

Annexe A : Distances utilisées

Soient $X = (x_1, .., x_n)$ n vecteurs de dimension d et $Y = (y_1, .., y_m)$ m vecteurs de dimension d. On suppose que les séquences X et Y sont modélisées par des processus gaussiens multi- dimensionnels P_X et P_Y.

$$P_X(x) = \frac{1}{(2\pi)^{d/2} |\sum_X|^{1/2}} \exp(-1/2(x-\mu_X)^T \sum_X^{-1}(x-\mu_X))$$

Distance de Kullback Leibler Pondérée : La distance de Kullback-Leibler entre X et Y s'écrit :

$$KL(X,Y) = \int p_X(X) \log \frac{P_X(X)}{P_Y(X)} dX \tag{6.2}$$

Comme cette mesure n'est pas symétrique, la distance de Kullback-Leibler symmetrisée est définie par :

$$KL2(X,Y) = KL(X,Y) + KL(Y,X) \tag{6.3}$$

La démonstration de la formule de la distance de KL pour les distributions gaussiennes est reportée dans [15]. Elle montre qu'elle peut s'écrire :

$$KL2(X,Y) = \frac{1}{2} tr(\Sigma_X \Sigma_Y^{-1} + \Sigma_Y \Sigma_X^{-1}) + \frac{1}{2}(\mu_X - \mu_Y)^T (\Sigma_X^{-1} + \Sigma_Y^{-1})(\mu_X - \mu_Y) - d \tag{6.4}$$

La distance de Kullback-Leibler Pondérée (KLP) que nous proposons, n'est autre que la distance de Kullback-Leibler symétrisée, lorsque les densités de probabilité P_X et P_Y sont pondérées par leurs données d'apprentissage respectives n et m. Dans ce cas il s'agit de nP_X et mP_Y.
En adoptant la même démarche que dans [15], on trouve :

$$KLP(X,Y) = \frac{1}{2} tr(n\Sigma_X \Sigma_Y^{-1} + m\Sigma_Y \Sigma_X^{-1}) + \frac{1}{2}(\mu_X - \mu_Y)^T (n\Sigma_X^{-1} + m\Sigma_Y^{-1})(\mu_X - \mu_Y) - (n+m)d \tag{6.5}$$

Distance de Perte de vraisemblance : Dans le domaine logarithmique, la vraisemblance ll_1 de X, en dimension 1, s'écrit :

$$ll_1 = \sum_{xinX} -\log\sigma_x - \log\sqrt{2\pi} - \frac{1}{2}\frac{(x-\mu_x)^2}{\sigma_x^2} = -n\log\sigma_x - nlog\sqrt{2\pi} - \frac{n}{2} \tag{6.6}$$

Si on désigne par $ll2$ la vraisemblance de Y et ll celle de $Z = (X + Y)$, alors la perte en vraisemblance engendrée par le remplacement de X et Y par Z peut s'exprimer par :

$$ll_1 + ll_2 - ll = \log\frac{\sigma^{(n+m)/2}}{\sigma_x^{n/2}\sigma_y^{m/2}} \tag{6.7}$$

Publications

L. Zouari, K. McTait et G. Chollet. "Sélection de Gaussiennes pour une Transcription Rapide des Émissions Radiophoniques de la Campagne ESTER". Conférence Internationale "Sciences Electroniques, Technologies de l'Information et des Télécommunications" SETIT. Sousse. Mars 2005

F. Brugger, L. Zouari, H. Bredin, A. Amehraye, G. Chollet, D. Pastor et Y. Ni. "Reconnaissance Audiovisuelle de la Parole par VMike". 26 ème Journées d'Etude sur la Parole JEP. Dinard. Juin 2006.

L. Zouari et G. Chollet, "Efficient Gaussian Mixture for Speech Recognition". International Conference in Pattern Recognition ICPR. Hongkong. Août 2006.

L. Zouari et G. Chollet. "Sélection de Paramètres pour la Discrimination Parole/non Parole d'Emissions Radio Diffusées". Ateliers de travail sur le Traitement et l'Analyse de l'Information : Méthodes et Applications, TAIMA 2007.

Rémi Landais, Hervé Bredin, L. Zouari et G. Chollet. " Vérification Audiovisuelle de l'identité". Ateliers de travail sur le Traitement et l'Analyse de l'Information : Méthodes et Applications, TAIMA 2007.

Efficient Gaussian Mixture for Speech Recognition

Leila Zouari and Gérard Chollet
GET-ENST/CNRS-LTCI
6 rue Barrault 75634 Paris cedex 13, France

zouari, chollet@tsi.enst.fr

Abstract

This article presents a clustering algorithm to determine the optimal number of components in a Gaussian mixture. The principle is to start from an important number of mixture components then group the multivariate normal distributions into clusters using the divergence, a weighted symmetric, distortion measure based on the Kullback-Leibler distance. The optimal cut in the tree, i.e. the clustering, satisfies criteria based on either the minimum amount of available training data or dissimilarities between clusters. The performance of this algorithm is compared favorably against a reference system and a likelihood loss based clustering system. The tree cutting criteria are also discussed. About an hour of Ester, a French broadcast News database is used for the recognition experiments. Performance are significantly improved and the word error rate decreases by about 4.8%, where the confidence interval is 1%.

1. Introduction

Nowadays, state of the art Hidden Markov Models based large vocabulary speech recognition systems make use of a important number of Gaussian distributions to improve their acoustic modeling accuracy. One of the disadvantages of this practice is the increase of the complexity of the system making it unsuitable for practical, embedded or even real time applications. Several criteria are generally used to stop growing the mixture, i.e. the number of Gaussian distributions. Mainly a tradeoff is to be found between the model precision and the ability to accurately estimate the model parameters. In the literature three classes of approaches are used to stop growing a mixture:

- when the amount of training data is insufficient (a threshold is placed on the number of frames used to estimate the mixture components),
- if no significant likelihood increase is observed,
- and when the Bayesian Information Criterion (BIC) gain becomes negative or below a threshold. This criterion controls the model complexity by penalizing the likelihood with the number of parameters.

An alternative proposed by Messina [1] is to grow a mixture only when it's distance to a frame is important. So, distances between a frame and Gaussians are computed and the minimum is selected. If this minimum distance is less than a threshold the component is updated with this frame otherwise a new mixture component is created. To decrease the number of mixture components in a phonetically tied mixture system, Digalakis [4] classifies the Gaussian distributions and re-estimates the obtained clusters. The clustering metric is based on the increase of entropy due to merging distributions. This way, the number of Gaussians is reduced to less than 40% with a little degradation of performance. Besides, this system is more accurate (+0.8%) than the reference one using the same number of Gaussians.
In the present work it is proposed to determine the optimal number of components in a Gaussian mixture models using a growing-clustering process, following the same principle of clustering as Digalakis, and we introduce the weighted cross entropy metric for better distributions classification. The idea driving this procedure is to explore a large set of components, then the set dimension is reduced by merging close elements. So, for each Gaussian mixture, distributions are grouped into a binary tree structure and every cut in the tree defines a possible clustering. To determine the optimal cut in the tree, two criteria are experimented: a data driven one and a dissimilarity based other. For each criterion, the weighted Kullback-Leibler divergence performance is compared to the initial system and also to a loss likelihood based clustering system.
The remainder of this paper is organized as follows: section 2 outlines the classification process, presents the proposed weighted Kullback-Leibler metric and details the tree cutting criteria, section 3 reports on tests protocols and results, the conclusions and prospective work are described in section 5.

2. Gaussian distributions classification

2.1. Clustering process

In order to build Gaussian trees, hierarchical bottom-up classification algorithm is applied to each mixture. It performs in many steps:

- Compute distances between all pairs of distribution.
- Merge the closest two distributions as follows:
  Let $g_1(n_1, \mu_1, \sigma_1)$ and $g_2(n_2, \mu_2, \sigma_2)$ two Gaussians to which n_1 and n_2 frames have been associated during the training. If g_1 and g_2 are merged into $g_3(n_3, \mu_3, \sigma_3)$ then :

  $$n_3 = n_1 + n_2$$
  $$\mu_3 = \frac{n_1}{n_1+n_2}\mu_1 + \frac{n_2}{n_1+n_2}\mu_2$$
  $$\sigma_3 = \frac{n_1}{n_1+n_2}\sigma_1 + \frac{n_2}{n_1+n_2}\sigma_2 + \frac{n_1 n_2}{(n_1+n_2)^2}(\mu_1-\mu_2)(\mu_1-\mu_2)^T$$

  g_3 replaces (g_1, g_2) in the set whose size is reduced by one.
- If the number of Gaussians is greater than 1 go to step1.

2.2. Metrics

Two distances are used: likelihood loss based distance and weighted relative entropy based metric.
- Loss likelihood based metric: If g1 and g2 are merged into g3 then the likelihood loss (pv) is the difference between the likelihoods of g1 and g2 and the likelihood of g3 on the training data:

$$PV(g_1, g_2, g_3) = log\frac{\|\sigma_3^{(n_1+n_2)/2}\|}{\|\sigma_1^{n_1/2}\|\|\sigma_2^{n_2/2}\|}$$

This metric is somewhat similar to the loss of entropy based distance used by Digalakis [4]. It was successfully used in model adaptation [2].
- The weighted symmetric Kullback Leibler divergence (Klp): It is expressed as the distance between two probability density functions weighted by the amount of training data.

$$KLp(g_1; g_2) = \frac{1}{2}tr(n_1\frac{\sigma_1}{\sigma_2} + n_2\frac{\sigma_2}{\sigma_1}) + \frac{1}{2}(\mu_1-\mu_2)^T(\frac{n_1}{\sigma_1}+\frac{n_2}{\sigma_2})(\mu_1-\mu_2) - (n_1+n_2)d$$

d is the dimension of the parameters vectors.
The use of the information provided by the amount of training data is advantageous if training and testing data have the same proportions otherwise it can be harmful.

2.3. Tree cutting

From the root of the tree to the leaves, different cuts can be defined allowing many classifications. Three cutting ways are proposed:

- Fixed: We consider a constant number of classes for each mixture. Beginning from the leaves, traverse each level till the number of nodes at the corresponding stage reaches the predefined value of classes and cut.
- Weight based: The number of classes depends on the amount of the available data to estimate the distributions of each class. Starting from the root, the tree is processed and we stop at node for which the children weight is less than a predefined threshold.
- Distance based: tree cutting is performed when the distance between two levels reach a maximum value. Considering only the maximum can lead to a very little (or large) number of clusters, besides many important distances can be close to the maximum value. for all these reasons, several cuttings per tree have been considered. Each cutting is operated in a particular level of the tree.

For weight and distance criteria, as the number of Gaussians per state can be different, a mean value is computed.
The resulting mixtures are re-estimated by means of Baum Welch algorithm.

3. Experiments and results

3.1. Resources

All the experiments are conducted using parameter vectors with 12 MFCC coefficients, energy, and their first and second derivatives. The 40 acoustic models are context independent with 3 states per model. For the training task, about 82 manually transcribed hours of the Ester train database [3] are used. The dictionary contains 118000 words (with 65000 distinct words). The language model is formed by 4 millions of bigrams and trigrams. Tests are conducted using an hour of Broadcast News extracted from the Ester test data set.
The initial system contains 256 Gaussians per state. For each mixture, the 256 Gaussians are classified by a bottom up hierarchical algorithm. Depending on the experiments, likelihood loss or weighted cross entropy based metric is used for clustering. Then classes are obtained by cutting the binary tree following a criterion: fixed, weight based or distance based number of clusters.
In order to compare the different systems, several reference systems 32, 64, 80, 128, 180, 256, 220 and 512 Gaussians per mixture are produced and evaluated.

3.2. Fixed classes

The number of classes is fixed and is the same for the reference (Ref), the loss likelihood (pv) and the weighted Kullback-Leibler (klp) based systems. After clustering, the obtained pv and klp models are trained. We find that au maximum two iterations are needed to estimate these models parameters. Results within a confidence interval of 1% are as follows:

Table 1. wer for Ref, PV, and KLp systems

Gaussians nbr	Ref (%)	PV (%)	KLp (%)
32	42.6	40.6	39.5
64	40.4	38.0	37.5
80	38.3	37.4	36.9
128	37.3	36.2	36.2
180	36.4	36.1	36.2
220	36.3	35.8	35.5
256	36.3	-	-
512	35.5	-	-

Table1 and figure1 show that both pv and klp systems outperform the reference one, with a little advantage for the latter. Especially, for the klp models with 32 and 64 Gaussians per state, the word error rate (wer) decreases by about 3% compared to the reference system. With a large number of clusters differences are less interesting.
Performance of the klp system using 220 Gaussians per state are similar to the 512 reference one.

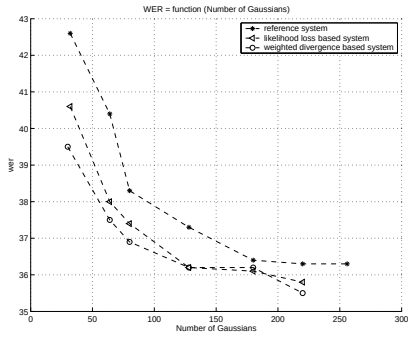


Figure 1. wer vs number of Gaussians, for ref, pv and klp systems

3.3. Weight based classes

By using this criterion, the number of clusters in the different mixtures is variable and depends on the acoustic variability of each state. Besides, this way we ensure that each cluster has sufficient amount of training data to estimate it. So, in each level of the tree, when a node reaches the global minimum of this level, we cut at his parent level. Results are as reported in table 2 and figure 2.

Table 2. pv and klp weight based cutting

Metric	Gaussians	wer (%)
KLp	28	40.0
	53	36.6
	150	35.9
	195	36.0
PV	53	39.5
	101	36.8
	156	36.5

We notice that using the weight criterion, the klp system outperforms both pv and the reference system. Especially, with only a mean of 53 Gaussians per state, it's performance is close to that of the reference system with 256 Gaussians. Besides, the wer decreases by about 4.8% compared to the initial system using the same number of Gaussians.
For klp system with 28 Gaussians per state, the wer is also better than the initial 64 Gaussians one. Finally, klp models with 150 Gaussians are quite performing as the 512 reference system.

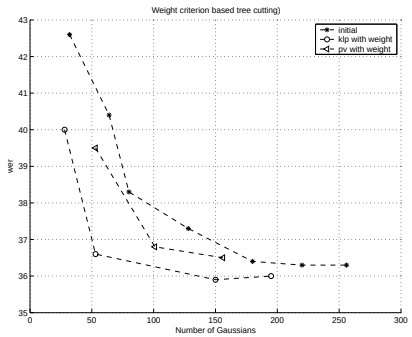


Figure 2. Weight based tree cutting

3.4. Distance based classes

This criterion prevents clustering too distant Gaussians:

- if distributions are too different in the case of klp distance
- or merging them leads to a big likelihood loss if pv based metric is employed

We consider several levels of the tree and cut when the distance between two clusterings is the maximum in this level. The obtained results are reported in table 3.

Table 3. pv and klp distance based cutting

Metric	Gaussians nbr	wer (%)
KLp	30	40.7
	59	37.7
	101	36.1
	196	35.9
PV	44	39.4
	94	36.7
	204	35.8

Once again, we see that pv and klp systems outperform the reference one, and that the klp divergence based system is the best. Applying klp or pv clustering process, we obtain globally the same performance as the reference system using only about 40% of the total number of Gaussians. These results are interesting but they remain less important than the previous experiments (53 Gaussians) in which this number is reduced to 20%.

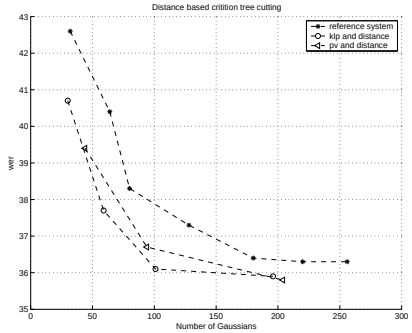


Figure 3. Distance based tree cutting

3.5. Weight versus distance

To compare distance and weight criteria we plot the correspondent curves using either klp or pv metric.

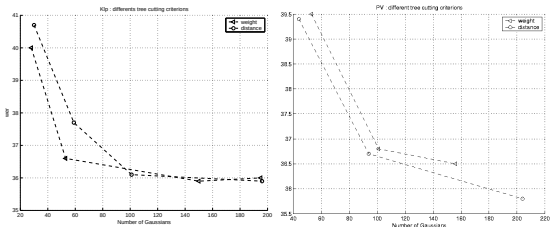


Figure 4. Weight and distance based tree cutting for respectively klp and pv systems

In the klp system, the weight criterion performs better than distance one, especially when the number of clusters is low. In the case of pv clustering, it is the opposite situation and the distance is better. These results can be interpreted as follows:
- when klp clustering metric is used, no particular attention is given to the amount of training data available for each cluster. Only resembling Gaussians are merged, ensuring that at each level clusters are as distant as possible. So in some levels many clusters could not have enough training data, and cutting at these levels is not interesting.
- In the case of pv based clustering, the loss of likelihood is minimum at each level. So the resulting clusters are as representative as possible of the training data. Knowing that no information about similarity of clusters to each others is taken into account, many resemblant clusters can be present in the same level. In this case the distance based cutting criterion can remove the redundant information.

4. Conclusion and discussion

An hierarchical Gaussians clustering algorithm for optimal mixture dimension determination based on a weighted Kullback Leibler distance (klp) is described. Experiments varying the tree cutting criterion show that in all the cases the proposed metric outperforms the loss likelihood based clustering (pv) system and the initial one. We also notice that the tree cutting criterion depends of the clustering distance. While the weight based tree cutting criterion is better for the klp system, the distance based cutting is more interesting for the pv system. In both cases, the good criterion of cut is that which brings information the distance does not take in consideration. As a perspective to this work, a linear discriminant analysis per state can be deduced from the Gaussian classification. This way, more separate and hence discriminant parameter vectors can be constructed and tested for better recognition.

References

[1] R. Messina and D. Jouvet. Sequential clustering algorithm for gaussian mixture initialisation. In *In proceedings ICASSP*, 2004.

[2] C. Mokbel. Online adaptation of hmms to real life conditions: A unified framework. In *IEEE Transaction on Speech and Audio Processing*, 2001.

[3] J. B. S. Galliano E. Geoffrois D. Mostefa, K. Choukri and G. Gravier. The ester phase ii campaign for the rich transcription of french broadcast news. In *In proceedings Eurospeech Interspeech*, Lisboa, 2005.

[4] H. M. V. Digalakis, P. Monaco. Genones : Generalized mixture tying in continuous hidden markov model- based specch recognizers. In *IEEE Transactions on Speech and audio Processing p 294-300*, 1996.

Sélection de Paramètres pour la Discrimination Parole/non Parole d'Emissions Radio Diffusées

Leila Zouari et Gérard Chollet

CNRS-LTCI
École Nat. Sup. des Télécommunications de Paris,
Département Traitement du Signal et des Images,
46 rue Barrault, 75314 Paris, France.
Tél : int+ 33 1 45 81 71 44, Fax : int+ 33 1 45 81 37 94
zouari,chollet@enst.fr

Résumé En reconnaissance automatique de la parole grand vocabulaire d'émissions radio diffusées, une étape crutiale de segmentation en parole/non parole est nécessaire. Or souvent, les segments de parole sont mélangés avec d'autres sons tels que la musique. Par conséquent, dans cet article, on se propose de trouver une paramétrisation adéquate aussi bien pour la parole que pour un mélange de parole+musique afin de bien les discriminer. On s'intéressera en particulier aux paramètres MFCC (Mel Frequency Cepstral Coefficients), LFCC (Linear Frequency Cepstral Coefficients) et à leurs combinaisons, et on évaluera trois types de combinaison, par fusion des paramètres, par fusion des scores et par fusion des décisions.
Nos expériences montrent que les coefficients MFCC sont plus performants en détection de la parole, que les paramètres LFCC le sont en reconnaissance de la musique+parole et que leur combinaison constitue un bon compromis lorsque des signaux de parole et de parole + musique sont tous les deux présents.
Mots clés Segmentation audio, Parole, Musique, MFCC, LFCC.

1 Introduction

Souvent dans les émissions radio, plusieurs types de signaux sont présents : parole, musique, parole + musique, jingles, silence, etc. Suivant le type d'application, différentes segmentations sont envisageables :

- segmentation musique/non-musique pour le traitements de la musique (classification par genre ou par instrument, etc),
- séparation parole/fond musical des segments de parole + musique pour le mixage audio ou la séparation des sources, etc,
- segmentation parole/non-parole pour la transcription orthographique et éventuellement la recherche d'information.

Ce travail s'insère dans le cadre du développement d'un système de transcription automatique des émissions radio. De ce fait, on se propose dans un premier temps de réaliser une segmentation du flux audio dans le but d'extraire les parties contenant de la parole ou de la parole mélangée avec la musique. La construction d'un tel système suscite le choix d'une

paramétrisation adéquate aussi bien pour la parole que pour la parole et la musique.
Les coefficients MFCC ont prouvé leur efficacité en traitement automatique de la parole. Leur succès provient, entre autres, de l'utilisation de l'échelle MEL qui favorise les basses fréquences. Dernièrement, Logan [5], a montré que les MFCC peuvent aussi représenter la musique sans pour autant se prononcer sur leur optimalité. L'échelle MEL n'étant pas optimale pour la musique puisqu'il peut y avoir autant d'information en basses fréquences qu'en hautes fréquences. Par conséquent, dans cet article, on s' est proposé de trouver la meilleure paramétrisation de la musique mélangée avec la parole. Les paramètres retenus seront combinés avec les coefficients MFCC pour mieux discriminer la parole et les mélanges de parole et de musique. Trois techniques de fusion ont été évaluées : fusion des paramètres, fusion des scores et fusion des décisions.
Après avoir présenté notre système de segmentation et décrit notre corpus de données, nous détaillerons les expériences de discrimination de la parole et de la parole+musique et nous en tirerons les conclusions.

2 Système de segmentation

Classiquement, la segmentation du flux audio est réalisée en deux temps. Dans un premier temps, on extrait du signal les paramètres jugés pertinents. Ces derniers doivent caractériser au mieux les classes à discriminer. Puis un processus de segmentation/classification permet d'affecter chaque partie du signal à une classe.

2.1 Paramétrisation

Notre paramétrisation est basée sur les coefficients cepstraux. Le signal audio est extrait de la séquence vidéo, échantionné à 16khz, puis les coefficients MFCC (MEL Frequency Cepstral Coefficients) et LFCC (Linear Frequency Cepstral Coefficients) sont calculés à partir d'un banc de 24 filtres. Ces filtres, de type MEL (échelle logarithmique) ou linéaires, sont appliqués toutes les 10 ms sur une fenêtre glissante de durée 20ms. Aux coefficients statiques (12 MFCC ou LFCC + énergie) nous rajoutons les dérivées première et seconde, ce qui permet d'obtenir des vecteurs de paramètres de dimension 39.

2.2 Classification

Les systèmes de segmentation de l'état de l'art font appel à des techniques tels que les modèles de Markov cachés (HMM) [4], les Modèles de Mélange Gaussien (GMM) [1,3], les k-plus proches voisins (KNN) [6], les réseaux de neurones, et plus récemment les Machines à Vectors de Supports (SVM) [2]. Comme ces travaux sont menés dans un objectif de reconnaissance de la parole, une approche de type GMM est adoptée. Ainsi, chaque classe est modélisée par un Modèle de Mélange Gaussien (GMM).
La classification, réalisée toutes les 10ms, est basée sur la règle suivante : soient N classes $C_1, C_2, .., C_N$ et le vecteur de test O. Le vecteur O est assigné à la classe la plus vraisemblable càd celle pour laquelle la vraisemblance $P(O/C_i)$ est maximale.
Quatre classes ont été utilisées : parole, musique, parole + musique et autres. Un nombre de composants de 256 gaussiennes par GMM est choisi empiriquement.

3 Expériences et résultats

3.1 Base de données

Nous avons utilisé une base de données de la variété télé "le grand échiquier". L'enregistrement dure trois heures et demi. Il contient de la parole, de la musique, la combinaison des deux (chants, jingles, ..) et des sons divers tels que les rires, les applaudissements, les effets spéciaux. Après avoir étiqueté manuellement cette base, nous l'avons découpée comme suit : 2h30 pour l'apprentissage et le reste pour l'évaluation. Le contenu de ces deux parties est explicité sur la figures Fig.1.

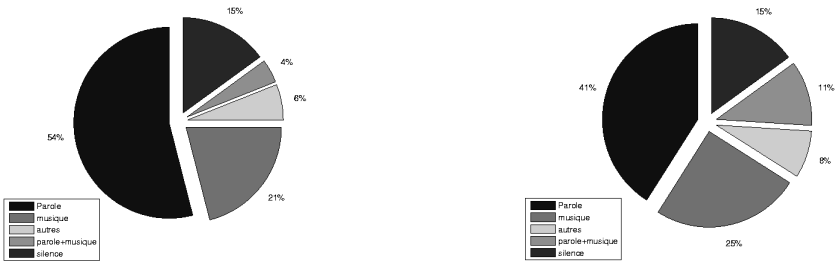


Fig. 1. Contenu des bases d'apprentissage (à gauche) et de test (à droite)

3.2 Mesure de performance

L'évaluation est réalisée trame par trame (toutes les 10ms). Les performances sont mesurées sur la base du rappel (R) et de la précision (P).

$$R = \sum_c T(c|c) / \sum_c T(c) \tag{1}$$

$$P = \sum_c T(c|c) / \sum_c T(c) + T(c|nc) \tag{2}$$

où $T(c|nc)$ est le temps où l'évènement c a été détecté à tort, $T(nc|c)$ le temps où c n'a pas été détecté à tort, $T(c)$ le temps où c est présent et $T(nc)$ le temps où c n'est pas présent.

Les évènements sont la parole, la musique et la parole+musique. Les performances des systèmes seront comparées sur la base de la F-mesure définie par :

$$F - mesure = 2 * R * P/(R + P) \tag{3}$$

Les temps seront mesurés en secondes. Dans les expériences qui suivent, on notera les valeurs de $F - mesure$ pour différentes $marges$, où $marge$ correspond à l'écart toléré (des limites) entre la segmentation automatique et la segmentation manuelle (en millisecondes).

3.3 Systèmes de MFCC/LFCC

Comme nous l'avons précisé dans l'introduction, l'objectif de ce travail et de trouver une paramétrisation adéquate et pour la parole et pour la parole mélangée avec la musique. Pour ce faire, nous avons commencé par développer deux systèmes de segmentation parole/ musique/ parole+musique/ autres en utilisant les paramétrisations MFCC et LFCC.

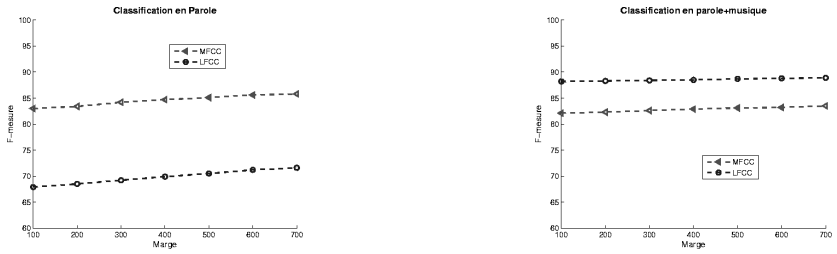


Fig. 2. Performances des paramétrisations MFCC et LFCC

Sur la figure Fig.2 sont reportées les performances du système en classification en parole et en parole + musique. On remarque un écart important entre la classification de la parole en utilisant les MFCC et celle en utilisant les LFCC. Pour la classification en parole + musique, les LFCC sont plus performants que les MFCC. Néanmoins, la différence n'est pas très importante.
Bien qu'intéressantes, ces constatations ne nous permettent pas de trancher entre MFCC et LFCC car dans les données de test, on ne connaît pas a priori les proportions de segments de parole et de parole + musique.

3.4 Combinaison des paramètres

Il s'agit d'une simple concaténation des paramètres MFCC et LFCC.

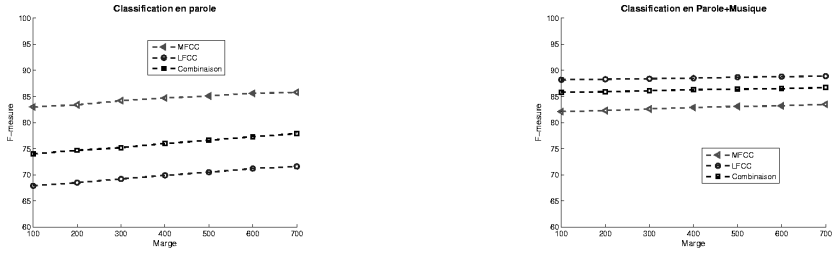


Fig. 3. Classification parole/musique/parole+musique/autres

La figure Fig.3 montre que les performances du système issu de la combinaison des paramètres sont entre celles du système MFCC et celles du système LFCC.

3.5 Combinaison des scores

Disposant des systèmes MFCC pour la parole et LFCC pour la parole et la musique, la combinaison des scores est réalisée en leur affectant des poids différents afin de privilégier l'une ou l'autre des paramétrisations. A chaque instant t, si $P(O_{MFCC};t)$ et $P(O_{LFCC};t)$ sont les vraisemblances d'une observation O calculées avec les systèmes MFCC et LFCC, alors son score de fusion peut s'exprimer par : $P(O_{fusion};t) = \lambda P(O_{MFCC};t) \times (1-\lambda) P(O_{LFCC};t)$

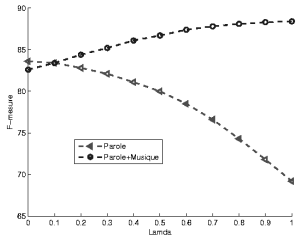


Fig. 4. Performances de la classification de la parole et de la parole + la musique en fonction de λ

Nous avons fait varier le poids λ entre 0 et 1. D'après la figure Fig.4, on peut constater que les performances de détection de la parole se dégradent lorsque λ augmente et le contraire pour la parole + la musique. Mais, vue la forte dégradation des performances de détection de la parole, on est tenté par l'utilisation de valeurs faibles de λ.

3.6 Combinaison des décisions

Il s'agit de fusionner les meilleurs systèmes de segmentation parole/non parole (P/NP) et parole+musique/non parole+musique (MP/NMP) pour en déduire une segmentation en 4 classes : parole (P), parole+musique (MP), musique (M) et autres (A). Les règles de fusion sont explicitées dans le tableau Tab.1.

Tab. 1. Règles de combinaison des décisions

Système MFCC	Fusion	Système LFCC
P → P	P	NMP ← P
MP → P	MP	MP ← MP
A → NP	A	NMP ← A
M → NP	M	MP ← M

On remarque (Fig.5) que la combinaison des systèmes apporte une amélioration par rapport aux coefficients LFCC pour la détection de la parole sans dépasser les performances avec les MFCC ce qui confirme leur supériorité en représentation de la parole. Pour la classification de la parole + la musique les performances de la fusion dépassent celles des

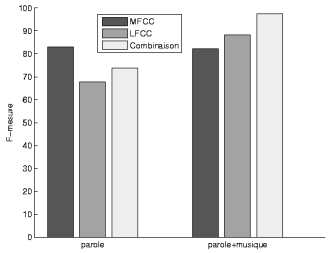


Fig. 5. Histogrammes de la fusion des décisions

deux systèmes de base. Par conséquent, la fusion des décisions pourrait constituer le meilleur compromis dans le cas où la nature des données de test est inconnue.

4 Conclusion

Dans le cadre d'une transcription automatique des émissions radio diffusées, une segmentation audio est réalisée. L'objectif d'une telle segmentation est de caractériser au mieux les segments de parole et de parole+musique qui seront par la suite transcrits. Les paramètres testés sont les coefficients MFCC, LFCC et leurs combinaisons. Les résulats de nos expériences montrent que les coefficients MFCC sont plus adéquats pour la discrimination de la parole, les coefficients LFCC le sont pour la musique+parole et leur combinaison l'est pour un flux audio contenant à la fois la parole et la parole mélangée avec de la musique.

5 Remerciements

Ce travail a été réalisé dans le cadre du pôle de compétitivité Cap-Digital et du réseau d'excellence KSpace. Nous tenons à remercier Youssef Boukhabrine pour sa contribution aux expériences.

Références

1. G. Linares C. Fredouille, D. Matrouf and P. Nocera. Segmentation en Macro-classes Acoustiques d'Emissions Radiophoniques dans le cadre d'ESTER. In *Journées d'Etude sur la Parole JEP*, 2004.
2. G.D. Guo and S.Z. Li. Content-based Audio Classification and Retrieval by Support Vector Machines. In *IEEE transactions on Neural Network*, Janvier 2003.
3. JL. Rouas J. Pinquier and R. A. Obrecht. A Fusion Study in Speech/Music Classification. In *proceedings ICASSP*, 2003.
4. O. Mella J. Razik, D. Fohr and P. Valles. Segmentation Parole/Musique pour la Transcription Rapide. In *Journées d'Etude sur la Parole JEP*, 2004.
5. B. Logan. Mel Frequency Cepstral Coefficients for Music Modelling. In *proceedings of the International Symposium on Music Information Retrieval*, 2000.
6. E. Scheirer and M. Slaney. Construction and Evaluation of a Robust Mainframe Speech/Music Discriminator. In *IEEE International Conference on Audio Speech and Signal Processing*, number 1331-1334, Munich, Germany, 1997.

Bibliographie

[1] A. Aiyer, MJF. Gales, and MA Picheny. Rapid Likelihood Calculation of Subspace Clustered Gaussian Components. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, volume 3, pages 1519–1522, Istanboul, 2000.

[2] J. Ajmera, IA. Mccowan, and H. Bourlard. Robust HMM-based Speech/Music Segmentation. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, pages 297–300, Orlando, Mai 2002.

[3] F. Alleva, X. Huang, and MY. Hwang. Improvements on the Pronunciation Prefix Tree Search Organization. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, volume 1, pages 134–136, 1996.

[4] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A Compact Model for Speaker-Adaptive Training. In *proceedings ICSLP*, pages 1137–1140, 1996.

[5] E. Bocchieri. Vector Quantization for the Efficient Computation of Continuous Density Likelihoods. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, volume 2, pages 692–695, Minneapolis, Avril 1993.

[6] E. Bocchieri and B. Mak. Subspace Distribution Clustering Hidden Markov Model. *IEEE transactions on Speech and Audio Processing*, pages 264–275, Mars 2001.

[7] MJ. Carey, ES. Parris, and H. Lloyd-Thomas. A Comparison of Features for Speech Music Discrimination. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, pages 149–152, Phoenix, 1999.

[8] A. Chan, M. Ravishankar, and AI. Rudnicky. On Improvements to CI based GMM Selection. In *European Conference on Speech Communication and Technology*, pages 565–568, Lisbon, Septembre 2005.

[9] A. Chan and J. Sherwani. Sphinx 3.4 Development Progress. www-2.cs.cmu.edu, 2004.

[10] A. Chan, J. Sherwani, R. Mosur, and A. Rudnicky. Four Layer Categorization Scheme of Fast GMM Computation Techniques in Large Vocabulary Continuous Speech Recognition Systems. In *proceedings ICSLP*, Jesu Island - Korea, 2004.

[11] G. Chollet. Evaluation of ASR Systems Algorithms and Databases. In Antonio RUBIO-AYUSO, editor, *NATO-ASI : Speech Recognition and Coding. New Advances and Trends*, 1995.

[12] GD. Cook, JD. Christie, PR. Clarkson, and M.M. Hochberg. Real-Time Recognition of Broadcast News. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, pages 141–144, 1996.

[13] DARPA. DARPA Broadcast News Transcription and Understanding Workshop. http ://www.nist.gov/speech/publications/darpa98, 1998.

[14] DARPA. DARPA, Broadcast News Workshop. http ://www.nist.gov/speech/publilcations/darpa99, 1999.

[15] P. Delacourt. *La Segmentation and le Regroupement par Locuteurs pour l'Indexation de Documents Audio*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Paris, 2000.

[16] L. Deng and X. Huang. Challenges in Adopting Speech Recognition. In *Communications of the ACM*, volume 47, pages 69–75, 2004.

[17] V. Digalakis, P. Monaco, and H. Murveit. Genones : Generalized Mixture Tying in Continuous Hidden Markov Model-Based Speech Recognizers. In *IEEE Transactions on Speech and audio Processing*, volume 4, pages 281–289, 1996.

[18] V. Digalakis, S. Tsakalidis, C. Harizakis, and L. Neumeyer. Efficient Speech Recognition using Subvector Quantization and Discrete-Mixture HMMs. In *Computer Speech and Language*, volume 14, pages 33–46, Janvier 2000.

[19] JM. Dolmazon, F. Bimbot, G. Adda, M. El-Bèze, JC. Caërou, J. Zeiligera, and M. Adda. Organisation de la Première Campagne AUPELF pour l'Evaluation des Systèmes de Dictée Vocale. In *Journées Scientifiques and Techniques Francil*, pages 13–18, 1997.

[20] K. El-Maleha, M. Kleina, G. Petrucci, and P. Kabal. Speech/Music Discrimination for Multimedia Applications. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, pages 2445–2448, 2000.

[21] JG. Fiscus. A Post-Processing System to Yield Reduced Word Error Rates : Recognizer Output Voting Error Reduction (ROVER). In *IEEE workshop on Automatic Speech Recognition and Understanding ASRU*, pages 347–354, Santa Barbara, Décembre 1997.

[22] C. Fredouille, D. Matrouf, G. Linares, and P. Nocera. Segmentation en Macro-classes Acoustiques d'Emissions Radiophoniques dans le cadre d'ESTER. In *Journées d'Etude sur la Parole JEP*, pages 225–228, Fès, Avril 2004.

[23] MJF. Gales, DY. Kim, PC. Woodlet, HY. Chan, D. Mrva, R. Sinha, and SE. Tranter. Progress in the CU-HTK Broadcast News Transcription System. In *IEEE Transactions on Audio, Speech, and Language Processing*, volume 14, pages 1513–1525, 2006.

[24] MJF. Gales, KM. Knill, and S. Young. Use of Gaussian Selection in Large Vocabulary Continuous Speech Recognition using HMMs. In *Proceedings ICSLP*, pages 470–473, Philadelpie, Octobre 1996.

[25] S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, JF. Bonastre, and G. Gravier. The Ester Phase II Campaign for the Rich Transcription of French Broadcast News. In *European Conference on Speech, Communication and Technology Eurospeech*, Lisboa, 2005.

[26] JL. Gauvain, G. Adda, L. Lamel, F. Lefèvre, and H. Schwenk. Transcription de la Parole Conversationnelle. In *Journées d'Etude sur la Parole JEP*, Fès, Avril 2004.

[27] JL. Gauvain and L. Lamel. Large - Vocabulary Continuous Speech Recognition : Advances and Applications. In *proceedings of the IEEE*, volume 88, pages 1181–1200, 2000.

[28] G. Gravier. Spro4.0. http ://www.irisa.fr/metiss/guig/software.html, 2007.

[29] G. Gravier, JF. Bonastre, S. Galliano, E. Geoffrois, K. Mc-Tait, and K. Choukri. ESTER Une Campagne d'Evaluation des Systèmes d'Indexation d'Emissions Radiophoniques. In *Journées d'Etude sur la Parole JEP*, pages 253–256, Fès, Avril 2004.

[30] G. Gravier, F. Yvon, B. Jacob, and F. Bimbot. Sirocco : un Système Ouvert de Reconnaissance de la Parole. In *XXIVème Journées d'Etude sur la Parole*, pages 273–276, Nancy, 2002.

[31] GD. Guo and SZ. Li. Content-Based Audio Classification and Retrieval by Support Vector Machines. In *IEEE Transactions on Neural Network*, volume 14, pages 209–215, Janvier 2003.

[32] JP. Haton, C. Cerisara, D. Fohr, Y. Laprie, and K. Smaïli. *Reconnaissance Automatique de la Parole du Signal à son Interprétation*. Dunod, Paris, 2006.

[33] M.Y. Hwang. *Sub-phonetic Acoustic Modeling for Speaker Independent Continuous Speech Recognition*. PhD thesis, Université de Carnegie Mellon, Pittesburg, 1993.

[34] F. Jelinek. Continuous Speech Recognition by Statistical Methods. In *Proceedings of the IEEE*, volume 64, pages 532 – 556, 1976.

[35] F. Jürgen and R. Ivica. The Bucket Box Intersection (BBI) Algorithm for Fast Approximative Evaluation of Diagonal Mixture Gaussians. In *Proceedings International Conference on Acoustics, Speech, and Signal Processing ICASSP*, volume 2, pages 837–840, Atlanta, May 1996.

[36] F. Jürgen, R. Ivica, and S. Tilo. Speeding up the Score Computation of HMM Speech Recognizers with the Bucket Voronoi Intersection Algorithm. In *European Conference on Speech, Communication and Technology Eurospeech*, pages 1091–1094, Madrid, 1995.

[37] KM. Knill, MJF. Gales, and S. Young. State based Gaussian Selection in Large Vocabulary Continuous Speech Recognition using HMMs. In *IEEE Transactions on Speech and Audio Processing*, volume 7, pages 152–161, Mars 1999.

[38] A. Lee, T. Kawahara, and K. Shikano. Gaussian Mixture Selection Using Context Independent HMM. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, volume 7, Scandinavia, 2001.

[39] A. Lee, T. Kawahara, K. Takeda, and K. Shikano. A New Phonetic Tied-Mixture Model for Efficient Decoding. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, volume 3, pages 1269–1272, Istanboul, 2000.

[40] J. Leppänen and I. Kiss. Gaussian Selection with Non-overlapping Clusters for ASR in Embedded Devices. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, France, May 2006.

[41] X. Li and J. Bilmes. Feature Pruning in Likelihood Evaluation of HMM-Based Speech Recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding ASRU*, pages 303– 308, Decembre 2003.

[42] X. Li and J. Bilmes. Feature Pruning for Low-Power ASR Systems in Clean and Noisy Environments. In *IEEE Signal Processing Letters*, volume 12, Juillet 2005.

[43] G. Linares, P. Nocera, and D. Matrouf. Partitionnement Dynamique des Distributions pour le Calcul des Emissions dans un Décodeur Acoustico-Phonétique Markovien. In *Journées d'Etude sur la Parole JEP*, 2000.

[44] B. Logan. Mel Frequency Cepstral Coefficients for Music Modelling. In *International Symposium on Music Information Retrieval*, Izmir, 2000.

[45] L. Lu, HJ. Zhang, and H. Jiang. Content Analysis for Audio Classification and Segmentation. In *IEEE Transactions on Speech and Audio Processing*, volume 10, pages 504– 516, Octobre 2002.

[46] B. Mak. *Towards A Compact Speech Recognizer : Subspace Distribution Clustering Hidden Markov Model*. PhD thesis, Oregon Graduate Institute of Science and technology, Avril 1998.

[47] B. Mak, E. Bocchieri, and E. Barnard. Stream Derivation and Clustering Scheme for Subspace Distribution Clustering Hidden Markov Model. In *IEEE Automatic Speech Recognition and Understanding Workshop ASRU*, pages 339–346, Santa Barbara, Decembre 1997.

[48] L. Mangu, E. Brillet, and A. Stolke. Finding Consensus Among Words : Lattice-Based Word Error Minimization. In *proceedings Eurospeeech*, pages 495–498, 1999.

[49] M. Manta, F. Antoine, S. Galliano, and C. Barras. Transcriber. http ://www.etca.fr/CTA/gip/Projets/Transcriber, 2007.

[50] R. Messina and D. Jouvet. Sequential Clustering Algorithm for Gaussian Mixture Initialisation. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, pages 264–275, 2004.

[51] C. Mokbel. Online Adaptation of HMMs to Real Life Conditions : A Unified Framework. In *IEEE Transaction on Speech and Audio Processing*, volume 9, pages 342–357, Mai 2001.

[52] N. Morgan, E. Luissier, A. Janin, and B. Kingsbury. Reducing errors by increasing the error rate : MLP Acoustic Modeling for Broadcast News Transcription. In *DARPA Broadcast News Workshop*, Herndon, Février 1999.

[53] H. Murveit, P. Monaco, V. Digalakis, and J. Butzberger. Techniques to Achieve an Accurate Real-time Large Vocabulary Speech Recognition System. In *Proceedings of the workshop Human Language Technology*, pages 393–398, New Jersey, 1994.

[54] H. Ney and S. Ortmann. Progress in Dynamic Programming Search for LVCSR. In *IEEE Signal Processing Magazine*, volume 88, pages 1224–1240, 2000.

[55] H. Ney, R. Uaeb-Umbach, B.-H Tran, and M. Oerder. Improvements in Beam Search for 10000-Word Continuous Speech Recognition. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, volume I, pages 9–12, San Francisco, Mars 1992.

[56] Nist. National institute of standards and technology. www.nist.giv/speech, 2003.

[57] J. James Odell. *The Use of Context in Large Vocabulary Speech Recognition.* PhD thesis, Université de Cambridge, Cambridge, 1995.

[58] J. Olsen. Gaussian Selection using Multiple Quantisation Indexes. In *IEEE Nordic Processing symposium*, 2000.

[59] S. Ortmanns, A. Eiden, H. Ney, and N. Coenen. Look-Ahead Techniques for Fast Beam Search. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, pages 1783–1789, Germany, April 1997.

[60] S. Ortmanns, A. Eiden, H. Ney, and N. Coenen. Look-Ahead Techniques for Improved Beam Search. In *CRIM-FORWISS Workshop*, pages 10–22, Montreal, April 1997.

[61] S. Ortmanns, H. Ney, and A. Eiden. Language-Model Look- Ahead for Large Vocabulary Speech Recognition. In *International Conference on Spoken Language Processing*, pages 2091–2094, Philadelphie, Octobre 1996.

[62] S. Ortmanns, H. Ney, and T. Firslaff. Fast Likelihood Computation Methods for Continuous Mixture Densities in Large Vocabulary Speech Recognition. In *European Conference on Speech Communication and Technology*, pages 139–142, Rhodès, Septembre 1998.

[63] M. Padmanabhan and M. Picheny. Large Vocabulary Speech Recognition Algorithms. In *Computer Magazine*, volume 35, 2002.

[64] M. Padmanablan, L. Bahl, and D. Nahamoo. Partitioning the Feature Space of a Classifier with Linear Hyperplanes. In *IEEE Transactions on Speech and Audio Processing*, volume 7, pages 282–288, May 1999.

[65] M. Padmanablan, E.E Jan, L. Bahl, and M.Picheny. Decision-Tree based Feature Space Quantization for Fast Gaussian Computation. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 325–330, Santa Barbara, Decembre 1997.

[66] D. Pallett. A Look at NIST's Benchmark ASR Tests : Past, Present, and Future. In *IEEE Woorkshop on Automatic Speech Recognition and understanding ASRU*, pages 483– 488, Novembre 2003.

[67] D. Pallett, J. Fiscus, W. Fisher, J. Garofolo, B. Lund, A. Martin, and M. Przybocki. 1994 Benchmark Tests for the ARPA Spoken Language Program. In *Proceedings ARPA Spoken Language Systems Technology Workshop*, pages 5–36, Austin, Janvier 1995.

[68] C. Panagiotakis and G. Tziritas. A Speech/Mmusic Discriminator based on RMS and Zero-Crossings. In *IEEE Transactions on Multimedia*, 2005.

[69] DB. Paul. An Investigation of Gaussian Shortlists. In *IEEE workshop on Automatic Speech Recognition and Understanding ASRU*, Décembre 1999.

[70] J. Pinquier. *Indexation Sonore : Recherche de Composantes Primaires pour une Structuration Audiovisuelle*. PhD thesis, Université de Toulouse, France, Décembre 2004.

[71] J. Pinquier, JL. Rouas, and R. Obrecht. A Fusion Study in Speech/Music Classification. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, volume 2, pages 17–20, Avril 2003.

[72] M. Ravishankar. Sphinx-3 s3.x decoder (x=5). Sphinx Speech Group School of Computer Science Carnegie Mellon University, Juillet 2004.

[73] M. Ravishankar, R. Bisiani, and E. Thayer. Sub-vector Clustering to Improve Memory and Speed Performance of Acoustic Likelihood Computation. In *European Conference on Speech, Communication and Technology Eurospeech*, 1997.

[74] M. Ravishankar, R. Singh, B. Raj, and R.M Stern. The 1999 CMU 10X Real Time Broadcast News Transcription System. In *Nist Speech Transcription Workshop*, 2000.

[75] J. Razik, D. Fohr, O. Mella, and P. Valles. Segmentation Parole/Musique pour la Transcription Rapide. In *Journées d'Etude sur la Parole JEP*, Fès, Avril 2004.

[76] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco. Connectionist Probability Estimators in HMM speech Recognition. In *IEEE Transactions on Speech and Audio Processing*, volume 2, pages 161 – 174, 1994.

[77] A. Sankar and V. Ramana. Parameter Tying and Gaussian Clustering for Faster, Better, and Smaller Speech Recognition. In *European Conference on Speech, Communication and Technology Eurospeech*, Grèce, 1999.

[78] A. Sankar, V. Ramana, A. Slolcke, and F. Weng. Improved Modeling and Efficiency for Automatic Transcription of Broadcast News. In *Speech Communication*, volume 37, pages 133–158, 2002.

[79] J. Saunders. Real-time Discrimination of Broadcast Speech/Music. In *IEEE International Conference on Audio Speech and Signal Processing ICASSP*, pages 993–996, Atlanta, 1996.

[80] E. Scheirer and M. Slaney. Construction and Evaluation of a Robust Mainframe Speech/Music Discriminator. In *IEEE International Conference on Audio Speech and Signal Processing*, pages 1331–1334, Munich, 1997.

[81] M. Seck. *Détection de Ruptures and Suivi de Classes de Sons pour l'Indextaion Sonore.* PhD thesis, Université Rennes I, France, Janvier 2001.

[82] R. Singh. Sphinxtrain. http ://fife.speech.cs.cmu.edu/sphinxman/scriptman1.html, Novembre 2000.

[83] S. Srivastava. Fast Gaussian Evaluation in Large Vocabulary Continuous Speech Recognition. Master's thesis, Mississippi State University, Décembre 2002.

[84] J. Suontausts, J. Hakkinen, and O. Viikki. Fast Decoding Techniques for Practical Realtime Speech Recognition Systems. In *IEEE Workshop on Automatic Speech Recognition and Understanding ASRU*, décembre 1999.

[85] S. Takahashi and S. Sagayama. Four-level Tied-structure for Efficient Representation of Acoustic Modeling. In *IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP*, volume 1, pages 520–523, 1995.

[86] M. Woszczyna. *Fast Speaker Independent Large Vocabulary Speech Recognition.* PhD thesis, Université de Karlsruhe, Allemagne, 1998.

[87] M. Woszczyna and M. Finke. Minimizing Search Errors due to Delayed Bigrams in Real-Time Speech Recognition Systems. In *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, volume 1, pages 137–140, Mai 1996.

[88] S. Young. Statistical Modelling in Continuous Speech Recognition (CSR). In *Proceedings of the 17th International Conference on Uncertainty in Artificial Intelligence*, Seattle, WA, Août 2001.

[89] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book version 3.2.* Cambridge University. Engineering Department, 2002.

[90] S. Young, J. Odell, and P. Woodland. Tree-based State Tying for High Accuracy Acoustic Modeling. In *proceedings ARPA Workshop on Human Language Technology*, pages 307–312, 1994.

[91] Q. Zhu, A. Stolckea, BY. Chen, and N. Morgan. Using MLP Features in SRI's Conversational Speech Recognition System. In *European Conference on Speech Communication and Technology*, pages 2141–2144, Lisbone, Septembre 2005.

[92] L. Zouari and G. Chollet. Efficient Mixture for Speech Recognition. In *International Conference in Pattern Recognition, ICPR*, pages 294–297, Hongkong, Août 2006.

[93] L. Zouari and G. Chollet. Sélection des Paramètres pour la Discrimination Parole/non Parole d'Émissions Radio Diffusées. In *Cinquième édition des Ateliers de Travail sur le Traitement and l'Analyse de l'Information TAIMA*, Hammamet, Mai 2007.