THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Cotutelle internationale avec :
Université Mohammed V-Agdal de Rabat

Présentée et soutenue par :
Reda JOURANI

Le 6 septembre 2012

Titre :

Reconnaissance automatique du locuteur par des GMM à grande marge

---

École doctorale et discipline ou spécialité :
ED MITT : Image, Information, Hypermedia

Unité de recherche :
Institut de Recherche en Informatique de Toulouse

Directeur(s) de Thèse :

Régine ANDRÉ-OBRECHT, Professeur de l'Université de Toulouse, France
Driss ABOUTAJDINE, Professeur à la Faculté des Sciences de Rabat, Maroc

Rapporteurs :

Jean-François BONASTRE, Professeur à l'Université d'Avignon et des pays de Vaucluse, France
Alexandros POTAMIANOS, Associate Professor at the Technical University of Crete, Greece

Autre(s) membre(s) du jury :

Abdelhak MOURADI, Professeur à l'ENSIAS, Maroc	Président
Khalid DAOUDI, Chargé de Recherches à l'INRIA, France	Encadrant scientifique
Mohammed HAMRI, Professeur à la Faculté des Sciences de Rabat, Maroc	Examinateur

Remerciements

Les travaux présentés dans le mémoire ont été effectués à l'Institut de Recherche en Informatique de Toulouse - Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier).

Cette thèse de doctorat a été préparée en cotutelle internationale avec l'Université Mohammed V-Agdal de Rabat, Maroc.

Mes remerciements vont en premier lieu à mes encadrants – la Professeure *de l'Université de Toulouse* Régine ANDRÉ-OBRECHT, le Professeur *à la Faculté des Sciences de Rabat* Driss ABOUTAJDINE et le Docteur Khalid DAOUDI Chargé de Recherches *à l'INRIA* – pour leur confiance et l'attention qu'ils ont bien voulu porter à mon travail.

Je remercie également le Professeur Abdelhak MOURADI ex-directeur de *l'ENSIAS*, d'avoir accepter de présider le jury de ma soutenance.

Je remercie aussi les autres membres du jury, notamment le Professeur *à l'Université d'Avignon et des pays de Vaucluse* Jean-François BONASTRE, le Professeur *à l'Université technique de Crète* Alexandros POTAMIANOS et le Professeur *à la Faculté des Sciences de Rabat* Mohammed HAMRI pour leurs remarques et leurs encouragements.

Merci à tous les membres de l'équipe SAMoVA de l'IRIT, Benjamin, Christine, Hélène, Isabelle, Jérôme, Julien, Khalid, Lionel, Maxime, Patrice, Philippe, Philippe et Régine. Une pensée également à Elie, Frédérick, Hervé, Ioannis et José.

Je remercie également tous les membres du laboratoire LRIT, spécialement Ahmed, Amir, Hajar, Hassan, Hicham, Hind, Khalid, Nabil, Rachid et Samira.

Un grand merci à mes parents et à ma famille pour leur soutien et leur affection.

Sans oublier finalement mes amis ; Ana-Maria, Anass, Dieudonné, Gregory ...

Table des matières

Table des figures

Liste des tableaux

Glossaire

AT-norm	Adaptive Test normalization
CMLLR	Constrained Maximum Likelihood Linear Regression
CMS	Cepstral Mean Subtraction
CMVN	Cepstral Mean and Variance Normalization
DET	Detection Error Tradeoff
EER	Equal Error Rate
EM	Expectation-Maximisation
FM	Feature Mapping
FSMS	Feature Space Mahalanobis Sequence
FT	Feature Transformation
GLDS	Generalized Linear Discriminant Sequence
GMM	Gaussian Mixture Model
GSL	GMM Supervector Linear kernel
HMM	Hidden Markov Model
HT-PLDA	Heavy-Tailed Probabilistic Linear Discriminant Analysis
IPDF	Inner Product Discriminant Functions
JFA	Joint Factor Analysis
KBM	Binary-key background model
L-BFGS	Limited-memory Broyden-Fletcher-Goldfarb-Shanno
LDA	Linear Discriminant Analysis
LFCC	Linear-Frequency Cepstral Coefficients
LM-dGMM	Large Margin diagonal GMM
LM-GMM	Large Margin Gaussian Mixture Models
LPCC	Linear Predictive Cepstral Coefficients
LPC	Linear Predictive Coding
MAP	Maximum A Posteriori
MFCC	Mel-Frequency Cepstral Coefficients
minDCF	minimum of Detection Cost Function
MLE	Maximum Likelihood Estimation
MLLR	Maximum Likelihood Linear Regression
NAP	Nuisance Attribute Projection
NIST	U.S. National Institute of Standards and Technology

PCA	Principal Component Analysis
PIUBM	Phonetically Inspired Universal Background Model
PLDA	Probabilistic LDA
PLP	Perceptual Linear Prediction
RASTA	RelAtive SpecTrAl
SFA	Symmetrical "Latent" Factor Analysis
S-norm	Symmetric normalization
SRE	Speaker Recognition Evaluation
SVM	Support Vector Machine
T-norm	Test normalization
UBM	Universal Background Model
VQ	Vector Quantisation
WCCN	Within-Class Covariance Normalization
WNAP	Weighted NAP
Z-norm	Zero normalization

Chapitre 1

Introduction

Durant la dernière décennie, on a vu l'apparition de plus en plus de domaines et d'applications nécessitant l'authentification des personnes : contrôle d'accès à des bâtiments sécurisés, transmission de données personnelles, recherche et surveillance d'individus, contrôle aux frontières, justice, domotique ... Ces applications utilisent généralement des technologies d'authentification basées sur la biométrie. On parle dans ce cas là d'une authentification biométrique. L'authentification biométrique est la reconnaissance automatique d'une personne en utilisant des traits distinctifs, i.e., des caractéristiques physiques (biologiques) ou traits comportementaux personnels automatiquement mesurables, robustes et distinctifs qui peuvent être utilisés pour identifier ou vérifier l'identité prétendue d'un individu. Certaines technologies biométriques constituent une solution efficace relativement simple et pas chère qui assure de bonnes performances en reconnaissance. Elles offrent beaucoup plus d'avantages que les méthodes classiques d'authentification telles que l'utilisation de mots de passe ou de clés et badges d'accès, qui sont très vulnérables au vol et à la falsification. Les technologies biométriques fournissent en effet encore plus de sûreté et de confort d'utilisation.

On trouve deux grandes tâches en reconnaissance : l'identification et la vérification. Dans la première, le système biométrique pose et essaye de répondre à la question suivante : *qui est la personne X ?*. Le système requiert une information biométrique et la compare avec chaque information stockée dans la base de données des utilisateurs autorisés ; c'est une comparaison un parmi plusieurs. Une des applications d'identification est la recherche de criminels et de terroristes en utilisant des données de surveillance. Dans la vérification, le système biométrique demande à l'utilisateur son identité et essaye de répondre à la question : *est ce la personne X ?*. Dans cette tâche, l'utilisateur annonce son identité par l'intermédiaire, par exemple, d'un numéro d'identification ou d'un nom d'utilisateur, puis le système sollicite une information biométrique provenant de cette personne pour la comparaître avec la donnée préalablement enregistrée qui correspond à l'identité prétendue. C'est une comparaison vrai/faux. Le système trouvera finalement ou

non une correspondance entre les deux. La vérification est communément employée dans des applications de contrôle d'accès et de paiement par authentification.

Plusieurs informations biométriques ont été utilisées dans diverses applications et secteurs, à savoir : l'empreinte digitale, la main, l'iris, la rétine, le visage, la voix, les veines, la signature ... Ces modalités se distinguent par leur fiabilité, le niveau d'acceptation par l'utilisateur, le coût et l'effort de mise en œuvre. Des modalités comme l'iris ou la rétine donnent les meilleurs taux de reconnaissance en comparaison avec les autres informations biométriques. Cependant elles sont peu acceptées par les utilisateurs et sont lourdes à mettre en place. Selon le niveau de sécurité requis, les ressources financières et les contraintes techniques et d'utilisation, une modalité peut être privilégiée par rapport à une autre.

1.1 Cadre de l'étude

1.1.1 La reconnaissance automatique du locuteur

Depuis plusieurs dizaines d'années, la reconnaissance automatique du locuteur, i.e., la reconnaissance par la voix, fait l'objet de travaux de recherche entrepris par de nombreuses équipes de recherche dans le monde. En effet, NIST (U.S. National Institute of Standards and Technology) coordonne depuis 1996 une série d'évaluations SRE (Speaker Recognition Evaluation), dans le but de promouvoir la recherche en reconnaissance automatique du locuteur. Chaque année, NIST encourage les universitaires et industriels à participer à sa campagne d'évaluation, en soumettant des systèmes état de l'art et innovants aux tâches définies de reconnaissance. 58 structures de recherches établies dans les cinq continents ont participé par exemple à la campagne d'évaluation NIST-SRE 2010, qui était dédiée à la vérification du locuteur. D'autres travaux se sont attaqués à l'identification et l'indexation des locuteurs. Cet engouement pour la reconnaissance automatique du locuteur revient aux avancés enregistrées en traitement de la parole qui ont permis de concevoir des systèmes de plus en plus fiables. De plus, la reconnaissance automatique du locuteur offre beaucoup d'avantages par rapport aux autres technologies biométriques, notamment le facile recueil d'enregistrements que ce soit au su ou à l'insu de l'utilisateur et la simple mise en œuvre d'un système d'authentification. Un téléphone ou microphone fait facilement office de capteur (de dispositif d'acquisition), ce qui rend cette technologie relativement économique et facilement déployable.

Les systèmes de reconnaissance du locuteur utilisent la variabilité inter-locuteurs de production de la parole pour séparer les individus entre eux. Mais il y a aussi d'autres facteurs de variation qui s'ajoutent et qui altèrent la parole et nuisent à la robustesse des systèmes de reconnaissance du locuteur, comme les variabilités intra-locuteur et inter-sessions d'enregistrement.

Pour améliorer les performances de reconnaissance, la voix peut être utilisée en complément d'une autre modalité ; on parle dans ce cas là d'une authentification multimodale. Dans une application où les locuteurs cibles sont coopératifs, on peut envisager l'utilisation de phrases type mot de passe connues au préalable dans le système, ce qui permet d'augmenter sa fiabilité. La connaissance a priori ou non du contenu prononcé permet de distinguer entre deux modes de fonctionnement : dépendant et indépendant du texte. La reconnaissance indépendante du texte est plus difficile, c'est dans ce contexte que se situe ce travail de thèse.

1.1.2 Les approches scientifiques en reconnaissance automatique du locuteur

La majorité des systèmes actuels de reconnaissance du locuteur sont basés sur l'utilisation de modèles de mélange de Gaussiennes (GMM). Ces modèles de nature générative sont généralement appris en utilisant les techniques de Maximum de Vraisemblance et de Maximum A Posteriori (MAP) [Reynolds et al., 2000]. Cependant, cet apprentissage génératif ne s'attaque pas directement au problème de classification étant donné qu'il fournit un modèle de la distribution jointe. Ceci a conduit récemment à l'émergence d'approches discriminantes qui tentent de résoudre directement le problème de classification [Keshet and Bengio, 2009], et qui donnent généralement de bien meilleurs résultats. Par exemple, les machines à vecteurs de support (SVM), combinées avec les supervecteurs GMM sont parmi les techniques les plus performantes en reconnaissance automatique du locuteur [Campbell et al., 2006b].

1.2 Cadre des recherches proposées

Nos travaux ont pour objectif général la proposition de nouveaux modèles GMM à grande marge (modèles discriminants) pour la reconnaissance automatique du locuteur qui soient une alternative aux modèles GMM génératifs classiques et la technique discriminante état de l'art GMM-SVM.

1.2.1 GMM à grande marge

Récemment, une nouvelle approche discriminante pour la séparation multi-classes a été proposée et appliquée en reconnaissance de la parole, les GMM à grande marge (LM-GMM) [Sha and Saul, 2006], [Sha and Saul, 2007], [Sha, 2007]. Cette dernière utilise la même notion de marge que les SVM et possède les mêmes avantages que les SVM en terme de la convexité du problème à résoudre. Mais elle diffère des SVM car elle construit une frontière non-linéaire entre les classes directement dans l'espace des données. De manière très schématique, cette modélisation construit une frontière non-linéaire de type

quadratique entre les classes dans l'espace des données en maximisant une marge entre ces dernières. Les classes sont représentées par des ellipsoïdes, et sont initialisées par des modèles GMM.

1.2.2 Application à la reconnaissance automatique du locuteur

En reconnaissance automatique du locuteur, les systèmes GMM de l'état de l'art utilisent des matrices de covariance diagonales et sont appris par adaptation MAP des vecteurs moyennes d'un modèle du monde. Exploitant cette propriété, nous présentons dans cette thèse une version simplifiée des LM-GMM, les modèles LM-GMM à matrices de covariance diagonales (LM-dGMM). Nous évaluons nos modèles LM-dGMM dans des tâches de reconnaissance du locuteur en utilisant les données de la campagne d'évaluation de NIST-SRE 2006 [NIST, 2006].

À partir de notre système initial, i.e., le modèle de mélange LM-dGMM, nous proposons également d'autres variantes : une revisite de l'initialisation des modèles qui permet d'éviter de faire appel à l'adaptation MAP et à l'utilisation d'un ensemble de développement, une prise en compte uniquement des k meilleures composantes du mélange LM-dGMM qui rend accessible par cette modélisation le traitement de grands volumes de données, et la définition d'une nouvelle stratégie de traitement des données aberrantes qui occurrent souvent dans les enregistrements audio.

Cette thèse de doctorat a été préparée en cotutelle entre l'Université Toulouse 3 Paul Sabatier et l'Université Mohammed V-Agdal de Rabat, au sein de l'équipe SAMoVA [1] de l'Institut de Recherche en Informatique de Toulouse (IRIT) et le Laboratoire de Recherche en Informatique et Télécommunications (LRIT) de la Faculté des Sciences de Rabat. Cette thèse se veut d'enrichir d'anciens travaux réalisés dans l'équipe SAMoVA traitant la vérification du locuteur [Louradour, 2007] et la caractérisation et reconnaissance des rôles des locuteurs [Bigot, 2011], et une thèse préparée au laboratoire LRIT sur l'indexation de documents audio au sens du locuteur [Rougui, 2008].

1.3 Organisation du mémoire

Ce document se divise en deux grandes parties. La première partie fait un état de l'art de la reconnaissance automatique du locuteur et la deuxième correspond aux travaux de recherche réalisés durant notre préparation de cette thèse, à savoir, la présentation des modèles LM-dGMM et leur évaluation dans le contexte reconnaissance du locuteur.

La première partie est formée d'un seul chapitre qui s'attache à faire un tour complet des approches et techniques majoritairement utilisées en reconnaissance automatique du locuteur. C'est un rapide survol des fondements des systèmes état de l'art. Les différents

[1] Structuration, Analyse et MOdélisation de documents Vidéo et Audio

modules/parties d'un système classique de reconnaissance du locuteur sont décrits en détail.

La seconde partie de ce document concerne la problématique abordée dans cette thèse au cours de deux chapitres. Le chapitre 3 introduit les modèles GMM à grande marge. S'imprégnant des recherches menées par Fei SHA, nous commençons le chapitre par une description des modèles LM-GMM originaux avant de présenter par la suite les modèles LM-dGMM que nous avons développés. Une première évaluation satisfaisante dans le contexte de l'identification du locuteur, nous a conduit à approfondir ces modèles et proposer des variantes. Elles sont déclinées dans le chapitre 4 et évaluées dans le cadre de tâches de reconnaissance du locuteur.

Nous concluons finalement ce manuscrit par une synthèse générale de l'ensemble des travaux menés durant nos recherches, en s'ouvrant sur des perspectives de travaux qui pourront être effectués dans de futures recherches.

Chapitre 2

Reconnaissance automatique du locuteur

2.1 Introduction

Les systèmes de reconnaissance automatique du locuteur utilisent la variabilité inter-locuteurs due à la production de la parole pour séparer les individus les uns des autres. La production de la parole fait intervenir plusieurs organes (poumons, cordes vocales, conduit vocal, langue, lèvres ...) qui agissent comme un filtre linéaire, dont le spectre est caractérisé par des maxima d'énergie, appelés formants. Le signal de parole est un signal non stationnaire, qui est considéré comme quasi stationnaire sur des fenêtres d'analyse de très courte durée (de l'ordre de 10 à 30 ms). On peut décomposer la parole en un ensemble d'unités sonores de base. Ces unités sonores ont des équivalents linguistiques, les phonèmes, qui sont caractéristiques d'une langue donnée. On distingue entre sons voisés dont le signal temporel est quasi-périodique et sons non voisés, où le signal est considéré comme bruité. Dans le premier cas, la source d'excitation est modélisée par une série d'impulsions périodiques de fréquence (de voisement) F0, correspondante à la fréquence de vibration des cordes vocales, appelée aussi la fréquence fondamentale ou le pitch. Dans le deuxième cas, la source est modélisée par un bruit blanc gaussien.

Les performances des systèmes de reconnaissance dépendent de la bonne représentation et caractérisation de la variabilité inter-locuteurs, qui résulte en réalité des différences morphologiques, physiologiques et de prononciation entre individus. Il y a aussi d'autres facteurs de variation qui s'ajoutent et qui altèrent cependant la parole : les variabilités intra-locuteur et inter-sessions, qui compliquent davantage les tâches de reconnaissance. En effet, il est impossible pour un même individu de reproduire exactement un même signal de parole. Il existe une variabilité propre à chaque locuteur, dépendante de son état physique et psychologique. Ainsi, le facteur âge, l'état de santé ou des changements émotionnels altèrent les performances. On observe également une dégradation importante des performances quand les canaux d'enregistrement et de transmission changent entre

les sessions d'apprentissage et de test. Cette variabilité inter-sessions reste la difficulté majeure à surmonter en reconnaissance automatique du locuteur.

Ce chapitre fait un tour complet des approches et techniques majoritairement utilisées en reconnaissance automatique du locuteur ; en extraction des paramètres, modélisation et normalisation des scores. Nous abordons ensuite le problème de compensation de la variabilité inter-sessions, en parlant des principaux méthodes et formalismes proposés. Nous citerons après quelques exemples de nouvelles technologies puis nous donnerons finalement une synthèse des spécificités de certains des meilleurs systèmes de vérification du locuteur, soumis à la campagne d'évaluation NIST-SRE 2010 [NIST, 2010].

2.2 Les outils de la reconnaissance automatique du locuteur

Un système d'identification du locuteur est basé sur la connaissance de C clients du système, représentés chacun par un modèle. À l'arrivée d'un signal de parole, le système doit déterminer l'identité de la personne qui parle dans cet enregistrement, parmi les C connues. Un système de vérification répond quand à lui à une autre question, en se basant sur la connaissance (du modèle) d'une identité clamée i et d'un modèle du monde (UBM), qui représente en réalité l'hypothèse opposée de production. Le système détermine si le locuteur i parle ou non dans l'enregistrement actuel. La majorité des systèmes de reconnaissance du locuteur que se soit dans les tâches d'identification ou de vérification, utilise les Modèles de Mélange de lois Gaussiennes (GMM) dans la modélisation des locuteurs, que ce soit exclusivement ou en combinaison avec d'autres techniques.

Un système de reconnaissance (identification ou vérification) comporte plusieurs composantes : un module d'extraction de paramètres, un bloc d'appariement, un module de normalisation des scores d'appariement et un module de décision. Les figures Fig. 2.1 et Fig. 2.2 donnent l'architecture d'un système d'identification et d'un système de vérification du locuteur.

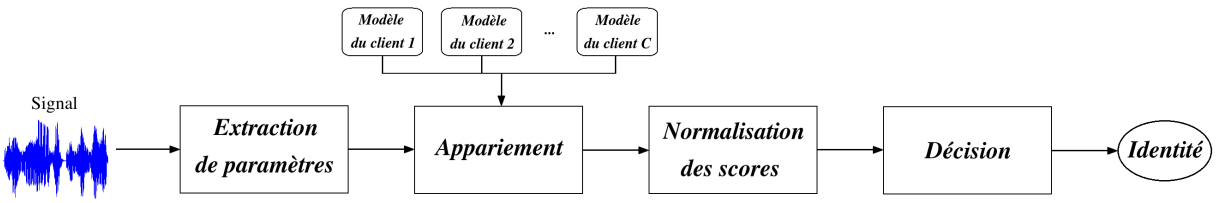


FIG. 2.1: *Architecture d'un système d'identification du locuteur.*

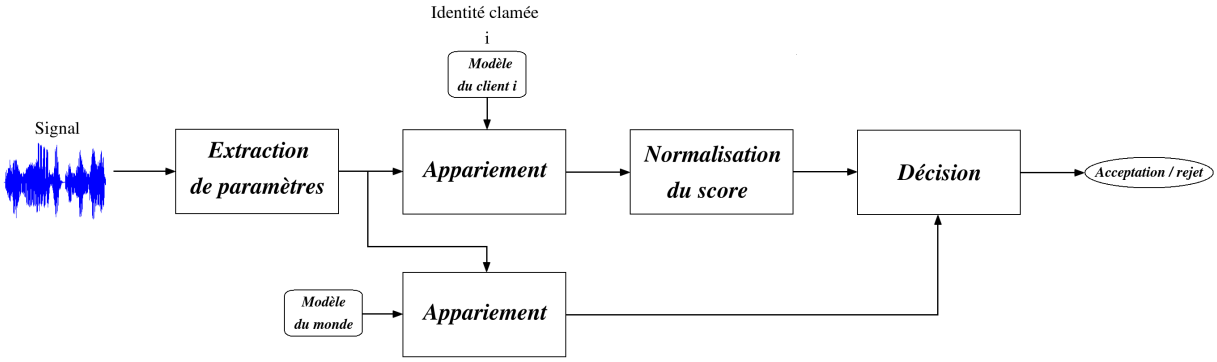


FIG. 2.2: *Architecture d'un système de vérification du locuteur.*

Le module d'extraction de paramètres transforme un signal de parole en une séquence de vecteurs acoustiques utiles à la reconnaissance. Ce module comporte différents sous-modules à savoir, la paramétrisation, la segmentation parole / non parole et des prétraitements.

Dans une application d'identification, on calcule les log-vraisemblances des vecteurs de paramètres par rapport aux C modèles des clients du système. Ces scores d'appariement sont ensuite normalisés pour réduire les effets de la variabilité inter-sessions, et le module de décision sélectionne le modèle (l'identité) le plus vraisemblable a posteriori. Dans une application de vérification, on calcule à la fois la log-vraisemblance (normalisée) des vecteurs de paramètres par rapport au modèle de l'identité clamée i, et par rapport au modèle du monde. Le module de décision compare finalement le rapport des deux scores d'appariement par rapport à un seuil de décision pour déterminer si le locuteur cible i parle ou non dans le signal de parole. Ce seuil de décision dépend de l'application.

Durant la phase d'apprentissage des modèles des locuteurs, l'estimation par maximum de vraisemblance (l'algorithme EM [Bishop, 2006]) donne de bons résultats quand on dispose d'une grande quantité de données, suffisamment nécessaire pour estimer robustement les paramètres d'un modèle GMM. Cependant en reconnaissance automatique du locuteur, peu de données sont généralement disponibles pour apprendre directement ces modèles. On utilise donc les méthodes d'adaptation de modèles, où on adapte un modèle du monde aux données d'apprentissage du locuteur. Ce modèle du monde est appris sur une grande quantité de données appartenant à plusieurs locuteurs, et enregistrées durant diverses sessions. Cet ensemble d'apprentissage doit être le plus riche et varié possible.

La figure Fig. 2.3 montre le processus classique d'apprentissage d'un locuteur cible dans les systèmes de reconnaissance du locuteur.

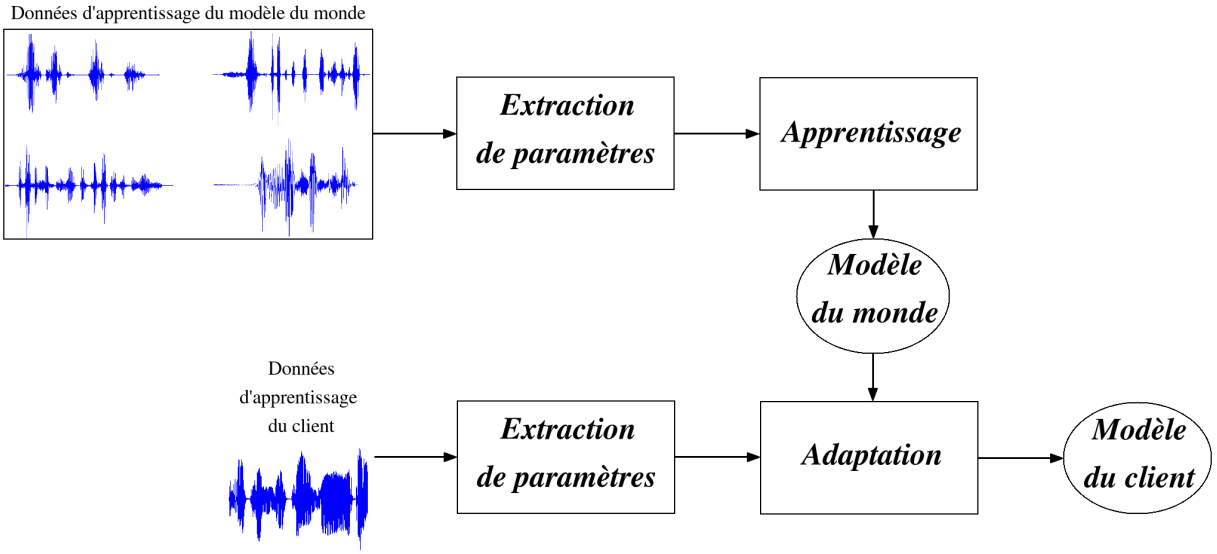


FIG. 2.3: *Architecture d'un module d'apprentissage de systèmes de reconnaissance automatique du locuteur.*

Nous présentons dans les prochaines sous-sections les différentes approches et techniques utilisées en extraction de paramètres, modélisation et normalisation des scores.

2.2.1 Extraction de paramètres

Cette sous-section commence par évoquer les différents paramètres du signal de parole qui sont utilisés en reconnaissance automatique du locuteur, à savoir des paramètres spectraux, des paramètres liés à la source vocale, des paramètres prosodiques et des paramètres de haut-niveau. Nous parlerons ensuite de la segmentation *parole / non parole* qui tend à ne garder que les trames utiles au processus de reconnaissance. Bien qu'elle soit faite généralement après la phase de paramétrisation, la segmentation peut aussi intervenir avant celle ci. Nous exposerons finalement les principales techniques de normalisation des paramètres acoustiques qui ont été proposées en reconnaissance automatique du locuteur.

2.2.1.1 Paramétrisation

Divers paramètres du signal de parole ont été proposés en reconnaissance automatique du locuteur. Idéalement, ces paramètres doivent avoir une forte variabilité inter-locuteurs et une faible variabilité intra-locuteur, permettant ainsi de discriminer plus facilement différents individus. De plus, ces paramètres doivent être robustes aux différents bruits et variations inter-sessions, et difficiles à reproduire par un imposteur.

Peuvent être utilisés, des paramètres spectraux à court terme, des paramètres liés à la source vocale, des paramètres prosodiques et des paramètres de haut-niveau. Les

paramètres spectraux sont calculés dans des fenêtres d'analyse à très courte durée, et reflètent le timbre de la voix et la résonance du conduit vocal supralaryngé. Les paramètres de la source vocale caractérisent la source vocale (le débit glottique). Les paramètres prosodiques se calculent quand à eux sur des dizaines ou des milliers de millisecondes. Finalement, le groupe des paramètres de haut-niveau (phonèmes, paramètres idiolectaux, sémantique, accent, prononciation ...) est sensé être très robuste face au bruit et aux variations inter-sessions. Cependant, ces paramètres restent difficiles à extraire et nécessitent de très grandes quantités de données, ce qui les rend peu utilisés dans les systèmes actuels de reconnaissance du locuteur.

Paramètres spectraux à court terme

Le signal de parole varie continuellement au cours du temps, en fonction des mouvements articulatoires. Par conséquent, sa paramétrisation doit être effectuée sur de courtes fenêtres d'analyse (typiquement 10 à 30ms) où le signal est considéré comme quasi stationnaire. L'analyse utilise des fenêtres glissantes qui se recouvrent, à décalage régulier de 5 à 10ms. Le signal de parole est tout d'abord filtré par un filtre numérique de préaccentuation (un filtre passe-haut) pour intensifier les hautes fréquences, qui sont toujours plus faibles en énergie que les basses fréquences. Pour améliorer l'analyse et limiter les effets de bord, on pondère ensuite les trames du signal par une fenêtre temporelle aplatie aux extrémités, ce qui permet de réduire les discontinuités dans le signal. Différentes fenêtres ont été proposées et étudiées par la communauté du traitement de signal, e.g., Hann, Blackman, Kaiser et Hamming.

Dans la représentation source/filtre du signal de parole, ce signal résulte d'une convolution (dans le domaine temporel) de la source et du conduit vocal (filtre) : $s(n) = e(n) * h(n)$ [Rabiner and Schafer, 1978]. Le passage dans le domaine log-spectral permet de remplacer cette convolution par une somme : $\log|S(f)| = \log|E(f)| + \log|H(f)|$. Le cepstre réel d'un signal numérique est obtenu en appliquant une transformation de Fourier inverse au logarithme de son spectre. Dans ce nouveau domaine, la séparation source - conduit peut être faite aisément via un simple fenêtrage temporel (appelé liftrage). Plusieurs coefficients cepstraux sont utilisés en reconnaissance automatique du locuteur.

Les MFCC (Mel-Frequency Cepstral Coefficients) [Davis and Mermelstein, 1980] sont des coefficients cepstraux calculés par une transformée en cosinus discrète DCT appliquée sur des coefficients d'énergie. Une analyse en banc de filtres selon l'échelle de Mel transforme le spectre de puissance du signal en coefficients d'énergie par bandes de fréquences, avant de leur appliquer ensuite une compression logarithmique. Si les coefficients cepstraux sont issus d'une analyse en banc de filtres sur une échelle linéaire, on parle dans ce cas là des coefficients LFCC (Linear-Frequency Cepstral Coefficients).

Le signal de parole présente des spécificités liées à la production même, qui sont utiles à exploiter lors de son analyse. Par conséquent, on a proposé des méthodes d'analyse basées sur le processus de production de la parole. Le modèle source/filtre de production de la

parole [Fant, 1970] modélise le signal de parole comme la sortie d'un filtre tout-pôle excité par une suite d'impulsions régulières dans le cas d'un son voisé, ou par un bruit blanc dans le cas d'un son non-voisé. L'analyse par prédiction linéaire (Linear Predictive Coding) LPC [Markel and Gray, 1976], [Rabiner and Schafer, 1978] se fonde sur la corrélation entre les échantillons successifs du signal vocal et fait l'hypothèse du modèle acoustique linéaire.

En introduisant une excitation $e(n)$ de variance σ^2 à l'entrée d'un modèle d'ordre p, l'échantillon du signal à l'instant n $s(n)$ s'écrit comme une combinaison linéaire des p précédents échantillons :

$$s(n) \;=\; \left(\sum_{i=1}^{p} a_i s(n-i) \right) \;+\; e(n). \tag{2.1}$$

La fonction de transfert associé à ce filtre linéaire de prédiction est donnée par :

$$H(z) \;=\; \frac{S(z)}{E(z)} \;=\; \frac{\sigma^2}{1 - \sum_{i=1}^{p} a_i z^{-i}}. \tag{2.2}$$

Les coefficients de prédiction a_i de ce modèle Auto-Régressif AR, sont calculés en minimisant l'erreur quadratique moyenne induite par le modèle.

Le choix de l'ordre de prédiction résulte d'un compromis entre le temps, la quantité de données et la qualité d'analyse. Il est à noter que ce modèle assez "simplifié" peut être généralisé en un modèle plus "réel" ARMA (Auto-Régressif à Moyenne Ajustée), qui prend en compte le conduit nasal mis en parallèle du conduit vocal et le phénomène de rayonnement aux lèvres. Néanmoins, ce modèle ARMA est plus délicat à estimer que le modèle AR. Pour une qualité donnée de la modélisation, on préfère apprendre un modèle AR avec un ordre un peu surestimé à un modèle ARMA d'ordre inférieur.

L'analyse LPC permet de représenter l'enveloppe spectrale du signal à partir des coefficients de prédiction. La réponse fréquentielle du filtre linéaire de prédiction reflète les pics du spectre du signal de parole, ce qui rend cette analyse très utilisée pour la détermination des formants. Les coefficients de prédiction a_i sont rarement utilisés directement comme paramètres acoustiques. Ils sont plutôt transformés en de plus robustes et moins corrélés paramètres, comme les LPCC et les coefficients PLP.

Les LPCC (Linear Predictive Cepstral Coefficients) [Rabiner and Juang, 1993] sont des coefficients cepstraux c_i calculés à partir des coefficients de prédiction a_i :

$$\begin{aligned} c_0 \;&=\; \ln(\sigma^2), \\ c_i \;&=\; a_i \;+\; \sum_{k=1}^{i-1} \frac{k}{i} c_k a_{i-k}, \quad 1 \le i \le p, \\ c_i \;&=\; \sum_{k=1}^{i-1} \frac{k}{i} c_k a_{i-k}, \quad i > p, \end{aligned} \tag{2.3}$$

où σ^2 est le gain du modèle LPC.

L'analyse par prédiction linéaire perceptuelle (Perceptual Linear Prediction) PLP [Hermansky, 1990] prend en compte la perception humaine de la parole. Cette technique utilise des connaissances issues de la psychoacoustique lors de l'estimation d'un modèle AR, à savoir, une résolution non linéaire en fréquence à l'aide de bandes critiques sur une échelle de Bark, une préaccentuation du signal selon une courbe d'isosonie, et une compression en racine cubique pour simuler la loi de perception humaine en puissance sonore. Le spectre résultant est sujet finalement à une modélisation Auto-Régressive.

Paramètres de la source vocale

Les paramètres de la source vocale caractérisent le signal d'excitation glottique des sons voisés. On suppose que ces paramètres contiennent de l'information spécifique au locuteur. Dans le modèle source/filtre, la manière la plus évidente d'estimer le flux glottique (source du signal de parole) est d'appliquer l'inverse du filtre du conduit vocal estimé par prédiction linéaire. Divers systèmes de reconnaissance du locuteur utilisant des paramètres du flux glottique ont été proposés dans la littérature [Thévenaz and Hügli, 1995], [Plumpe et al., 1999], [Murty and Yegnanarayana, 2006], [Mahadeva Prasanna et al., 2006], [Zheng et al., 2007], [Gudnason and Brookes, 2008], [Kinnunen and Alku, 2009]. Les résultats obtenus montrent que la combinaison des paramètres du conduit vocal (MFCC, LPCC ...) avec les paramètres de la source vocale améliore les performances des systèmes.

Paramètres prosodiques

Contrairement aux paramètres spectraux à court terme traditionnels, les paramètres prosodiques peuvent impliquer de plus longs segments de parole (syllabe, mot, expression). Comme paramètres prosodiques, on trouve par exemple : la fréquence fondamentale, l'intonation, l'accentuation, l'énergie, le rythme et le débit de parole ; ils caractérisent le style d'élocution du locuteur. L'association de paramètres prosodiques aux paramètres spectraux à court terme permet d'améliorer les performances d'identification et de vérification des systèmes de reconnaissance du locuteur [Bartkova et al., 2002], [Adami et al., 2003], [Shriberg et al., 2005], [Ferrer et al., 2007] [Dehak et al., 2007].

La fréquence fondamentale (pitch) est directement liée à l'anatomie des cordes vocales (poids, taille, rigidité ...). Les cordes vocales des femmes sont généralement plus petites que celles des hommes, ce qui rend leur fréquence fondamentale plus haute en comparaison avec celle des hommes. De ce fait, elle peut être utilisée pour faire de la classification en genres des locuteurs. Le pitch peut être estimé en utilisant par exemple l'algorithme YIN [De Cheveigné and Kawahara, 2002]. Diverses caractéristiques descriptives de la fréquence fondamentale et de sa dynamique temporelle ont été étudiées dans la littérature, en utilisant différentes approches de modélisation [Atal, 1972], [Markel et al., 1977], [Carey et al., 1996], [Cheng and Leung, 1998], [Sönmez et al., 1998], [Shriberg et al., 2005],

[Chen et al., 2005], [Kinnunen and González-Hautamäki, 2005], [Laskowski and Jin, 2009]. D'après [Sönmez et al., 1997], le log de la fréquence fondamentale apparaît plus intéressant que la fréquence fondamentale elle même, et il a été utilisé dans [Cheng and Leung, 1998], [Sönmez et al., 1998], [Kinnunen and González-Hautamäki, 2005].

L'énergie est extraite directement du signal temporel. Sur une fenêtre d'analyse, elle est calculée par : $E = \sum_{n=1}^{N} s(n)^2$, et elle est généralement exprimée en décibels $E_{dB} = 10 \log E$. la variation de l'énergie et de sa dérivée première et seconde est liée à l'intonation du locuteur.

D'après [Shriberg et al., 2005], les paramètres statistiques de la fréquence fondamentale capturant le niveau du pitch sont plus performants que les paramètres de l'énergie et de la durée. La durée regroupe des caractéristiques relatives à l'organisation temporelle du discours (durée des phones et des silences ...). L'utilisation conjointe de ces trois informations prosodiques améliorent d'avantage les performances [Dehak et al., 2007]. Cependant l'estimation et la modélisation des différents niveaux de l'information prosodique reste relativement difficile, surtout dans les applications de reconnaissance indépendante du texte.

Paramètres de haut-niveau

En plus des différences physiologiques, chaque personne a sa propre habitude linguistique, sa propre manière de parler, et qui diffère en fonction de son statut socio-professionnel. Le système linguistique propre à une personne donnée est appelé idiolecte. Il se manifeste par des choix particuliers dans le vocabulaire et la grammaire (utilisation de mots, de phrases et de tours de parole particuliers) ainsi que dans des variantes dans l'intonation et la prononciation.

Le premier travail en reconnaissance automatique du locuteur à s'être intéressé à des caractéristiques de haut-niveau est [Doddington, 2001]. La modélisation haut-niveau tend à considérer chaque expression comme une séquence de lexèmes, et de se baser après sur des occurrences et des alignements de séquences. Cette modélisation utilise souvent des approches statistiques comme les modèles n-grammes [Jelinek and Mercer, 1980]. Sont utilisés par exemple comme lexèmes, les mots [Doddington, 2001], les phones [Andrews et al., 2002], [Navrátil et al., 2003], [Jin et al., 2003], [Campbell et al., 2004a], et les labels des k-meilleures gaussiennes de modèles GMM [Xiang, 2003], [Ma et al., 2006]. Dans [Leung et al., 2006], il est proposé un modèle de prononciation basé sur des correspondances entre des propriétés articulatoires (le point et le lieu d'articulation) et les séquences de phonèmes, tandis que sont alignées des séquences de phones et de phonèmes dans [Klusáček et al., 2003], pour modéliser la prononciation du locuteur.

La dynamique d'élocution d'une personne contient des informations spécifiques à cette personne, et qui sont utiles à sa reconnaissance. Pour modéliser la dynamique temporelle

(l'évolution) du signal de parole, on calcule le plus souvent les dérivées premières et secondes des paramètres statistiques sur une fenêtre temporelle centrée sur ces paramètres la [Furui, 1981], [Soong and Rosenberg, 1988], et on regroupe l'ensemble de ces paramètres dans un même vecteur acoustique.

2.2.1.2 Segmentation parole / non parole

Les trames résultantes de la phase de paramétrisation ne sont pas toutes utiles au processus de reconnaissance de locuteurs. La paramétrisation est suivie d'une identification *parole / non parole* (Voice Activity Detection) VAD, où le label non parole désigne en réalité toute trame jugée non utile au sens reconnaissance. La VAD peut aussi intervenir avant la paramétrisation.

La VAD tend à éliminer les trames à faible et à moyenne énergie, qui représentent du silence, du bruit ou de l'écho. Ces trames rendent la reconnaissance des locuteurs plus difficiles. On ne garde donc que les trames à haute énergie et qui correspondent principalement aux zones stables des voyelles. En pratique, il est difficile d'avoir une VAD indépendante des différentes conditions d'enregistrement. Par conséquent, on fait une VAD par condition.

Plusieurs techniques de VAD ont été proposées dans la littérature, principalement dans la communauté de la reconnaissance de la parole. Nous citons par exemple les travaux : [Rabiner and Sambur, 1975] utilise l'énergie et le taux de passage par zéro (zero crossing rate) et [Li et al., 2002], [Ying et al., 2011] se servent de l'énergie uniquement ; [Lamel et al., 1981] se base sur la détection de l'impulsion de l'énergie (Energy pulse detection) ; [Tucker, 1992], [Ishizuka and Nakatani, 2006] s'appuient sur la périodicité des trames ; [Shen et al., 1998] se base sur l'entropie ; [Nemer et al., 2001] utilise des statistiques d'ordre supérieur du signal de parole dans le domaine résiduel LPC ; [Ramírez et al., 2004] mesure la divergence spectrale à long terme (long-term spectral divergence) LTSD entre la parole et le bruit et opère la segmentation en comparant l'enveloppe spectrale à long terme avec le spectre moyen du bruit ; [Davis et al., 2006] repose sur une mesure du rapport signal sur bruit.

La segmentation la plus simple et la plus utilisée reste celle basée sur l'énergie du signal. La distribution énergétique des trames est généralement modélisée par un modèle GMM à 2 (ou 3) composantes dont la loi gaussienne à faible (respectivement à haute) énergie modélise les trames à faible (respectivement à haute) énergie. L'apprentissage des lois gaussiennes se solde par le calcul de seuils de décision (deux seuils de décision pour un modèle à 3 composantes) dépendant des paramètres du modèle, qui permet d'assigner les trames à l'une des classes. La figure Fig. 2.4 montre la distribution de la log-énergie des trames d'un signal de parole. La modélisation de la log-énergie se fait en pratique après une normalisation moyenne variance (qui sera décrite dans la section 2.2.1.3).

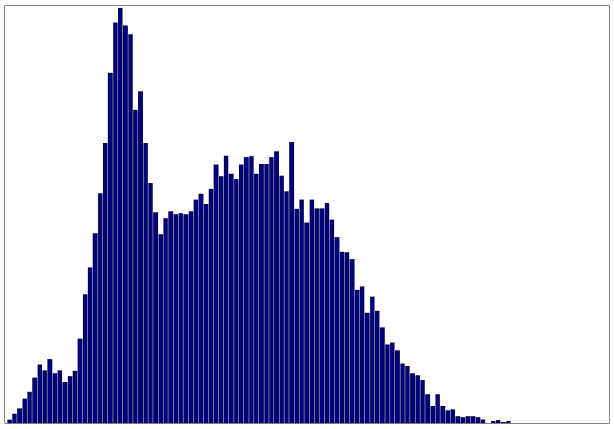


FIG. 2.4: *Distribution énergétique des trames d'un signal de parole.*

2.2.1.3 Normalisation des paramètres acoustiques

Différentes techniques de normalisation des paramètres acoustiques ont été proposées en reconnaissance automatique du locuteur, pour augmenter la robustesse des systèmes face aux variations des conditions d'enregistrement et de transmission. Cette partie en exposera les principales (appliquées aux trames labellisées parole). En pratique, différentes techniques de normalisation sont utilisées ensemble dans les systèmes de reconnaissance du locuteur.

Normalisation moyenne variance cepstrale

La normalisation moyenne variance cepstrale (Cepstral Mean and Variance Normalization) CMVN [Viikki and Laurila, 1998] est une technique très simple et très répondue en reconnaissance automatique du locuteur. Elle consiste à retirer la moyenne de la distribution de chacun des paramètres cepstraux (la composante continue), et à ramener la variance à une variance unitaire en les divisant par l'écart type global des paramètres acoustiques. Quand seule la moyenne est normalisée, on parle alors de la *Cepstral Mean Subtraction* (CMS) ou *Cepstral Mean Normalization* (CMN) [Furui, 1981].

Le passage du domaine temporel aux domaines log-spectral et cepstral, transforme les bruits convolutifs en des bruits additifs. Des bruits convolutifs variant lentement dans le temps seront alors représentés par une composante additive presque constante tout au long de l'enregistrement de parole. Par conséquent, la suppression de la composante continue permet de réduire l'effet de ces bruits. L'estimation de la moyenne et de la variance peut être réalisée sur l'intégralité de la séquence de parole, comme elle peut être faite sur des fenêtres glissantes ce qui permet d'atténuer des variations temporelles plus rapides du bruit. Une fenêtre de 3-5 secondes représente un bon compromis entre une bonne estimation des statistiques et une bonne prise en compte des variations des conditions.

Filtrage RASTA

Le filtrage RASTA (RelAtive SpecTrAl) [Hermansky and Morgan, 1994] a été proposé comme alternative à la CMS. Il exploite le fait que les bruits convolutifs engendrés par le canal de communication, sont généralement à variation beaucoup plus lente que la parole. Cette technique applique donc un filtrage passe-bande (avec une bande passante de 1 à 12Hz) dans le domaine log-spectral (ou cepstral) pour réduire les bruits stationnaires et à lente variation, ainsi que les distorsions à rapide variation (au dessus de 12Hz).

"Gaussianisation"

Le *Feature warping* [Pelecanos and Sridharan, 2001] est une technique qui modifie les paramètres acoustiques afin que leur distributions suivent une autre distribution donnée, typiquement une distribution gaussienne de moyenne nulle et de variance unité. Chaque paramètre est "gaussianisé" séparément, sous l'hypothèse de l'indépendance des paramètres.

La *Short-time Gaussianization* [Xiang et al., 2002] est une autre technique qui se base sur le même principe que le *Feature warping*, mais qui applique cependant une transformation linéaire globale avant la gaussianisation. Cette transformation a pour but de décorréler les paramètres acoustiques, validant ainsi leur indépendance. La *Short-time Gaussianization* donne de bien meilleurs résultats que le *Feature warping*. Néanmoins, son implémentation reste plus difficile à mettre en œuvre.

Feature mapping

La CMVN, le filtrage RASTA et la "gaussianisation" sont toutes des techniques non-supervisées, qui n'utilisent aucune connaissance sur le canal. Le *Feature mapping* (FM) [Reynolds, 2003] est quand à lui une technique supervisée de normalisation qui projète les paramètres acoustiques liés à une condition d'enregistrement donnée dans un nouvel espace de caractéristiques indépendant du canal. Cette transformation permet de réduire les effets de la variabilité du canal, entre conditions d'apprentissage et de test.

Pour ce faire, le FM apprend un modèle GMM pour chaque condition (canal) connue, par adaptation MAP d'un modèle indépendant du canal (un modèle du monde). Le modèle dépendant du canal modélise en réalité un sous-espace de l'espace acoustique global. Il en est déduit une transformation représentant la relation entre le modèle indépendant du canal et le modèle dépendant du canal. Lors de la phase d'apprentissage des modèles, on cherche en premier lieu à identifier le canal le plus vraisemblable pour l'enregistrement traité, et on applique par la suite la transformation qui lui est associée sur les vecteurs paramétriques. L'étape de recherche du canal le plus vraisemblable ne se fait que lorsqu'on ne dispose de cette information, information qui n'est généralement pas disponible.

Signalons finalement que dans [Heck et al., 2000], le réseau de neurones a été utilisé pour transformer les paramètres acoustiques en de nouveaux paramètres plus robustes aux variations du canal. Le réseau de neurones est appris dans l'objectif de maximiser les performances de reconnaissance et la robustesse du système, en minimisant une fonction d'entropie croisée.

2.2.2 Modélisation

Dans les applications de reconnaissance automatique du locuteur dépendantes du texte, la modélisation du locuteur tient compte des dépendances temporelles entre les vecteurs paramétriques extraits. On peut ainsi envisager d'aligner temporellement les séquences de vecteurs d'apprentissage et de test, car elles doivent contenir la même séquence de phones. Néanmoins dans les applications indépendantes du texte, la modélisation tient compte de la seule distribution des paramètres acoustiques. Les techniques de modélisations peuvent dérivées de différentes grandes approches, comme l'approche vectorielle, connexionniste, prédictive et statistique.

2.2.2.1 Approche vectorielle

Dans l'approche vectorielle, les vecteurs paramétriques d'apprentissage et de test sont (directement ou indirectement) comparés, sous l'hypothèse que les vecteurs d'une des séquences sont une réalisation imparfaite des vecteurs de l'autre séquence. La distorsion entre les deux séquences représente leur degré de similarité.

Cette approche compte deux grandes techniques, e.g., l'alignement temporelle (Dynamic Time Warping) DTW [Furui, 1981] et la quantification vectorielle (Vector Quantisation) VQ [Soong et al., 1985], qui ont été respectivement proposées pour les applications dépendantes et indépendantes du texte. La DTW aligne temporellement les suites d'observations, tandis que la VQ représente le locuteur par un dictionnaire de codes.

Quantification vectorielle

La quantification vectorielle décompose l'espace acoustique d'un locuteur donné X, en un ensemble de M sous-espaces représentés par leur vecteurs centroïdes $C = \{c_1, c_2, ..., c_M\}$. Ces vecteurs centroïdes forment un dictionnaire (de taille M) qui modélise ce locuteur, et sont calculés en minimisant l'erreur de quantification moyenne (distorsion) induite par le dictionnaire sur les données d'apprentissage du locuteur $\{x_1, x_2, ..., x_T\}$:

$$D(X, C) = \frac{1}{T} \sum_{t=1}^{T} \min_{1 \leq m \leq M} d(x_t, c_m), \tag{2.4}$$

où $d(.,.)$ est une mesure de distance au sens d'une certaine métrique liée à la paramétrisation. L'apprentissage vise à réduire l'erreur de quantification. On peut mieux représenter

le locuteur en augmentant la taille du dictionnaire, mais le système sera moins rapide et plus demandeur de mémoire. Il faut trouver donc un bon compris. La construction du dictionnaire peut être faite en utilisant par exemple l'algorithme LBG [Linde et al., 1980] ou l'algorithme des K-moyennes (K-means) [MacQueen, 1967]. Lors de la phase de reconnaissance, la décision tient compte de l'erreur de quantification moyenne. Plus l'erreur est faible plus vraisemblablement la séquence de parole a été dite par le locuteur.

Plusieurs variantes d'utilisation de la VQ ont été proposées dans la littérature dont [Soong et al., 1985], [Burton, 1987], [Soong and Rosenberg, 1988] et [He et al., 1999]. Une modélisation VQ-UBM qui combine les concepts de la quantification vectorielle et de l'adaptation MAP a été proposée dans [Hautamäki et al., 2008]. Le système résultant est plus rapide qu'un système classique GMM-UBM. De plus, une comparaison plus détaillée entre ces deux systèmes dans [Kinnunen et al., 2009] montre que le système VQ-UBM donne de bien meilleurs résultats que le système GMM-UBM quand on dispose de relativement beaucoup de données d'apprentissage (tests effectués dans le cadre de la condition principale des campagnes d'évaluation NIST-SRE 2005, 2006 et 2008) et inversement quand on dispose de très peu de données (tests effectués dans le cadre de la condition 10sec-10sec de ces mêmes campagnes d'évaluation).

2.2.2.2 Approche prédictive

Les modèles prédictifs reposent sur le principe qu'une trame d'un signal de parole peut être générée à partir des trames précédentes du signal. Un locuteur donné est représenté par une fonction de prédiction estimée sur ses données d'apprentissage. Deux stratégies peuvent être ensuite adoptées pour la reconnaissance : soit calculer une erreur de prédiction comme mesure de similarité, entre les trames prédites (en utilisant la fonction de prédiction du locuteur concerné) et les trames réellement observées ; soit comparer la fonction de prédiction du locuteur concerné avec une nouvelle fonction de prédiction estimée cette fois ci sur les nouvelles données, selon une mesure de distance donnée.

On a utilisé dans la littérature deux grandes familles de fonctions de prédiction :

- Les modèles Auto-Régressifs vectoriels (AR-Vector Models) ARVM [Grenier, 1980], [Bimbot et al., 1992], [Montacié and Le Floch, 1993], [Griffin et al., 1994], [Magrin-Chagnolleau et al., 1996] qui modélisent l'évolution du spectre du signal par un modèle Auto-Régressif.
- Les réseaux de neurones prédictifs [Hattori, 1992], [Artieres and Gallinari, 1993], [Paoloni et al., 1996], i.e., des réseaux de neurones de type perceptron multicouche utilisés non pour faire de la classification mais plutôt comme modèles prédictifs (on apprend un modèle par locuteur).

2.2.2.3 Approche connexionniste

Un modèle connexionniste, ou modèle neuromimétique ou réseau de neurones artificiels est un modèle discriminant formé d'un grand nombre de cellules élémentaires (neurones)

fortement interconnectées, dont la sortie de chaque neurone est fonction de ses entrées [McCulloch and Pitts, 1943]. Un réseau de neurones est défini par plusieurs paramètres : la topologie des connexions entre les neurones, les fonctions d'agrégation des entrées, et l'algorithme d'apprentissage utilisé.

En reconnaissance automatique du locuteur, les locuteurs sont modélisés par un réseau de neurones appris sur l'ensemble des données d'apprentissage de tous les clients, qui tend à discriminer entre les différents clients du système. Différents types de réseaux de neurones ont été utilisés en reconnaissance automatique du locuteur : le perceptron multicouche (Multi-Layer Perceptron) MLP, le modèle LVQ (Learning Vector Quantization), le modèle RBF (Radial Basis function), le réseau de neurones auto-associatif (autoassociative neural network) AANN ..., sans pour autant réussir à s'imposer parmi les systèmes classiques. Nous renvoyons le lecteur aux travaux [Bennani et al., 1990], [Oglesby and Mason, 1990], [Oglesby and Mason, 1991], [Farrell et al., 1994], [Lapidot et al., 2002], [Yegnanarayana and Kishore, 2002], [Ganchev et al., 2004] pour un aperçu des travaux menés avec les modèles connexionnistes.

2.2.2.4 Approche statistique

Les techniques statistiques considèrent le locuteur comme étant une source probabiliste et le modélisent par une densité de probabilité connue. La phase d'apprentissage consiste à estimer les paramètres de la fonction de densité de probabilité. La décision est prise en calculant la vraisemblance des données par rapport au modèle appris préalablement. Empruntés à la reconnaissance automatique de la parole, les modèles de Markov cachés (Hidden Markov Models) HMM [Rabiner, 1989], [De Veth and Bourlard, 1995] ont été utilisés dans des applications dépendantes du texte. Les Modèles de Mélange de lois Gaussiennes (Gaussian Mixture Models) GMM [Reynolds and Rose, 1995], [Reynolds et al., 2000], un HMM à un seul état, restent les modèles état de l'art utilisés dans la grande majorité des systèmes de reconnaissance du locuteur indépendants du texte. Parmi les modèles les plus utilisés en reconnaissance automatique du locuteur, on trouve également les machines à vecteurs de support (Support Vector Machine) SVM [Louradour, 2007].

Selon l'approche d'apprentissage, on peut aussi classifier les différents modèles en modèles génératifs et modèles discriminants. Les modèles génératifs comme les GMM et les VQ estiment la distribution des paramètres acoustiques de chaque locuteur séparément ; tandis que les modèles discriminants comme les SVM et les réseaux de neurones estiment des frontières entre les locuteurs. Nous détaillerons à présent, les principaux modèles utilisés en reconnaissance automatique du locuteur indépendante du texte.

2.2.2.5 Modèle de mélange de Gaussiennes

Le modèle de mélange de Gaussiennes est un modèle statistique où la distribution des données est un mélange de plusieurs lois Gaussiennes. Le GMM est le modèle de référence en reconnaissance du locuteur [Reynolds and Rose, 1995], [Reynolds et al., 2000].

La $m^{ème}$ loi gaussienne d'un mélange λ à M composantes est paramétrée par un vecteur de moyennes μ_m de dimension D (D étant la dimension de l'espace des données), une matrice de covariance Σ_m de dimension $D \times D$ et un poids $w_m \geq 0$. La fonction de densité de probabilité s'écrit sous forme de :

$$P(x|\lambda) = \sum_{m=1}^{M} w_m \, \mathcal{N}(x|\mu_m, \Sigma_m), \tag{2.5}$$

où

$$\mathcal{N}(x|\mu_m, \Sigma_m) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_m|}} \, exp\Big(-\frac{1}{2} \, (x-\mu_m)^T \, \Sigma_m^{-1} \, (x-\mu_m) \Big), \tag{2.6}$$

et $\sum_{m=1}^{M} w_m = 1$.

L'apprentissage d'un modèle GMM consiste en l'estimation de l'ensemble des paramètres $\lambda = \{\mu_m, \Sigma_m, w_m\}_{m=1}^{M}$, en utilisant un ensemble de données d'apprentissage $X = \{x_1, x_2, ..., x_T\}$ ($x_t \in \mathcal{R}^D$). Cet apprentissage fait souvent appel à la technique d'estimation par maximum de vraisemblance (Maximum Likelihood Estimation) MLE [Fisher, 1925] ; on utilise souvent l'algorithme Espérance-Maximisation (Expectation-maximisation) EM [Dempster et al., 1977], [Bishop, 2006].

Estimation par maximum de vraisemblance

L'algorithme EM est un algorithme itératif sous optimal d'estimation de paramètres de modèles probabilistes selon le critère du maximum de vraisemblance :

$$\lambda^{MLE} = \underset{\lambda}{argmax} \; P(X|\lambda). \tag{2.7}$$

Dans la pratique, on cherche à maximiser la log-vraisemblance des données X par rapport au modèle λ, qui est définie comme :

$$LL_{avg}(X|\lambda) = \frac{1}{T} \sum_{t=1}^{T} \log P(x_t|\lambda). \tag{2.8}$$

Chacune des itérations de l'algorithme comporte deux étapes :

– L'étape d'espérance (E), où on calcule les probabilités a posteriori que les gaussiennes aient généré les données d'apprentissage :

$$P(m|x_t) = \frac{w_m \mathcal{N}(x_t|\mu_m, \Sigma_m)}{\sum_{g=1}^{M} w_g \mathcal{N}(x_t|\mu_g, \Sigma_g)}. \tag{2.9}$$

– L'étape de maximisation (M), où on ré-estime les paramètres du modèle afin de maximiser la vraisemblance :

$$\begin{aligned} w_m &= \frac{1}{T}\sum_{t=1}^{T} P(m|x_t), \\ \mu_m &= \frac{\sum_{t=1}^{T}\Big(P(m|x_t)x_t\Big)}{\sum_{t=1}^{T} P(m|x_t)}, \\ \Sigma_m &= \frac{\sum_{t=1}^{T}\Big(P(m|x_t)\big(x_t-\mu_m\big)\big(x_t-\mu_m\big)^T\Big)}{\sum_{t=1}^{T} P(m|x_t)}. \end{aligned} \tag{2.10}$$

Le critère d'arrêt de l'algorithme est défini à partir de la variation de la vraisemblance ou l'atteinte d'un nombre maximum d'itérations. L'algorithme EM assure une croissance monotone de la vraisemblance jusqu'à convergence vers un maximum local. L'initialisation de l'algorithme EM peut être effectuée efficacement à l'aide de la méthode des K-moyennes [MacQueen, 1967].

Les GMM sont généralement à matrices de covariance diagonales. L'apprentissage de matrices de covariance pleines augmente considérablement le nombre de paramètres du modèle, compliquant ainsi leur estimation.

Approche GMM-UBM

L'estimation par maximum de vraisemblance nécessite une grande quantité de données, pour estimer robustement les paramètres d'un modèle GMM. Dans le cas où la quantité de données n'est pas suffisante pour un apprentissage "direct" du GMM, on utilise des méthodes d'adaptation de modèles. En reconnaissance automatique du locuteur, peu de données sont généralement disponibles pour apprendre directement ces modèles. Un modèle GMM du monde ou UBM (Universal Background Model) à matrices diagonales est ainsi appris par l'algorithme EM sur des centaines voire des milliers d'heures d'enregistrements appartenant à plusieurs locuteurs et dans différentes conditions d'enregistrement. Ensuite, le modèle d'un locuteur est appris par adaptation (généralement par la méthode de Maximum A Posteriori (MAP) [DeGroot, 1970], [Gauvain and Lee, 1994]) de l'UBM aux données de ce locuteur [Reynolds et al., 2000]. Les systèmes état de l'art de reconnaissance automatique du locuteur exploitent l'information du genre du locuteur en apprenant deux modèles du monde dépendants du genre. L'adaptation tient compte

donc de cette information. Le modèle estimé selon le critère MAP s'écrit comme :

$$\lambda^{MAP} = \underset{\lambda}{argmax}\ P(X|\lambda)P(\lambda). \tag{2.11}$$

On peut adapter l'ensemble des paramètres de l'UBM, comme on peut se limiter à n'en adapter que certains. On a montré dans [Reynolds et al., 2000] que l'adaptation des moyennes uniquement donne de bons résultats. Les matrices de covariance (diagonales) et les poids restent inchangés. Notons respectivement par $\lambda_{UBM} = \{\mu_{Um}, \Sigma_m, w_m\}_{m=1}^{M}$ et $\lambda_c = \{\mu_{cm}, \Sigma_m, w_m\}_{m=1}^{M}$, le modèle du monde et le modèle d'un locuteur c. Les vecteurs de moyennes adaptés selon le critère MAP, s'écrivent sous forme de :

$$\mu_{cm} = \alpha_m E_m(x) + (1-\alpha_m)\mu_{Um}, \tag{2.12}$$

où

$$\begin{aligned} \alpha_m &= \frac{n_m}{n_m + r}, \\ E_m(x) &= \frac{1}{n_m}\sum_{t=1}^{T}\Big(P(m|x_t)x_t\Big), \\ n_m &= \sum_{t=1}^{T} P(m|x_t), \end{aligned} \tag{2.13}$$

et r un facteur de régulation qui règle l'équilibre entre la distribution a priori et l'importance accordée aux données d'adaptation.

Durant la phase de reconnaissance, la décision se base sur les log-vraisemblances des données. Dans une application d'identification, on cherche parmi l'ensemble des clients du système, le modèle (le locuteur) qui maximise la log-vraisemblance des données de test $\acute{X} = \{x_1, x_2, ..., x_T\}$:

$$Identité = \underset{c}{argmax}\ LL_{avg}(\acute{X}|\lambda_c). \tag{2.14}$$

La tâche de vérification utilise conjointement à la fois le modèle de l'identité clamée i (appris par adaptation MAP) et le modèle du monde. Ce système est communément appelé GMM-UBM. Le score d'appariement (de vérification) s'exprime par la formule suivante :

$$LLR_{avg} = \frac{1}{T}\sum_{t=1}^{T}\Big(\log P(x_t|\lambda_i) - \log P(x_t|\lambda_{UBM})\Big). \tag{2.15}$$

Le module de décision compare (la moyenne du) logarithme du rapport des vraisemblances par rapport à un seuil de décision Θ :

- Si $LLR_{avg} > \Theta$ alors le locuteur i parle.
- Sinon, un autre locuteur parle.

Cette métrique reflète la différence d'implication des deux modèles (locuteur / "non-locuteur") dans la génération des observations du test, et peut être interprétée comme une normalisation des log-vraisemblances de différents locuteurs cibles par rapport à un même modèle non-locuteur (représentant l'hypothèse inverse), ce qui permet finalement de comparer différents scores par rapport à un même seuil de décision.

En réalité, les systèmes GMM de l'état de l'art effectuent la décision en ne tenant compte que des k-meilleures gaussiennes et non pas de l'ensemble des gaussiennes. Il a été observé dans [Reynolds et al., 2000], que peu de gaussiennes contribuent significativement dans la valeur de la vraisemblance. La vraisemblance peut être bien approximée en se limitant uniquement aux k-meilleures gaussiennes.

Autres approches gaussiennes

Une autre technique d'adaptation issue de la reconnaissance automatique de la parole a été expérimentée en reconnaissance du locuteur. Des systèmes basés sur la méthode d'adaptation par régression linéaire avec maximisation de la vraisemblance (Maximum likelihood linear regression) MLLR ont été proposés dans [Mariéthoz and Bengio, 2002], [Stolcke et al., 2005], [Mak et al., 2006].

La MLLR [Leggetter and Woodland, 1995] transforme les vecteurs moyennes du modèle du monde μ_{Um} en les vecteurs adaptés μ_{cm}, par le biais d'une transformation linéaire (à paramètres A et b) commune à toute les gaussiennes du modèle (de la classe c) :

$$\forall m, \qquad \mu_{cm} \; = \; \mathrm{A}\mu_{Um} \; + \; \mathrm{b}. \tag{2.16}$$

Les paramètres A et b (qu'on regroupe en une seule matrice de transformation) sont appris sur les données d'adaptation par un algorithme type EM. Généralement, seules les moyennes sont adaptées. Mais lorsqu'on a suffisamment de données d'adaptation, les variances peuvent être également adaptées, soit indépendamment des moyennes ou soit en utilisant la même matrice d'adaptation que pour les moyennes, on parle dans ce dernier cas d'une *Constrained MLLR* (CMLLR) [Gales, 1998]. Les résultats montrent que cette technique d'adaptation donne de bons résultats quand on dispose de peu de données d'adaptation (de l'ordre de quelques secondes).

D'un autre coté, des travaux en reconnaissance automatique du locuteur ont utilisé l'information phonémique dans la modélisation des clients du système, en se basant sur des modèles GMM appris par classe de phonèmes ou de syllabes. Nous renvoyons le lecteur par exemple aux travaux [Faltlhauser and Ruske, 2001], [Park and Hazen, 2002], [Hebert and Heck, 2003], [Chaudhari et al., 2003], [Hansen et al., 2004], [Bocklet and Shriberg, 2009].

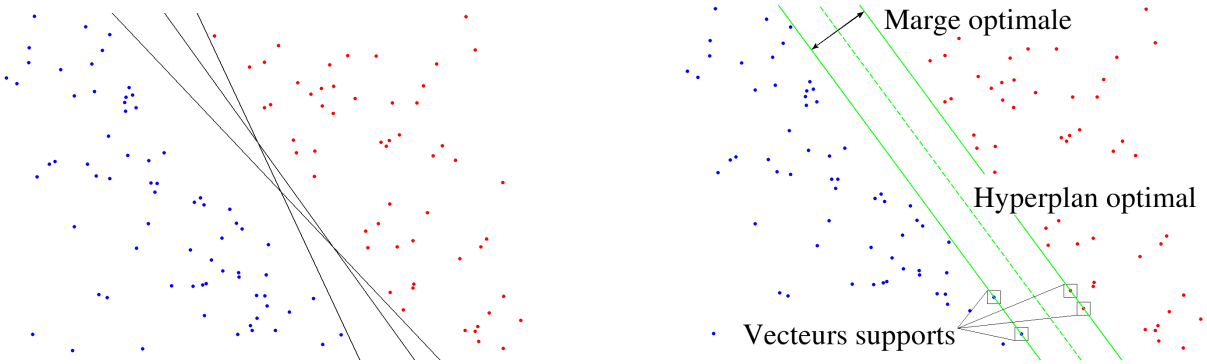
2.2.2.6 Machine à vecteurs de support

La machine à vecteurs de support (Support Vector Machine) SVM est un classifieur discriminant qui sépare deux classes (ayant comme labels $+1$ et -1) par un hyperplan de séparation [Cortes and Vapnik, 1995], [Vapnik, 1998]. C'est une technique très utilisée en classification et régression, grâce à son bon pouvoir de généralisation.

Généralités

Pour un ensemble de vecteurs paramétriques de deux classes (linéairement séparables), il existe une multitude d'hyperplans séparateurs qui séparent les données de ces deux classes. Mais seulement un seul de ces hyperplans maximise la marge entre les données et la frontière de séparation. La marge étant la distance entre l'hyperplan et les données les plus proches (appelées vecteurs supports). Cet hyperplan optimal donne les meilleures performances en généralisation. Disposant de données d'apprentissage $\{x\}$ (de classe ± 1), le problème dans la modélisation par SVM est de trouver l'hyperplan optimal.

La figure Fig. 2.5 montre le principe de l'hyperplan optimal et de la marge optimale dans la modélisation par SVM.



(a) Hyperplans séparateurs (non optimaux)

(b) Hyperplan optimal

FIG. 2.5: *Principe des machines à vecteurs de support.*

La plupart des applications pratiques correspondent en réalité à des classes non linéairement séparables, où il n'existe aucune frontière de séparation linéaire capable de séparer les données des deux classes. Le problème est dit non linéairement séparable. La figure Fig. 2.6 donne un exemple de ce type de problèmes. Afin de pouvoir traiter le cas de données non linéairement séparables, le principe des SVM est de transposer le problème dans un espace de dimension supérieure, où il est probable de trouver une frontière de séparation linéaire (un hyperplan de séparation). Ceci est réalisé grâce à l'utilisation d'une fonction noyau (kernel function) respectant les conditions de Mercer. La fonction noyau s'exprime sous forme de $\mathrm{K}(x, y) = \phi(x)^T \phi(y)$, où $\phi(x)$ est une expansion de l'espace de données

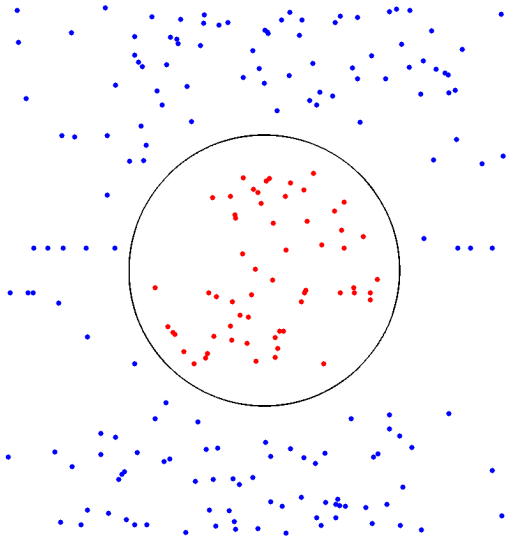


FIG. 2.6: *Classes non linéairement séparables.*

(l'espace d'entrée) à l'espace des caractéristiques de plus haute dimension (l'espace de redescription). La fonction discriminante de la SVM est donnée par :

$$\begin{aligned} f(x) &= \sum_{i=1}^{L} \alpha_i y_i \mathrm{K}(x, x_i) + b \\ &= \left(\sum_{i=1}^{L} \alpha_i y_i \phi(x_i) \right)^T \phi(x) + b = W^T \phi(x) + b, \end{aligned} \tag{2.17}$$

où $y_i \in \{+1, -1\}$ sont les sorties (les labels) idéales, $\sum_{i=1}^{L} \alpha_i y_i = 0$ et $\alpha_i > 0$. Notons que l'équation de l'hyperplan de séparation $H(W, b)$ est $f(x) = 0$. Le modèle SVM, i.e., les L vecteurs supports x_i, leur poids correspondants α_i et le paramètre b (un décalage), est obtenu en résolvant un problème d'optimisation quadratique. Durant la phase de reconnaissance, le score du modèle SVM f_{x} sur les données y est :

$$f_{\mathrm{x}}(y) = W_{\mathrm{x}}^T \phi(y) + b_{\mathrm{x}}. \tag{2.18}$$

Étant de base des classifieurs binaires, deux principales approches ont été proposées pour utiliser les SVM dans la séparation multi-classes : l'approche une contre toutes (one against all) et l'approche par paires (one against one, appelée aussi pairwise) [Hsu and Lin, 2002]. On apprend dans la première, C SVM (C étant le nombre de classes) séparant chacune entre une classe donnée et l'ensemble des autres classes. Alors qu'on apprend dans la deuxième, $C(C-1)/2$ SVM séparant chacune les deux classes de toute paire (de classes) possible, et construire ensuite un arbre de décision à partir de ces SVM.

Utilisation des SVM en reconnaissance automatique du locuteur

En reconnaissance automatique du locuteur, le modèle SVM est appris en utilisant les données d'un client (ayant comme label +1) et des données appartenant à d'autres locuteurs (ayant le label −1). Les données utilisées peuvent être soit des séquences de vecteurs paramétriques où soit issues d'une modélisation des locuteurs (la combinaison des SVM avec d'autres techniques de modélisation). Cette deuxième utilisation des SVM sera abordée dans la section 2.2.2.7.

Des SVM apprises sur des séquences de vecteurs paramétriques ont été utilisées dans la littérature, avec différentes fonctions du noyau et avec toutes sortes de paramètres acoustiques, e.g., paramètres spectraux [Schmidt and Gish, 1996], [Wan and Campbell, 2000], [Campbell, 2002], [Wan and Renals, 2002], [Staroniewicz and Majewski, 2004], [Campbell et al., 2006a], paramètres prosodiques [Shriberg et al., 2005], et paramètres de haut-niveau [Campbell et al., 2004a]. Ainsi par exemple :

- Des noyaux polynomiaux (non normalisés et normalisés) et un noyau radial gaussien (Radial Basis Function) RBF ont été évalués dans [Wan and Campbell, 2000].
- Des noyaux incorporant le principe d'alignement dynamique ont été proposés dans [Shimodaira et al., 2001], [Wan and Carmichael, 2005] pour des applications dépendantes du texte.
- Un noyau (Generalized Linear Discriminant Sequence) GLDS basé sur une expansion polynomiale de vecteurs, a été proposé dans [Campbell, 2002], [Campbell et al., 2006a].
- Un noyau linéaire a été utilisé dans [Shriberg et al., 2005].
- Une nouvelle famille de noyaux est proposée dans [Louradour et al., 2007], les noyaux FSMS (Feature Space Mahalanobis Sequence).

2.2.2.7 Modèles hybrides

Modèles d'ancrage

Les modèles d'ancrage ont été introduits dans [Merlin et al., 1999] afin de rendre les tâches d'indexation et de reconnaissance des locuteurs plus rapides. C'est une technique de représentation "relative" des clients du système par rapport à un ensemble de modèles de références bien appris précédemment, appelés modèles d'ancrage (anchor models).

Dans l'espace des modèles d'ancrage, un locuteur est caractérisé par un vecteur V (de dimension N) contenant les scores de vraisemblance de ses données d'apprentissage X, par rapport aux N modèles d'ancrage $\{\lambda_{Ai}\}_{i=1}^{\mathrm{N}}$ définissant l'espace :

$$\mathrm{V} = \begin{bmatrix} LLR_{avg}(X|\lambda_{A1}) \\ \vdots \\ LLR_{avg}(X|\lambda_{A\mathrm{N}}) \end{bmatrix}. \tag{2.19}$$

Durant la phase de test, les données de test sont projetées dans l'espace des modèles d'ancrage, et une mesure de similarité (une distance) entre vecteurs de cet espace donnera le score d'évaluation.

Le choix des modèles d'ancrage est crucial. Les locuteurs de référence doivent être choisis de façon à avoir le meilleur espace de représentation. Pour plus de détails sur cette technique, nous renvoyons le lecteur aux travaux [Mami and Charlet, 2006], [Naini et al., 2010]. Dans [Collet et al., 2005], une autre utilisation des modèles d'ancrage a été proposée. Les auteurs présentent une approche statistique dans l'espace des modèles d'ancrage, qui consiste à modéliser le locuteur par une distribution normale dans cet espace.

Combinaison des SVM avec des GMM

La deuxième utilisation des SVM en reconnaissance automatique du locuteur consiste en leur combinaison avec les GMM (elles ont aussi été combinées avec des HMM dans des applications dépendantes du texte). Des premiers travaux ont appris des modèles SVM sur de nouveaux paramètres qui sont en fonction de scores de vraisemblances GMM, i.e., on fait un post-traitement sur des scores GMM par les SVM [Bengio and Mariéthoz, 2001], [Fine et al., 2001b], [Kharroubi et al., 2001], [Le and Bengio, 2003], [Liu et al., 2006]. Dans [Krause and Gazit, 2006], des SVM ont été appris sur les différences entre les vecteurs de moyennes du modèle GMM du client et de ceux de l'UBM. D'autres travaux se sont appuyés sur le score de Fisher $U_\lambda(X) = \nabla_\lambda \log P(X|\lambda)$ qui transforme une séquence de données X en un seul vecteur dans l'espace du gradient de la log-vraisemblance, pour apprendre des SVM dans ce nouvel espace en utilisant des fonctions du noyau fondées sur les scores de Fisher, e.g., un noyau de Fisher dans [Fine et al., 2001a] et des noyaux basés sur la divergence de Kullback-Leibler (KL) dans [Moreno and Ho, 2003], [Ho and Moreno, 2004], [Dehak and Chollet, 2006]. Aussi, une généralisation du noyau de Fisher a été proposée dans [Wan and Renals, 2005].

On a proposé ensuite dans [Campbell et al., 2006b], [Campbell et al., 2006c], une toute autre approche considérant directement les paramètres d'un modèle GMM comme vecteur d'observation de la SVM. On construit un supervecteur GMM (de taille $M \times D$) en regroupant les M vecteurs de moyennes (de dimension D) du GMM appris par adaptation MAP. Cette représentation peut être considérée comme une expansion de séquences de vecteurs (de tailles variables) en un seul vecteur de haute dimension (de taille fixe) via la modélisation par GMM :

$$\phi(X) = \begin{bmatrix} \mu_{x1} \\ \vdots \\ \mu_{xM} \end{bmatrix}, \tag{2.20}$$

où le GMM $\{\mu_{xm}, \Sigma_m, w_m\}$ a été appris sur la séquence X. Les supervecteurs GMM s'accordent bien avec la philosophie des SVM. [Campbell et al., 2006b], [Campbell et al., 2006c]

proposent donc d'utiliser un noyau linéaire dans l'espace des supervecteurs :

$$\mathrm{K}(X,Y) = \sum_{m=1}^{M} \left(\sqrt{w_m}\Sigma_m^{-(1/2)}\mu_{xm}\right)^T \left(\sqrt{w_m}\Sigma_m^{-(1/2)}\mu_{ym}\right). \quad (2.21)$$

Les poids et les variances des gaussiennes servent à normaliser les vecteurs de moyennes avant l'apprentissage. Ce système *GMM Supervector Linear Kernel* (GSL) (appelé aussi dans la littérature *Gaussian Supervector SVM* (GSV-SVM) et GMM-SVM) est parmi les techniques les plus performantes en reconnaissance automatique du locuteur. Pour plus de détails sur l'utilisation des SVM en reconnaissance automatique du locuteur, nous conseillons [Louradour, 2007].

Dans [Karam and Campbell, 2007], [Stolcke et al., 2007], les auteurs se servent des paramètres de la transformation MLLR pour former des supervecteurs MLLR qui serviront à l'apprentissage de modèles SVM. Un travail similaire dans [Zhu et al., 2008] caractérise un locuteur par une transformation de caractéristiques (Feature Transformation) FT, qui représente le locuteur par sa différence par rapport à un ensemble donné de locuteurs. La FT est une fonction de régression linéaire (une transformation affine) qui transforme les données (dépendantes) du locuteur en des données indépendantes du locuteur. La FT consiste en deux ensembles de paramètres : des vecteurs de biais (bias vectors) et des matrices de transformation, représentant respectivement l'information du premier et du second ordre. Les vecteurs de biais et les matrices de transformation sont assignés séparément aux deux classes de régression, ce qui rend la FT plus flexible que la CMLLR. Les paramètres de la FT sont estimés selon un critère MAP. Ces paramètres sont ensuite regroupés en supervecteurs qui serviront finalement à l'apprentissage des modèles SVM. Dans [Zhu et al., 2009], une extension de cette technique a été proposée, qui utilise une adaptation MAP jointe des paramètres de la FT et des paramètres GMM. L'ensemble de ces paramètres formeront après les supervecteurs.

2.2.3 Normalisation des scores

La variabilité inter-sessions induit dans la phase de test une variabilité des scores de vérification. Cependant le seuil de décision qui est fixé empiriquement lors de la phase de développement, est commun à toutes les conditions de test rencontrées et il est indépendant du locuteur. De ce fait, on a introduit des techniques de normalisation des scores pour renforcer la robustesse des systèmes de reconnaissance. Ces techniques permettent d'atténuer la variabilité des scores (non compensée lors de la paramétrisation et modélisation), rendant finalement différents scores comparables. Elles se basent sur l'analyse des distributions des scores des clients et des imposteurs.

Généralement, la normalisation suis la forme suivante :

$$\check{s} = \frac{s-\mu_I}{\sigma_I}, \quad (2.22)$$

où $\check{s}$ est le score normalisé, s le score original, et μ_I et σ_I sont respectivement la moyenne et l'écart type des scores imposteurs. Les techniques les plus couramment utilisées sont la

zero normalization (Z-norm) et la *test normalization* (T-norm) [Auckenthaler et al., 2000], et se différencient par l'estimation des μ_I et σ_I. Leurs combinaisons, une Z-norm suivie par une T-norm et inversement, sont respectivement appelées la ZT-norm et la TZ-norm.

2.2.3.1 Z-norm

La Z-norm estime μ_I et σ_I en calculant les scores d'appariement entre le locuteur cible et un ensemble d'énoncés dites par des personnes imposteurs. C'est une normalisation dépendante du locuteur, qui ne nécessite pas la connaissance des énoncés de test. Le calcul de μ_I et σ_I peut se faire donc avant la phase d'évaluation.

Une variante de la Z-norm, appelée H-norm pour *handset normalization*, a été proposée dans [Reynolds, 1997]. Cette technique de normalisation prend en compte l'information du type du canal.

2.2.3.2 T-norm

La T-norm utilise les énoncés des locuteurs imposteurs pour leur apprendre des modèles. On estime μ_I et σ_I en calculant les scores d'appariement entre ces modèles des imposteurs et le segment de test. C'est une normalisation dépendante de l'énoncé de test, qui ne peut se faire que durant la phase d'évaluation. La T-norm permet de compenser les variations des conditions d'enregistrement de l'énoncé.

Des expériences menées dans [Barras and Gauvain, 2003] ont montré que la Z-norm et T-norm ont des effets de rotation sur la courbe DET qui se font dans deux directions opposées. Quand on a un point de fonctionnement du système qui se situe dans la zone à faibles taux de fausses acceptations, la T-norm améliore les performances de vérification. Alors que la Z-norm améliore les performances du système, quand on a un point de fonctionnement se situant dans la zone à faibles taux de faux rejets.

2.2.3.3 S-norm

La *symmetric normalization* (S-norm) [Kenny, 2010] est une technique récente de normalisation qui normalise le score original en appariant à la fois l'information d'apprentissage et de test avec la cohorte (la liste) d'imposteurs :

$$\check{s} \;=\; \frac{s-\mu_I^a}{\sigma_I^a} + \frac{s-\mu_I^t}{\sigma_I^t}, \tag{2.23}$$

où les statistiques des scores imposteurs μ_I^a et σ_I^a invoquent l'information d'apprentissage, tandis que μ_I^t et σ_I^t invoquent celle du test.

2.2.3.4 Normalisation adaptative

Les techniques de normalisation sont d'autant plus performantes quand la cohorte d'imposteurs est bien choisie ; on parle de normalisation adaptative. Pour une séquence

de test ou un modèle client donné, on sélectionne respectivement les modèles imposteurs ou les séquences imposteurs qui donnent les plus grands scores d'appariement.

L'*Adaptive T-norm* (AT-norm) a été proposée dans [Sturim and Reynolds, 2005]. On recherche à partir d'un grand ensemble de modèles imposteurs, ceux qui sont similaires au modèle client. Les modèles sélectionnés partagent des caractéristiques communes avec le locuteur cible (le sexe, l'âge, le type du canal ...), ce qui permet d'avoir des scores imposteurs d'appariement proches à ceux du client.

Les techniques adaptatives peuvent se combiner avec les autres techniques de normalisation. Par exemple faire une Z-norm suivie d'une AT-norm pour une ZAT-norm, ou encore une S-norm avec une AS-norm pour une SAS-norm.

2.2.4 Compensation de la variabilité inter-sessions

On observe une baisse considérable des performances quand les canaux d'enregistrement et de transmission changent entre les sessions d'apprentissage et de test. Le passage à l'espace des supervecteurs rend possible la directe représentation et compensation de toute variabilité indésirable des supervecteurs GMM. Toute différence en un ensemble de supervecteurs GMM représentant un même locuteur, ne résulte que des variabilités intra-locuteur et inter-sessions. Les formalismes de l'analyse de facteur (Factor Analysis) et de la variabilité totale (Total Variability) ont été introduits pour compenser ces variabilités dans les systèmes basés sur les GMM, tandis que les méthodes de compensation *Nuisance Attribute Projection* (NAP) et *Within-class covariance normalization* (WCCN) ont été proposées pour les systèmes à base de SVM.

2.2.4.1 Compensation NAP

La NAP est une méthode de compensation qui essaye de compenser la variabilité des supervecteurs, avant l'apprentissage des modèles SVM [Solomonoff et al., 2005]. Cette technique de pré-traitement peut être considérée comme une normalisation des supervecteurs, avant leur modélisation par des SVM [Campbell et al., 2006c]. La NAP peut être appliquée à tout supervecteur SVM, i.e., à tout vecteur de donnée servant à l'apprentissage d'une SVM indépendamment du noyau utilisé. La méthode suppose que l'indésirable variabilité peut être efficacement estimée dans un espace de très haute dimension, en utilisant des statistiques du second ordre (matrice de covariance). Elle suppose aussi que cette variabilité se concentre dans un sous-espace engendré par les vecteurs propres de la matrice de covariance. Il suffit donc d'estimer et de retirer ce sous-espace pour compenser la variabilité. Notons par la suite (h, s) la session (l'enregistrement) h du locuteur s.

La NAP transforme un supervecteur ϕ en un supervecteur compensé $\hat{\phi}$:

$$\hat{\phi} \;=\; \phi \;-\; \mathbf{S}(\mathbf{S}^T\phi), \tag{2.24}$$

en utilisant la matrice des canaux propres (eigenchannel matrix) $\mathbf{S}$, qui est apprise en utilisant plusieurs enregistrements (sessions) par locuteur. Disposant d'un ensemble d'en-

registrements représentés par leur supervecteurs associés :

$$\{\phi(1,s_1)\cdots\phi(h_1,s_1)\cdots\phi(1,s_N)\cdots\phi(h_N,s_N)\}, \tag{2.25}$$

de N locuteurs s_i, ayant chacun h_i différentes sessions, la NAP commence par supprimer la variabilité des locuteurs en enlevant la moyenne du sous-ensemble de supervecteurs appartenant à chaque locuteur $\{\overline{\phi(s_i)}\}$:

$$\forall s_i, \forall h, \qquad \phi(h,s_i) \;=\; \phi(h,s_i) \;-\; \overline{\phi(s_i)}. \tag{2.26}$$

Les supervecteurs résultants sont ensuite regroupés en une matrice unique :

$$\mathbf{C} \;=\; \big[\phi(1,s_1)\cdots\phi(h_1,s_1)\cdots\phi(1,s_N)\cdots\phi(h_N,s_N)\big], \tag{2.27}$$

qui représente les variations inter-sessions. La NAP cherche après à identifier le sous-espace de dimension R où les variations sont les plus larges (les principales directions de la variabilité), en résolvant le problème aux valeurs propres (eigenvalue problem) associé à la matrice de covariance $\mathbf{CC}^T$. On obtient donc finalement la matrice de projection $\mathbf{S}$ de taille $MD \times R$.

Dans [Vogt et al., 2008], une NAP discriminante (SD-NAP) a été proposée. Ce travail part du fait que la NAP standard peut supprimer une information (variabilité) spécifique au locuteur. La SD-NAP vise alors à garder cette "désirable" variabilité, en utilisant un critère basé sur la *scatter difference analysis*. Les résultats montrent une modeste amélioration des performances par rapport à la NAP standard.

Le système SVM (à base de supervecteurs GMM) intégrant la méthode de compensation NAP sera appelé par la suite GSL-NAP.

2.2.4.2 Normalisation WCCN

La WCCN est une autre méthode de normalisation des supervecteurs SVM, similaire à la NAP [Hatch and Stolcke, 2006], [Hatch et al., 2006]. Cette méthode utilise les statistiques d'ordre 1 et 2 de chaque classe (locuteur) pour construire un ensemble de bornes supérieures des taux de faux positifs et de faux négatifs (les erreurs de classification), pour les minimiser par rapport aux paramètres d'un classificateur linéaire.

Pour un noyau linéaire généralisé de la forme $\mathrm{K}(x,y) \;=\; x^T\mathrm{R}y$, où R est une matrice semidéfinie positive, la WCCN utilise la matrice de transformation (normalisation) $\mathrm{R} \;=\; \mathrm{W}^{-1}$, où W est une sommation pondérée des matrices de covariance intra-classe des locuteurs :

$$\mathrm{W} \;\triangleq\; \sum_{i=1}^{C} P(i)\mathrm{V}_i, \tag{2.28}$$

où $P(i)$ et V_i sont respectivement la probabilité a priori et la matrice de covariance de la classe i (C étant le nombre de classes).

La WCCN vise à identifier les directions orthonormales dans l'espace des caractéristiques qui maximisent l'information pertinente pour la classification, i.e., qui minimisent les erreurs de classification, tout en leur associant des poids optimaux. Et c'est ce qui fait la différence avec la NAP, qui supprime complètement certaines dimensions en projetant les supervecteurs dans un sous-espace à variabilité maximale ; la WCCN garde quand à elle toutes les dimensions, mais en leurs associant des poids différents.

Vu sa complexité calculatoire (l'estimation et l'inversion de la matrice W), la WCCN s'applique en combinaison avec l'analyse en composantes principales (Principal Component Analysis) PCA, dans les applications à grand volume de données.

2.2.4.3 Analyse de facteur

L'analyse jointe de facteur (Joint Factor Analysis) JFA est l'une des techniques phares de la compensation des variations entre conditions (séquences) d'apprentissage et de test. L'avantage de cette technique est de proposer des outils puissants pour modéliser la variabilité due au locuteur et au canal. Ce formalisme conjugue les deux méthodes de voix propres (Eigenvoices) et de canaux propres (Eigenchannels).

Voix propres

La première méthode cherchant à exprimer la variabilité inter-locuteurs dans un espace à dimension réduite est celle des voix propres [Kuhn et al., 2000], [Kenny et al., 2005a]. Comme leur homologues visages propres proposées initialement en reconnaissance de visages, les voix propres peuvent être obtenues à partir d'une analyse PCA. Dans l'espace des supervecteurs, le supervecteur du locuteur s $\mathbf{M}_s$ (de taille MD) s'écrit sous forme de :

$$\mathbf{M}_s \;=\; \mathbf{M} \;+\; \mathbf{V}\mathbf{y}_s, \tag{2.29}$$

où $\mathbf{M}$ est le supervecteur de l'UBM (de taille MD aussi), $\mathbf{V}$ est la matrice de faible rang ($MD \times R_v$) caractérisant la variabilité inter-locuteurs et $\mathbf{y}_s$ sont les facteurs du locuteur (speaker factors) ; un vecteur réduit de taille $R_v \times 1$.

Analyse jointe de facteur

Le formalisme de la JFA [Kenny et al., 2005b], [Kenny et al., 2006], [Kenny et al., 2007b] tend à modéliser la variabilité inter-locuteurs et à compenser la variabilité inter-sessions en décomposant le modèle du locuteur en deux composantes, l'une dépendante du locuteur et l'autre dépendante du canal :

$$\mathbf{M}_{(h,s)} \;=\; \mathbf{S} \;+\; \mathbf{C}, \tag{2.30}$$

où

$$\mathbf{S} \;=\; \mathbf{M} \;+\; \mathbf{V}\mathbf{y}_s \;+\; \mathbf{D}\mathbf{z}_s, \tag{2.31}$$

et

$$\mathbf{C} = \mathbf{U}\mathbf{x}_{h,s}. \tag{2.32}$$

$\mathbf{D}$ est une matrice diagonale de taille $MD \times MD$. Le terme $\mathbf{D}\mathbf{z}_s$ est un résidu qui compense le fait qu'en pratique, on risque de ne pas estimer d'une manière fiable la matrice de faible rang $\mathbf{V}$. Ce terme décrit ainsi la variabilité restante des locuteurs. Les composantes du vecteur $\mathbf{z}_s$ sont appelées les facteurs communs (common factors). Les vecteurs $\mathbf{y}_s$ et $\mathbf{z}_s$ sont des vecteurs aléatoires indépendants qui suivent une distribution normale. La matrice de faible rang $\mathbf{U}$ est la matrice du canal (de taille $MD \times R_u$), qui génère le sous-espace vectoriel associé aux variations du canal (session), dont les vecteurs colonnes sont les canaux propres. $\mathbf{x}_{h,s}$ est le vecteur des facteurs du canal (channel factors) ; un vecteur de taille $R_u \times 1$ qui suit une distribution normale.

Les matrices $\mathbf{V}$, $\mathbf{U}$ et $\mathbf{D}$ sont appelées hyperparamètres du modèle JFA. Elles sont estimées en amont de l'évaluation sur de très grandes quantités de données réunissant plusieurs sessions par locuteur, et ceci pour différents locuteurs. On commence par estimer la matrice $\mathbf{V}$ en supposant que les deux autres matrices sont nulles. Étant donné la matrice $\mathbf{V}$, on estime ensuite la matrice $\mathbf{U}$ tout en continuant à considérer la matrice $\mathbf{D}$ nulle. Et on finit par estimer la matrice $\mathbf{D}$, connaissant les deux autres matrices. Nous renvoyons le lecteur aux [Kenny et al., 2007a], [Kenny et al., 2008], pour une description détaillée de la procédure d'estimation des hyperparamètres du formalisme JFA. Théoriquement, la matrice $\mathbf{S}$ de la NAP est similaire à la matrice du canal $\mathbf{U}$. Et les deux demandent une riche quantité de données.

Durant la phase d'apprentissage, les facteurs latents $\mathbf{x}_{h,s}$ et $\mathbf{y}_s$ sont conjointement estimés, avant l'estimation ensuite de $\mathbf{z}_s$. Finalement le supervecteur du canal $\mathbf{C}$ est supprimé et le supervecteur du locuteur $\mathbf{S}$ est utilisé comme modèle du locuteur. Diverses manières de scoring ont été proposées dans la littérature. Une comparaison de ces méthodes a été faite dans [Glembek et al., 2009].

Analyse latente de facteur

Lorsque seule la décomposition en Eigenchannels est réalisée (appelée LFA pour Latent Factor Analysis, connue aussi sous le nom de SFA pour Symmetrical "Latent" Factor Analysis), seule la variabilité inter-sessions est estimée.

SFA décompose le supervecteur de la session h du locuteur s $\mathbf{M}_{(h,s)}$ en trois composantes : une indépendante du locuteur et de la session (introduite par l'utilisation de l'UBM), une dépendante du locuteur, et une composante dépendante de la session. Les données d'apprentissage du locuteur, spécifiques au locuteur, mais aussi à la session d'enregistrement, introduisent les deux dernières composantes [Matrouf et al., 2007], [Fauve et al., 2007].

La décomposition SFA s'écrit comme :

$$\mathbf{M}_{(h,s)} = \mathbf{M} + \mathbf{D}\mathbf{y}_s + \mathbf{U}\mathbf{x}_{(h,s)}. \tag{2.33}$$

La matrice diagonale $\mathbf{D}$ est de taille $MD \times MD$, où $\mathbf{DD}^T$ représente la matrice de covariance a priori de $\mathbf{y}_s$. Les termes $\mathbf{Dy}_s$ et $\mathbf{Ux}_{(h,s)}$ représentent respectivement la composante dépendante du locuteur et la composante dépendante de la session.

La technique SFA commence par estimer la matrice $\mathbf{U}$ en utilisant un nombre important de locuteurs et de sessions pour chaque locuteur. Disposant des paramètres SFA $(\mathbf{M}, \mathbf{D}, \mathbf{U})$, l'apprentissage des modèles compensés des clients se fait en éliminant directement la dissemblance de session dans le domaine du modèle (model domain). Alors que la compensation dans la phase de test, se fait dans le domaine des caractéristiques (feature domain).

Pour une description détaillée de SFA, voir la section 2 de la partie Annexes.

2.2.4.4 Variabilité totale

La JFA définit deux espaces séparés, à savoir l'espace du locuteur (défini par les voix propres) et l'espace du canal (défini par les canaux propres). Ce puissant formalisme a conduit à la définition d'un nouvel espace unique appelé l'espace de la variabilité totale, englobant simultanément les variabilités inter-locuteurs et inter-sessions. Cet espace est défini par la matrice de la variabilité totale qui contient les vecteurs propres associés aux plus grandes valeurs propres de la matrice de covariance de la variabilité totale. Il n'y a plus de distinction entre les différentes variabilités dans cette approche [Dehak et al., 2009].

Le supervecteur GMM dépendant du locuteur et du canal se décompose maintenant en :

$$\mathbf{M}_{(h,s)} = \mathbf{M} + \mathbf{Tw}, \tag{2.34}$$

où $\mathbf{M}$ est toujours le supervecteur de l'UBM, $\mathbf{T}$ est une matrice à faible rang de taille $MD \times R_T$ et $\mathbf{w}$ est un vecteur contenant les facteurs de la variabilité totale (total variability factors), appelé aussi dans la littérature un i-vecteur (i-vector). Ce dernier englobe la plupart des informations pertinentes sur l'identité du locuteur. Le vecteur $\mathbf{w}$ suit une distribution normale. La matrice $\mathbf{T}$ résulte en pratique de la concaténation de deux matrices, apprises sur des données téléphoniques et sur des données enregistrées via microphone :

$$\mathbf{T} = \left[\mathbf{T}_{tel}\ \mathbf{T}_{mic}\right], \tag{2.35}$$

on parle d'i-vecteurs téléphone/microphone [Senoussaoui et al., 2010].

L'analyse de facteur joue dans cette technique de modélisation, le rôle d'un extracteur de paramètres. La compensation de canal ne se fait plus dans l'espace des supervecteurs GMM, mais plutôt dans l'espace (à faible dimension) des facteurs de la variabilité totale. On commence par effectuer une analyse discriminante linéaire (Linear Discriminant Analysis) LDA pour définir de nouveaux axes minimisant la variance intra-classe causée par les effets du canal, et maximisant par la même occasion la variance entre locuteurs. Puis on applique après, une normalisation WCCN sur les vecteurs résultants de la projection par la LDA.

Durant la phase de test, on utilise une distance cosinus (cosine distance) entre i-vecteurs comme mesure de similarité. Les expériences menées ont montré que cette approche de modélisation améliore les performances de la JFA. Une autre approche a été testée qui consiste à apprendre des modèles SVM sur les i-vecteurs, en utilisant un noyau cosinus (cosine kernel). Ce système appelé SVM-JFA, donne de bien meilleurs résultats que la JFA. Mais il reste moins bon que l'approche sans modélisation SVM [Dehak et al., 2009], [Dehak et al., 2011b].

2.2.4.5 LDA probabiliste

La LDA probabiliste (Probabilistic LDA) PLDA [Prince and Elder, 2007] est une technique qui a été initialement proposée en reconnaissance de visages. En reconnaissance automatique du locuteur, elle peut être considérée comme une version simplifiée de la JFA qui s'applique dans l'espace des facteurs de la variabilité totale.

Disposant d'un ensemble de R enregistrements représentés par les i-vecteurs $\mathbf{w}_r$ d'un locuteur donné, le modèle PLDA est défini comme :

$$\mathbf{w}_r \;=\; \mathcal{M} \;+\; \mathbf{U}_1\mathbf{x}_1 \;+\; \mathbf{U}_2\mathbf{x}_{2r} \;+\; \epsilon_r. \tag{2.36}$$

Les paramètres de ce modèle sont :

- Un vecteur moyen $\mathcal{M}$ de taille R_T.
- Une matrice $\mathbf{U}_1$ de taille $R_T \times N_1$, dont les colonnes sont appelées des voix propres.
- Une matrice $\mathbf{U}_2$ de taille $R_T \times N_2$, dont les colonnes sont appelées des canaux propres.
- Une matrice de précision Λ de taille $R_T \times R_T$, caractérisant la distribution de ϵ_r.

$\mathbf{x}_1$, $\mathbf{x}_{2r}$ et ϵ_r sont des variables cachées représentant respectivement les facteurs du locuteur, les facteurs du canal et un bruit résiduel [Kenny, 2010].

On fait deux suppositions sur les distributions de probabilité a priori de ces variables. Soit on considère qu'elles suivent une distribution gaussienne, et on parle dans ce cas là d'une PLDA gaussienne, ou soit qu'elles suivent une distribution à queue lourde (heavy-tailed distribution), une loi de Student (Student's t-distribution) par exemple [Svensén and Bishop, 2005], [Bishop, 2006], [Archambeau and Verleysen, 2007], et on a alors une PLDA à queue lourde (heavy-tailed PLDA), que nous allons noté HT-PLDA. En supportant de grandes déviations de la moyenne, la distribution à queue lourde permet de mieux traiter les données aberrantes qui occurrent souvent en reconnaissance automatique du locuteur.

Notons par $\mathcal{N}(\mu, \Sigma)$, une distribution gaussienne de moyenne μ et de matrice de covariance Σ, et par $\Gamma(k, \theta)$, une distribution Gamma de paramètres k et θ. Et considérons les paramètres scalaires n_1, n_2 et ν, appelés degrés de liberté, et les variables scalaires

cachées u_1, u_{2r} et v_r. On assume dans la HT-PLDA que :

$$\begin{aligned}
\mathbf{x}_1 &\sim \mathcal{N}\big(0, u_1^{-1}\mathrm{I}\big) \quad \text{où} \quad u_1 \sim \Gamma\big(n_1/2, n_1/2\big), \\
\mathbf{x}_{2r} &\sim \mathcal{N}\big(0, u_{2r}^{-1}\mathrm{I}\big) \quad \text{où} \quad u_{2r} \sim \Gamma\big(n_2/2, n_2/2\big), \\
\epsilon_r &\sim \mathcal{N}\big(0, v_r^{-1}\Lambda^{-1}\big) \quad \text{où} \quad v_r \sim \Gamma\big(\nu/2, \nu/2\big).
\end{aligned} \tag{2.37}$$

La PLDA permet de réduire la dimension d'i-vecteurs téléphone/microphone tout en transformant ces i-vecteurs (appartenant initialement à un espace regroupé téléphone-microphone) en un espace homogène. Cette transformation élimine certaines informations redondantes et nuisances du canal, ce qui permet d'améliorer les performances par rapport aux i-vecteurs originaux [Senoussaoui et al., 2011], [Dehak et al., 2011a].

Notons par $\mathbf{w}_a$ et $\mathbf{w}_t$, les deux i-vecteurs définissant l'unité d'évaluation. Dans une application de vérification de locuteurs, on évalue les hypothèses de production H_s, i.e., $\mathbf{w}_a$ et $\mathbf{w}_t$ sont associés à la même variable latente d'identité $\mathbf{x}_1^c$ du locuteur c, et l'hypothèse inverse H_d, i.e., $\mathbf{w}_a$ et $\mathbf{w}_t$ correspondent à deux variables latentes d'identité $\mathbf{x}_1^a$ et $\mathbf{x}_1^t$. Le score d'appariement s'exprime sous forme de :

$$score \;=\; \log \frac{P(\mathbf{w}_a, \mathbf{w}_t|H_s)}{P(\mathbf{w}_a|H_d)P(\mathbf{w}_t|H_d)}. \tag{2.38}$$

Ce logarithme du rapport des vraisemblances se calcule facilement dans le cas d'une PLDA gaussienne (les vraisemblances marginales sont gaussiennes). Le calcul des vraisemblances se base cependant sur l'utilisation d'une borne inférieure dans le cas du modèle complexe HT-PLDA.

Une forme spéciale de la PLDA a été proposée en [Burget et al., 2011], appelée "two-covariance model", qui utilise des matrices de covariance pleines intra-classe et inter-classes dans la modélisation des variabilité du locuteur et inter-sessions.

Les résultats expérimentaux montrent que la HT-PLDA donne de bien meilleurs résultats que la PLDA gaussienne et que la JFA [Kenny, 2010], [Matějka et al., 2011]. Cependant, le système à base de la HT-PLDA est plus complexe à mettre en œuvre. On a proposé dans [Garcia-Romero and Espy-Wilson, 2011], une approche permettant de traiter le comportement non-gaussien des i-vecteurs par l'intermédiaire d'une normalisation de longueur. Cette transformation non-linéaire qu'on applique sur les i-vecteurs atténue le comportement non-gaussien, ce qui permet de rendre la PLDA gaussienne à performances égales de la HT-PLDA.

2.2.5 Nouvelles technologies

2.2.5.1 Fonctions discriminantes de produit scalaire

Les fonctions discriminantes de produit scalaire (Inner Product Discriminant Functions) IPDF [Campbell et al., 2009] constituent un nouveau formalisme (représentation)

qui étend des techniques communes en reconnaissance automatique du locuteur, pour la comparaison de modèles locuteurs. L'IPDF compare entre vecteurs de paramètres de modèles GMM (compensés via des transformations linéaires) en utilisant des fonctions de produit scalaire. La comparaison entre vecteurs de paramètres ne nécessite plus la satisfaction des conditions de Mercer, qui sont requises dans les systèmes SVM standards.

Le formalisme de l'IPDF apprend des modèles GMM, par adaptation MAP des vecteurs moyennes de l'UBM et par estimation par maximum de vraisemblance des poids des composantes. Les poids et les moyennes des gaussiennes $\{\mu_{cm}, w_{cm}\}_{m=1}^{M}$ d'un locuteur c sont ensuite regroupés en un vecteur de paramètres a_c :

$$\mathrm{a}_c \;=\; \left[w_{c1}\cdots w_{cM}\ \mu_{c1}^T\cdots\mu_{cM}^T\right]^T. \tag{2.39}$$

Diverses fonctions de comparaison $C(\mathrm{a}_x, \mathrm{a}_y)$ comparant entre deux vecteurs de paramètres a_x et a_y et produisant une mesure de similarité entre les deux locuteurs x et y, ont été proposées dans [Campbell et al., 2009]. Parmi ces fonctions, on retrouve la fonction C_{GM} (appelée Geometric Mean comparison) qui se base sur la divergence de Kullback-Leibler entre deux modèles GMM :

$$C_{GM}(\mathrm{a}_x, \mathrm{a}_y) \;=\; (\mu_x - \mathbf{M})^T\big(w_x^{1/2}\otimes \mathrm{I}_D\big)\Sigma^{-1}\big(w_y^{1/2}\otimes \mathrm{I}_D\big)(\mu_y - \mathbf{M}), \tag{2.40}$$

où μ_i est le supervecteur (de moyennes) du locuteur i, $\mathbf{M}$ est le supervecteur de l'UBM, Σ est la matrice diagonale par blocs construite à partir des matrices de covariance de l'UBM, $\otimes$ est le produit de Kronecker, I_D est la matrice d'identité de taille $D \times D$, et w_i est la matrice diagonale regroupant les poids du modèle du locuteur i.

Les résultats expérimentaux montrent que la tenue en compte des poids des composantes dans le calcul du produit scalaire permet d'améliorer les performances par rapport à un système SVM de base.

2.2.5.2 NAP pondérée

Une nouvelle forme de la NAP, appelée NAP pondérée (Weighted NAP) WNAP est introduite dans [Campbell, 2010]. La WNAP optimise le critère suivant :

$$\min_U \sum_n W_n \|Q_{U,D}\delta_n^s\|_D^2, \tag{2.41}$$

où

- U est le sous-espace de nuisance.
- W_n est un poids associé à la séquence de parole n.
- D est une matrice définie positive diagonale, dépendante potentiellement des poids des composantes.
- $Q_{U,D}$ est la matrice de projection WNAP, de forme :

$$Q_{U,D} \;=\; \mathrm{I} - \Big(U\big(U^TD^2U\big)^{-1}U^TD^2\Big). \tag{2.42}$$

– Les δ_n^s sont les données servant à l'apprentissage de la matrice WNAP.

Le nombre de trames parole dans la séquence de parole peut être utilisé comme poids W_n. Notons par $\{z_n^s\}$, un ensemble d'apprentissage qui comprend plusieurs séquences n, appartenant à plusieurs locuteurs s. Et supposons qu'on dispose pour chaque locuteur s, d'un vecteur sans nuisances $\overline{z^s}$. Les δ_n^s se calculent alors avec :

$$\delta_n^s = z_n^s - \overline{z^s}. \tag{2.43}$$

De ce fait, les δ_n^s expriment donc des nuisances.

Dans le cas d'un système IPDF utilisant la fonction C_{GM}, la métrique D s'exprime sous forme de :

$$D = \left(w^{1/2} \otimes \mathrm{I}_D\right)\Sigma^{-1/2}, \tag{2.44}$$

où w et Σ sont les matrices regroupant les poids et matrices de covariance de l'UBM.

2.2.5.3 Approches alternatives d'apprentissage du modèle du monde

On a proposé dans [Omar and Pelecanos, 2010], trois critères alternatives pour l'apprentissage de l'UBM.

Dans la première approche, l'UBM est construit à partir du modèle acoustique d'un système de reconnaissance de la parole via un *clustering K-means*, qui se base sur les distances de Kullback-Leibler entre composantes gaussiennes. Cette approche essaie d'exploiter l'information phonétique dépendante du contexte du modèle acoustique en l'apprentissage de l'UBM. Ce modèle est appelé un PIUBM pour "*Phonetically Inspired UBM*".

Les deux autres approches ajoutent quand à elles, un terme de régularisation à la fonction objective de l'algorithme du maximum de vraisemblance. Dans l'une des ces deux techniques, les paramètres de l'UBM sont estimés en utilisant une fonction objective qui favorise une représentation parcimonieuse de chaque locuteur sur les données d'apprentissage (sparse speaker representation), tandis que la deuxième technique intègre un terme de régularisation qui tend à augmenter les scores des clients et à diminuer ceux des imposteurs. Cette approche de régularisation discriminante permet un apprentissage discriminant de l'UBM.

Les résultats expérimentaux montrent que ces trois approches améliorent les performances par rapport à un système de base.

2.2.5.4 Clé binaire

Une nouvelle approche proposée dans [Anguera and Bonastre, 2010], représente le locuteur (une séquence de parole) par une clé acoustique à valeurs binaires (binary key). Les caractéristiques principales de cette approche sont la petite taille des vecteurs représentatifs, comparée aux autres approches de modélisation, et son faible coût calculatoire en ne maniant que des vecteurs binaires. La représentation par clés binaires fait appel à deux processus. On commence par apprendre un "grand" modèle du monde KBM (binary-key

background model), qui sera ensuite utiliser lors de l'évaluation pour convertir les séquences de parole en des clés binaires.

On utilise un UBM classique à N_g composantes pour apprendre des modèles GMM de N_s locuteurs d'ancrage (anchor speakers), couvrant idéalement tout l'espace acoustique et faisant sortir les spécificités des locuteurs. Le KBM est obtenu après en concaténant les N_s modèles GMM.

On définit ensuite la clé du locuteur comme un vecteur binaire de dimension $N = N_s \times N_g$:

$$\mathbf{v}_c \;=\; [v_1 \;\cdots\; v_N], \tag{2.45}$$

où les bits $\mathbf{v}_c[i]$ mis à 1 indiquent les composantes du KBM qui modélisent la séquence acoustique. Pour ce faire, on définit un vecteur accumulateur $\mathbf{v}_s$ initialisé par 0. Pour chacun des vecteurs acoustiques de la séquence, on calcule les vraisemblances par rapport aux N composantes du KBM pour en sélectionner les $p_1\%$ plus vraisemblables gaussiennes. Les composantes sélectionnées se voient leur valeurs d'accumulation augmentées de 1. À la fin de la séquence, les scalaires $\mathbf{v}_s[i]$ reflètent l'importance relative de chaque composante i en la modélisation de la séquence de parole. Un deuxième paramètre p_2 permet d'obtenir la clé $\mathbf{v}_c$ à partir de $\mathbf{v}_s$, en associant aux $p_2\%$ plus importantes composantes de $\mathbf{v}_s$ une valeur VRAI (dans $\mathbf{v}_c$) et une valeur FAUX pour les composantes restantes.

Finalement on utilise comme score d'appariement, une simple mesure de similarité entre les clés binaires $\mathbf{v}_a$ et $\mathbf{v}_t$ qui définissent l'unité d'évaluation :

$$S(\mathbf{v}_a, \mathbf{v}_t) \;=\; \frac{1}{N} \sum_{i=1}^{N} \big(\mathbf{v}_a[i] \wedge \mathbf{v}_t[i]\big), \tag{2.46}$$

où $\wedge$ est l'opérateur du ET logique.

2.2.6 Fusion

Divers systèmes de reconnaissance du locuteur combinent plusieurs sources d'information en fusionnant différents "sous" systèmes de reconnaissance. Cette fusion améliore incessamment les performances d'un sous-système unique, mais nécessite en contre partie beaucoup plus de ressources techniques et calculatoires. Son succès dépend bien des performances des sous-systèmes ainsi que de leur complémentarité. La fusion permet aussi de concevoir des systèmes biométriques multimodaux, combinant différentes modalités.

Typiquement, un ou plusieurs ensembles de paramètres du signal de parole sont utilisés avec un ou plusieurs classificateurs. Puis, les différents scores ou décisions sont combinés. Parmi les travaux traitant la fusion en reconnaissance du locuteur, nous renvoyons le lecteur aux [Chen et al., 1997], [Fredouille et al., 2000], [Ramachandran et al., 2002], [Garcia-Romero et al., 2003], [Campbell et al., 2003], [Kinnunen et al., 2004], [Campbell et al., 2004b], [Scheffer and Bonastre, 2006], [Mashao and Skosan, 2006], [Huenupán et al., 2007], [Solewicz and Koppel, 2007], [Ferrer et al., 2008].

La fusion de 9 systèmes de reconnaissance du locuteur utilisant des informations acoustiques, prosodiques, phonétiques et lexicales a permis d'améliorer considérablement les performances du simple système à paramètres acoustiques, montrant ainsi la complémentarité de ces différentes informations (paramétrisations) [Reynolds et al., 2003].

La forme de fusion la plus élémentaire consiste à combiner les scores d'appariement s_i de N_c différents systèmes :

$$s = \sum_{i=1}^{N_c} p_i s_i, \tag{2.47}$$

où les poids p_i mesurent la contribution de chacun des classificateurs. Ces poids sont optimisés sur des données de développement, et sont maintenus après fixes lors de l'évaluation. Si les scores d'appariement peuvent être interprétés comme des probabilités a posteriori, la fonction du produit peut être utilisée à la place de la fonction de sommation, mais au risque d'amplifier les erreurs d'estimation.

Une technique efficace de recherche des poids par régression logistique (logistic regression) a été proposée dans [Brümmer and du Preez, 2006], dont l'implémentation libre est disponible grâce à l'outil FoCal [Brümmer, 2005]. D'autres travaux se sont basés sur une post-classification (modélisation) des scores des différents sous systèmes (regroupés en un vecteur de scores) en utilisant par exemple une SVM ou un réseau de neurones. Dans [Ferrer et al., 2008], une procédure modifiée de régression linéaire logistique a été proposée, qui utilise des caractéristiques auxiliaires (auxiliary features) en l'optimisation des poids de fusion. Cette procédure donne de bien meilleurs résultats par rapport aux autres approches de fusion.

2.3 Systèmes de la campagne d'évaluation NIST-SRE 2010

Cette section donne une synthèse des spécificités de certains des meilleurs systèmes de vérification du locuteur, soumis à la campagne d'évaluation NIST-SRE 2010 [NIST, 2010] : SRI[2], SVIST[3], iFly[4], ABC[5], LPT[6], I4U[7], IIR[8], MITLL[9], IBM[10], LIA[11]. Le tableau Tab. 2.1 regroupe les différents paramètres du signal de parole et leurs techniques de norma-

[2]SRI International (http ://www.sri.com/).
[3]Shanghai Voice Info Science and Technology.
[4]The USTC iFly Speech Laboratory (http ://wwwcnc.ustc.edu.cn/en/article/41/42fe1a02/).
[5]**A**gnitio , **B**rno University of Technology , **C**entre de recherche en informatique de Montréal.
[6]Loquendo - Politecnico di Torino.
[7]Institute for Infocomm Research , University of Science and Technology of China/iFly , University of Joensuu (Eastern Finland) , University of New South Wales , Nanyang Technological University.
[8]Institute for Infocomm Research (http ://www.i2r.a-star.edu.sg/).
[9]MIT Lincoln Laboratory (http ://www.ll.mit.edu/).
[10]IBM Thomas J. Watson Research Center (http ://www.watson.ibm.com/index.shtml).
[11]Laboratoire Informatique d'Avignon (http ://lia.univ-avignon.fr/).

lisation, les diverses approches de modélisation et compensation ainsi que les différentes techniques de normalisation des scores utilisés par ces systèmes. Nous donnerons aussi à titre indicatif le nombre de (s-)sous-systèmes utilisant ces paramètres, techniques et modèles. Ces systèmes sont décrits en détail dans la partie Annexes. Les quelques paramètres et modèles non présentés précédemment seront exposés dans cette dernière.

Les meilleurs systèmes fusionnent plusieurs sous-systèmes, qui peuvent résulter eux mêmes de la fusion de divers s-sous-systèmes. Utilisés seuls, certains sous-systèmes peuvent donner de moins bons résultats que les (sous-)systèmes classiques de la littérature. Néanmoins, leur fusion avec ces (sous-)systèmes état de l'art permet d'améliorer les résultats globaux.

D'après la description des meilleurs systèmes de la campagne d'évaluation NIST-SRE 2010, on peut constater que les paramètres acoustiques les plus utilisés restent de loin les standards MFCC. Les systèmes essaient également d'exploiter d'autres sources d'information comme la prosodie. Le filtrage RASTA et les techniques de gaussianisation sont très utilisés pour la normalisation des paramètres descriptifs du signal de parole. Certains systèmes les utilisent même conjointement.

En modélisation, les systèmes compensés basés sur le formalisme de l'analyse de facteurs, e.g., JFA et Variabilité totale, sont les plus utilisés en reconnaissance automatique du locuteur et donnent de très bons résultats. On trouve ensuite le système classique répandu à base de SVM, le GSL-NAP. D'autres systèmes basés sur les GMM, SVM ou combinant ces deux approches de modélisation ont été aussi expérimentés. Malgré l'intégration de techniques de compensation de la variabilité inter-sessions, on peut remarquer que tous les systèmes continuent de faire appel aux méthodes de normalisation des scores. Parmi toutes ces méthodes, la ZT-norm est celle qui est la plus utilisée, puis vient après la TZ-norm.

	SRI	SVIST	iFly	ABC	LPT	I4U	IIR	MITLL	IBM	LIA
MFCC	3	1		7	4	1	1	6	2	
PLP	1	1	5		4	1	1			
LPCC		1	4			1	1	5	1	
LFCC										4
Paramètres prosodiques	1			1				1		
Paramètres ASR									3	
Word N-gram	1									
CMLLR-MLLR				1						
SCM-SCF						1				
SWLP						1				
RASTA		3	9			3	3			
Feature Warping		3		1	8			+	2	
Gaussianisation			9			2	2			
Short-time Gaussianization				5						
CMVN	4					1	1			4
CMS						2	2			
JFA	5	1	5	3	4	1	1	4	2	
GSL-NAP		1	4			1	1	2		
Variabilité totale		1		2	4			5		
GMM-SFA										3
MLLR-SVM & NAP	1									
SVM	1									
SVM & NAP				1						
GMM-SVM-BHATT & NAP						1	1			
GMM-SVM-FT & NAP						1	1			

"GMM-SFA" Supervector Linear kernel										1
Two-covariance modeling				1						
PLDA				1						
HT-PLDA				1						
GMM & NAP								4		
IPDF & WNAP								4		
Comparaison de facteurs de locuteurs								1		
ZT-norm	4			4	+			10	+	4
TZ-norm		3	9				1			
S-norm				3				1		
AT-norm					+					
ZAT-norm								4		
T-norm						2				
Z-norm						1				
SAS-norm								1		

TAB. 2.1: Caractéristiques de certains des meilleurs systèmes de vérification du locuteur soumis à la campagne d'évaluation NIST-SRE 2010.

Le signe + indique l'utilisation de la technique de normalisation dans certains ou dans l'ensemble des sous-systèmes fusionnés.

Diverses combinaisons {paramétrisation, normalisation des vecteurs de paramètres, modélisation – compensation, normalisation des scores} ont été testées. Il est difficile d'en considérer une comme étant la meilleure ; à notre connaissance, aucune étude comparative détaillée n'a été faite dans la littérature. Il ressort par exemple de l'analyse de tests menés par le I4U que sur 13 sous-systèmes, le sous-système (PLP – JFA) est le meilleur système unique. Puis vient juste après le sous-système (MFCC – GSL-NAP), qui est un tout petit peu moins bon. Néanmoins d'après les tests de SIVST, le sous-système (MFCC – GSL-NAP) donne de meilleurs résultats que le sous-système (PLP – JFA) ; tous deux restent moins bons que le sous-système (LPCC – Variabilité totale). Mais en survolant les travaux publiés en la littérature, on peut dire que le système (MFCC – JFA) est celui qui est le plus utilisé en la reconnaissance automatique du locuteur.

Chapitre 3

Les modèles GMM à grande marge

Récemment, une nouvelle approche discriminante pour la séparation multi-classes a été proposée et appliquée en reconnaissance de la parole, les LM-GMM pour *Large Margin Gaussian Mixture Models* [Sha and Saul, 2006], [Sha and Saul, 2007], [Sha, 2007]. Cette méthode introduit la notion de marge exploitée dans les outils de type Machine à vecteurs de support (SVM) afin de rendre plus discriminante l'approche générative probabiliste basée sur des lois gaussiennes couramment utilisée. De manière très schématique, elle permet de construire une frontière non-linéaire de type quadratique entre les classes, représentées par des ellipsoïdes, dans l'espace des données lui-même, en maximisant une marge entre ces classes ; de ce fait cette modélisation discriminante est plus adaptée à des applications multi-classes que l'approche classique SVM. L'apprentissage de ces modèles implique une initialisation de type Modèle de Mélanges de lois gaussiennes (GMM).

La modélisation LM-GMM diffère des autres approches d'apprentissage discriminant des GMM qui sont utilisées en traitement de la parole, comme les techniques d'estimation par information mutuelle maximale (Maximum Mutual Information Estimation) MMIE [Valtchev et al., 1997] et d'estimation par maximum de vraisemblance conditionnel (Conditional Maximum Likelihood Estimation) CMLE [Nadas, 1983], où l'optimisation des paramètres des modèles n'est pas convexe. De plus, ces méthodes d'apprentissage n'intègrent pas la notion de marge.

En reconnaissance automatique du locuteur, les systèmes GMM utilisent des matrices de covariance diagonales et sont appris par adaptation MAP des vecteurs moyennes d'un modèle du monde. Exploitant cette propriété, nous proposons une version simplifiée des LM-GMM, les modèles LM-GMM à matrices de covariance diagonales (LM-dGMM). L'algorithme d'apprentissage résultant est plus simple et plus rapide que la version originale, et donne de bien meilleurs résultats que les modèles LM-GMM originaux. Au cours de ce chapitre, après avoir rappelé la notion originale de modèle LM-GMM [Sha and Saul, 2006], nous présentons les modèles LM-dGMM [Jourani et al., 2010] accompagnés de l'algorithme d'apprentissage, et une première application dans le cadre de la reconnaissance automatique du locuteur.

3.1 Le modèle original Large Margin GMM (LM-GMM)

Remarque : l'essentiel du contenu de cette section est issu de la thèse de Fei SHA [Sha, 2007], mais la présentation est revue en énonçant la règle de décision systématiquement avant de définir la fonction de perte et l'algorithme d'optimisation utilisé en phase d'apprentissage.

Dans un problème à C classes et des observations appartenant à $\mathbb{R}^D$, l'approche LM-GMM consiste à supposer que les frontières des régions attachées à chaque classe sont de type quadratique résultant de la caractérisation de chaque classe par un ellipsoïde. L'ellipsoïde associé à la classe c est paramétré par un vecteur centroïde $\mu_c \in \mathbb{R}^D$, le centre de l'ellipsoïde, et une matrice semidéfinie positive $\Psi_c \in \mathbb{R}^{D\times D}$ qui détermine son orientation.

La règle de décision attribue à un vecteur d'observation $x \in \mathbb{R}^D$ la classe dont le centroïde est le plus proche en terme de distance de Mahalanobis :

$$y = \underset{c}{argmin} \left\{(x-\mu_c)^T\Psi_c(x-\mu_c)+\theta_c\right\}. \tag{3.1}$$

Dans cette expression, est introduit un facteur positif θ_c (offset) qui correspond à un décalage des frontières pour prendre en compte une connaissance a priori sur les classes [Sha and Saul, 2006], [Sha and Saul, 2007]. Ce paramètre équivaut à l'existence d'une probabilité a priori dans l'approche bayésienne. Il s'en suit que chaque classe c est caractérisée par le triplet $(\mu_c, \Psi_c^{-1}, \theta_c)$.

L'argument de la partie droite de l'équation eq. 3.1 n'est pas linéaire par rapport aux paramètres de l'ellipsoïde μ_c et Ψ_c^{-1}. C'est pourquoi il est introduit pour chaque classe c, une matrice élargie $\Phi_c \in \mathbb{R}^{(D+1)\times(D+1)}$:

$$\Phi_c = \begin{pmatrix} \Psi_c & -\Psi_c\mu_c \\ -\mu_c^T\Psi_c & \mu_c^T\Psi_c\mu_c+\theta_c \end{pmatrix}. \tag{3.2}$$

La précédente règle de décision s'écrit alors sous forme matricielle :

$$y = \underset{c}{argmin} \left\{z^T\Phi_c z\right\}, \tag{3.3}$$

où $z = \begin{bmatrix} x \\ 1 \end{bmatrix}$. L'argument de la partie droite de la règle de décision en l'équation eq. 3.3 est à présent linéaire par rapport aux paramètres Φ_c.

Par la suite, à chaque observation x_n sera associé le vecteur augmenté d'une coordonnée égale à 1, noté z_n.

3.1.1 Apprentissage discriminant : Maximisation de la marge

L'apprentissage des modèles LM-GMM vise à trouver les matrices Φ_c qui minimisent le risque empirique sur les données d'apprentissage ; cela signifie que les modèles trouvés

doivent non seulement permettre de minimiser l'erreur de classification, mais assurer que les données d'apprentissage soient les plus éloignées possible des frontières de décision. C'est pourquoi à l'image des SVM, est introduite la notion de marge d'une donnée x_n, définie comme étant sa distance à la plus proche frontière de décision.

Disposant des données d'apprentissage $\{(x_n, y_n)\}_{n=1}^{N}$, où $y_n \in \{1, 2, \cdots, C\}$ est la classe à laquelle appartient la donnée x_n, les modèles LM-GMM sont appris de telle sorte que chaque vecteur x_n soit distant d'au moins une distance unitaire des frontières de décision des autres classes :

$$\forall c \neq y_n, \qquad z_n^T \Phi_c z_n \; - \; z_n^T \Phi_{y_n} z_n \; \geq \; 1. \tag{3.4}$$

La figure Fig. 3.1 illustre le principe de la marge dans la modélisation LM-GMM.

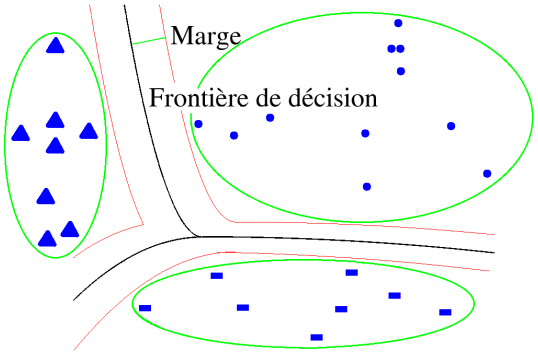


FIG. 3.1: *Frontières de décision dans les modèles LM-GMM.*

Dans le cas où il existe une solution, la satisfaction de l'ensemble des contraintes ne conduit pas à une unique solution (Φ_c). Il a été proposé une optimisation convexe qui sélectionne les plus "petits" paramètres, satisfaisant les contraintes de marge de l'équation eq. 3.4 ; le terme "petit" étant à prendre au sens de la norme donnée par la trace d'une matrice. De plus, dans le cas où il n'existe pas de solution, les contraintes sont relâchées au travers d'un facteur de régularisation et la fonction de perte dont la minimisation conduit à l'obtention des LM-GMM, s'écrit sous la forme de :

$$\mathbf{L} \;\; = \;\; \sum_{n=1}^{N} \sum_{c \neq y_n} \max\left(0 \;,\; 1 + z_n^T (\Phi_{y_n} - \Phi_c) z_n\right) \;\; + \;\; \alpha \sum_{c=1}^{C} trace(\Psi_c). \tag{3.5}$$

Cette fonction de perte est linéaire par morceaux et convexe en Φ_c. Le premier terme de $\mathbf{L}$ pénalise les violations de marge, tandis que le deuxième terme régularise les matrices d'orientation des C classes. La minimisation de la trace de la matrice Ψ_c a pour effet d'augmenter le volume de l'ellipsoïde de la classe c.

En résumé, l'apprentissage des modèles LM-GMM consiste à rechercher les matrices Φ_c solutions du problème d'optimisation sous contraintes suivantes :

$$\begin{array}{ll} \min & \mathbf{L} \\ s.c. & \Phi_c \succeq 0, \quad c = 1, 2, ..., C. \end{array} \tag{3.6}$$

Ce problème d'optimisation semidéfinie (Semidefinite Programming SDP [Vandenberghe and Boyd, 1996]) peut être résolu en utilisant les méthodes du point intérieur (interior point methods), mais cette solution n'est pas envisageable dans le cas de volumes d'apprentissage trop importants.

3.1.2 Extension des modèles LM-GMM dans le cadre segmental

Dans certains problèmes de classification, il est souhaitable de vouloir prendre une décision sur un ensemble d'observations que l'on sait appartenir à une même classe, au lieu de prendre une décision sur chacune des observations prise individuellement ; c'est un cas fréquent en traitement automatique de la parole et plus particulièrement en reconnaissance de segments phonétiques ou reconnaissance de locuteurs. Dans ce cas, la décision est dite prise sur un segment, suite d'observations élémentaires, et la taille de ces segments est variable. Les précédents modèles LM-GMM ont été étendus pour pouvoir aborder le cas de cette reconnaissance segmentale. Supposons que $\{x_{n,t}\}_{t=1}^{T_n}$ représente la séquence de T_n vecteurs paramétriques qui composent un $n^{ème}$ segment ; la règle de décision proposée est donnée par :

$$y \;=\; \underset{c}{argmin} \left\{ \sum_{t=1}^{T_n} z_t^T \Phi_c z_t \right\}. \tag{3.7}$$

Il s'en suit que les contraintes de grande marge lors de l'apprentissage, se voient alors réécrites comme :

$$\forall c \neq y_n, \qquad \left(\frac{1}{T_n} \sum_{t=1}^{T_n} z_{n,t}^T \Phi_c z_{n,t} \right) \;-\; \left(\frac{1}{T_n} \sum_{t=1}^{T_n} z_{n,t}^T \Phi_{y_n} z_{n,t} \right) \;\geq\; 1, \tag{3.8}$$

en supposant que le segment $\{x_{n,t}\}_{t=1}^{T_n}$ de l'ensemble d'apprentissage appartient à la classe y_n.

La fonction de perte devient alors :

$$\mathbf{L} \;=\; \sum_{n=1}^{N} \sum_{c \neq y_n} \max \left(0 \;,\; 1 + \left(\frac{1}{T_n} \sum_{t=1}^{T_n} z_{n,t}^T (\Phi_{y_n} - \Phi_c) z_{n,t} \right) \right) \;+\; \alpha \sum_{c=1}^{C} trace(\Psi_c). \tag{3.9}$$

3.1.3 Traitement des données aberrantes

La présence de données aberrantes pénalise l'apprentissage. L'algorithme d'apprentissage est ramené à plus se focaliser sur les "quelques" données aberrantes qui existent, que de bien classer les autres données. Classer correctement une seule donnée aberrante peut diminuer davantage la fonction de perte que modéliser correctement les autres données.

Afin de privilégier l'apprentissage sur les données correctes, Fei SHA propose de détecter les données susceptibles d'être aberrantes et de réduire leur effet. Un modèle gaussien

est estimé par maximum de vraisemblance, pour chaque classe c. L'analogie GMM et LM-GMM permet de déduire de l'ensemble de ces lois gaussiennes, un modèle LM-GMM avec en particulier l'ensemble des matrices (Φ_c^{MLE}). Finalement pour chaque vecteur x_n de l'ensemble d'apprentissage, la perte accumulée due aux violations des contraintes par rapport à ces modèles est calculée à partir de l'équation eq. 3.4 et donne la valeur :

$$h_n^{MLE} = \sum_{c \neq y_n} \max\left(0\ ,\ 1 + z_n^T(\Phi_{y_n}^{MLE} - \Phi_c^{MLE})z_n\right). \tag{3.10}$$

$h_n^{MLE} \geq 0$ représente la baisse de la valeur de la fonction de perte $\mathbf{L}$, quand une donnée initialement mal classée x_n est prise en compte durant l'apprentissage des modèles LM-GMM. Les données aberrantes ont une grande valeur de h_n^{MLE}.

On pondère les termes de perte de $\mathbf{L}$ par les facteurs multiplicatifs suivants :

$$l_n = \min\left(1, \frac{1}{h_n^{MLE}}\right). \tag{3.11}$$

Cette stratégie permet d'équilibrer l'influence des données durant l'apprentissage. Les poids l_n sont calculés une seule fois au départ, puis maintenus fixes tout au long de l'apprentissage des modèles LM-GMM. La fonction de perte devient alors :

$$\mathbf{L} = \sum_{n=1}^{N} l_n \sum_{c \neq y_n} \max\left(0\ ,\ 1 + z_n^T(\Phi_{y_n} - \Phi_c)z_n\right) + \alpha \sum_{c=1}^{C} trace(\Psi_c). \tag{3.12}$$

3.2 Modèle de mélange LM-GMM

3.2.1 Définition du modèle de mélange LM-GMM

A l'image de l'utilisation de Mélanges de Lois Gaussiennes pour modéliser une classe et prendre en compte sa variabilité, il est supposé que chaque classe est représentée par un ensemble de M ellipsoïdes ; un ellipsoïde est une *sous-classe*, représentée comme précédemment par un triplet (μ, Ψ^{-1}, θ), reformulé en une matrice élargie Φ. Par simplicité, on suppose que le nombre d'ellipsoïdes est le même pour toutes les classes et égal à M.

La règle de décision devient :

$$y = \underset{c}{argmin} \left\{ -\log \sum_{m=1}^{M} e^{-z^T \Phi_{cm} z} \right\}, \tag{3.13}$$

où Φ_{cm} est le $m^{ème}$ ellipsoïde de la classe c.

La figure Fig. 3.2 donne l'exemple d'un mélange à trois composantes par classe.

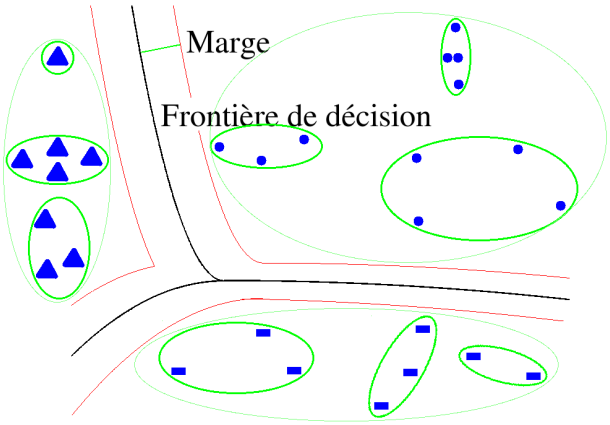


FIG. 3.2: *Mélange LM-GMM à 3 ellipsoïdes.*

3.2.2 Apprentissage du modèle de mélange LM-GMM

Dans le cas de modèles de mélange LM-GMM, on désire définir des ellipsoïdes représentés par l'ensemble des $\Phi_{cm}, c = 1, 2, ..., C, m = 1, 2, ..., M$, de telle sorte qu'une marge maximale existe entre ellipsoïdes appartenant à des classes différentes. On recherche des frontières de décision discriminantes entre les éléments du mélange. Il s'en suit qu'en phase d'apprentissage, chaque donnée x_n est supposée appartenir à une classe y_n et à une sous classe m_n, et on cherche à ce que cette donnée x_n soit plus proche du centroïde associé $\mu_{y_n m_n}$ (en terme de distance de Mahalanobis) d'au moins une distance unitaire (une marge minimale unitaire) que de tout centroïde μ_{cm} de toute autre classe c. Disposant de l'ensemble des données d'apprentissage $\{(x_n, y_n, m_n)\}_{n=1}^{N}$, les contraintes LM-GMM à satisfaire deviennent :

$$\forall c \neq y_n, \forall m, \qquad z_n^T(\Phi_{cm} - \Phi_{y_n m_n})z_n \;\geq\; 1. \tag{3.14}$$

Afin de regrouper les M précédentes contraintes pour chaque classe c, l'inégalité soft-max $\min_m a_m \geq -\log \sum_m e^{-a_m}$ est utilisée (formulation non équivalente), l'équation eq. 3.14 devient alors :

$$\forall c \neq y_n, \qquad -\log \sum_{m=1}^{M} e^{-z_n^T \Phi_{cm} z_n} \;-\; z_n^T \Phi_{y_n m_n} z_n \;\geq\; 1. \tag{3.15}$$

Bien que non-linéaires en les matrices Φ_{cm}, les contraintes de l'équation eq. 3.15 restent convexes. La fonction de perte des LM-GMM s'écrit maintenant sous forme de :

$$\mathbf{L} \;=\; \sum_{n=1}^{N} \sum_{c \neq y_n} \max\left(0\ ,\ 1 + z_n^T \Phi_{y_n m_n} z_n \;+\; \log \sum_{m=1}^{M} e^{-z_n^T \Phi_{cm} z_n}\right) \;+\; \alpha \sum_{c=1}^{C} \sum_{m=1}^{M} trace(\Psi_{cm}). \tag{3.16}$$

Le problème d'optimisation est résolu en utilisant la méthode du sous-gradient projeté (projected subgradient method) [Bertsekas, 1999], qui est une méthode itérative pour la

résolution de problèmes de minimisation convexe[12]. Il s'agit d'appliquer la méthode du sous-gradient et de vérifier la semidéfinie positivité de chaque matrice Φ_{cm} à chaque itération ; à chaque matrice non semidéfinie positive est substitué son projeté dans l'ensemble de toutes les matrices semidéfinies positives.

Remarques :

- Ces modèles de mélange peuvent être étendus facilement pour faire de la reconnaissance segmentale ou pour traiter l'impact de données aberrantes.
- Par similitude avec les mélanges classiques de lois gaussiennes, le vecteur centroïde et la matrice d'orientation de l'ellipsoïde sont analogues au vecteur moyenne et à l'inverse de la matrice de covariance d'une loi gaussienne. Le facteur θ_{cm} dépend quand à lui du poids w_{cm} de la composante gaussienne :
  $\theta_{cm} = \frac{1}{2}\big(D\log(2\pi) \;+\; \log|\Psi_{cm}^{-1}|\big) \;-\; \log(w_{cm})$.

3.2.3 Mise en œuvre

3.2.3.1 Étiquetage des données d'apprentissage

Lors de la phase d'apprentissage, chaque donnée est étiquetée par sa classe d'appartenance. On peut ne pas disposer a priori des labels $\{m_n\}_{n=1}^{N}$. Dans ce cas, la détermination de ces labels peut être faite en apprenant par maximum de vraisemblance un GMM par classe sur les données d'apprentissage de cette classe, et en sélectionnant la composante ayant la plus grande probabilité a posteriori :

$$m_n \;=\; \underset{m}{argmin}\; \left\{z_n^T \Phi_{y_n m}^{MLE} z_n\right\}, \tag{3.17}$$

où la matrice $\Phi_{y_n m}^{MLE}$ est construite à partir du modèle GMM correspondant à la classe y_n à laquelle appartient la donnée x_n.

Les labels m_n sont sélectionnés une seule fois au départ, puis ils sont maintenus fixes tout au long de la phase d'apprentissage des mélanges LM-GMM. Des résultats expérimentaux montrent qu'une mise à jour de ces labels n'améliore pas considérablement les performances. Par contre, elle augmente la complexité calculatoire de l'algorithme d'apprentissage.

L'ensemble des modèles GMM ainsi appris pour chaque classe servira en règle générale pour initialiser le mélange LM-GMM lors de son apprentissage.

3.2.3.2 Factorisation matricielle

La méthode du sous-gradient projeté garantit la convergence vers le minimum global. Néanmoins, elle converge très lentement en pratique. Pour accélérer la convergence de l'algorithme d'apprentissage, spécialement au début de l'optimisation, on utilise une approche plus agressive qui se base sur le gradient conjugué (conjugate gradient) [Avriel, 2003].

[12] Dans des applications réelles où on a des millions de données et des centaines de classes, la complexité de ce problème rend l'utilisation des méthodes du point intérieur difficile.

On reformule le problème d'optimisation précédent comme étant un problème d'optimisation sans contraintes, en écrivant chaque matrice Φ_{cm} sous forme du produit d'une matrice Λ_{cm} par sa transposée (racine carrée d'une matrice semidéfinie positive) :

$$\Phi_{cm} \quad = \quad \Lambda_{cm}\Lambda_{cm}^{T}. \tag{3.18}$$

La fonction de perte, fonction des Λ_{cm}, n'est pas convexe. L'algorithme du gradient conjugué peut donc converger vers un minimum local. De plus, la racine carrée d'une matrice n'est pas unique. En pratique, les tests menés montrent qu'une optimisation combinée qui initialise la méthode du sous-gradient projeté par les matrices résultats du gradient conjugué, fonctionne très bien.

Notons qu'on peut aussi utiliser l'algorithme L-BFGS au lieu du gradient conjugué, pour la résolution du problème d'optimisation. Cette méthode sera rappelée en section 3.4.1.

3.3 Les mélanges LM-GMM à matrices de covariance diagonales (LM-dGMM)

Notre objectif applicatif est la reconnaissance automatique du locuteur ; or dans ce domaine, dans les systèmes GMM de l'état de l'art, les lois gaussiennes ont des matrices de covariance diagonales qui, de plus, sont supposées indépendantes du locuteur ainsi que leur pondération. La transposition de ces propriétés nous a conduit à proposer une version simplifiée des mélanges LM-GMM, pour devenir des mélanges LM-GMM à matrice diagonale, notés par la suite mélange "LM-dGMM" [Jourani et al., 2010] : les matrices d'orientation sont diagonales et indépendantes des classes ainsi que les offsets. La réécriture de la règle de décision est donnée par :

$$y \quad = \quad \underset{c}{argmin} \left\{ -\log \sum_{m=1}^{M} \exp\Big(- d(x, \mu_{cm}) - \theta_m \Big) \right\}. \tag{3.19}$$

3.3.1 Apprentissage d'un mélange LM-dGMM

L'apprentissage est inspiré de l'apprentissage des GMM, réalisé par adaptation de type MAP d'un modèle universel. Ce positionnement entraîne que les contraintes de grande marge ne porteront, lors de l'apprentissage, que sur les centroïdes, dès lors qu'un modèle initial sera défini.

3.3.1.1 Initialisation de l'apprentissage

Pour réaliser cet apprentissage, il est nécessaire d'initialiser le mélange LM-dGMM. Nous supposons que nous avons pu apprendre pour chaque classe c un GMM à M composantes, noté GMM_c appris par adaptation MAP d'un modèle GMM général, où seules les

moyennes ont été ré-estimées. L'analogie entre GMM et LM-GMM nous permet d'avoir ce mêle initial. Il est rappelé que le $m^{ème}$ ellipsoïde de la classe c correspond à la $m^{ème}$ loi gaussienne du GMM_c, son centroïde correspond au vecteur moyen μ_{cm}. La matrice d'orientation est l'inverse de la matrice de covariance diagonale $\Sigma_m = diag(\sigma^2_{m1}, ..., \sigma^2_{mD})$ et le facteur $\theta_m = \frac{1}{2}\big(D\log(2\pi) + \log|\Sigma_m|\big) - \log(w_m)$, où w_m est le poids de la loi gaussienne. Ces deux paramètres ne dépendent pas de c.

Le modèle GMM_{y_n} est exploité pour déterminer pour chaque donnée x_n le label m_n de la composante du $y_n^{ème}$ mélange ; comme précédemment, il correspond à la plus grande probabilité a posteriori.

3.3.1.2 Nouvelle version de la fonction de perte

Disposant des données d'apprentissage $\{(x_n, y_n, m_n)\}_{n=1}^N$, les contraintes LM-dGMM à satisfaire lors de l'apprentissage se déduisent simplement de l'équation eq. 3.14 :

$$\forall c \neq y_n, \forall m, \qquad d(x_n, \mu_{cm}) + \theta_m \;\geq\; 1 \;+\; d(x_n, \mu_{y_n m_n}) + \theta_{m_n}, \tag{3.20}$$

où

$$d(x_n, \mu_{cm}) \;=\; \sum_{i=1}^{D} \frac{(x_{ni} - \mu_{cmi})^2}{2\sigma^2_{mi}} \tag{3.21}$$

est la distance euclidienne normalisée entre x_n et μ_{cm}. Comme précédemment, nous faisons appel à l'inégalité softmax pour dériver une borne inférieure à $\min\limits_m \Big(d(x_n, \mu_{cm}) + \theta_m\Big)$, regroupant ainsi les contraintes en :

$$\forall c \neq y_n, \qquad -\log \sum_{m=1}^{M} \exp\Big(-d(x_n, \mu_{cm}) - \theta_m\Big) \;\geq\; 1 \;+\; d(x_n, \mu_{y_n m_n}) + \theta_{m_n}. \tag{3.22}$$

La fonction de perte devient alors :

$$\begin{aligned} \mathbf{L} \;=\; & \sum_{n=1}^{N} \sum_{c \neq y_n} \max\left(0\ ,\ 1 + d(x_n, \mu_{y_n m_n}) + \theta_{m_n} + \log \sum_{m=1}^{M} \exp\Big(-d(x_n, \mu_{cm}) - \theta_m\Big)\right) \\ & + \alpha \sum_{m=1}^{M} \sum_{d=1}^{D} \frac{1}{\sigma^2_{md}}. \end{aligned} \tag{3.23}$$

Cette fonction de perte se simplifie une nouvelle fois, dans la mesure où les matrices d'orientation sont considérées comme constantes, pour devenir au final :

$$\mathbf{L} \;=\; \sum_{n=1}^{N} \sum_{c \neq y_n} \max\left(0\ ,\ 1 + d(x_n, \mu_{y_n m_n}) + \theta_{m_n} + \log \sum_{m=1}^{M} \exp\Big(-d(x_n, \mu_{cm}) - \theta_m\Big)\right). \tag{3.24}$$

Le terme de régularisation disparaît.

Comparé à l'algorithme original de [Sha and Saul, 2006], notre algorithme simplifié est beaucoup moins complexe. Notre système est plus rapide et moins demandeur de mémoire. En effet, les multiplications matricielles disparaissent avec les matrices Φ_{cm}, et les contraintes de semidéfinie positivité se voient relaxées. En général, on converge après un nombre réduit d'itérations. Un autre avantage important est que les offsets restent normalisés du fait de leur non-apprentissage, ce qui n'est pas le cas avec les modèles LM-GMM.

3.3.2 Extension des modèles LM-dGMM dans le cadre segmental

Dans le cadre d'une approche segmentale, où il est imposé à une séquence de vecteurs de données d'appartenir à une même classe, la règle de décision s'écrit :

$$y \;=\; \underset{c}{argmin} \;\left\{ \sum_{t=1}^{T} -\log \sum_{m=1}^{M} \exp\Big(-d(x_t, \mu_{cm}) - \theta_m \Big) \right\}. \tag{3.25}$$

En phase d'apprentissage, il s'en suit que si $\{x_{n,t}\}_{t=1}^{T_n}$ est la séquence de T_n vecteurs paramétriques de la classe y_n et $m_{n,t}$ est le label associé au vecteur $x_{n,t}$, les contraintes de grande marge sont à présent définies par :

$$\forall c \neq y_n, \forall m, \qquad \frac{1}{T_n} \sum_{t=1}^{T_n} \Big(d(x_{n,t}, \mu_{cm}) + \theta_m \Big) \;\geq\; 1 \;+\; \frac{1}{T_n} \sum_{t=1}^{T_n} \Big(d(x_{n,t}, \mu_{y_n m_{n,t}}) + \theta_{m_{n,t}} \Big), \tag{3.26}$$

ou encore

$$\forall c \neq y_n, \quad \frac{1}{T_n} \sum_{t=1}^{T_n} -\log \sum_{m=1}^{M} \exp\Big(-d(x_{n,t}, \mu_{cm}) - \theta_m \Big) \;\geq\; 1 + \frac{1}{T_n} \sum_{t=1}^{T_n} d(x_{n,t}, \mu_{y_n m_{n,t}}) + \theta_{m_{n,t}}, \tag{3.27}$$

après application de l'inégalité softmax.

La fonction de perte s'écrit :

$$\begin{aligned} \mathbf{L} \;=\; \sum_{n=1}^{N} \sum_{c \neq y_n} \max \Bigg(0 \;,\; 1 + \frac{1}{T_n} \sum_{t=1}^{T_n} \Bigg(& d(x_{n,t}, \mu_{y_n m_{n,t}}) + \theta_{m_{n,t}} \\ & + \log \sum_{m=1}^{M} \exp\Big(-d(x_{n,t}, \mu_{cm}) - \theta_m \Big) \Bigg) \Bigg). \end{aligned} \tag{3.28}$$

En utilisant la méthode L-BFGS [Nocedal and Wright, 1999] qui sera introduite dans la prochaine section, le module d'apprentissage cherche finalement à trouver les vecteurs de moyennes μ_{cm} qui minimisent cette fonction de perte.

3.3.3 Traitement des données aberrantes

Nous adoptons la même stratégie que [Sha and Saul, 2006] pour détecter et traiter les données aberrantes. Nous utilisons les modèles GMM qui ont servi en phase d'initialisation pour compléter cette initialisation[13], en calculant les pertes accumulées dues aux violations des contraintes de grande marge de l'équation eq. 3.27 :

$$\begin{aligned} h_n^{MAP} \;=\; \sum_{c \neq y_n} \max\Bigg(0\;,\; 1 + \frac{1}{T_n}\sum_{t=1}^{T_n}\Bigg(d(x_{n,t}, \mu^{MAP}_{y_n m_{n,t}}) + \theta_{m_{n,t}} \\ + \log \sum_{m=1}^{M} \exp\Big(-d(x_{n,t}, \mu^{MAP}_{cm}) - \theta_m\Big)\Bigg)\Bigg). \end{aligned} \tag{3.29}$$

Nous pondérons donc les termes de pertes de l'équation eq. 3.28 par les poids $l_n = \min\left(1, \frac{1}{h_n^{MAP}}\right)$:

$$\mathbf{L} \;=\; \sum_{n=1}^{N} l_n h_n. \tag{3.30}$$

Pour résumer, notre algorithme simplifié d'apprentissage du modèle de mélange LM-dGMM consiste à :

- Disposer d'un modèle GMM universel,
- Initialiser le mélange LM-dGMM après avoir appris par adaptation MAP pour chaque classe c un GMM_c,
- Déterminer pour chaque vecteur de l'ensemble d'apprentissage, son label complet,
- Calculer les pondérations des segments d'apprentissage,
- Résoudre en utilisant l'algorithme L-BFGS, le problème d'optimisation non-linéaire sans contraintes défini par la fonction de perte de l'équation eq. 3.30.

3.4 Optimisation par l'algorithme L-BFGS

Cette section introduit la méthode L-BFGS utilisée pour résoudre notre problème d'optimisation ; elle peut aussi être utilisée pour apprendre les modèles LM-GMM originaux.

3.4.1 Introduction de la Méthode L-BFGS

L'optimisation différentiable consiste à minimiser une fonction réelle différentiable définie sur un espace hilbertien. Les algorithmes à directions de descente cherchent un minimum d'une fonction $f(x)$, en générant une suite de points $(x_k)_{k\in\mathbb{N}}$ qui diminuent la valeur

[13] Comme seules les moyennes seront apprises lors de l'apprentissage du mélange LM-dGMM, nous n'affectons pas dans les équations suivantes, l'exposant MAP aux paramètres de variance et d'offsets.

de la fonction. Ces algorithmes partent d'un point initial x_0 en suivant une direction dite de descente d_k, avec un pas de descente α_k :

$$x_{k+1} \;=\; x_k \;+\; \alpha_k d_k. \tag{3.31}$$

Les trois principales directions de descente sont :
- La direction du gradient $d_k = -\nabla f(x_k)$ qui est une direction lente.
- La direction de Newton $d_k = -\big(\nabla^2 f(x_k)\big)^{-1}\nabla f(x_k)$ qui nécessite la matrice hessienne définie positive.
- La direction de Quasi-Newton $d_k = -\mathrm{M}_k^{-1}\nabla f(x_k)$, où M_k est une approximation convenable de la matrice hessienne.

L'un des algorithmes de minimisation les plus utilisés est la méthode de Newton. Cette méthode est souvent très efficace, mais par contre elle est très coûteuse. Les algorithmes de type Quasi-Newton, la méthode *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) par exemple, consistent justement à remplacer l'inverse de la matrice hessienne par une suite d'approximations symétriques définies positives, avec des corrections de faible rang à chaque itération. L'inconvénient de ces algorithmes reste cependant le coût de stockage des variables calculées à chaque itération qui sont utilisées pour l'approximation de l'inverse de la matrice hessienne. La méthode *limited-memory BFGS* (L-BFGS ou LM-BFGS) est une extension à mémoire limitée de la méthode BFGS, où on ne stocke que les K (souvent $\mathrm{K} < 10$) dernières variables. De ce fait, la méthode L-BFGS est très utilisée dans les applications traitant de grands volumes de données.

La détermination du bon pas de descente α_k obéit à des règles de recherche linéaire comme : les règles exactes, la règle d'Armijo, la règle de Goldstein et les règles de Wolfe [Nocedal and Wright, 1999]. Pour K fixé, sont définies les variables $\hat{\mathrm{K}} = \min\{k, \mathrm{K}-1\}$, $s_k = x_{k+1} - x_k$, $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$, $\rho_k = \dfrac{1}{y_k^T s_k}$ et $V_k = \mathrm{I} - \rho_k y_k s_k^T$, et une matrice initiale symétrique et définie positive H_0. La formule de mise à jour de l'inverse de la matrice hessienne H_k[14] dans la méthode L-BFGS est donnée par [Liu and Nocedal, 1989] :

$$\begin{aligned}
\mathrm{H}_{k+1} \;=\; & \Big(V_k^T \cdots V_{k-\hat{\mathrm{K}}}^T\Big)\mathrm{H}_0\Big(V_{k-\hat{\mathrm{K}}} \cdots V_k\Big) \\
& +\rho_{k-\hat{\mathrm{K}}}\Big(V_k^T \cdots V_{k-\hat{\mathrm{K}}+1}^T\Big)s_{k-\hat{\mathrm{K}}}s_{k-\hat{\mathrm{K}}}^T\Big(V_{k-\hat{\mathrm{K}}+1} \cdots V_k\Big) \\
& +\rho_{k-\hat{\mathrm{K}}+1}\Big(V_k^T \cdots V_{k-\hat{\mathrm{K}}+2}^T\Big)s_{k-\hat{\mathrm{K}}+1}s_{k-\hat{\mathrm{K}}+1}^T\Big(V_{k-\hat{\mathrm{K}}+2} \cdots V_k\Big) \\
& \vdots \\
& \rho_k s_k s_k^T.
\end{aligned} \tag{3.32}$$

3.4.2 Résolution dans le cas LM-dGMM

Dans notre problème d'optimisation, la méthode L-BFGS nécessite le calcul du gradient de la fonction de perte par rapport à l'ensemble des vecteurs de moyennes. Chaque

[14] $\mathrm{H}_k = \mathrm{M}_k^{-1}$

segment de données x_n induit, en cas de violations des contraintes segmentales de grande marge, les dérivées :

$$\forall c \neq y_n, \forall m, \qquad \frac{\partial \mathbf{L}}{\partial \mu_{cm}} = \frac{l_n}{T_n} \sum_{t=1}^{T_n} \left(-\frac{\partial d(x_{n,t}, \mu_{cm})}{\partial \mu_{cm}} \frac{\exp\Big(- d(x_{n,t}, \mu_{cm}) - \theta_m \Big)}{\sum_k \exp\Big(- d(x_{n,t}, \mu_{ck}) - \theta_k \Big)} \right), \quad (3.33)$$

$$\frac{\partial \mathbf{L}}{\partial \mu_{y_n m_{n,t}}} = l_n \frac{nloss}{T_n} \frac{\partial d(x_{n,t}, \mu_{y_n m_{n,t}})}{\partial \mu_{y_n m_{n,t}}}, \quad (3.34)$$

où

$$\frac{\partial d(x_{n,t}, \mu_{cm})}{\partial \mu_{cmi}} = \frac{\mu_{cmi} - x_{nti}}{\sigma^2_{mi}} \quad (3.35)$$

et $nloss$ ($nloss \leq C - 1$) est le nombre de classes concurrentes qui enfreignent les contraintes de l'équation eq. 3.27 pour le segment x_n.

Dans notre implémentation de la méthode L-BFGS, nous avons utilisé, comme Fei SHA, un pas de descente obéissant aux conditions de Wolfe.

3.5 Première application : l'identification du locuteur

3.5.1 Le système d'identification du locuteur

Le système de classification mis en place a pour objet l'identification du locuteur ; bien évidemment, chaque locuteur correspond à une classe. Ce système se compose classiquement d'un module de traitement acoustique permettant d'extraire les observations du signal de parole et d'un module de décision basé sur différentes versions de l'approche LM-GMM.

3.5.1.1 Extraction de paramètres

Dans la mesure où nous utilisons les données délivrées par le NIST lors des campagnes d'évaluation et afin de pouvoir comparer les résultats expérimentaux à ceux d'autres systèmes de la littérature, le traitement acoustique que nous avons mis en place reste classique.

Paramétrisation

Le signal de parole est filtré de manière à ne garder que la bande de fréquence [300-3400]Hz. Il est ensuite analysé localement à l'aide d'un fenêtrage temporel de type Hamming. Des fenêtres glissantes de 20ms sont utilisées, à décalage régulier de 10ms. Des coefficients cepstraux LFCC sont calculés à partir d'un banc de 24 filtres à échelle linéaire. Le vecteur de paramètres se compose de 50 coefficients incluant 19 LFCC, leurs

dérivées premières, les 11 premières dérivées secondes et la dérivée première de l'énergie : $\mathbf{C_1}$ ... $\mathbf{C_{19}}$, $\mathbf{\Delta C_1}$ ... $\mathbf{\Delta C_{19}}$, $\mathbf{\Delta E}$, $\mathbf{\Delta\Delta C_1}$ ... $\mathbf{\Delta\Delta C_{11}}$. Cette paramétrisation est similaire à celle utilisée par le système de reconnaissance automatique du locuteur du LIA [Fauve et al., 2007] et elle est faite avec l'outil SPro [Gravier, 2003].

Détection de l'activité vocale VAD

La segmentation parole / non parole se base sur l'énergie du signal, et se fait après une normalisation moyenne variance CMVN du paramètre de l'énergie (normalisation par centrage/réduction de manière à avoir une moyenne nulle et une variance unitaire sur la totalité de la séquence de parole). La distribution énergétique des trames est modélisée par un modèle GMM à 3 composantes, et ne sont considérées comme parole que les trames à haute énergie, i.e., les trames assignées à la loi gaussienne de plus forte moyenne ; cette stratégie revient à seuiller l'énergie des trames.

La figure Fig. 3.3 illustre un exemple de segmentation parole / non parole, où le pourcentage de trames assignées à la gaussienne centrale de moyenne énergie défini la classe non parole (les valeurs d'énergies à couleur grise) et la classe parole (les valeurs d'énergies à couleur bleue).

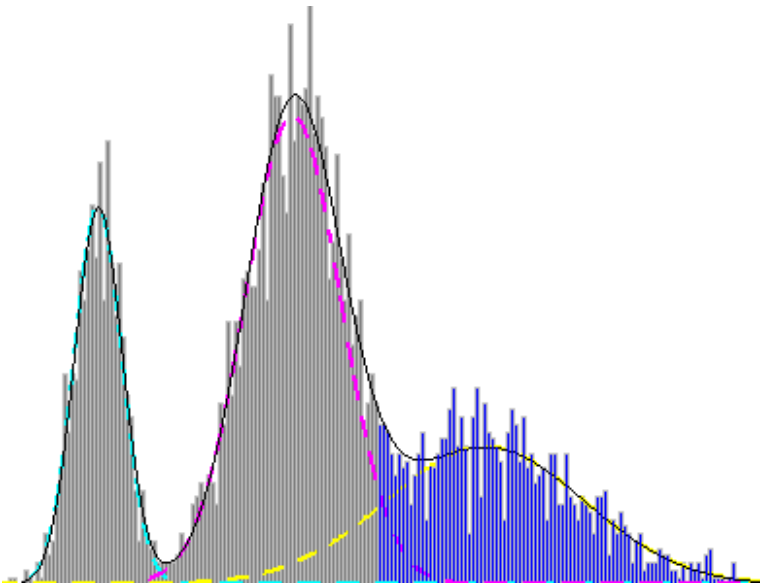


FIG. 3.3: *Exemple de VAD basée sur l'énergie.*

Normalisation des paramètres acoustiques

La normalisation des vecteurs paramétriques résultant de la VAD se fait en appliquant une normalisation CMVN. L'estimation de la moyenne et de la variance est réalisée sur l'intégralité de la séquence de parole [Viikki and Laurila, 1998].

3.5.1.2 Modélisation

Nous avons développé trois systèmes se différenciant au travers de leur module de décision :

- Le système, appelé par la suite système-GMM, est le système classique qui utilise l'approche probabiliste GMM-UBM, à savoir que chaque locuteur est représenté par un GMM, issu de l'adaptation MAP d'un modèle du monde.
- Le système, appelé par la suite système-LM-GMM, utilise un mélange LM-GMM.
- Le système, appelé par la suite système-LM-dGMM, utilise un mélange LM-dGMM.

Quelque soit le système, seule l'approche segmentale sera naturellement mise en oeuvre !

3.5.2 Protocole expérimental

Cette première famille d'expériences a pour but de vérifier la faisabilité et l'intérêt de l'approche LM-GMM par rapport à l'approche GMM et de régler les quelques paramètres nécessaires à la mise en oeuvre.

Corpus

Pour ce faire, le cadre retenu est celui de la tâche d'identification du locuteur de la campagne d'évaluation **NIST-SRE 2006** [NIST, 2006], sous la **condition principale (1conv4w-1conv4w)**. Nous sélectionnons un ensemble de **50 locuteurs masculins** (le nombre de classes est égal à 50) parmi les 349 locuteurs masculins présents dans la base. Le choix s'est porté sur les locuteurs qui ont le plus de segments de test disponibles.

Phase d'apprentissage

L'apprentissage des modèles de chaque locuteur est réalisé à partir des 5 minutes de parole disponibles, qui se réduisent, après segmentation (VAD), en moyenne à **3 minutes**. **Système-GMM : Les modèles GMM de ce système sont appris en utilisant l'outil ALIZE/Spkdet [Fauve et al., 2007], [Bonastre et al., 2008]. L'adaptation MAP utilise un UBM appris sur les données téléphoniques de tous les locuteurs masculins de NIST-SRE 2004 [NIST, 2004]. Les modèles des locuteurs sont dérivés par adaptation MAP de cet UBM avec un facteur de régulation de 14.**
Les GMM ainsi appris sont utilisés comme initialisation des modèles de base pour le système-LM-GMM et le système-LM-dGMM.

Pour des raisons de temps de calcul, le nombre maximal d'itérations de l'algorithme L-BFGS est fixé expérimentalement à 30 itérations dans le système-LM-dGMM et à 10 dans le système-LM-GMM.

Plusieurs systèmes sont appris pour un nombre de composantes par classe, égal à 16, 32 ou 64. Il est difficile d'envisager un ordre plus élevé compte tenu du volume des données d'apprentissage et du temps de calcul. Le coût en terme de ressources informatiques est linéaire en le nombre des composantes du mélange et en le nombre des vecteurs paramétriques. Dans le cas du modèle de mélange LM-GMM, le coût est aussi quadratique en la dimension des données, et il est linéaire en ce facteur dans le cas du mélange LM-dGMM.

A noter que, pour le système-LM-GMM, plusieurs systèmes sont appris avec différentes valeurs du paramètre de régularisation α pour juger de son impact.

Expériences

Les techniques de normalisation des scores et de compensation de la variabilité inter-sessions ne sont pas utilisées dans cette première série de tests.

Nous n'utilisons pas de données de développement, vu qu'on ne dispose que d'un seul fichier d'apprentissage dans la condition principale de NIST-SRE 2006 (1conv4w-1conv4w). Nous suivons l'évolution des performances des systèmes au cours des itérations successives de l'apprentissage, sur les données de test et nous donnons les résultats du système considéré le meilleur au cours de ce processus itératif et le résultat à l'issue de la dernière itération. À l'instar des systèmes état de l'art, la décision dans le système-GMM est faite en ne tenant compte que des 10-meilleures gaussiennes de chaque modèle.

La fiabilité de l'estimation d'un taux de reconnaissance sur une base donnée dépend du nombre de tests réalisés, i.e., du nombre d'entités à reconnaître. L'intervalle de confiance permet d'évaluer la précision de l'estimation, ce qui permet de juger si une variation du taux de reconnaissance est significative ou non. Dans le cas où le nombre d'échantillons à reconnaître N est grand, l'intervalle de confiance (en %) à 95% est approché par :

$$I_c \;=\; \left[P \;-\; 1,96\sqrt{\frac{P(1-P)}{N}}100 \;;\; P \;+\; 1,96\sqrt{\frac{P(1-P)}{N}}100 \right], \qquad (3.36)$$

où P est le taux de reconnaissance mesuré [Montacié and Chollet, 1987].

Les tests se font sur des segments d'une durée de 5 minutes réduite à 3 minutes en moyenne (après segmentation). Les évaluations des différents systèmes sont obtenues sur une base comprenant 232 segments, à raison de 4 en moyenne par locuteur.

3.5.3 Performances

Avant de donner les performances des différents systèmes, il convient de préciser le rôle du coefficient de régularisation proposé par Fei SHA.

3.5.3.1 Impact du terme de régularisation

Pour le système-LM-GMM, l'algorithme cherche à minimiser conjointement le terme pénalisant les violations de marge et le terme régularisant les matrices d'orientation. Plusieurs valeurs de l'hyperparamètre α, i.e., l' hyperparamètre pondérant les deux termes de la fonction objective, ont été essayées. L'analyse de l'ensemble des résultats montre que l'algorithme a tendance à minimiser le terme de régularisation au détriment de celui des pertes dues aux violations de marge. On retrouve ce comportement dans tous les tests réalisés, et ce phénomène se traduit par une baisse considérable des performances par rapport aux modèles initiaux.

Afin d'illustrer cette remarque, la figure Fig. 3.4 montre la variation de chacun des deux termes de la fonction de perte durant le processus d'optimisation, pour le système-LM-GMM à 16 composantes et $\alpha = 1$. Cette configuration donne un taux de bonne identification de 7.8%.

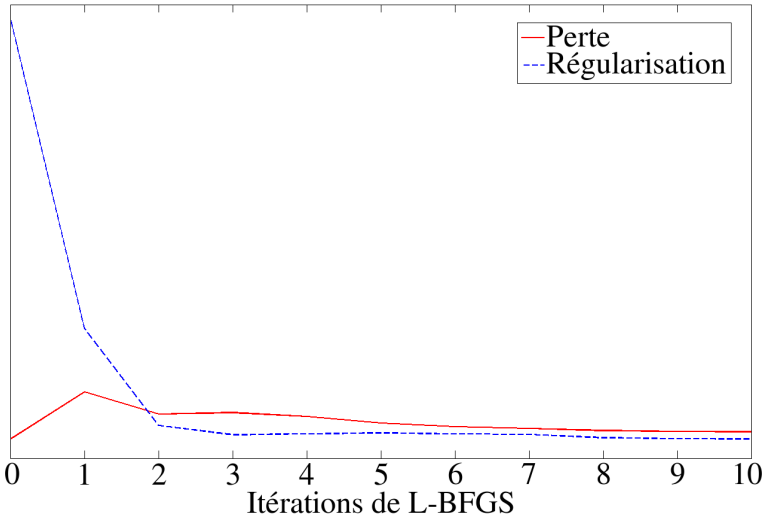


FIG. 3.4: *Variation des termes de perte et de régularisation de la fonction de perte du système-LM-GMM, pour 16 composantes et* $\alpha = 1$.

Nous éliminons par conséquent le terme de régularisation dans tous les systèmes LM-GMM.

3.5.3.2 Comparaison des systèmes

Comme dit précédemment, trois types de systèmes sont évalués : le système GMM traditionnel, le système mélange LM-GMM segmental et notre système LM-dGMM dans sa version segmentale ; trois configurations sont présentées avec 16, 32 et 64 composantes par classe. Avant d'analyser les résultats obtenus, il convient de signaler que les configurations testées du système-GMM ne peuvent être considérées comme "baselines" en

reconnaissance du locuteur, étant donné qu'en général les systèmes GMM utilisent de 512 à 2048 composantes. Notre objectif est de confronter l'apprentissage génératif (par adaptation MAP) aux apprentissages discriminants à grande marge.

L'ensemble des figures Fig. 3.5, Fig. 3.6, Fig. 3.7, Fig. 3.8, Fig. 3.9 et Fig. 3.10 montrent les variations de la valeur de la fonction de perte et des taux d'identification correcte sur la base de test, pour chacun des systèmes LM-GMM et LM-dGMM, obtenus après chaque itération, durant le processus d'optimisation. Sur chaque figure donnant le taux d'identification, est également reporté par une ligne horizontale le taux d'identification correcte obtenu par le système GMM, considéré comme système initial.

Les tableaux Tab. 3.1, Tab. 3.2 et Tab. 3.3 résument ces figures en donnant deux taux d'identification correcte pour chacun des systèmes LM-GMM et LM-dGMM : le meilleur taux d'identification au cours des itérations successives et le taux d'identification correcte à la dernière itération du processus d'optimisation (dernière itération de l'algorithme L-BFGS). Cette dernière est inférieure ou égale au nombre maximal d'itérations à atteindre (10 pour le système-LM-GMM et 30 pour le système-LM-dGMM) ; cela est dû à l'annulation de la fonction objective durant l'optimisation.

Les taux des systèmes LM-GMM et LM-dGMM sont comparés à celui obtenu par le système GMM. Compte tenu du nombre d'observations en test et du taux de reconnaissance de l'ordre de 65%, l'intervalle de confiance à 95% est en moyenne de $\pm$ 6%.

LM-GMM versus LM-dGMM

Dans les trois configurations testées, il apparaît clairement que l'approche LM-dGMM annule la fonction de perte plus rapidement que l'approche classique LM-GMM. Notre système surclasse clairement les modèles LM-GMM à 16 gaussiennes, et les deux systèmes affichent des performances comparables avec des modèles à 32 gaussiennes. Bien qu'on observe un aspect chaotique des taux d'identification correcte dans les deux cas, l'approche LM-dGMM se comporte mieux que l'approche LM-GMM pour deux raisons :

- Les taux d'identification restent au moins égaux aux taux des systèmes initiaux.
- Avec 16 ou 32 composantes, les taux d'identification correcte augmentent systématiquement au cours des premières itérations.

Ces résultats indiquent que la stratégie consistant à définir des modèles GMM à grande marge avec pour seuls paramètres les vecteurs de moyennes est très satisfaisante par rapport à l'approche originale.

Système		Taux de bonne identification
GMM		61.2 %
LM-GMM	meilleure itération (it. 1)	**60.3 %**
	dernière itération (it. 10)	59.1 %
LM-dGMM	meilleure itération (it. 6)	**67.7 %**
	dernière itération (it. 30)	59.5 %

TAB. 3.1: Taux de bonne identification de systèmes GMM, LM-GMM et LM-dGMM à 16 gaussiennes.

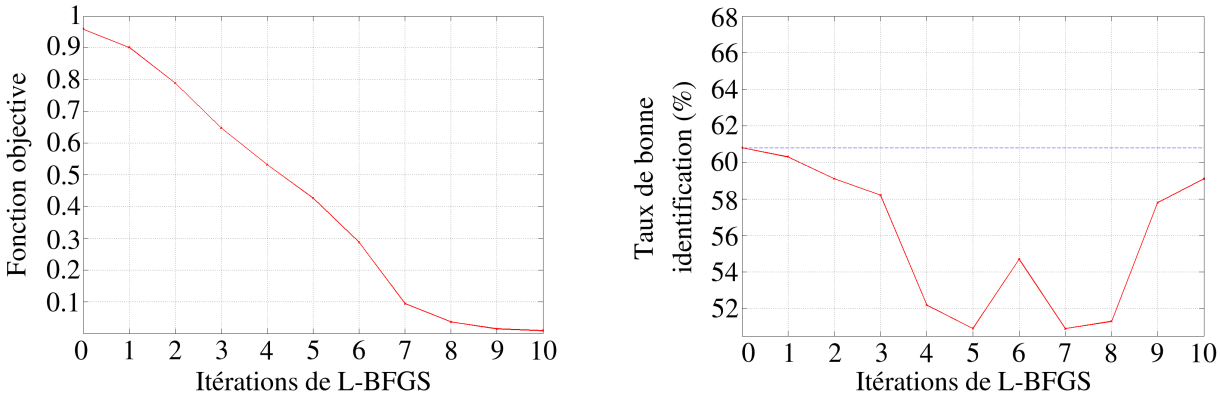


FIG. 3.5: *Variation de la fonction objective et du taux de bonne identification, des modèles LM-GMM à 16 gaussiennes.*

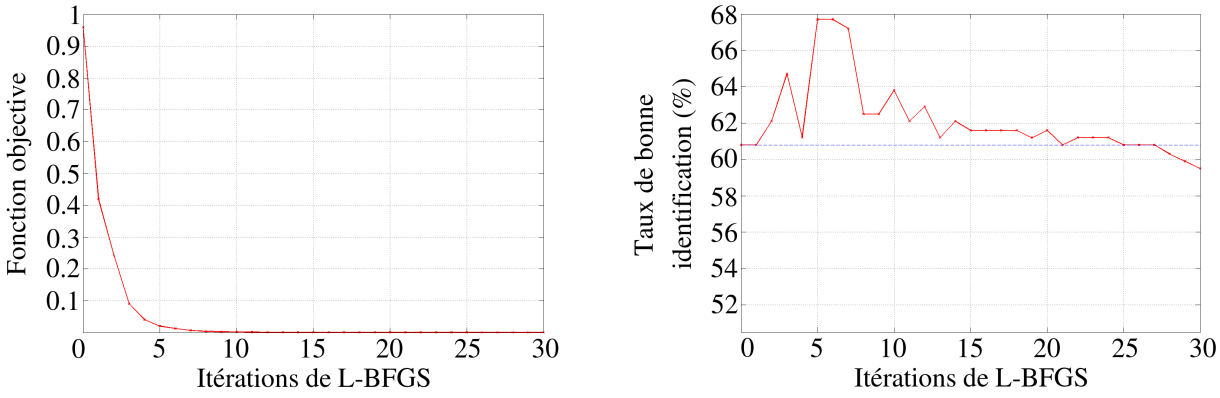


FIG. 3.6: *Variation de la fonction objective et du taux de bonne identification, des modèles LM-dGMM à 16 gaussiennes.*

Système		Taux de bonne identification
GMM		68.1 %
LM-GMM	meilleure itération (it. 3)	**72.0 %**
	dernière itération (it. 10)	65.9 %
LM-dGMM	meilleure itération (it. 2)	**71.6 %**
	dernière itération (it. 10)	69.0 %

TAB. 3.2: Taux de bonne identification de systèmes GMM, LM-GMM et LM-dGMM à 32 gaussiennes.

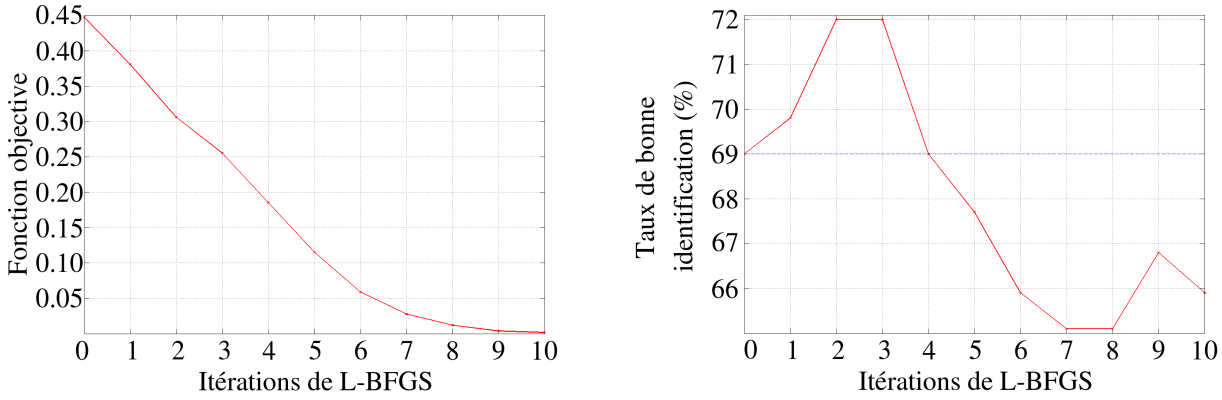


FIG. 3.7: *Variation de la fonction objective et du taux de bonne identification, des modèles LM-GMM à 32 gaussiennes.*

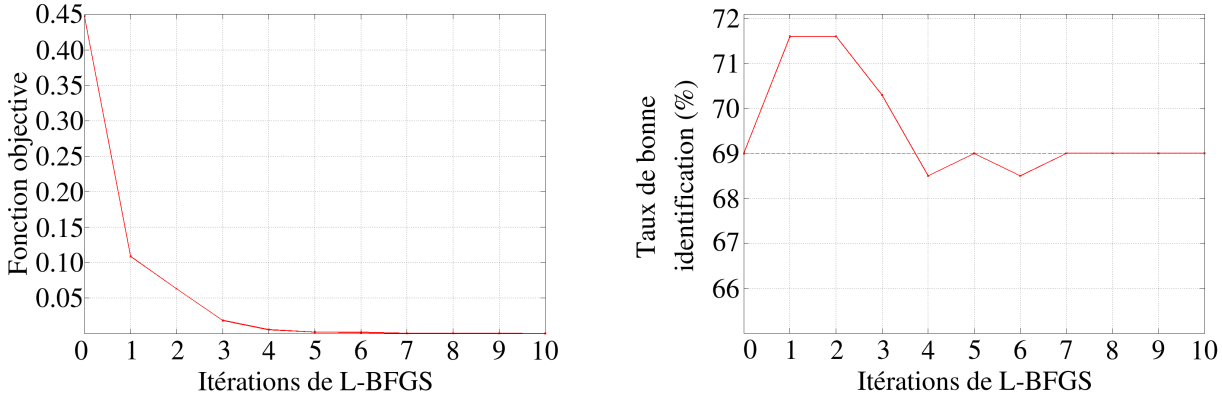


FIG. 3.8: *Variation de la fonction objective et du taux de bonne identification, des modèles LM-dGMM à 32 gaussiennes.*

LM-dGMM versus GMM

Les résultats des tableaux Tab. 3.1 et Tab. 3.2 révèlent que les deux algorithmes d'apprentissage discriminant permettent d'atteindre au cours des itérations de meilleures performances que le système GMM, et ce dans les deux configurations 16 et 32.

L'approche LM-dGMM permet cette amélioration tout en réduisant considérablement la complexité algorithmique de l'approche LM-GMM original.

L'analyse des courbes de variation des taux de bonne identification des systèmes à 16 et 32 gaussiennes, révèle que nos performances passent par un optimal avant de baisser en fonction du nombre d'itérations de l'algorithme L-BFGS. Nous améliorons les performances des GMM classiques dès les premières itérations. Sans utiliser de données de développement, il suffirait donc juste de prendre les modèles obtenus au début du processus d'optimisation pour améliorer les performances des modèles initiaux. Nous pouvons aussi constater que les modèles obtenus aux dernières itérations ont des performances proches ou égales à celles des modèles initiaux, ce qui n'est pas le cas avec les modèles de mélange LM-GMM de [Sha and Saul, 2006].

Lors du passage à des modèles à 64 gaussiennes, le tracé de la fonction de perte montre que les modèles GMM initiaux satisfont un nombre important de contraintes de grande marge, i.e., il y a très peu de violations enregistrées. C'est consistent avec le fait que l'augmentation du nombre de composantes permet d'approcher la vraie distribution. Cette augmentation fait que le nombre de données d'apprentissage par composante diminue. L'utilisation de plus de données d'apprentissage ou la définition de nouvelles contraintes de grande marge permettra de bénéficier dans ce cas là du pouvoir discriminant des modèles LM-dGMM.

Système		Taux de bonne identification
GMM		69.8 %
LM-GMM	meilleure itération (it. 8)	68.5 %
	dernière itération (it. 9)	67.7 %
LM-dGMM	meilleure itération (it. 2)	69.4 %
	dernière itération (it. 3)	69.0 %

TAB. 3.3: Taux de bonne identification de systèmes GMM, LM-GMM et LM-dGMM à 64 gaussiennes.

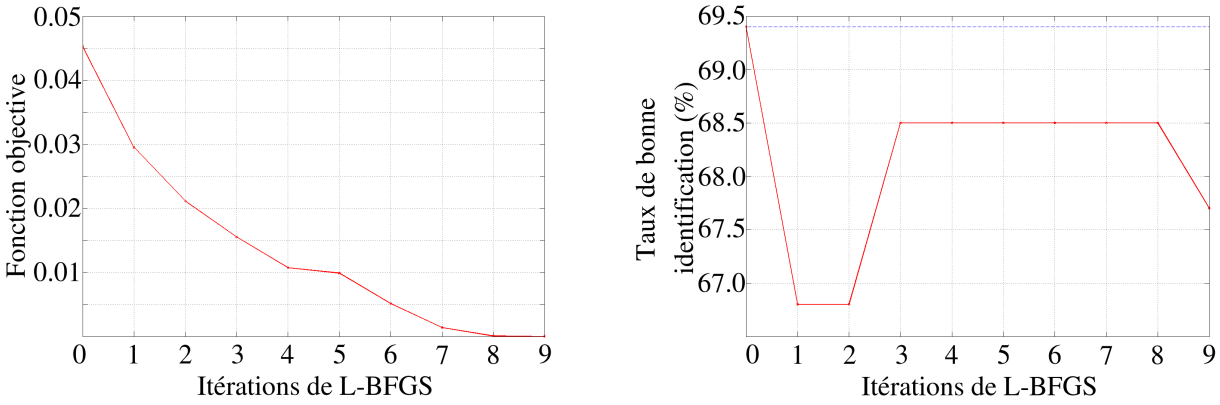


FIG. 3.9: *Variation de la fonction objective et du taux de bonne identification, des modèles LM-GMM à 64 gaussiennes.*

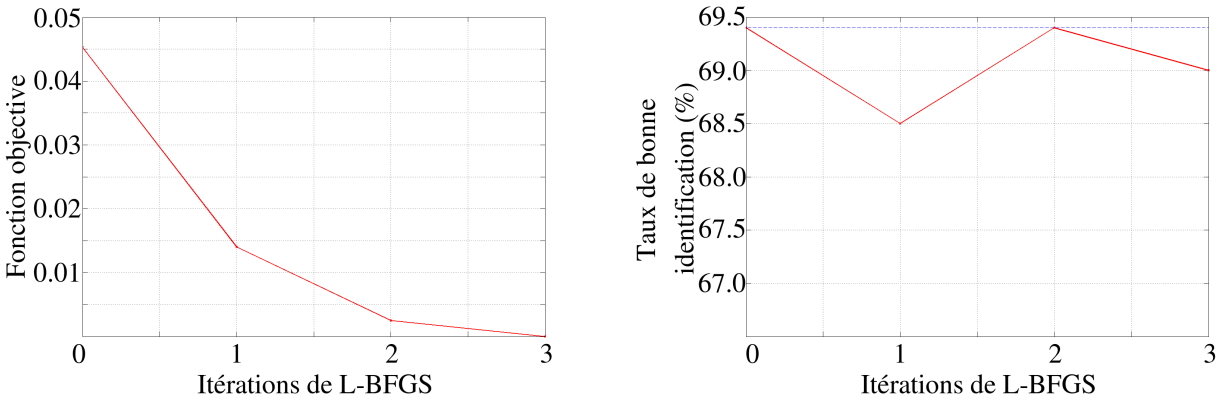


FIG. 3.10: *Variation de la fonction objective et du taux de bonne identification, des modèles LM-dGMM à 64 gaussiennes.*

3.6 Conclusion

Nous avons présenté dans ce chapitre, les modèles GMM à grande marge. Le modèle original proposé par Fei SHA représente dans l'espace des données chaque classe par un ellipsoïde construit à partir d'une loi gaussienne. L'apprentissage de ces modèles se fait au sens maximisation de la marge entre les différents ellipsoïdes. Le formalisme des LM-GMM allie l'approche probabiliste des lois gaussiennes et la notion de marge de séparation. S'inspirant de ces modèles discriminants, nous avons proposé de nouveaux GMM à grande marge qui exploitent les propriétés des systèmes GMM état de l'art utilisés en reconnaissance automatique du locuteur, à savoir leur apprentissage par adaptation MAP de uniquement des vecteurs moyennes d'un modèle du monde ; les matrices de

covariance **diagonales** et les poids restent inchangés. En définissant des contraintes de grande marge sur uniquement les vecteurs moyennes, l'algorithme d'apprentissage devient beaucoup moins complexe.

Dans une application d'identification du locuteur avec des modèles à 16 composantes, nos modèles LM-dGMM simplifiés donnent de bien meilleurs résultats que les modèles LM-GMM originaux. Ils atteignent également de meilleures performances que les modèles GMM génératifs classiques. Les taux de bonne identification des systèmes GMM, LM-GMM et LM-dGMM sont respectivement 61.2%, 60.3% et 67.7%. Avec des modèles à 32 gaussiennes, l'apprentissage discriminant permet encore une fois d'améliorer les performances. Quand on a des modèles à plus de gaussiennes, la meilleure initialisation et la diminution du nombre de données d'apprentissage associées à chaque vecteur moyenne font que l'apprentissage à grande marge faille à améliorer les performances.

Nous allons étudier dans le chapitre 4 des variantes de modèles LM-dGMM qui abordent ces deux problématiques, via une "moins" bonne initialisation des modèles à grande marge et une restriction des contraintes de grande marge à uniquement un sous ensemble des vecteurs moyennes.

Chapitre 4

Variantes à partir du système basé sur le modèle de mélange LM-dGMM

Afin d'approfondir notre connaissance du modèle de mélange LM-dGMM et éventuellement d'obtenir des performances supérieures, nous nous sommes penchés sur trois problèmes :

- Une revisite de l'initialisation des modèles, quelque soit l'approche, LM-GMM ou LM-dGMM, afin d'éviter de faire appel à l'adaptation MAP des GMM et d'avoir en toute rigueur à définir un ensemble de développement.
- Une prise en compte des "k" meilleures composantes dans chaque mélange LM-dGMM, pour réduire les temps d'apprentissage comme ceux de reconnaissance. L'objectif est de rendre accessible par cette modélisation le traitement de très grands volumes de données.
- Une nouvelle stratégie des données aberrantes.

De nombreuses expérimentations sont réalisées afin de valider ces propositions.

De plus, afin de pouvoir juger de la puissance de notre approche, nous avons également modifié la stratégie de décision et l'avons rendue compatible avec une tâche de vérification du locuteur. Bien que cette modification soit artificielle et soulève de nouveaux problèmes, elle permet de juger du réel pouvoir discriminant de ces modèles par rapport aux approches courantes (supervecteurs GMM, SVM, SFA ...).

4.1 Initialisation des mélanges LM-GMM et LM-dGMM par le modèle du monde

Comme il est dit au chapitre précédent (sections 3.2.3 et 3.3.1.1), l'initialisation des mélanges LM-GMM et LM-dGMM, avant apprentissage, est jusqu'alors réalisé avec des

modèles GMM obtenus par adaptation de type MAP d'un modèle du monde noté UBM. D'une part, cette adaptation MAP implique, en toute rigueur d'utiliser un autre ensemble de données pour réaliser l'apprentissage des mélanges LM-GMM et LM-dGMM ; d'autre part, comme le montre le suivi des itérations lors de l'apprentissage, il n'est pas certain de tirer le meilleur profit de cet apprentissage du fait d'une initialisation quasi idéale du point de vue probabiliste !

Au lieu d'initialiser lors de leur apprentissage les modèles à grande marge par les GMM appris par adaptation MAP (nous appellerons cet apprentissage des LM-GMM et LM-dGMM, l'apprentissage GMM), nous nous proposons de les initialiser par le modèle du monde lui même. De ce fait, toutes les classes sont représentées par le même mélange initial ! Nous appellerons l'apprentissage des LM-GMM et LM-dGMM, avec cette initialisation, l'apprentissage UBM.

Pour l'étude expérimentale, nous gardons le même protocole expérimental que celui de la section 3.5.2 du précédent chapitre : chaque classe correspond à un des 50 locuteurs, extraits de la campagne d'évaluation NIST-SRE 2006 et l'ensemble d'apprentissage de chaque mélange LM-GMM ou LM-dGMM est réalisé à partir de 3 minutes de parole effective en moyenne. L'évaluation est faite sur les mêmes 232 séquences de parole de test.

Les tableaux Tab. 4.1, Tab. 4.2 et Tab. 4.3 présentent les taux d'identification correcte des systèmes basés GMM, mélanges LM-GMM et mélanges LM-dGMM avec un nombre de composantes M égal respectivement à 16, 32 et 64, et ce en fonction du type d'apprentissage GMM ou UBM, à des fins de comparaison avec les résultats du chapitre précédent. Ces tableaux sont accompagnés des figures Fig. 4.1, Fig. 4.2, Fig. 4.3, Fig. 4.4, Fig. 4.5 et Fig. 4.6 qui montrent les variations de la valeur de la fonction objective et des taux d'identification correcte durant le processus itératif d'optimisation. La ligne horizontale des figures donnant le taux d'identification reporte le taux obtenu par l'approche classique GMM-UBM. Il est rappelé que compte tenu du nombre d'observations en test et du taux de reconnaissances de l'ordre de 65%, l'intervalle de confiance est en moyenne de $\pm$ 6%.

De manière générale, alors qu'il est rassurant de constater que les taux d'identification correcte obtenus après l'apprentissage UBM sont en général au moins aussi élevés qu'après l'apprentissage GMM. Il convient néanmoins de relever certaines différences de comportement selon que le modèle est un mélange LM-GMM ou un mélange LM-dGMM.

1. Pour des **modèles à 16 composantes**, l'apprentissage UBM ne modifie que très peu les performances obtenues par le mélange LM-GMM : une baisse non significative est observée. L'apprentissage UBM est plus favorable à l'approche LM-dGMM : la fonction de perte diminue moins rapidement mais le taux d'identification correcte est légèrement augmenté. La courbe du taux de reconnaissance montre que le critère d'arrêt de l'algorithme d'apprentissage basé sur un nombre maximum d'itérations se justifie tout à fait : même si cette courbe n'est pas strictement croissante, les variations restent très minimes sans valeurs significatives.

Système			Taux de bonne identification
GMM			61.2%
Initialisation par les GMM	LM-GMM	meilleure itération (it. 1)	**60.3%**
		dernière itération (it. 10)	59.1%
	LM-dGMM	meilleure itération (it. 6)	**67.7%**
		dernière itération (it. 30)	59.5%
Initialisation par l'UBM	LM-GMM	meilleure itération (it. 6)	**59.1%**
		dernière itération (it. 9)	57.8%
	LM-dGMM	meilleure itération (it. 29)	**69.4%**
		dernière itération (it. 30)	69.0%

TAB. 4.1: Taux de bonne identification de systèmes GMM, LM-GMM et LM-dGMM à 16 gaussiennes : Apprentissage GMM vs Apprentissage UBM.

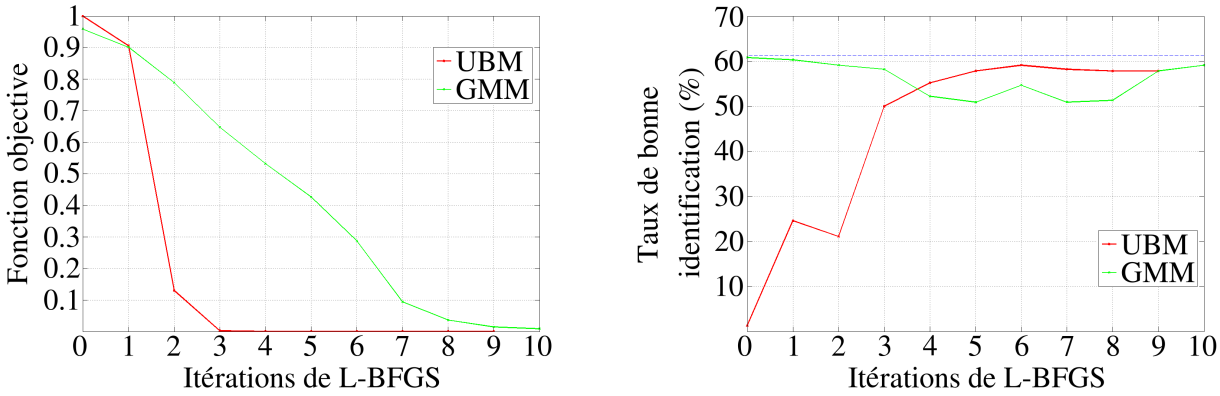


FIG. 4.1: *Variation de la fonction objective et du taux de bonne identification, des modèles LM-GMM à 16 gaussiennes : Apprentissage GMM vs Apprentissage UBM.*

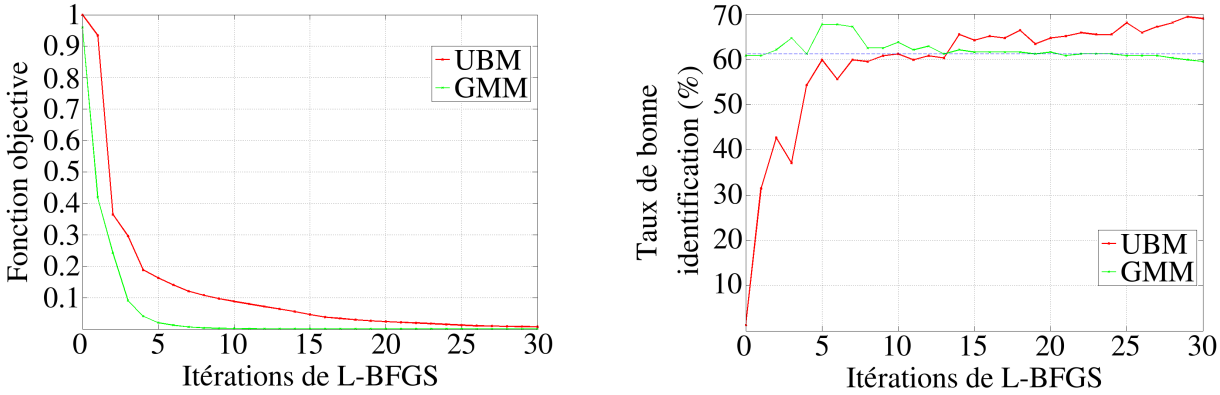


FIG. 4.2: *Variation de la fonction objective et du taux de bonne identification, des modèles LM-dGMM à 16 gaussiennes : Apprentissage GMM vs Apprentissage UBM.*

2. Pour des **modèles à 32 composantes**, l'apprentissage UBM n'est absolument pas bénéfique au mélange LM-GMM, et ce de manière significative : il est observé une perte du taux de reconnaissance de plus de 8% ! Ce comportement semble être corrélé à une forte diminution de la fonction de perte dès les premières itérations. L'apprentissage UBM et l'apprentissage GMM donnent des résultats très comparables pour le mélange LM-dGMM. La fonction de perte diminue moins rapidement avec l'approche UBM et elle permet une stabilisation du modèle pour atteindre un taux de reconnaissance autour de 69%.

Système			Taux de bonne identification
GMM			68.1%
Initialisation par les GMM	LM-GMM	meilleure itération (it. 3)	**72.0%**
		dernière itération (it. 10)	65.9%
	LM-dGMM	meilleure itération (it. 2)	**71.6%**
		dernière itération (it. 10)	69.0%
Initialisation par l'UBM	LM-GMM	meilleure itération (it. 7)	**57.8%**
		dernière itération (it. 7)	57.8%
	LM-dGMM	meilleure itération (it. 15)	**69.8%**
		dernière itération (it. 30)	68.5%

TAB. 4.2: Taux de bonne identification de systèmes GMM, LM-GMM et LM-dGMM à 32 gaussiennes : Apprentissage GMM vs Apprentissage UBM.

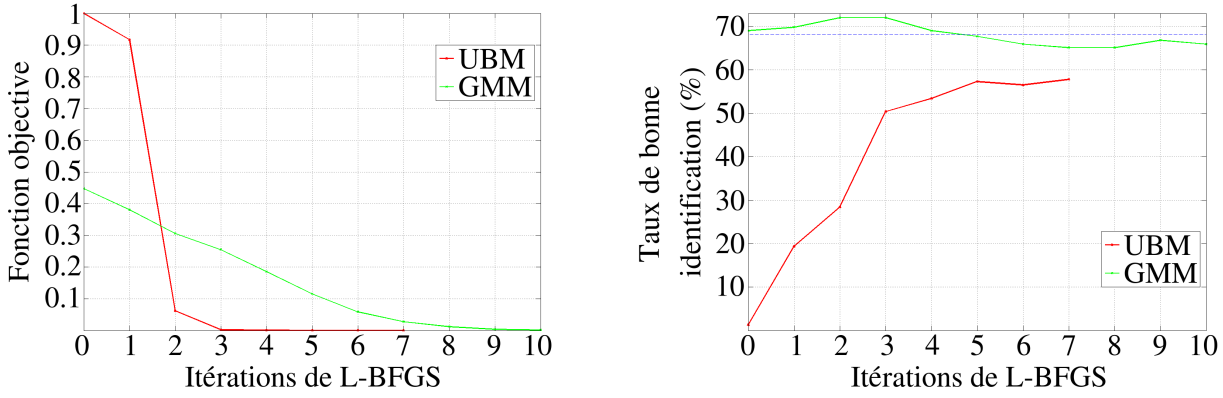


FIG. 4.3: *Variation de la fonction objective et du taux de bonne identification, des modèles LM-GMM à 32 gaussiennes : Apprentissage GMM vs Apprentissage UBM.*

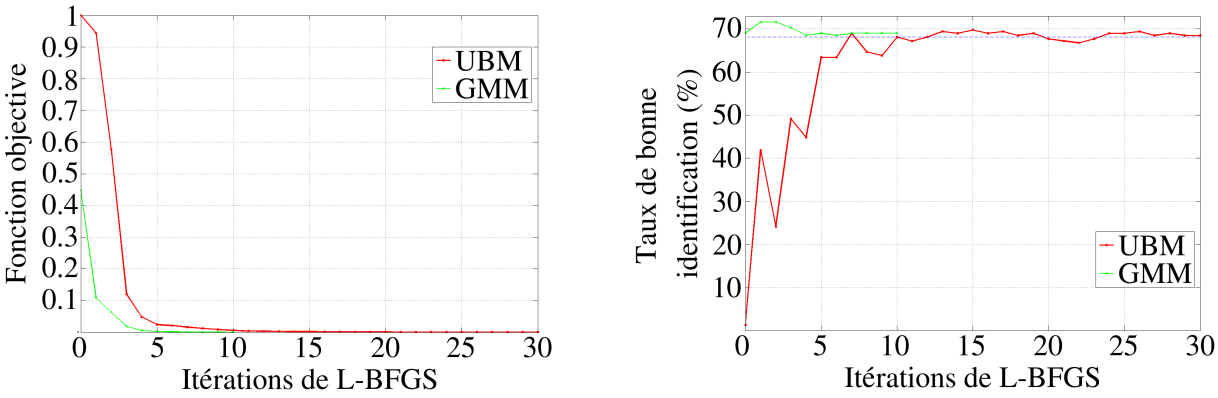


FIG. 4.4: *Variation de la fonction objective et du taux de bonne identification, des modèles LM-dGMM à 32 gaussiennes : Apprentissage GMM vs Apprentissage UBM.*

3. Pour des **modèles à 64 composantes**, la tendance observée précédemment s'accentue. L'apprentissage UBM est très défavorable au mélange LM-GMM alors qu'il est bénéfique pour le mélange LM-dGMM. Comme précédemment, avec l'apprentissage UBM, la fonction de perte s'annule très rapidement pour le mélange LM-GMM et le taux de reconnaissance plafonne à 51%, en chute de 18% par rapport à l'apprentissage GMM. A l'opposé, le taux de reconnaissance obtenu avec le mélange LM-dGMM est augmenté de plus de 4% pour dépasser la valeur de 73%. Cette augmentation est également très intéressante par rapport à l'approche classique GMM-UBM.

Système			Taux de bonne identification
GMM			69.8%
Initialisation par les GMM	LM-GMM	meilleure itération (it. 8)	68.5%
		dernière itération (it. 9)	67.7%
	LM-dGMM	meilleure itération (it. 2)	69.4%
		dernière itération (it. 3)	69.0%
Initialisation par l'UBM	LM-GMM	meilleure itération (it. 5)	**51.7%**
		dernière itération (it. 5)	51.7%
	LM-dGMM	meilleure itération (it. 14)	**73.7%**
		dernière itération (it. 17)	73.3%

TAB. 4.3: Taux de bonne identification de systèmes GMM, LM-GMM et LM-dGMM à 64 gaussiennes : Apprentissage GMM vs Apprentissage UBM.

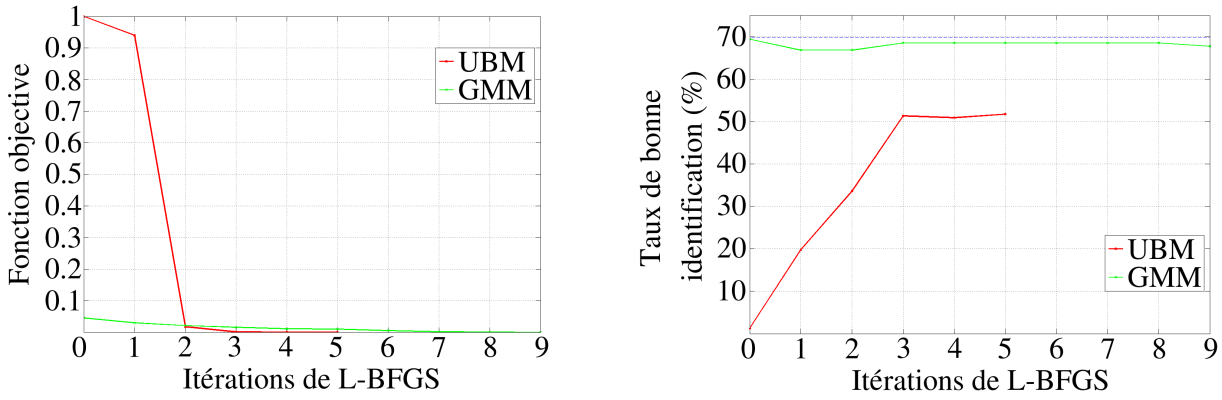


FIG. 4.5: *Variation de la fonction objective et du taux de bonne identification, des modèles LM-GMM à 64 gaussiennes : Apprentissage GMM vs Apprentissage UBM.*

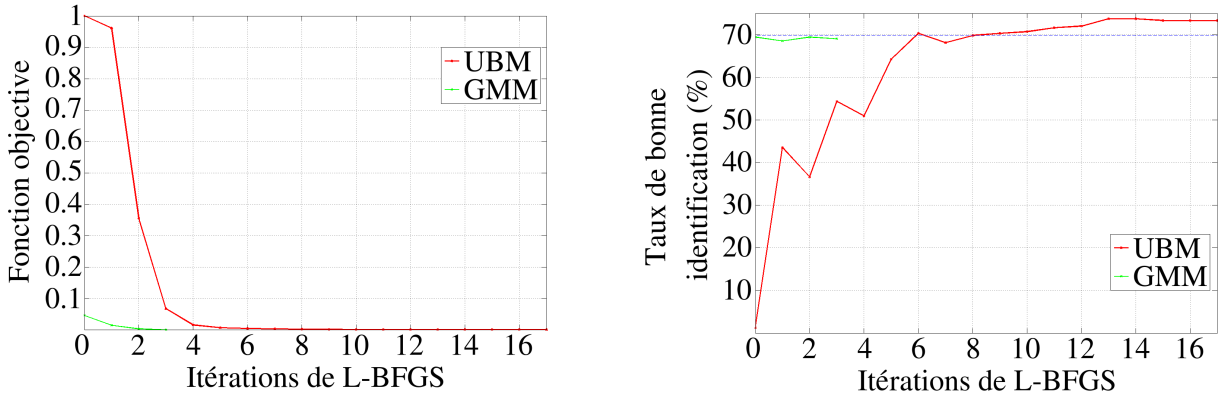


FIG. 4.6: *Variation de la fonction objective et du taux de bonne identification, des modèles LM-dGMM à 64 gaussiennes : Apprentissage GMM vs Apprentissage UBM.*

En conclusion, avec l'apprentissage UBM, l'algorithme d'apprentissage des LM-GMM semble nécessiter moins d'itérations pour converger, la fonction de perte diminue plus rapidement ; ce fait semble être lié à une diminution du taux de reconnaissance des modèles LM-GMM qui deviennent moins bons que les modèles GMM génératifs de base. Nous observons un comportement opposé avec les mélanges LM-dGMM ; la fonction de perte diminue moins rapidement, la convergence est plus lente mais elle assure une meilleure stabilité du taux de reconnaissance à une valeur de plus, plus élevée.

Ceci confirme le fait que la séparation entre classes est assurée majoritairement par l'éloignement des vecteurs de moyennes et que les matrices de covariance et les poids interviennent moindrement.

Nous notons aussi que les performances de classification obtenues dans les dernières itérations du processus d'optimisation sont les meilleures. Cet apprentissage progressif réalisé à partir du modèle UBM permet au processus d'optimisation d'atteindre en fin d'itérations un modèle correspondant aux meilleures performances. Cette approche permet

de profiter pleinement du pouvoir discriminant des modèles à grande marge, sans aucun recours aux données de développement. Il devient inutile d'apprendre des modèles GMM par adaptation MAP et par conséquent le temps nécessaire à l'apprentissage des modèles de locuteurs est diminué.

4.2 Apprentissage des modèles LM-dGMM restreint aux k-meilleures gaussiennes

Malgré le fait que l'apprentissage des LM-dGMM soit plus rapide que celui des LM-GMM, sa complexité algorithmique reste encore trop élevée pour traiter efficacement de grands volumes de données. Des expériences avec seulement une cinquantaine de locuteurs modélisés par des modèles "complexes" (modèles à plus de 64 composantes gaussiennes) nécessitent un temps de calcul considérable : l'apprentissage et l'évaluation de cinquante modèles LM-dGMM de base à 256 gaussiennes ($M = 256$) dure 9 jours et demi , sur un processeur Intel XEON 64bits 3.16GHz avec 6MO de cache L2. L'apprentissage des modèles a été fait en moyenne sur à peu près 3 minutes de parole effective, ce qui correspond à peu près à 7285 vecteurs paramétriques (observations). Toutes les durées que nous allons donner par la suite correspondront au temps de calcul nécessaire sur une architecture matérielle pareille.

Par conséquent, les applications impliquant un grand nombre de classes et de surcroît de grands volumes de données, e.g., comme dans les campagnes d'évaluation NIST-SRE où des centaines voire des milliers de locuteurs sont disponibles, seraient impossibles à réaliser dans un temps raisonnable.

4.2.1 Principe de réduction du nombre de composantes

Afin de pouvoir utiliser nos modèles LM-dGMM dans des applications réelles de reconnaissance de locuteurs, nous proposons de réduire considérablement le nombre de contraintes de grande marge à satisfaire (de l'équation eq. 3.27 du chapitre 3), réduisant ainsi la complexité calculatoire de la fonction de perte et de son gradient par rapport aux vecteurs de moyennes μ_{cm} ; le calcul de la fonction de perte et le calcul de son gradient sont les entités qui demandent le plus de temps de calcul. De plus, en réduisant le nombre de composantes sur lesquelles les contraintes portent, ceci nous permet de disposer de beaucoup plus de données (d'exploiter toutes les données d'apprentissage) en la discrimination de cet ensemble de composantes.

Pour ce faire, nous nous inspirons d'une heuristique classiquement utilisée dans les systèmes basés GMM, à savoir que la décision est inchangée si l'on prend en compte les scores issus des k-meilleures composantes gaussiennes et non ceux de l'ensemble des lois gaussiennes.

Pour chaque vecteur $x_{n,t}$ du nième segment d'apprentissage associé à la classe y_n, nous relaxons les contraintes à satisfaire en ne considérant pour chaque classe c autre que la classe y_n que les k lois gaussiennes les plus vraisemblables ; nous supposons de cette manière que les autres composantes en étant peu vraisemblables sont suffisamment éloignées de la composante dont est issu le plus probablement le vecteur.

Pour ce vecteur $x_{n,t}$ et chacune des $C-1$ classes autres que y_n, on note $S^{\mathbf{c}}_{n,t}$ l'ensemble des k composantes gaussiennes les plus vraisemblables. Les contraintes de grande marge deviennent :

$$\forall c \neq y_n,\ \frac{1}{T_n}\sum_{t=1}^{T_n} -\log \sum_{m\in S^{\mathbf{c}}_{n,t}} \exp\Big(-d(x_{n,t},\mu_{cm})-\theta_m\Big) \ \geq\ 1+\frac{1}{T_n}\sum_{t=1}^{T_n} d(x_{n,t},\mu_{y_n m_{n,t}})+\theta_{m_{n,t}}. \tag{4.1}$$

La fonction de perte à minimiser devient alors :

$$\begin{aligned}\mathbf{L} \ = \ & \sum_{n=1}^{N} l_n \sum_{c\neq y_n} max\Bigg(0\ ,\ 1+\frac{1}{T_n}\sum_{t=1}^{T_n}\Bigg(d(x_{n,t},\mu_{y_n m_{n,t}})+\theta_{m_{n,t}} \\ & +\log\sum_{m\in S^{\mathbf{c}}_{n,t}} \exp\Big(-d(x_{n,t},\mu_{cm})-\theta_m\Big)\Bigg)\Bigg),\end{aligned} \tag{4.2}$$

où

$$\begin{aligned} l_n \ &= \ \min\left(1,\frac{1}{h_n^{MAP}}\right), \\ h_n^{MAP} \ &= \ \sum_{c\neq y_n}\max\Bigg(0\ ,\ 1+\frac{1}{T_n}\sum_{t=1}^{T_n}\Bigg(d(x_{n,t},\mu^{MAP}_{y_n m_{n,t}})+\theta_{m_{n,t}} \\ &\qquad +\log\sum_{m\in S^{\mathbf{c}}_{n,t}} \exp\Big(-d(x_{n,t},\mu^{MAP}_{cm})-\theta_m\Big)\Bigg)\Bigg). \end{aligned} \tag{4.3}$$

La stricte mise en oeuvre de cette stratégie implique en phase d'apprentissage de connaître pour chaque donnée, les k composantes les plus vraisemblables pour chaque classe, et ce à chaque itération. Nous avons exploité le fait que les apprentissages que nous étudions, que ce soit UBM ou GMM, dérivent d'un même modèle UBM pour définir des procédures simplifiées. Nous allons préciser dans les deux paragraphes suivants comment nous avons exploité cette propriété et quelle a été la règle de décision associée.

4.2.2 Apprentissage UBM (initialisation des modèles LM-dGMM avec le modèle du monde)

4.2.2.1 Deux mises en oeuvre de l'apprentissage

Comme dit précédemment, une stratégie alternative à la stricte mise en oeuvre de l'algorithme d'apprentissage où les ensembles $S^{\mathbf{c}}_{n,t}$ sont recalculés à chaque itération, consiste à déterminer ces ensembles en utilisant les modèles initiaux, et les maintenir fixes durant toute la phase d'apprentissage.

De plus, compte tenu de l'initialisation du modèle LM-dGMM de chaque classe c par le modèle UBM, les ensembles initiaux $S^{\mathbf{c}}_{n,t}$ sont initialement indépendants de la classe. La fonction de perte, pour cette alternative, s'écrit donc :

$$\begin{aligned}\mathbf{L} &= \sum_{n=1}^{N} l_n \sum_{c \neq y_n} max\Bigg(0\ ,\ 1 + \frac{1}{T_n}\sum_{t=1}^{T_n}\Bigg(d(x_{n,t}, \mu_{y_n m_{n,t}}) + \theta_{m_{n,t}} \\ &\quad + \log \sum_{m \in S_{n,t}} \exp\Big(-d(x_{n,t}, \mu_{cm}) - \theta_m\Big)\Bigg)\Bigg),\end{aligned} \tag{4.4}$$

où

$$l_n = \min\left(1, \frac{1}{h_n^{MAP}}\right),$$

$$\begin{aligned}h_n^{MAP} &= \sum_{c \neq y_n} \max\Bigg(0\ ,\ 1 + \frac{1}{T_n}\sum_{t=1}^{T_n}\Bigg(d(x_{n,t}, \mu^{MAP}_{y_n m_{n,t}}) + \theta_{m_{n,t}} \\ &\quad + \log \sum_{m \in S_{n,t}} \exp\Big(-d(x_{n,t}, \mu^{MAP}_{cm}) - \theta_m\Big)\Bigg)\Bigg).\end{aligned} \tag{4.5}$$

Les ensembles $S_{n,t}$ sont les ensembles $S^{\mathbf{c}}_{n,t}$ obtenus lors de l'initialisation. Cette fonction de perte est convexe et peut être minimisée par des algorithmes classiques d'optimisation non-linéaire tels que l'algorithme L-BFGS [Nocedal and Wright, 1999].

Pour synthétiser, l'algorithme d'apprentissage des modèles LM-dGMM sans remise à jour des ensembles $S^{\mathbf{c}}_{n,t}$, se simplifie de la manière suivante :

- Initialiser chaque classe par le modèle UBM,
- Utiliser l'UBM pour sélectionner l'ensemble des k-meilleures gaussiennes associé à chaque vecteur de données,
- Calculer les poids des segments,
- Résoudre le problème de minimisation défini par la fonction de perte de l'équation eq. (4.4).

4.2.2.2 Règle de décision et résultats expérimentaux

Pour un segment de test donné $\{x_t\}_{t=1}^{T}$, et pour chaque vecteur x_t, deux alternatives sont également possibles :

- Soit nous utilisons le modèle UBM $\{\mu_{Um}, \Sigma_m, \theta_m\}$ pour sélectionner l'ensemble E_t des k composantes gaussiennes les plus vraisemblables et nous limitons la règle de décision aux scores des composantes associées dans chaque classe :

$$y = \underset{c}{argmin} \left\{ \sum_{t=1}^{T} -\log \sum_{m \in E_t} \exp\Big(- d(x_t, \mu_{cm}) - \theta_m \Big) \right\}. \tag{4.6}$$

- Soit nous calculons le score pour chacune des classes à partir de l'ensemble E_t^c des k composantes les plus vraisemblables du modèle de la classe en question :

$$y = \underset{c}{argmin} \left\{ \sum_{t=1}^{T} -\log \sum_{m \in E_t^c} \exp\Big(- d(x_t, \mu_{cm}) - \theta_m \Big) \right\}. \tag{4.7}$$

Comparaison des deux alternatives pour l'apprentissage : Très rapidement, nous avons constaté une différence énorme en termes de performances selon la mise en oeuvre de l'apprentissage et du test. Nous appellerons "Fixes" les conditions qui impliquent les k composantes les plus vraisemblables du modèle UBM seulement, et "À jour" lorsque les algorithmes sont appliqués dans leur version stricte avec remise à jours des ensembles des k composantes.

Le tableau Tab. 4.4 donne les taux d'identification correcte de modèles LM-dGMM à 16 gaussiennes ($M = 16$), pour un apprentissage et test restreints aux *10*-meilleures gaussiennes ($k = 10$). Nous rapportons aussi à titre indicatif, les durées des phases d'apprentissage et de test.

Sélect. k-meilleures gaussiennes		Taux de bonne identification		Durée
Apprentissage	Test			
Fixes	Fixes	meilleure itération (it. 29) dernière itération (it. 30)	**69.8%** 69.4%	4h37
À jour	À jour	meilleure itération (it. 1) dernière itération (it. 30)	31.9% 12.1%	7h21

TAB. 4.4: Sélection des k-meilleures gaussiennes : k-meilleures gaussiennes fixes vs k-meilleures gaussiennes à jour.

Ces résultats montrent clairement que la mise en oeuvre stricte conduit à de mauvais résultats. Fixer l'ensemble des k composantes gaussiennes dès l'initialisation de l'algorithme d'apprentissage s'avère être très bénéfique. En plus de donner largement les meilleurs résultats, l'algorithme résultant est beaucoup plus rapide.

Étude des durées d'apprentissage et de test : Une série d'expérimentations a été faite pour évaluer l'influence du nombre de composantes gaussiennes sur les durées de l'apprentissage et de test.

Un test croisé a été réalisé pour constater l'importante influence du nombre de gaussiennes sur la durée sollicitée par l'apprentissage de modèles LM-dGMM. Le tableau Tab. 4.5 révèle les taux d'identification de modèles LM-dGMM à 16 gaussiennes avec les durées des phases d'apprentissage et de test, dans diverses configurations.

Conditions		Taux de bonne identification		Durée
Apprentissage	Test			
$k = 1$	$k = 1$	meilleure itération (it. 7) dernière itération (it. 7)	45.3% 45.3%	0h13
$k = 1$	$k = M$	meilleure itération (it. 1) dernière itération (it. 7)	31.9% 20.7%	1h20
$k = M$	$k = 1$	meilleure itération (it. 9) dernière itération (it. 30)	56.5% 51.7%	3h18
LM-dGMM de base ($k = M$)		meilleure itération (it. 29) dernière itération (it. 30)	69.4% 69.0%	**7h16**

TAB. 4.5: Influence du nombre de gaussiennes sélectionnées : Performances vs durées d'apprentissage et de test.

Notons que la complexité algorithmique des calculs de la fonction de perte et des gradients est linéaire en le nombre de composantes. Mais les vraies durées d'apprentissage et de test ne reflètent pas trop cet aspect linéaire, du fait que le processus d'optimisation (le nombre d'itérations jusqu'à convergence) varie d'un cas à un autre.

Conditions		Durées	
Apprentissage	Test	Apprentissage	Test
$k = 10$	$k = 10$	0h59	0h05
$k = 20$	$k = 20$	1h53	0h16
$k = 32$	$k = 32$	2h57	0h24
$k = 64$	$k = 64$	6h01	0h42
$k = 128$	$k = 128$	12h19	1h38

TAB. 4.6: Durées d'apprentissage et de test de systèmes LM-dGMM à 512 composantes aux *10*, *20*, *32*, *64* et *128* meilleures gaussiennes.

Partant d'un modèle du monde à 512 composantes, nous avons fait varier la valeur de k entre 10 et 128. Il est rappelé que l'apprentissage est fait à partir de 3 minutes de parole pour chacune des 50 classes (50 locuteurs) et le test est réalisé sur 232 segments de

parole. Dans les cinq configurations testées, l'apprentissage se termine après 7 itérations de L-BFGS, i.e. La fonction de perte s'annule après 7 itérations de L-BFGS. Le tableau Tab. 4.6 regroupe l'ensemble des durées.

Il apparaît clairement que réduire la valeur de k réduit considérablement le temps d'apprentissage, mais aussi la durée du test : il faut compter 1,39 seconde par segment de test en moyenne pour $k = 10$ et 25,47 secondes pour $k = 128$!

Étude de la valeur de k en fonction de M : Une série d'expériences a été conduite afin de déterminer quelles sont les valeurs intéressantes de k par rapport aux valeurs possibles de M. Les expériences sont faites avec 32, 64, 128 et 256 composantes. Les tableaux Tab. 4.7 et Tab. 4.8 présentent un comparatif des performances des systèmes GMM génératifs, des systèmes LM-dGMM de base et des systèmes LM-dGMM simplifiés avec k-meilleures gaussiennes. Il est rappelé que le choix des k-meilleures composantes est fait avec le modèle UBM.

Système	Taux de bonne identification	
GMM	68.1%	
LM-dGMM de base	meilleure itération (it. 15)	69.8%
	dernière itération (it. 30)	68.5%
LM-dGMM aux *10*-meilleures gaussiennes	meilleure itération (it. 29)	69.4%
	dernière itération (it. 30)	69.0%
LM-dGMM aux *20*-meilleures gaussiennes	meilleure itération (it. 21)	69.4%
	dernière itération (it. 30)	68.5%

(a) Modèles à 32 gaussiennes.

Système	Taux de bonne identification	
GMM	69.8%	
LM-dGMM de base	meilleure itération (it. 14)	73.7%
	dernière itération (it. 17)	73.3%
LM-dGMM aux *10*-meilleures gaussiennes	meilleure itération (it. 16)	71.6%
	dernière itération (it. 16)	71.6%
LM-dGMM aux *20*-meilleures gaussiennes	meilleure itération (it. 16)	72.4%
	dernière itération (it. 16)	72.4%

(b) Modèles à 64 gaussiennes.

TAB. 4.7: Taux de bonne identification de systèmes GMM et LM-dGMM à 32 et 64 gaussiennes.

Système	Taux de bonne identification	
GMM	74.1%	
LM-dGMM de base	meilleure itération (it. 11)	74.1%
	dernière itération (it. 11)	74.1%
LM-dGMM aux *10*-meilleures gaussiennes	meilleure itération (it. 10)	**74.1%**
	dernière itération (it. 10)	74.1%
LM-dGMM aux *20*-meilleures gaussiennes	meilleure itération (it. 11)	73.3%
	dernière itération (it. 11)	73.3%

(a) Modèles à 128 gaussiennes.

Système	Taux de bonne identification	
GMM	73.3%	
LM-dGMM de base	meilleure itération (it. 11)	72.0%
	dernière itération (it. 11)	72.0%
LM-dGMM aux *10*-meilleures gaussiennes	meilleure itération (it. 11)	71.1%
	dernière itération (it. 11)	71.1%
LM-dGMM aux *20*-meilleures gaussiennes	meilleure itération (it. 10)	71.6%
	dernière itération (it. 10)	71.6%

(b) Modèles à 256 gaussiennes.

TAB. 4.8: Taux de bonne identification de systèmes GMM et LM-dGMM à 128 et 256 gaussiennes.

En regardant les résultats, on constate que les systèmes LM-dGMM aux *10*-meilleures gaussiennes donnent des performances proches ou égales aux systèmes LM-dGMM de base où la totalité des composantes est utilisée. La restriction aux k-meilleures gaussiennes n'handicape pas en réalité les modèles à grande marge. Avec une valeur de $k = 10$, nous obtenons des performances comparables tout en réduisant considérablement les temps de calcul.

Une série d'expériences a été faite avec des modèles à 512 composantes. Le tableau Tab. 4.9 donne les taux d'identification correcte de systèmes LM-dGMM à différentes valeurs de k. Les temps de calcul explosent et de ce fait nous n'avons pas obtenu la performance pour le système de base à 512 composantes. Les performances constatées sont très stables dès les premières valeurs de k, et elles augmentent de manière non significative avec celles de k.

Conditions		Taux de bonne identification	
Apprentissage	Test		
$k = 10$	$k = 10$	meilleure itération (it. 6)	62.5%
		dernière itération (it. 6)	62.5%
$k = 20$	$k = 20$	meilleure itération (it. 7)	63.8%
		dernière itération (it. 7)	63.8%
$k = 32$	$k = 32$	meilleure itération (it. 6)	64.2%
		dernière itération (it. 7)	63.8%
$k = 64$	$k = 64$	meilleure itération (it. 6)	63.8%
		dernière itération (it. 7)	63.4%
$k = 128$	$k = 128$	meilleure itération (it. 7)	63.8%
		dernière itération (it. 7)	63.8%

TAB. 4.9: Performances de systèmes LM-dGMM à 512 composantes aux *10*, *20*, *32*, *64* et *128* meilleures gaussiennes.

Étude des performances en fonction de la valeur de M : Lorsque nous examinons les performances des systèmes LM-dGMM de base, nous remarquons que le bénéfice de la séparation à grande marge se perd avec des modèles à plus de 128 composantes. C'est consistent avec le fait connu que l'approche discriminante est plus efficace quand la "vraie" distribution n'est pas convenablement modélisée par les modèles génératifs. Dans le cas de modèles complexes, un autre enfreint se présente en plus de la complexité algorithmique. Il est certain que le peu de données d'apprentissage est un facteur très limitatif et ne permet pas un apprentissage correct que ce soit pour les modèles GMM ou les modèles LM-dGMM. Rappelons que dans notre cas expérimental, nous ne disposons que de 3 minutes de parole pour apprendre le modèle de chaque classe !

4.2.3 Apprentissage GMM (initialisation des modèles LM-dGMM avec les modèles GMM génératifs)

Rappelons que, dans le cadre de l'apprentissage GMM, les mélanges LM-dGMM sont appris après une initialisation faite à partir de modèles GMM différents pour chaque classe : ces modèles GMM sont issus d'une adaptation MAP du modèle UBM.

4.2.3.1 Des variantes pour la mise en oeuvre de l'apprentissage

Deux mises en oeuvre de l'apprentissage peuvent être envisagées selon la définition des ensembles $S_{n,t}^{\mathbf{c}}$. Dans la stricte mise en oeuvre de la procédure décrite ci dessus, ces ensembles sont redéfinis à chaque itération et dépendent effectivement de la classe c, et ce dès l'initialisation. Dans la deuxième version, pour réduire le temps de calcul et l'espace

mémoire requis, nous proposons d'exploiter comme précédemment, la correspondance qui existe entre les composantes des modèles GMM appris par adaptation MAP et celles de l'UBM [Reynolds et al., 2000] : un vecteur de données proche d'une certaine composante de l'UBM sera aussi proche de la composante correspondante en GMM. Nous utilisons donc l'UBM pour sélectionner un ensemble $S_{n,t}$ unique des k-meilleures gaussiennes pour chaque vecteur $x_{n,t}$ d'apprentissage. Nous avons donc une sélection $(C-1)$ fois plus rapide et moins demandeuse de mémoire (plus le nombre de classes est grand plus le gain est important).

Avec cette approche, la fonction de perte à minimiser est celle décrite précédemment (l'équation eq. 4.4), mais les modèles initiaux LM-dGMM ne sont pas identiques [Jourani et al., 2011a].

4.2.3.2 Règle de décision et résultats expérimentaux

Comme précédemment, deux règles de décision sont possibles :
- À chaque vecteur x_t sont associées les k-meilleures gaussiennes de chacun des C modèles LM-dGMM, correspondant à chacune des classes.
- À chaque vecteur x_t sont associées les k-meilleures gaussiennes du modèle UBM qui composent E_t.

Le score du segment est calculé avec les composantes ainsi sélectionnées pour chaque vecteur le composant.

Comparaison des alternatives proposées en apprentissage et en test : Les tableaux Tab. 4.10, Tab. 4.11, Tab. 4.12, Tab. 4.13, Tab. 4.14 et Tab. 4.15 exposent les taux d'identification correcte des systèmes LM-dGMM avec $k = 10$ dans les quatre alternatives envisagées pour la sélection des composantes les plus vraisemblables. Il est noté UBM lorsque la sélection des composantes est faite à partir du modèle UBM, et GMM lorsque la sélection respecte le cadre théorique.

Sélect. k-meilleures gaussiennes		Taux de bonne identification	
Apprentissage	Test		
UBM	UBM	meilleure itération (it. 17)	**67.7%**
		dernière itération (it. 30)	66.8%
UBM	GMM	meilleure itération (it. 28)	67.2%
		dernière itération (it. 30)	66.8%
GMM	UBM	meilleure itération (it. 5)	65.9%
		dernière itération (it. 30)	60.3%
GMM	GMM	meilleure itération (it. 5)	65.9%
		dernière itération (it. 30)	61.2%

Tab. 4.10: Sélection des k-meilleures gaussiennes en utilisant UBM et GMM ($M = 16$).

Sélect. k-meilleures gaussiennes		Taux de bonne identification	
Apprentissage	Test		
UBM	UBM	meilleure itération (it. 13)	69.8%
		dernière itération (it. 30)	66.8%
UBM	GMM	meilleure itération (it. 13)	**71.1%**
		dernière itération (it. 30)	67.7%
GMM	UBM	meilleure itération (it. 1)	70.7%
		dernière itération (it. 7)	66.8%
GMM	GMM	meilleure itération (it. 1)	71.1%
		dernière itération (it. 7)	66.4%

TAB. 4.11: Sélection des k-meilleures gaussiennes en utilisant UBM et GMM ($M = 32$).

Sélect. k-meilleures gaussiennes		Taux de bonne identification	
Apprentissage	Test		
UBM	UBM	meilleure itération (it. 8)	**72.8%**
		dernière itération (it. 15)	71.6%
UBM	GMM	meilleure itération (it. 10)	72.4%
		dernière itération (it. 15)	71.1%
GMM	UBM	meilleure itération (it. 3)	70.3%
		dernière itération (it. 3)	70.3%
GMM	GMM	meilleure itération (it. 3)	69.4%
		dernière itération (it. 3)	69.4%

TAB. 4.12: Sélection des k-meilleures gaussiennes en utilisant UBM et GMM ($M = 64$).

Sélect. k-meilleures gaussiennes		Taux de bonne identification	
Apprentissage	Test		
UBM	UBM	meilleure itération (it. 10)	**77.2%**
		dernière itération (it. 10)	77.2%
UBM	GMM	meilleure itération (it. 9)	76.7%
		dernière itération (it. 10)	76.3%
GMM	UBM	meilleure itération (it. 0)	74.1%
		dernière itération (it. 0)	74.1%
GMM	GMM	meilleure itération (it. 0)	74.1%
		dernière itération (it. 0)	74.1%

TAB. 4.13: Sélection des k-meilleures gaussiennes en utilisant UBM et GMM ($M = 128$).

Sélect. k-meilleures gaussiennes		Taux de bonne identification	
Apprentissage	Test		
UBM	UBM	meilleure itération (it. 9)	**77.6%**
		dernière itération (it. 9)	77.6%
UBM	GMM	meilleure itération (it. 9)	77.2%
		dernière itération (it. 9)	77.2%
GMM	UBM	meilleure itération (it. 0)	73.3%
		dernière itération (it. 0)	73.3%
GMM	GMM	meilleure itération (it. 0)	73.7%
		dernière itération (it. 0)	73.7%

TAB. 4.14: Sélection des k-meilleures gaussiennes en utilisant UBM et GMM ($M = 256$).

Sélect. k-meilleures gaussiennes		Taux de bonne identification	
Apprentissage	Test		
UBM	UBM	meilleure itération (it. 3)	**75.4%**
		dernière itération (it. 8)	74.1%
UBM	GMM	meilleure itération (it. 2)	75.4%
		dernière itération (it. 8)	75.0%
GMM	UBM	meilleure itération (it. 0)	74.1%
		dernière itération (it. 0)	74.1%
GMM	GMM	meilleure itération (it. 0)	73.7%
		dernière itération (it. 0)	73.7%

TAB. 4.15: Sélection des k-meilleures gaussiennes en utilisant UBM et GMM ($M = 512$).

L'utilisation de l'UBM pour la détermination des ensembles de lois gaussiennes $S_{n,t}$ intervenant dans la phase d'apprentissage, donne des performances en général meilleures que l'approche GMM. Elle engendre les meilleurs résultats, tout en étant l'approche la plus rapide. L'utilisation du modèle UBM pour déterminer l'ensemble E_t en phase de test donne des performances similaires à l'approche GMM. Il s'en suit une amélioration des performances en combinant l'utilisation des ensembles $S_{n,t}$ et E_t, ce que nous ferons dans les expériences suivantes.

Étude de la valeur de k en fonction de la valeur de M : Les expériences présentées consistent à faire varier k en fonction du nombre total M de composantes par modèle LM-dGMM. Apprentissage et test utilisent le modèle UBM pour la sélection des meilleures composantes.

Les Tableaux Tab. 4.16, Tab. 4.17 et Tab. 4.18 présentent les taux d'identification correcte de systèmes GMM et LM-dGMM aux *10* et *20*-meilleures gaussiennes, pour des modèles à $M = 16, 32, 64, 128, 256$ et 512 composantes.

Système	Taux de bonne identification	
GMM	61.2%	
LM-dGMM aux *10*-meilleures gaussiennes	meilleure itération (it. 17)	67.7%
	dernière itération (it. 30)	66.8%
LM-dGMM aux *20*-meilleures gaussiennes		–

(a) Modèles à 16 gaussiennes.

Système	Taux de bonne identification	
GMM	68.1%	
LM-dGMM aux *10*-meilleures gaussiennes	meilleure itération (it. 13)	69.8%
	dernière itération (it. 30)	66.8%
LM-dGMM aux *20*-meilleures gaussiennes	meilleure itération (it. 5)	70.3%
	dernière itération (it. 30)	69.0%

(b) Modèles à 32 gaussiennes.

TAB. 4.16: Taux de bonne identification de systèmes GMM et LM-dGMM à 16 et 32 composantes, aux *10* et *20* meilleures gaussiennes.

Système	Taux de bonne identification	
GMM	69.8%	
LM-dGMM aux *10*-meilleures gaussiennes	meilleure itération (it. 8)	72.8%
	dernière itération (it. 15)	71.6%
LM-dGMM aux *20*-meilleures gaussiennes	meilleure itération (it. 11)	75.0%
	dernière itération (it. 14)	74.6%

(a) Modèles à 64 gaussiennes.

Système	Taux de bonne identification	
GMM	74.1%	
LM-dGMM aux *10*-meilleures gaussiennes	meilleure itération (it. 10)	77.2%
	dernière itération (it. 10)	77.2%
LM-dGMM aux *20*-meilleures gaussiennes	meilleure itération (it. 3)	76.3%
	dernière itération (it. 11)	75.9%

(b) Modèles à 128 gaussiennes.

TAB. 4.17: Taux de bonne identification de systèmes GMM et LM-dGMM à 64 et 128 composantes, aux *10* et *20* meilleures gaussiennes.

Système	Taux de bonne identification	
GMM	73.3%	
LM-dGMM aux *10*-meilleures gaussiennes	meilleure itération (it. 9)	**77.6%**
	dernière itération (it. 9)	**77.6%**
LM-dGMM aux *20*-meilleures gaussiennes	meilleure itération (it. 10)	77.2%
	dernière itération (it. 10)	77.2%

(a) Modèles à 256 gaussiennes.

Système	Taux de bonne identification	
GMM	74.1%	
LM-dGMM aux *10*-meilleures gaussiennes	meilleure itération (it. 3)	75.4%
	dernière itération (it. 8)	74.1%
LM-dGMM aux *20*-meilleures gaussiennes	meilleure itération (it. 7)	75.0%
	dernière itération (it. 7)	75.0%

(b) Modèles à 512 gaussiennes.

TAB. 4.18: Taux de bonne identification de systèmes GMM et LM-dGMM à 256 et 512 composantes, aux *10* et *20* meilleures gaussiennes.

Le meilleur système discriminant a un taux de 77.6% de bonne identification, il correspond à $k = 10$ pour $M = 256$. Le meilleur système GMM a un taux de 74.1%, pour $M = 128$ et 512. L'amélioration obtenue par le modèle LM-dGMM est de l'ordre de 5%, ce qui est significatif.

En comparant les résultats de l'apprentissage UBM avec ceux de l'apprentissage GMM, on constate que l'initialisation des modèles LM-dGMM "aux k-meilleures gaussiennes" avec les modèles GMM génératifs donne les meilleurs résultats. Nous adopterons donc dans les prochaines sections ce type d'initialisation.

Étude de la marge : En regardant les valeurs de la fonction de perte du système LM-dGMM à 512 composantes aux *10*-meilleures gaussiennes (la figure Fig. 4.7), i.e., $M = 512$ et $k = 10$, on s'aperçoit que les modèles initiaux satisfont un nombre important des contraintes de grande marge imposées ; cette observation explique comme précédemment l'amélioration modérée des performances par rapport aux système GMM de base. Il est donc naturellement de durcir les contraintes en jouant sur la valeur de la marge minimale, et en espérant ainsi augmenter le pouvoir discriminant des modèles LM-dGMM. Rappelons que la marge minimale a été choisie tout au long de ce travail unitaire, comme l'a proposé Fei SHA. Les tests que nous avons menés montrent qu'on peut améliorer légèrement les performances des modèles en choisissant des marges non-unitaires.

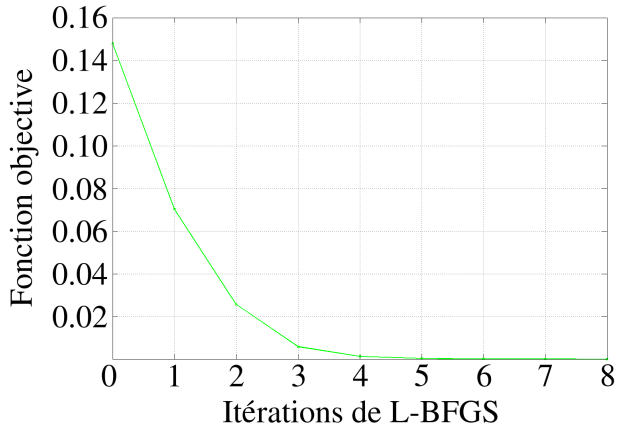


FIG. 4.7: *Fonction de perte du système LM-dGMM à 512 composantes aux 10-meilleures gaussiennes (marge unitaire).*

Le tableau Tab. 4.19 donne les taux d'identification correcte de systèmes à 512 gaussiennes, à différentes marge de séparation.

Marge		1	1.5	2	2.5	3
LM-dGMM ($k = 10$)	meilleure itér.	75.4%	76.3%	75.4%	74.1%	74.1%
	dernière itér.	74.1%	76.3%	75.4%	71.1%	70.7%
LM-dGMM ($k = 20$)	meilleure itér.	75.0%	77.2%	74.6%	74.6%	74.1%
	dernière itér.	75.0%	77.2%	74.6%	69.8%	71.1%

TAB. 4.19: Taux de bonne identification de systèmes LM-dGMM à 512 composantes, à différentes marges de séparation.

Les améliorations n'étant significatives, nous avons gardé une marge minimale unitaire.

4.3 Étude comparative dans le cas de grands volumes de données

L'apprentissage restreint aux k-meilleures gaussiennes réduit considérablement la complexité algorithmique des modèles LM-dGMM, tout en améliorant les performances par rapport aux modèles GMM classiques. Nous pouvons ainsi utiliser ce cadre afin de comparer notre système et certaines de ses variantes à d'autres approches classiques dans un scénario plus exigeant.

4.3.1 Description des systèmes d'identification du locuteur

Tous les systèmes de reconnaissance du locuteur utilisent le même module de traitement acoustique que celui décrit dans la section 3.5.1.1 du précédent chapitre. Les

différents systèmes de reconnaissance étudiés se différencient au travers de leur module de décision et de l'utilisation ou non d'une technique de compensation de variabilité. Pour le module de décision, nous avons comparé trois approches, à savoir :

- Le système probabiliste traditionnel de l'état de l'art GMM-UBM, i.e., le système-GMM.
- Le système discriminant avec le mélange LM-dGMM, i.e., le système-LM-dGMM.
- Le système discriminant, appelé par la suite système-GSL, basé sur des SVM à noyau linéaire avec pour paramétrisations les supervecteurs issus des GMM.

La technique de compensation de la variabilité inter-sessions SFA (eigenchannel) [Matrouf et al., 2007] (voir la section 2.2.4.3 du chapitre 2 et la section 2 de la partie Annexes) est expérimentée avec les deux systèmes à base de GMM :

- Le système compensé GMM, appelé par la suite système-GMM-SFA, intègre la technique de compensation de la variabilité inter-sessions SFA.
- Le système compensé à mélange LM-dGMM, appelé par la suite système-LM-dGMM-SFA, intègre la technique SFA, dans le sens où les modèles GMM-SFA sont utilisés comme initialisation dans l'algorithme d'apprentissage des LM-dGMM. Les données compensées dans le domaine des caractéristiques sont ensuite utilisées pour discriminer les différents locuteurs ; la compensation est faite directement dans le domaine des caractéristiques, i.e., post-traitement des données. La phase de test utilise également des données compensées.

Le système compensé GSL sera évalué également. Appelé par la suite système-GSL-NAP, ce système intègre la méthode de compensation de la variabilité des supervecteurs NAP.

4.3.2 Protocole expérimental

Corpus

Nos expérimentations sont toujours effectuées sur la tâche d'identification du locuteur de la **campagne d'évaluation NIST-SRE 2006** [NIST, 2006], mais **les tests sont réalisés cette fois-ci sur tous les locuteurs masculins de la condition principale (1conv4w-1conv4w), à savoir 349 locuteurs**. L'apprentissage et le test se font sur 3 minutes de parole effective en moyenne.

Mise en oeuvre

Tous les systèmes sont comparés en utilisant deux configurations à $\mathbf{M = 256}$ et $\mathbf{M = 512}$ gaussiennes. Pour nos systèmes, l'apprentissage de nos modèles LM-dGMM est restreint aux *10*-meilleures gaussiennes ($k = 10$). Il est réalisé à partir de l'initialisation GMM. l'UBM servant à sélectionner les *10*-meilleures gaussiennes du système-LM-dGMM-SFA est celui appris sur les données non compensées (la correspondance entre composantes gaussiennes persiste).

Les modèles GMM-SFA, GSL et GSL-NAP sont appris en utilisant l'outil ALIZE/Spkdet [Bonastre et al., 2008]. 200 imposteurs de NIST-SRE 2004 représentent les entrées négatives des SVM. La matrice **U** de SFA et la matrice **S** de NAP sont de rang 40, et sont estimées sur 2934 sessions de 124 locuteurs masculins de NIST-SRE 2004. Les techniques de normalisation des scores ne sont pas utilisées dans ces tests.

Évaluation des systèmes

Les taux d'identification correcte sont calculés sur une base comprenant 1546 fichiers de test. Les tableaux Tab. 4.20 et Tab. 4.21 regroupent les performances en identification du système-GMM, système-LM-dGMM, système-GSL, système-GSL-NAP, système-GMM-SFA et système-LM-dGMM-SFA. L'intervalle de confiance à 95% des différents systèmes est en moyenne de ± 1.5%.

Système	Taux de bonne identification	
GMM	75.87%	
LM-dGMM	meilleure itération (it. 3)	77.62%
	dernière itération (it. 30)	75.23%
GSL	81.18%	
GSL-NAP	87.19%	
GMM-SFA	89.26%	
LM-dGMM-SFA	meilleure itération (it. 2)	89.65%
	dernière itération (it. 22)	85.83%

TAB. 4.20: Taux de bonne identification de modèles GMM, LM-dGMM et GSL à 256 gaussiennes, avec et sans compensation de la variabilité inter-sessions.

Système	Taux de bonne identification	
GMM	77.88 %	
LM-dGMM	meilleure itération (it. 9)	78.40 %
	dernière itération (it. 9)	78.40 %
GSL	82.21 %	
GSL-NAP	87.77 %	
GMM-SFA	90.75 %	
LM-dGMM-SFA	meilleure itération (it. 4)	**91.27 %**
	dernière itération (it. 14)	90.30%

TAB. 4.21: Taux de bonne identification de modèles GMM, LM-dGMM et GSL à 512 gaussiennes, avec et sans compensation de la variabilité inter-sessions.

Une première remarque s'impose si on compare les tableaux Tab. 4.20 et Tab. 4.21 et le tableau Tab. 4.18 de la section précédente : les taux de bonne identification augmentent que ce soit pour le système GMM ou le système LM-dGMM et ce de manière plus marquée pour 512 composantes, alors que le nombre de locuteurs est passé de 50 à 349.

Comme prévu, les techniques de compensation de la variabilité inter-sessions permettent d'améliorer considérablement les performances des systèmes. Sans compensation, les résultats obtenus montrent que le système-GSL donne de bien meilleurs résultats en identification que les deux systèmes à base de GMM. Par contre quand on utilise la technique de compensation SFA, les systèmes à base de GMM deviennent meilleurs que le système-GSL-NAP discriminant.

Notre système-LM-dGMM-SFA a un taux de 91.27% de bonne identification. Ses performances sont comparables au système-GMM-SFA et sont meilleures de manière significative que celles du système-GSL-NAP [Daoudi et al., 2011].

4.4 Nouvelle règle de décision pour la reconnaissance du locuteur

S'inspirant du protocole expérimental de la vérification, nous proposons de faire une authentification C fois plus rapide, en prenant la décision en effectuant une seule comparaison au lieu de C comparaisons ; C étant le nombre de locuteurs connus. Le module d'apprentissage reste le même en considérant toutes les classes de la population ; seul le processus de décision change par rapport au système d'identification, en demandant au locuteur inconnu de réclamer son identité.

4.4.1 Règle de décision pour une identification rapide de locuteur.

Comme dit précédemment, l'algorithme d'apprentissage permet d'avoir un mélange LM-dGMM pour l'ensemble des locuteurs connus du système. Nous nous sommes inspirés de la technique GMM-UBM pour reformuler la décision : pour une prononciation donnée, associée à un segment $\{x_t\}_{t=1}^{T}$, nous comparons sa proximité au modèle LM-dGMM du locuteur proclamé i et sa proximité au modèle LM-dGMM du monde déduit du modèle GMM UBM $\{\mu_{Um}, \Sigma_m, \theta_m\}$. Le score s'écrit :

$$\begin{aligned} LLR_{avg} \;=\; & \frac{1}{T}\sum_{t=1}^{T}\Bigg(\log\sum_{m}\exp\Big(-d(x_t,\mu_{im})-\theta_m\Big) \\ & -\;\log\sum_{m}\exp\Big(-d(x_t,\mu_{Um})-\theta_m\Big)\Bigg). \end{aligned} \tag{4.8}$$

La tâche de reconnaissance utilise conjointement à la fois le modèle LM-dGMM de l'identité clamée i et le modèle du monde. Plus le score de vérification est élevé, plus les

chances que la séquence de parole x soit dite par le locuteur (modèle) i sont grandes. Par analogie avec les modèles GMM, ce score d'appariement peut être considéré comme un rapport de log-vraisemblances.

Afin de pouvoir exploiter nos modèles dans des scénarios à grandes quantités de données et à un nombre de composantes important, nous réutilisons l'approche des k-meilleures gaussiennes. Pour un segment donné $\{x_t\}_{t=1}^{T}$, pour chaque vecteur x_t, nous utilisons le modèle UBM $\{\mu_{Um}, \Sigma_m, \theta_m\}$ pour sélectionner l'ensemble E_t des k-meilleures gaussiennes et le score devient [Jourani et al., 2011b] :

$$\begin{aligned} LLR_{avg} &= \frac{1}{T}\sum_{t=1}^{T}\Bigg(\log\sum_{m\in E_t}\exp\Big(-d(x_t,\mu_{im})-\theta_m\Big) \\ &\quad -\log\sum_{m\in E_t}\exp\Big(-d(x_t,\mu_{Um})-\theta_m\Big)\Bigg). \end{aligned} \tag{4.9}$$

4.4.2 Protocole expérimental et performances

Les caractéristiques des modèles GMM et LM-dGMM sont celles du paragraphe précédent : **256 ou 512 composantes** sont utilisées, la valeur de k est fixée à 10. Avant que les techniques de compensation n'atteignent leur suprématie, les techniques de normalisation étaient fortement utilisées en vérification du locuteur, c'est pourquoi nous avons étudié les deux familles de techniques en les appliquant à nos modèles et nous donnons ci-près les performances pour chacune d'elles.

Les performances sont mesurées en terme de taux d'erreurs égales (Equal Error Rate) EER et de minima de la fonction de coût de détection (minimum of detection cost function) minDCF de NIST [Przybocki and Martin, 2004]. Elles sont calculées sur une liste de **22123 unités d'évaluation (trials)**. Ce jeu de test de NIST implique les 349 locuteurs masculins et 1601 fichiers de test de la condition principale de NIST-SRE 2006.

4.4.2.1 Utilisation de la normalisation des scores

Nous avons choisi d'expérimenter l'influence de la T-normalisation des scores [Auckenthaler et al., 2000]. La normalisation utilise une cohorte d'imposteurs, incluant 200 locuteurs masculins de NIST-SRE 2004 [NIST, 2004]. C'est la même liste d'entrées négatives utilisée pour l'apprentissage des SVM.

Le tableau Tab. 4.22 rassemble l'ensemble des résultats obtenus en terme de EER(%), des modèles GMM et LM-dGMM à $M = 256$ et $M = 512$ gaussiennes, avec et sans la normalisation des scores.

Système			EER
Sans T-norm	GMM		9.43%
	LM-dGMM	meilleure itération (it. 3)	8.97%
		dernière itération (it. 30)	13.36%
Avec T-norm	GMM		8.91%
	LM-dGMM	meilleure itération (it. 1)	**8.40%**
		dernière itération (it. 30)	12.80%

(a) Modèles à 256 gaussiennes.

Système			EER
Sans T-norm	GMM		9.74%
	LM-dGMM	meilleure itération (it. 1)	9.66%
		dernière itération (it. 9)	10.47%
Avec T-norm	GMM		8.90%
	LM-dGMM	meilleure itération (it. 1)	8.85%
		dernière itération (it. 9)	9.66%

(b) Modèles à 512 gaussiennes.

TAB. 4.22: EER de systèmes GMM et LM-dGMM, avec et sans une T-normalisation des scores.

Les résultats confirment que la technique de normalisation T-norm permet d'améliorer les résultats des différents systèmes, à la fois à base de GMM et de LM-dGMM. Le meilleur système-GMM obtient un EER égal à 8.90% et enregistre un minDCF égal à $3.55 * 10^{-2}$. C'est le système à $M = 512$ gaussiennes. Alors que notre meilleur système-LM-dGMM obtient un EER égal à 8.40% et affiche un minDCF égal à $3.49 * 10^{-2}$. Ces résultats sont enregistrés pour un modèle à $M = 256$ composantes. Il y a donc réduction de l'EER et du minDCF de respectivement 5.6% et 1.7%.

4.4.2.2 Utilisation de la compensation ; étude comparative

Comme observé dans [Fauve et al., 2007], le puissant formalisme SFA permet d'éliminer efficacement la variabilité inter-sessions, sans faire appel à la T-normalisation des scores. Nous n'allons donc plus utiliser cette technique de normalisation dans les prochaines expériences.

Les tableaux Tab. 4.23 et Tab. 4.24 rappellent les performances du système-GMM et du système-LM-dGMM sans compensation, et affichent ceux des système-GMM-SFA et système-LM-dGMM-SFA où la compensation est utilisée, avec respectivement $M = 256$ et $M = 512$ gaussiennes. La figure Fig. 4.8 représente les EER et minDCF des systèmes GMM et GMM-SFA et des meilleurs systèmes LM-dGMM et LM-dGMM-SFA, à 512 composantes gaussiennes [Jourani et al., irst].

Système		EER
GMM		9.43%
LM-dGMM	meilleure itération (it. 3)	8.97%
	dernière itération (it. 30)	13.36%
GSL		7.53%
GSL-NAP		6.45%
GMM-SFA		6.15%
LM-dGMM-SFA	meilleure itération (it. 5)	5.58%
	dernière itération (it. 22)	10.98%

TAB. 4.23: EER de modèles GMM, LM-dGMM et GSL à 256 gaussiennes, avec et sans compensation de la variabilité inter-sessions.

Système		EER
GMM		9.74%
LM-dGMM	meilleure itération (it. 1)	9.66%
	dernière itération (it. 9)	10.47%
GSL		7.23%
GSL-NAP		5.90%
GMM-SFA		5.53%
LM-dGMM-SFA	meilleure itération (it. 4)	**5.02%**
	dernière itération (it. 14)	6.38%

TAB. 4.24: EER de modèles GMM, LM-dGMM et GSL à 512 gaussiennes, avec et sans compensation de la variabilité inter-sessions.

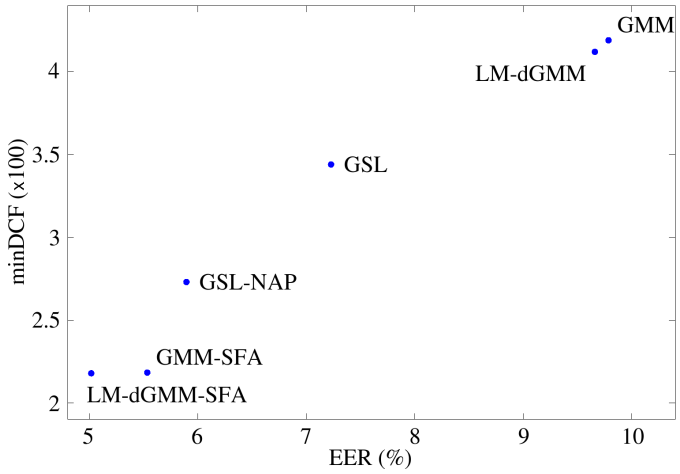


FIG. 4.8: *EER et minDCF des systèmes GMM, LM-dGMM, GSL, GSL-NAP, GMM-SFA et LM-dGMM-SFA à 512 composantes gaussiennes.*

Comme en identification, les techniques de compensation de la variabilité inter-sessions permettent d'améliorer considérablement les performances des systèmes. Notons que la technique SFA permet de réduire les erreurs d'un facteur proche de 2, ce qui est consistant avec la littérature portant sur les GMM-UBM. Les performances du système LM-dGMM-SFA étant très encourageantes, nous allons exploré plus en profondeur l'influence de la marge.

La table Tab. 4.25 donne les EER de modèles LM-dGMM-SFA à différentes marge minimale de séparation. Il est clair qu'une meilleure sélection de la marge peut améliorer significativement la performance de nos modèles à grande marge (par exemple, un EER de 4.85% au lieu du 5.02% obtenu avec une marge unitaire).

Marge	1	1.125	1.25	1.5	2	3	5	9
meilleure itération	5.02%	**4.85%**	5.02%	5.59%	5.58%	5.52%	5.55%	5.55%
dernière itération	6.38%	8.78%	7.65%	8.84%	28.29%	5.52%	5.55%	5.55%

TAB. 4.25: EER de systèmes LM-dGMM-SFA à 512 composantes aux *10*-meilleures gaussiennes, à différentes marges de séparation.

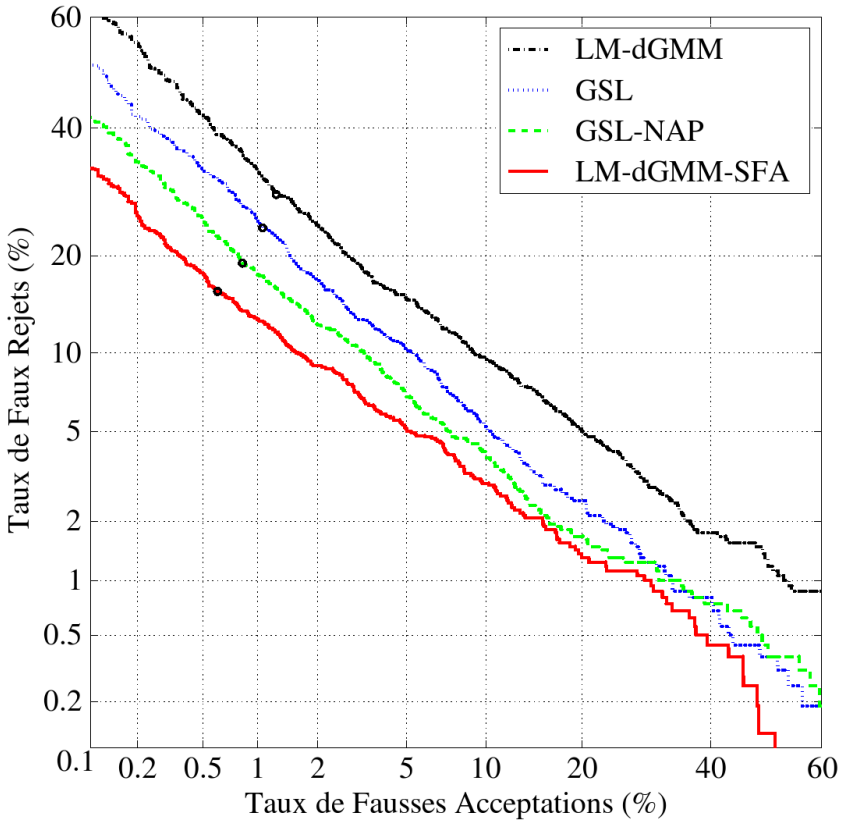


FIG. 4.9: *Courbes DET des systèmes LM-dGMM, GSL, GSL-NAP et LM-dGMM-SFA à 512 composantes gaussiennes.*

4.4.3 Comparaison et complémentarité entre les modélisations à grande marge et par SVM

Nous comparons les précédents systèmes système-GMM, système-LM-dGMM, système-GMM-SFA, système-LM-dGMM-SFA aux systèmes système-GSL et système-GSL-NAP dans le cadre de la précédente tâche de vérification du locuteur, qui utilise plus de séquences de test. Ces comparaisons sont rassemblées dans les tableaux Tab. 4.23 et Tab. 4.24 pour respectivement $M = 256$ et $M = 512$ gaussiennes. Les EER et minDCF des différents systèmes à 512 composantes gaussiennes sont donnés dans la figure Fig. 4.8.

Les résultats obtenus confirment que le système-GSL donne de bien meilleurs résultats que les deux systèmes à base de GMM, et que lorsqu'on utilise la technique de compensation SFA, les systèmes à base de GMM deviennent meilleurs que le système-GSL-NAP discriminant. Notre meilleur système-LM-dGMM-SFA à $M = 512$ composantes a un EER et minDCF égaux à 5.02% et $2.18 * 10^{-2}$. Le meilleur système GMM-SFA et GSL-NAP donnent un EER égal à 5.53% et 5.90%, pour $M = 512$. L'amélioration obtenue par le modèle LM-dGMM-SFA est respectivement de l'ordre de 9% et 15%.

La figure Fig. 4.9 affiche les courbes *Detection Error Tradeoff* (DET) des modèles discriminants GSL et GSL-NAP et celles des meilleurs modèles LM-dGMM et LM-dGMM-SFA, à 512 composantes gaussiennes. On peut s'apercevoir que les points de fonctionnement du meilleur système-LM-dGMM-SFA sont tous plus bas que ceux du système-GSL-NAP.

Nous avons réalisé un système de reconnaissance automatique du locuteur qui combine la modélisation LM-dGMM avec celle par SVM. Nous avons construit des supervecteurs par concaténation des vecteurs moyennes de nos meilleurs modèles LM-dGMM-SFA. Ces supervecteurs sont classés en utilisant des SVM à noyau linéaire. Nous notons ce système (LM-dGMM-SFA) – SVM.

Le système (LM-dGMM-SFA) – SVM à 512 composantes gaussiennes donne un EER égal à **4.39%**, améliorant ainsi les performances de chaque modélisation utilisée toute seule. Ce résultat révèle que ces deux approches de modélisation sont complémentaires. En plus d'améliorer les performances, cette combinaison permet aussi d'accélérer l'évaluation des modèles des locuteurs durant la phase de test.

4.5 Traitement des données aberrantes au niveau de la trame

Dans les systèmes LM-dGMM et LM-dGMM-SFA précédemment évalués, le traitement des données aberrantes est inclus. Rappelons que la stratégie consiste à calculer une pondération au niveau segmental et les différents poids servent à équilibrer l'influence des segments de données durant l'apprentissage. Un segment de données correctes a un poids égal à 1, tandis qu'un segment contenant des données aberrantes aura un poids inférieur à 1. Les formules de calcul de la fonction de perte et des poids sont données par les équations

eq. 4.4 et 4.5 de la section 4.2.2.1. Une simple expérience a été menée pour juger de l'importance de cette pondération avec le système LM-dGMM avec 512 composantes. Quand on n'utilise pas ces pondérations, i.e., on attribue un poids $l_n = 1$ à tous les segments, les performances se dégradent : sans aucun traitement, on enregistre un EER égal à 9.72%. Il s'agit d'une légère dégradation si on se rappelle que ce système affichait un ERR égal à 9.66%. Ce système à grande marge reste comparable au système-GMM de base, qui enregistre quand à lui un EER égal à 9.74%. Ceci confirme bel et bien la présence de données aberrantes dans le corpus d'apprentissage.

Nous proposons de revoir le traitement des données aberrantes en adoptant une stratégie différente pour les détecter et pondérer leur impact, pour plus réduire leur effet nuisible lors de la phase d'apprentissage.

4.5.1 Fonction de perte segmentale à points aberrants

Notre nouvelle stratégie de traitement des données aberrantes consiste à associer des poids d'influence au niveau de la trame et non du segment, tout en gardant nos contraintes segmentales :

$$\forall c \neq y_n,\ \frac{1}{T_n}\sum_{t=1}^{T_n} -\log \sum_{m\in S_{n,t}} \exp\Big(-d(x_{n,t},\mu_{cm})-\theta_m\Big) \ \geq\ 1+\frac{1}{T_n}\sum_{t=1}^{T_n} d(x_{n,t},\mu_{y_n m_{n,t}})+\theta_{m_{n,t}}. \tag{4.10}$$

Nous associons à chaque vecteur $x_{n,t}$ du segment n appartenant à la classe y_n, un ensemble de $(C-1)$ poids $l^{\mathbf{c}}_{n,t}$ calculés par rapport à chaque autre classe c, $(c \neq y_n)$. Pour ce faire, nous calculons pour tout vecteur $x_{n,t}$ les pertes induites $\big(h^{MAP}_{n,t}\big)^{\mathbf{c}} \geq 0$ par rapport aux différentes autres classes c :

$$\big(h^{MAP}_{n,t}\big)^{\mathbf{c}} \ =\ \frac{1+d(x_{n,t},\mu_{y_n m_{n,t}})+\theta_{m_{n,t}}+\log\sum_{m\in S_{n,t}}\exp\Big(-d(x_{n,t},\mu_{cm})-\theta_m\Big)}{T_n}. \tag{4.11}$$

Nous considérons ensuite la donnée $x_{n,t}$ comme étant aberrante si la perte est d'une valeur supérieure à 1 : $\big(h^{MAP}_{n,t}\big)^{\mathbf{c}} > 1$. Nous lui associons, dans ce cas là, le poids $l^{\mathbf{c}}_{n,t} = \frac{1}{\big(h^{MAP}_{n,t}\big)^{\mathbf{c}}}$. Les données correctes gardent un poids unitaire $l^{\mathbf{c}}_{n,t} = 1$. Avec cette approche de pondération par trame, la fonction de perte devient [Jourani et al., irst] :

$$\begin{aligned}\mathbf{L} \ =\ \sum_{n=1}^{N}\sum_{c\neq y_n}\max\Bigg(0\ ,\ \sum_{t=1}^{T_n}\frac{l^{\mathbf{c}}_{n,t}}{T_n}\Bigg(1+d(x_{n,t},\mu_{y_n m_{n,t}})+\theta_{m_{n,t}} \\ +\log\sum_{m\in S_{n,t}}\exp\Big(-d(x_{n,t},\mu_{cm})-\theta_m\Big)\Bigg)\Bigg).\end{aligned} \tag{4.12}$$

4.5.2 Évaluation

Pour évaluer cette proposition, nous avons repris les meilleurs systèmes développés dans les deux sections précédentes, à savoir les systèmes LM-dGMM et LM-dGMM-SFA avec 512 composantes. Le tableau Tab. 4.26 donne les performances en reconnaissance de ces systèmes à grande marge (compensés ou non), en utilisant l'une des deux approches de pondération.

Système			T. bonne identification	EER
Poids segmentaux	LM-dGMM	meill. itér.	78.40%	9.66%
		dern. itér.	78.40%	10.47%
	LM-dGMM-SFA	meill. itér.	91.27%	5.02%
		dern. itér.	90.30%	6.38%
Poids par trame	LM-dGMM	meill. itér.	78.40%	**9.47%**
		dern. itér.	78.40%	10.23%
	LM-dGMM-SFA	meill. itér.	91.27%	**4.89%**
		dern. itér.	90.30%	8.78%

TAB. 4.26: Traitement des données aberrantes : Poids segmentaux vs Poids par trame.

Dans la tâche de vérification, les résultats expérimentaux montrent que l'approche consistant à faire des pondérations par trame donne des résultats sensiblement meilleurs que l'approche à poids segmentaux. En effet, elle réduit le EER des systèmes LM-dGMM et LM-dGMM-SFA respectivement d'à peu près 2% et $2,6\%$. Par contre, aucune amélioration n'est enregistrée en identification.

Comme dans les précédentes expériences avec les modèles à grande marge (de base ou aux k-meilleures gaussiennes) initialisés par des GMM, un critère d'arrêt doit être élaboré. Les performances passent par un pic avant de se dégrader. Sans un critère d'arrêt établi, il faut prendre les modèles obtenus au début du processus d'optimisation[15] pour améliorer les performances des modèles initiaux.

Notre approche de traitement des données aberrantes peut être adaptée aux modèles GMM classiques en s'appuyant sur les vraisemblances des données. On peut par exemple considérer une donnée comme étant aberrante si sa vraisemblance par rapport à (la composante la plus vraisemblable de) sa classe est plus petite que celles par rapport aux autres classes. Les probabilités a posteriori seront par la suite pondérées par des poids dépendants du rapport des vraisemblances, pour équilibrer l'influence de l'ensemble des données.

[15] Expérimentalement, il semble en moyenne que les modèles obtenus à la quatrième itération de L-BFGS sont les meilleurs.

4.6 Conclusion

Nous avons présenté dans ce chapitre, des variantes de notre système initial de reconnaissance de locuteurs qui nous ont progressivement permis d'améliorer les performances.

- L'initialisation de l'apprentissage des modèles en utilisant pour chaque classe un LM-dGMM identique au modèle du monde donne de très bons résultats. Avec ce système, les meilleurs modèles à grande marge sont obtenus en zone de convergence, ce qui permet sa facile utilisation sans nécessiter des données de développement.
- L'utilisation des seules k-meilleures gaussiennes des modèles LM-dGMM permet d'étendre et d'utiliser la modélisation à grande marge dans des scénarios impliquant de grands volumes de données. L'apprentissage restreint aux k-meilleures gaussiennes de chaque classe allège l'apprentissage, tout en améliorant les performances des modèles GMM classiques.
- S'appuyant sur la technique de compensation de la variabilité inter-sessions SFA, le système LM-dGMM compensé enregistre des résultats en reconnaissance du locuteur meilleurs que ceux obtenus avec la technique discriminante GSL-NAP.
- La stratégie de pondération des données en fonction des pertes dues aux violations de marge, au niveau des trames, permet de traiter plus efficacement l'occurrence de données aberrantes dans le corpus d'apprentissage.

Finalement, utiliser les paramètres des modèles LM-dGMM-SFA en entrée d'un système de type SVM donne un système encore plus performant en reconnaissance du locuteur.

Chapitre 5

Conclusion

Divers modèles génératifs et discriminants ont été étudié en reconnaissance automatique du locuteur. La majorité des systèmes actuels sont basés sur l'utilisation de modèles de mélange de Gaussiennes (GMM) appris par adaptation MAP d'un modèle du monde UBM et/ou de machines à vecteurs de support (SVM). Le système hybride GSL associant les supervecteurs GMM formés en concaténant les vecteurs moyennes d'un modèle GMM à une SVM à noyau linéaire est parmi un des systèmes les plus performants. Cependant, dans les différentes applications de reconnaissance automatique du locuteur, le changement des conditions d'enregistrement entre les phases d'apprentissage et de reconnaissance reste une cause de dégradation considérable des performances ; la variabilité de session est le problème majeur à résoudre.

Les techniques d'analyse de facteurs comme SFA, ont été introduites pour compenser cette variabilité dans les systèmes basés sur les GMM, tandis que la méthode de compensation NAP a été proposée pour les systèmes à base de SVM.

Au cours de ce travail de thèse, nous avons proposé d'utiliser une nouvelle approche discriminante pour la reconnaissance automatique du locuteur qui consiste à utiliser des modèles GMM à grande marge, appelés LM-dGMM. Nos modèles reposent sur une récente approche discriminante pour la séparation multi-classes, qui a été appliquée en reconnaissance de la parole ; les modèles LM-GMM. Les LM-dGMM sont définis par un vecteur centroïde, une matrice de covariance diagonale et un offset. Initialisés par des modèles GMM-UBM, on apprend nos modèles d'une manière discriminante en introduisant des contraintes "à grande marge" sur les distances entre vecteurs centroïdes et observations. En comparaison avec les modèles LM-GMM originaux, l'apprentissage des LM-dGMM est beaucoup moins complexe : l'algorithme d'apprentissage simplifié est plus rapide et moins demandeur de mémoire. De plus, ces modèles donnent de bien meilleurs résultats que les modèles LM-GMM originaux.

Dans une application d'identification du locuteur sous les conditions d'évaluation de NIST-SRE 2006, les systèmes LM-GMM et LM-dGMM à 16 composantes atteignent respectivement 60.3% et 67.7% de taux de bonne identification, alors que le système GMM

classique enregistre un taux de 61.2%. Dans le cas de modèles à 32 composantes, l'apprentissage à grande marge permet d'améliorer encore les performances. L'analyse de la variation des taux de bonne identification au fil des itérations d'optimisation montre qu'on améliore les résultats obtenus avec les GMM classiques dès les premières itérations, avant d'enregistrer après une baisse des performances.

Quand on passe à des modèles à plus de composantes, l'apprentissage des GMM classiques fait que les modèles initiaux satisfont un nombre important de contraintes et l'apprentissage à grande marge ne permet pas d'améliorer les performances. Dans ce cas de figure, nous avons proposé d'initialiser nos modèles directement par l'UBM. Ceci assure une meilleure stabilité du taux de reconnaissance à une valeur de plus, plus élevée. Pour des modèles à 64 gaussiennes, le système GMM a un taux de bonne identification de 69.8%, tandis que notre système LM-dGMM enregistre un taux de 73.7%. Nous notons qu'avec cette initialisation, les performances de classification obtenues dans les dernières itérations du processus d'optimisation sont les meilleures. Cet apprentissage progressif réalisé à partir de l'UBM permet au processus d'optimisation d'atteindre en fin d'itérations un modèle correspondant aux meilleures performances, ce qui permet de profiter pleinement du pouvoir discriminant des modèles à grande marge sans aucun recours à des données de développement. Ce procédé rend inutile l'apprentissage des modèles GMM par adaptation MAP et par conséquent le temps nécessaire à l'apprentissage des modèles de locuteurs est diminué.

Malgré le fait que l'apprentissage de nos modèles LM-dGMM soit plus rapide que celui des modèles LM-GMM originaux, sa complexité algorithmique reste encore trop élevée pour traiter efficacement de grands volumes de données. Pour pouvoir utiliser nos modèles dans ce genre de scénario, tel que dans les campagnes d'évaluations NIST-SRE, nous avons proposé de réduire considérablement le nombre de contraintes de grande marge à satisfaire, en réduisant le nombre de composantes sur lesquelles portent ces contraintes. Cette réduction diminue la complexité algorithmique et permet aussi de disposer de "plus" de données en la discrimination de ce sous-ensemble de composantes. Pour ce faire, nous nous sommes basé sur deux principes : la décision de classification utilise généralement uniquement les k-meilleures gaussiennes, et la correspondance entre les composantes des GMM appris par adaptation MAP et celles de l'UBM.

Dans des tests effectués dans le cadre de tâches de reconnaissance sur tous les locuteurs masculins de la condition principale de NIST-SRE 2006, le meilleur système LM-dGMM à 512 composantes aux *10*-meilleures gaussiennes ($k = 10$) affiche un taux de bonne identification de 78.40% et un EER égal à 9.66%, alors que le système GMM a un taux de 77.88% de bonne identification et un EER égal à 9.74%. L'apprentissage restreint aux k-meilleures gaussiennes sélectionnées avec l'UBM est très rapide par rapport aux modèles LM-dGMM de base, tout en donnant de bonnes performances. Cependant, les performances restent meilleures avec une initialisation GMM (on initialise les LM-dGMM par des GMM) qu'avec une initialisation UBM, lors de l'apprentissage.

Nous avons évalué ensuite nos modèles en intégrant le formalisme SFA, en initialisant nos modèles par des GMM compensés et en utilisant des données compensées dans le domaine des caractéristiques. Ces modèles LM-dGMM-SFA donnent de bons résultats : le meilleur système LM-dGMM-SFA à 512 composantes aux *10*-meilleures gaussiennes enregistre un taux de bonne identification et un EER égaux respectivement à 91.27% et 5.02%. Le système GMM-SFA a un taux de 90.75% de bonne identification et un EER égal à 5.53%, et le meilleur système GSL-NAP affiche des taux de l'ordre de 87.77% et 5.90%. Nos modèles atteignent de meilleures performances que l'approche discriminante de référence GSL-NAP. Même s'il n'y a pas une grande amélioration par rapport aux modèles GMM-SFA, notre approche reste intéressante et prometteuse.

Nous avons réalisé un système de reconnaissance du locuteur qui combine la modélisation LM-dGMM avec celle par SVM, en construisant des supervecteurs par concaténation des vecteurs moyennes des modèles LM-dGMM-SFA, puis en les classant par des SVM à noyau linéaire. En vérification, ce système à 512 composantes gaussiennes et $k = 10$ donne un EER égal à 4.39%, améliorant ainsi les performances de chaque modélisation utilisée seule. Ce résultat révèle que ces deux approches de modélisation sont complémentaires.

En vue de regrouper l'ensemble des résultats obtenus et de permettre une plus claire et globale comparaison entre les différentes approches de modélisation proposées et étudiées, les tableaux Tab. 5.1, Tab. 5.2, Tab. 5.3, Tab. 5.4, Tab. 5.5 et Tab. 5.6, exposent les résultats des systèmes GMM, LM-GMM, LM-dGMM de base et LM-dGMM aux 10-meilleures gaussiennes à $M = 16, 32, 64, 128, 256$ et 512 composantes, qui ont été évalués dans le cadre du premier protocole expérimental :

- 50 locuteurs masculins de la condition principale (1conv4w-1conv4w) de NIST-SRE 2006.
- 232 séquences de parole de test.

Les systèmes à grande marge sont utilisés dans les deux cadres d'initialisation : initialisation des modèles par les GMM (apprentissage GMM) et initialisation par l'UBM (apprentissage UBM). Dans les systèmes aux k-meilleures gaussiennes, les résultats des quatre alternatives de sélection de ces composantes sont donnés :

- Sélection des composantes des phases d'apprentissage et de test en utilisant l'UBM ; notée dans les tableaux (UBM , UBM).
- Sélection des composantes des phases d'apprentissage et de test en utilisant respectivement l'UBM et les GMM ; notée dans les tableaux (UBM , GMM).
- Sélection des composantes des phases d'apprentissage et de test en utilisant respectivement les GMM et l'UBM ; notée dans les tableaux (GMM , UBM).
- Sélection des composantes des phases d'apprentissage et de test en utilisant les GMM ; notée dans les tableaux (GMM , GMM).

Signalons finalement que nous donnons directement pour tous ces modèles, les résultats à la meilleure itération de L-BFGS.

Système			T. bonne identification
GMM			61.2%
LM-GMM	Apprentissage GMM		60.3%
	Apprentissage UBM		59.1%
LM-dGMM de base	Apprentissage GMM		67.7%
	Apprentissage UBM		69.4%
LM-dGMM aux *10*-meilleures gauss.	Apprentissage UBM		69.8%
	Apprentissage GMM	Sélect. composantes : (UBM , UBM)	67.7%
		Sélect. composantes : (UBM , GMM)	67.2%
		Sélect. composantes : (GMM , UBM)	65.9%
		Sélect. composantes : (GMM , GMM)	65.9%

TAB. 5.1: Synthèse des résultats de systèmes à 16 composantes, évalués dans le cadre du premier protocole expérimental.

Système			T. bonne identification
GMM			68.1%
LM-GMM	Apprentissage GMM		72.0%
	Apprentissage UBM		57.8%
LM-dGMM de base	Apprentissage GMM		71.6%
	Apprentissage UBM		69.8%
LM-dGMM aux *10*-meilleures gauss.	Apprentissage UBM		69.4%
	Apprentissage GMM	Sélect. composantes : (UBM , UBM)	69.8%
		Sélect. composantes : (UBM , GMM)	71.1%
		Sélect. composantes : (GMM , UBM)	70.7%
		Sélect. composantes : (GMM , GMM)	71.1%

TAB. 5.2: Synthèse des résultats de systèmes à 32 composantes, évalués dans le cadre du premier protocole expérimental.

Système			T. bonne identification
GMM			69.8%
LM-GMM	Apprentissage GMM		68.5%
	Apprentissage UBM		51.7%
LM-dGMM de base	Apprentissage GMM		69.4%
	Apprentissage UBM		73.7%
LM-dGMM aux *10*-meilleures gauss.	Apprentissage UBM		71.6%
	Apprentissage GMM	Sélect. composantes : (UBM , UBM)	72.8%
		Sélect. composantes : (UBM , GMM)	72.4%
		Sélect. composantes : (GMM , UBM)	70.3%
		Sélect. composantes : (GMM , GMM)	69.4%

TAB. 5.3: Synthèse des résultats de systèmes à 64 composantes, évalués dans le cadre du premier protocole expérimental.

Système			T. bonne identification
GMM			74.1%
LM-dGMM de base	Apprentissage GMM		-
	Apprentissage UBM		74.1%
LM-dGMM aux *10*-meilleures gauss.	Apprentissage UBM		74.1%
	Apprentissage GMM	Sélect. composantes : (UBM , UBM)	77.2%
		Sélect. composantes : (UBM , GMM)	76.7%
		Sélect. composantes : (GMM , UBM)	74.1%
		Sélect. composantes : (GMM , GMM)	74.1%

TAB. 5.4: Synthèse des résultats de systèmes à 128 composantes, évalués dans le cadre du premier protocole expérimental.

Système			T. bonne identification
GMM			73.3%
LM-dGMM de base	Apprentissage GMM		-
	Apprentissage UBM		72.0%
LM-dGMM aux *10*-meilleures gauss.	Apprentissage UBM		71.1%
	Apprentissage GMM	Sélect. composantes : (UBM , UBM)	77.6%
		Sélect. composantes : (UBM , GMM)	77.2%
		Sélect. composantes : (GMM , UBM)	73.3%
		Sélect. composantes : (GMM , GMM)	73.7%

Tab. 5.5: Synthèse des résultats de systèmes à 256 composantes, évalués dans le cadre du premier protocole expérimental.

Système			T. bonne identification
GMM			74.1%
LM-dGMM de base	Apprentissage GMM		-
	Apprentissage UBM		-
LM-dGMM aux *10*-meilleures gauss.	Apprentissage UBM		62.5%
	Apprentissage GMM	Sélect. composantes : (UBM , UBM)	75.4%
		Sélect. composantes : (UBM , GMM)	75.4%
		Sélect. composantes : (GMM , UBM)	74.1%
		Sélect. composantes : (GMM , GMM)	73.7%

Tab. 5.6: Synthèse des résultats de systèmes à 512 composantes, évalués dans le cadre du premier protocole expérimental.

Les tableaux Tab. 5.7 et Tab. 5.8 affichent les performances des systèmes GMM, LM-dGMM, GSL, GSL-NAP, GMM-SFA, LM-dGMM-SFA et (LM-dGMM-SFA) – SVM, qui ont été évalués dans le cadre du deuxième protocole expérimental :

- 349 locuteurs masculins de la condition principale (1conv4w-1conv4w) de NIST-SRE 2006.
- 1546 fichiers de test en tâche d'identification.
- 22123 unités d'évaluation en tâche de vérification.

Système	Taux de bonne identification	EER	minDCF(x100)
GMM	75.87%	9.43%	4.26
LM-dGMM	77.62%	8.97%	3.97
GSL	81.18%	7.53%	3.40
GSL-NAP	87.19%	6.45%	2.71
GMM-SFA	89.26%	6.15%	2.41
LM-dGMM-SFA	89.65%	5.58%	2.29

Tab. 5.7: Performances en reconnaissance du locuteur de modèles GMM, LM-dGMM et GSL à 256 gaussiennes, avec et sans compensation de la variabilité inter-sessions.

Système	Taux de bonne identification	EER	minDCF(x100)
GMM	77.88%	9.74%	4.18
LM-dGMM	78.40%	9.66%	4.12
GSL	82.21%	7.23%	3.44
GSL-NAP	87.77%	5.90%	2.73
GMM-SFA	90.75%	5.53%	2.18
LM-dGMM-SFA	91.27%	5.02%	2.18
(LM-dGMM-SFA) – SVM	-	4.39%	2.16

Tab. 5.8: Performances en reconnaissance du locuteur de modèles GMM, LM-dGMM, GSL, GSL-NAP, GMM-SFA, LM-dGMM-SFA et (LM-dGMM-SFA) – SVM à 512 gaussiennes.

À la fin de nos travaux de recherche, nous avons proposé une nouvelle stratégie de traitement des données aberrantes au niveau de la trame. Elle consiste à associer des poids d'influence aux différentes trames du signal, ce qui permet de pondérer leur impact et de réduire tout effet nuisible lors de la phase d'apprentissage. En vérification, cette pondération par trame donne des résultats sensiblement meilleurs que l'approche initiale à poids segmentaux, qui consistait à définir des pondérations par segment. Nous réduisons le EER des systèmes LM-dGMM et LM-dGMM-SFA respectivement d'à peu près 2% et 2.6%. Cependant, on obtient avec ces deux approches les mêmes résultats en identification.

Il serait donc intéressant d'adapter cette approche aux modèles GMM classique. Le problème d'occurrence et de traitement des données aberrantes reste un problème important en reconnaissance automatique du locuteur.

Perspectives

Plusieurs perspectives restent ouvertes à ce travail de thèse.

La perspective principale est par rapport au processus de développement. L'algorithme d'apprentissage ne garantit d'obtenir le meilleur modèle à la dernière itération du processus d'optimisation. Ne disposant dans la condition principale des évaluations NIST-SRE que d'un seul enregistrement pour apprendre un modèle de locuteur, nous n'avons pas étudié directement cette problématique. L'initialisation des modèles à grande marge de base (à contraintes non restreintes aux k-meilleures gaussiennes) par le modèle du monde permet de régler ce problème. Mais la complexité calculatoire de ces modèles rend leur utilisation dans les scénarios à grands volumes de données, à plusieurs classes et plusieurs composantes gaussiennes, très difficile.

Parmi les autres perspectives, on peut citer :

- L'approfondissement de l'étude de la complémentarité entre les modélisation LM-dGMM et SVM, en étudiant notamment le comportement du système au fil des itérations de L-BFGS. En plus d'améliorer les performances de reconnaissance, la fusion des deux approches peut éventuellement régler le problème du critère d'arrêt, en proposant théoriquement aux SVM en fin d'itérations du processus d'optimisation, des données (des supervecteurs) "mieux" séparables, i.e., plus faciles à modéliser.
- La définition d'une nouvelle marge minimale à satisfaire ; des tests ont mis en évidence qu'une sélection d'une meilleure marge de séparation permet d'améliorer la performance de nos modèles, ce qui est consistent avec ce qui est fait dans la modélisation par des SVM.

Il serait intéressant aussi d'appliquer la notion de marge dans un espace autre que celui des caractéristiques, à l'image des systèmes GSL : définir des contraintes à grande marge par exemple dans l'espace des supervecteurs et de la variabilité totale. Cette reformulation aura des conséquences directes sur la complexité algorithmique et pourra de plus donner de très bons résultats. Dans l'espace des supervecteurs, on représentera chaque segment de données par un supervecteur formé à partir d'un GMM appris sur ces données là. On modélisera par la suite chaque classe par un ensemble d'ellipsoïdes. Les paramètres de ce mélange (les vecteurs centroïdes) seront ré-estimés en satisfaisant des contraintes de grande marge sur les distances entre les "supervecteurs-données" et les "supervecteurs-centroïdes". Cette approche de modélisation revient à remplacer les demi-plans définis par les SVM par un mélange d'ellipsoïdes, i.e., à construire une frontière de décision non-linéaire de type quadratique au lieu d'un hyperplan de séparation ; de ce fait cette approche est plus adaptée à des applications multi-classes que l'approche SVM. Dans l'espace de la variabilité totale, un locuteur est représenté par un vecteur contenant les facteurs de

la variabilité totale (i-vecteur) qui englobe la plupart des informations pertinentes sur l'identité du locuteur. Dans cette espace à faible dimension[16], les locuteurs sont symbolisés par des vecteurs discriminants ; l'analyse de facteur joue ici le rôle d'un extracteur de paramètres discriminants. De ce fait, l'utilisation des GMM à grande marge dans le cadre des systèmes à variabilité totale est très prometteuse. L' apprentissage de modèles LM-dGMM sur les i-vecteurs paraît donc être plus judicieux que leur apprentissage par des SVM classiques ou le simple calcul de distances cosinus entre i-vecteurs, qui sont faits actuellement en littérature.

Bien qu'on soit intéressé par la reconnaissance du locuteur, Il serait bien aussi d'évaluer nos modèles dans d'autres problèmes de classification, comme par exemple la reconnaissance de la langue parlée. Cette dernière empreinte les techniques traditionnelles utilisées en traitement de la parole : GMM, adaptation MAP, estimation MMI, analyse de facteur, variabilité totale, SVM ... [Rodríguez-Fuentes et al., 2012]. Ainsi, nos modèles à grande marge peuvent constituer une alternative à ces systèmes classiques à base de GMM et/ou SVM.

[16]La dimension de l'espace de la variabilité totale est expérimentalement de l'ordre de 500, tandis que celle des supervecteurs est de l'ordre de 25600.

Annexes

1 Description de systèmes de la campagne d'évaluation NIST-SRE 2010

Cette section décrit certains des meilleurs systèmes de vérification du locuteur, soumis à la campagne d'évaluation NIST-SRE 2010 [NIST, 2010].

1.1 SRI

VAD	- Une VAD pré-paramétrisation, qui utilise un décodeur HMM et diverses contraintes temporelles. - La segmentation des fichiers de type de parole interview, repose sur le résultat de la VAD et des transcriptions de la parole.
Sous-système 1 : Cepstral GMM-JFA	- (19 coef. MFCC + énergie) + dérivées premières et secondes. - Normalisation CMVN (intégralité de la séquence). - JFA : UBM à 1024 gaussiennes. Combinaison de deux matrices du canal $\mathbf{U}_{conv-tel}$ (apprise sur des données de type de parole conversation téléphonique) et $\mathbf{U}_{int}$ (apprise sur des données de type de parole interview). - ZT-norm dépendante du genre.
Sous-système 2 : Constrained Cepstral GMM-JFA	➢ *Restriction (durant les expériences menées) aux vecteurs de données des syllabes contenant le phonème [n] ou [ng] (18% de l'ensemble des données labellisées parole)*[17]. - (19 coef. MFCC + énergie) + dérivées premières et secondes. - Normalisation CMVN. - JFA : UBM à 1024 gaussiennes. Combinaison des deux matrices du canal $\mathbf{U}_{conv-tel}$ et $\mathbf{U}_{int}$. - ZT-norm dépendante du genre.

[17] L'idée est de garder les mêmes vecteurs de données, mais tout en les filtrant et les appariant (apprentissage/test).

Sous-système 3 : Cepstral GMM-JFA	➢ *Utilisation d'un UBM dépendant du genre.* ➢ *Les matrices* **V** *et* **U** *de la JFA sont indépendantes de la condition.* ➢ *La matrice* **D** *de la JFA n'est pas estimée.* ➢ *Lors du calcul des scores, les statistiques de la séquence de test sont normalisées par une valeur dérivée de la matrice* **V** *du locuteur.* - (19 coef. MFCC + énergie) + dérivées premières et secondes. - Normalisation CMVN. - JFA : UBM à 512 gaussiennes. Concaténation de matrices $\mathbf{U}_{tel}$ (apprise sur des données de type de parole conversation téléphonique, à canal téléphonique), $\mathbf{U}_{mic}$ (apprise sur des données de type de parole conversation téléphonique enregistrées via microphone), $\mathbf{U}_{int}$, $\mathbf{U}_{SRE10dev}$ (apprise sur les données de développement de NIST-SRE 2010). - ZT-norm dépendante des conditions.
Sous-système 4 : Cepstral GMM-JFA	➢ *(12 coef. PLP + énergie) + dérivées premières, secondes et troisièmes − > CMVN − > VTLN − > LDA + MLLT − > feature CMLLR − > 39 coefficients.* - Utilisation d'un UBM dépendant du genre. - Les matrices **V** et **U** de la JFA sont indépendantes de la condition. - La matrice **D** de la JFA n'est pas estimée. - JFA : UBM à 1024 gaussiennes. Concaténation de matrices $\mathbf{U}_{tel}$, $\mathbf{U}_{mic}$, $\mathbf{U}_{int}$, $\mathbf{U}_{SRE10dev}$. - ZT-norm dépendante des conditions. - Les statistiques de la séquence de test sont normalisées par une valeur dérivée de la matrice **V** du locuteur.
Sous-système 5 : MLLR-SVM	- La dimension des supervecteurs MLLR est 24960. - Normalisation dépendante du rang. - Compensation NAP. - Modélisation par des SVM à noyau linéaire.
Sous-système 6 : Word N-gram SVM	- Les 9000 plus fréquents bigrammes et trigrammes de mots sont utilisés comme caractéristiques. - Normalisation dépendante du rang. - Modélisation par des SVM à noyau linéaire.

Sous-système 7 : Prosodic	- Le sous-système prosodique est composé de 10 s-sous-systèmes, dont on combine les scores. - Ils utilisent tous le même "type" de paramètres : les coefficients de l'approximation polynomiale de Legendre de l'ordre 5 de la fréquence fondamentale et de l'énergie dans une certaine région, augmentés par la durée de cette région. - 3 régions sont utilisées : vallée de l'énergie, syllabe et "région uniforme". - Modélisation par JFA.
Fusion	- Combinaison dépendante de la condition (durée, type de parole, type de canal) par régression linéaire logistique, et en utilisant des métadonnées (nombre de mots, SNR, RMS, nativité, genre). - Données de développement choisies pour ressembler au maximum aux données de l'évaluation. - Fusion des 7 sous-systèmes : soumission 1. - Fusion de 6 sous-systèmes (le sous-système 2 n'est pas utilisé) : soumission 2.
Calibration	Données de développement choisies pour ressembler au maximum aux données de l'évaluation.

1.2 SVIST

Paramétrisation	➢ (19 coef. MFCC + énergie) + dérivées premières et secondes. ➢ 18 coef. LPCC + dérivées premières. ➢ 18 coef. PLP + dérivées premières.
VAD	- Une simple VAD qui est basée sur l'énergie. - ETSI VAD.
Norm. param. acoustiques	RASTA, Feature Warping.
Modélisation - Compensation	➢ JFA : UBM à 2048 gaussiennes. Concaténation de matrices **U** (tel, mic, int). ➢ Variabilité totale : Modèles à 2048 gaussiennes. Apprentissage des matrices **T**, A (la matrice de la projection LDA) et W (la matrice de la normalisation WCCN) sur des données de type conversation téléphonique (canal téléphonique et microphone) et interview.

	➢ GSL-NAP : Modèles à 512 gaussiennes. Apprentissage de la matrice **S** de NAP sur des données de type conversation téléphonique (canal téléphonique et microphone) et interview.
Norm. scores	TZ-norm dépendante des conditions d'apprentissage et de test.
Fusion	- Fusion linéaire des scores de 3 sous-systèmes (soumission 1) : (JFA – PLP), (Variabilité totale – LPCC) et (GSL-NAP – MFCC). - Fusion linéaire des scores de 2 sous-systèmes (soumission 2) : (JFA – PLP) et (GSL-NAP – MFCC).
Calibration	Simulation des conditions de l'évaluation, en sélectionnant des données de développement (de NIST SRE 2008) proches à celles de l'évaluation.

1.3 iFly

Sous-système 1 : PLP + JFA	- 13 coef. PLP + dérivées premières et secondes. - Une VAD basée sur l'énergie. - RASTA, Gaussianisation. - JFA : UBM à 1024 gaussiennes. Matrices JFA de rangs $R_v = 300$, $R_{u_{tel}} = 100$, $R_{u_{mic}} = 50$ et $R_{u_{int}} = 100$. - TZ-norm dépendante des conditions d'apprentissage et de test.
Sous-système 2 : LPCC et PLP + JFA	4 paramétrisations : - 18 coef. LPCC + dérivées premières et secondes. - Un filtre de Wiener est utilisé pour débruiter les données microphone et interview $->$ 18 coef. LPCC + dérivées premières et secondes. - 13 coef. PLP + dérivées premières et secondes. - Un filtre de Wiener est utilisé pour débruiter les données microphone et interview $->$ 13 coef. PLP + dérivées premières et secondes. - Microphone et interview : Une VAD basée sur l'énergie, couplée avec une autre technique basée sur un modèle du bruit. - Téléphone : Seule la VAD basée sur l'énergie est utilisée. - RASTA, Gaussianisation.

	- JFA : UBM à 1024 gaussiennes. $R_v = 300$. - Matrice du canal dépendante de la sous condition d'évaluation : tel-tel $->$ $R_{u_{tel}} = 100$. mic-mic $->$ $R_{u_{mic}} = 100$. int-int $->$ $R_{u_{int}} = 100$. int-tel $->$ **U** est apprise sur les données des locuteurs ayant à la fois des enregistrements de type conversation téléphonique et interview. int-mic $->$ concaténation de deux matrices **U** de rangs $R_{u_{int}} = 50$ et $R_{u_{mic}} = 50$. - TZ-norm dépendante des conditions d'apprentissage et de test. - Les scores des 4 s-sous-systèmes (utilisant chacun une des paramétrisations) sont fusionnés en leur associant des poids égaux.
Sous-système 3 : LPCC et PLP + GSL-NAP	4 paramétrisations : - 18 coef. LPCC + dérivées premières. - Un filtrage de Wiener $->$ 18 coef. LPCC + dérivées premières. - 13 coef. PLP + dérivées premières et secondes. - Un filtrage de Wiener $->$ 13 coef. PLP + dérivées premières et secondes. - Microphone et interview : Une VAD basée sur l'énergie, couplée avec une autre technique basée sur un modèle du bruit. - Téléphone : Seule la VAD basée sur l'énergie est utilisée. - RASTA, Gaussianisation. - GSL-NAP : Modèles à 512 gaussiennes. - Une matrice NAP de rang 64 par sous condition d'évaluation : $\mathbf{S}_{tel-tel}$, $\mathbf{S}_{mic-mic}$,$\mathbf{S}_{int-tel}$ (apprise sur des données interview + téléphone), $\mathbf{S}_{int-int}$, $\mathbf{S}_{int-mic}$ (apprise sur des données interview + microphone). - TZ-norm dépendante des conditions d'apprentissage et de test. - Fusion linéaire des scores des 4 s-sous-systèmes.
Fusion et calibration	- 3 systèmes soumis fusionnant chacun les 3 sous-systèmes : (PLP + JFA), (LPCC et PLP + JFA) et (LPCC et PLP + GSL-NAP). - La différence entre les 3 systèmes soumis se ramène au choix du seuil de décision : le seuil du système primaire est choisi par rapport au minDCF (sur les données de développement) et les seuils des deux autres systèmes sont choisis par rapport à l'EER et à la courbe DET.

1.4 ABC (Agnitio, But, Crim)

Sous-système 1 : BUT JFA'08	- (19 coef. MFCC + énergie) + dérivées premières et secondes. - Short-time Gaussianization. - La VAD utilise un système de reconnaissance des phonèmes Hongrois, et un post-traitement basé sur l'énergie. - JFA : UBM à 2048 gaussiennes. $R_v = 300$, une seule matrice du canal $\mathbf{U}_{allcond} = [\mathbf{U}_{tel}\mathbf{U}_{mic}\mathbf{U}_{int}]$ ($R_{u_{tel}} = 100$, $R_{u_{mic}} = 100$, $R_{u_{int}} = 20$). Scoring linéaire. - ZT-norm dépendante des conditions.
Sous-système 2 : BUT JFA'10	La seule différence par rapport au sous-système 1 résulte en de différentes matrices de la JFA : ➢ La matrice $\mathbf{D}$ de la JFA n'est pas estimée. ➢ Deux matrices du canal $\mathbf{U}_{tel-tel} = [\mathbf{U}_{tel}\mathbf{U}_{mic}]$ et $\mathbf{U}_{int-tel} = \mathbf{U}_{int-int} = [\mathbf{U}_{tel}\mathbf{U}_{mic}\mathbf{U}_{int}]$ ($R_{u_{tel}} = 100$, $R_{u_{mic}} = 100$, $R_{u_{int}} = 50$).
Sous-système 3 : Agnitio I-Vector	Système désigné uniquement pour les données téléphoniques. - 60 coef. MFCC. Two-covariance modeling : UBM indépendant du genre à 2048 gaussiennes $->$ i-vecteurs de dimension $R_T = 400$. Modélisation des i-vecteurs par un "two-covariance model". - Normalisation des scores par une version symétrique de la ZT-norm.
Sous-système 4 : BUT I-Vector Full Cov	- (19 coef. MFCC + énergie) + dérivées premières et secondes. - Short-time Gaussianization. - La VAD utilise un système de reconnaissance des phonèmes Hongrois, et un post-traitement basé sur l'énergie. - Variabilité totale : UBM indépendant du genre, à 2048 gaussiennes à matrices de covariance pleines. $R_T = 400$. - S-norm dépendante des conditions.

Sous-système 5 : BUT I-Vector LVCSR	➢ Seul l'UBM change par rapport au sous-système 4 : UBM indépendant du genre, à 2048 gaussiennes à matrices de covariance diagonales, dérivé d'un système de reconnaissance de la parole continue à large vocabulaire (via clustering).
Sous-système 6 : PLDA I-Vector	- Un système PLDA basé sur les i-vecteurs du sous-système 4. - Le système utilise une version différente de l'algorithme EM pour estimer les paramètres du modèle. Cette version utilise une mise-à-jour "minimum divergence" additionnelle, qui accélère la convergence.
Sous-système 7 : Prosodic JFA	- Caractéristiques : durée, fréquence fondamentale à "court terme", énergie à "court terme". - 6 coefficients DCT des trajectoires temporelles de la fréquence fondamentale et de l'énergie (en se limitant aux trames voisées sur des fenêtres de 300ms, à décalage de 50ms), augmentés par la durée, constituent les vecteurs paramétriques. - JFA : UBM à 512 gaussiennes. Scoring linéaire. - ZT-norm dépendante des conditions.
Sous-système 8 : SVM CMLLR-MLLR	- $LVCSR - PLP12_{0DAT}$ + VTLN + HLDA ... $->$ 39 coefficients. - Un 2-classes CMLLR modélise la parole et le silence, et un 3-classes MLLR modélise deux clusters de données parole et le silence. - Utilisation des transcriptions de la parole fournies par NIST. - Chaque séquence de parole est finalement représentée par un supervecteur normalisé, formé à partir des matrices $CMLLR_{parole}$ et $MLLR_{parole1,2}$ (normalisation du rang). - Compensation NAP. - Modélisation par des SVM à noyau linéaire.
Sous-système 9 : HT-PLDA (Système du Crim)	- (19 coef. MFCC + énergie) + dérivées premières et secondes. - Une VAD complexe : On y trouve des caractéristiques "d'initialisation", l'énergie, CMS, 2 modèles GMM "bruit" et "parole" à 4 et 16 composantes ... - Elle est coûteuse en calcul et ne peut être utilisée en temps-réel. - Feature Warping. - HT-PLDA. - La S-norm est utilisée dans les unités d'évaluation invoquant des données à type de canal microphone.

	- Développement sur uniquement les données des locutrices de SRE08.
Fusion	Utilisation de diverses mesures de qualité[18](d'Agnitio et/ou du BUT), qui s'avèrent très bénéfiques d'après les résultats.
Calibration	- Les données sont choisies dans le but de s'acclimater avec la nouvelle fonction DCF de NIST. - La calibration est optimisée en minimisant la cross-entropy.

1.5 LPT

Paramétrisation	4 paramétrisations : ➢ 300-3400Hz, Feature Warping (fenêtre de 3sec) post-VAD : - 12 coef. MFCC ($c_1 - c_{12}$) + 13 dérivées premières ($\Delta c_0 - \Delta c_{12}$). ➢ 0-4000Hz, Feature Warping (fenêtre de 3sec) pré-VAD : - 13 coef. PLP + dérivées premières, - 20 coef. MFCC + dérivées premières et secondes, - 20 coef. PLP + dérivées premières et secondes.
VAD	Décodeur phonétique (HMM-ANN).
Annulation de l'écho	- Conversation téléphonique : basée sur la VAD des deux canaux (comparaison de l'énergie). - Interview : basée sur les tours de paroles (transcription de la parole).
Modélisation - Compensation	UBM à 2048 gaussiennes. ➢ JFA : La matrice $\mathbf{V}$ est estimée sur des données téléphoniques uniquement. La matrice du canal résulte de la concaténation de trois matrices $\mathbf{U}_{tel}$, $\mathbf{U}_{mic}$ et $\mathbf{U}_{int}$. La matrice $\mathbf{D}$ n'est pas estimée. ➢ Variabilité totale.
Norm. scores	- ZT-norm. - AT-norm.
Fusion	- Fusion des scores des 8 systèmes via FoCal. - Les paramètres de fusion sont dépendants de la condition.
Calibration	Calibration dépendante de la condition (développement sur NIST-SRE 2008 , FoCal).

[18]Les mesures de qualité sont des statistiques issues du signal de parole, ne contenant presque aucune information discriminante, mais qui peuvent aider la calibration de scores discriminants. Parmi les mesures de qualité, on peut citer : discordance de genre, log du nombre de frames, SNR, log-vraisemblances de GMM Parole et Parole+Silence (appris sur de la parole et du silence), discordance du canal. Une basse qualité du signal (par exemple, un bas SNR) induira un score quasi nul.

1.6 I4U (IIR, USTC/iFly, UEF, UNSW, NTU)

Extraction de paramètres	➢ 13 coef. PLP + dérivées premières et secondes. - Une VAD basée sur la détection de l'impulsion de l'énergie. - RASTA, CMS, Gaussianisation. ➢ 18 coef. LPCC + dérivées premières. - Une VAD basée sur la détection de l'impulsion de l'énergie. - RASTA, CMS, Gaussianisation. ➢ 16 coef. MFCC + 16 dérivées premières + 14 dérivées secondes. - Une méthode de réduction du bruit basée sur la soustraction spectrale, est utilisée pour assister une VAD basée sur l'énergie. Les transcriptions de parole sont utilisées de plus, pour les données de type interview. - RASTA , CMVN. ➢ 28 coef. SCM-SCF : - 14 filtres de Gabor espacés selon l'échelle de Mel sont utilisés pour décomposer le signal de parole en 14 signaux sous-bande, pour calculer ensuite 28 coefficients SCM-SCF[19]. - VAD basée sur l'énergie. ➢ SWLP : La *Stabilized Weighted Linear Prediction* est une méthode d'analyse pour le calcul de coefficients MFCC, où le spectre FFT est remplacé par un spectre tout-pôle. Notons que l'ensemble des autres étapes restent inchangées[20].
Modélisation - Compensation	➢ JFA : UBM à 1024 gaussiennes. $R_v = 300$, $R_{u_{tel}} = 100$, $R_{u_{mic}} = 50$, $R_{u_{int}} = 100$. Scoring linéaire. - Une T-norm dépendante du canal d'apprentissage et une Z-norm dépendante du canal du test.

[19] Les SCM-SCF (Spectral Centroid Magnitude - Spectral Centroid Frequency) sont la magnitude et la fréquence moyenne pondérée, en chaque sous-bande. Ces deux mesures contiennent des informations relatives aux formants.

[20] L'idée est d'utiliser une fonction de pondération basée sur l'énergie à court terme pour pondérer le résidu de prédiction, de telle sorte que la modélisation tout-pôle se focalise sur les portions haute-énergie. Ces portions sont assumées être moins corrompues par le bruit et correspondent à la phase de fermeture glottale, permettant ainsi de mieux caractériser le conduit vocal.

	➢ GSL-NAP : GMM à 512 gaussiennes. La matrice **S** de NAP est de rang 60. - T-norm. ➢ GMM-SVM-BHATT : ✦ Utilisation de la distance de Bhattacharyya comme mesure de distance entre supervecteurs GMM, au lieu de la divergence KL. ✦ Adaptation MAP d'à la fois des vecteurs moyennes et des matrices de covariance. ✦ Le noyau de Bhattacharyya n'utilise l'information du poids, tandis qu'il utilise le supervecteur du modèle UBM. Les matrices de covariance servent à la normalisation. -GMM à 1024 gaussiennes. La matrice **S** de NAP est de rang 60. ➢ GMM-SVM-FT : GMM à 512 gaussiennes. La matrice **S** de NAP est de rang 60.
Fusion	- Soumission de deux systèmes combinant les 4 techniques de modélisation et les 5 ensembles de paramètres. Un sous-ensemble de 13 sous-systèmes (combinaisons) possibles sont utilisés. - Fusion linéaire des scores des sous-systèmes. Fusion dépendante des conditions d'apprentissage et de test. - Les poids de fusion choisis sont ceux minimisant la minDCF sur les données de développement (NIST-SRE 2008 et NIST-SRE 2008 follow-up).

1.7 IIR

Extraction de paramètres	Les 3 ensembles de paramètres PLP, LPCC et MFCC du système I4U.
Modélisation - Compensation	➢ JFA : UBM à 1024 gaussiennes. $R_v = 300$, $R_{u_{tel}} = 100$, $R_{u_{mic}} = 50$, $R_{u_{int}} = 100$. - TZ-norm dépendante des conditions d'apprentissage et de test. ➢ GSL-NAP : GMM à 1024 gaussiennes. La matrice **S** de NAP est de rang 60. ➢ GMM-SVM-BHATT : GMM à 1024 gaussiennes. La matrice **S** de NAP est de rang 60.

	➢ GMM-SVM-FT : GMM à 1024 gaussiennes. La matrice **S** de NAP est de rang 60.
Fusion	- Fusion linéaire de sous-systèmes. - Les poids de fusion choisis sont ceux minimisant la minDCF sur les données de développement (NIST-SRE 2008 et NIST-SRE 2008 follow-up).

1.8 MITLL

Pré-traitement	➢ Un pré-traitement des données est réalisé avant la paramétrisation : - Données téléphoniques : une annulation de l'écho standard (en utilisant les outils ISIP). - Données microphone : suppression de la tonalité continue et réduction du bruit à large bande.
Paramétrisation	➢ 20 coef. MFCC + dérivées premières et secondes. ➢ (18 coef. LPCC + énergie) + dérivées premières et secondes.
VAD	- Les fichiers interviews ont été segmentés en utilisant les transcriptions de parole fournies par NIST. - La segmentation des autres fichiers de parole utilise un *feature-based* GMM. Les résultats sont ensuite raffinés en utilisant un détecteur basé sur l'énergie uniquement.
Norm. param. acoustiques	Feature Warping.
Sous-système 1 : IPDF system	➢ Deux s-sous-systèmes sont appris (MFCC / LPCC), dont les scores sont ensuite fusionnés pour ne donner qu'un seul score. - L'approche IPDF. - Compensation par la WNAP. - ZT-norm.
Sous-système 2 : IZAT	Même configuration que dans le sous-système 1, mais avec une ZAT-norm à la place de la ZT-norm.
Sous-système 3 : JFA system	➢ Deux s-sous-système sont appris (MFCC / LPCC), dont les scores sont ensuite fusionnés pour ne donner qu'un seul score. - JFA : UBM à 1024 gaussiennes. - Une approche de validation croisée (utilisant la PCA) sert à estimer la matrice **V** (de rang 300), et deux matrices $\mathbf{U}_{tel}$ et $\mathbf{U}_{mic}$ (de rang 100 chacune). La matrice du canal résulte de la concaténation des deux précédentes matrices. - ZT-norm.

Sous-système 4 : ZAT3	Même configuration que dans le sous-système 3, mais avec une ZAT-norm à la place de la ZT-norm.
Sous-système 5 : Prosodic system	- Caractéristiques extraites au niveau pseudo-syllabique, correspondant à l'approximation polynomiale de Legendre de la fréquence fondamentale et des contours d'énergie. - Variabilité totale : UBM à 512 gaussiennes. La matrice $\mathbf{T}$ (de rang 200) est apprise sur uniquement des données téléphoniques. La LDA réduit la taille des i-vecteurs à 75. - ZT-norm.
Sous-système 6 : Eigenvoice comparaison system	➢ L'apprentissage consiste juste à estimer les facteurs du locuteur $\mathbf{y}_{s_a}$ de chaque locuteur, sans aucune modélisation de la variabilité de la session ou de compensation. Cette estimation utilise la matrice $\mathbf{V}$ du sous-système 3 (JFA). ➢ Durant le test, on commence par estimer les facteurs du locuteur de la séquence de test $\mathbf{y}_{s_t}$, et le scoring revient juste à calculer un produit scalaire entre des vecteurs $\mathbf{y}_{s_i}$ normalisés par une WCCN. Une matrice pleine W est calculée en utilisant la même liste d'apprentissage que celle qui a servie à l'apprentissage de la matrice $\mathbf{U}$ de la JFA. - Un seul système[21] à base des MFCC. - ZT-norm.
Sous-système 7 : GSL-NAP	➢ Deux s-sous-systèmes sont appris (MFCC / LPCC), dont les scores sont ensuite fusionnés pour ne donner qu'un seul score. - GSL-NAP : GMM à 2048 gaussiennes. 4000 imposteurs représentent les entrées négatives des SVM. La matrice $\mathbf{S}$ de NAP est de rang 64. - ZT-norm.
Sous-système 8 : TV system	➢ Deux sous-systèmes modélisant la Variabilité totale : le premier s'utilise pour des données téléphoniques, et le deuxième pour des données microphone ou interview. Sous-système TV_{tel} : UBM dépendant du genre à 2048 gaussiennes. Matrice $\mathbf{T}$ dépendante du genre de rang 600. La LDA réduit la taille des i-vecteurs à 250. - ZT-norm.

[21] Utilisé seul, ce système est moins bon que les autres systèmes acoustiques. Cependant, il améliore les performances globales lors de la fusion.

	Sous-système $TV_{mic-int}$ - Concaténation de deux matrices **T** dépendantes du genre : la précédente matrice à 600 vecteurs propres, et une nouvelle matrice (de rang 200) apprise sur des données microphone et interview. $->$ Projection par une PLDA dans un espace de dimension 600. $->$ réduction de l'espace à 250 par une LDA. - La matrice W de la WCCN est estimée sur des données téléphoniques, microphone et interview. - S-norm.
Sous-système 9 : SAS-TV	Même configuration que dans le sous-système 8, mais avec une SAS-norm à la place de la S-norm.
Fusion	Fusion en utilisant une régression logistique.

1.9 IBM

Paramétrisation	➢ 3 paramétrisations : - 12 MFCC + dérivées premières et secondes. - 12 LPCC + dérivées premières et secondes. - 40 paramètres basés sur l'utilisation d'un système de reconnaissance de la parole[22](l'estimation à partir de séquences de vecteurs 13 PLP, en utilisant une projection LDA et une transformation MLLT).
VAD	- Une VAD basée sur l'énergie (*fast dynamic energy noise floor tracking*), pour les paramètres cepstraux. - Dans les systèmes inspirés de la reconnaissance de parole, la segmentation est réalisée grâce à une composante du système.
Norm. param. acoustiques	Feature warping (post VAD) appliqué aux paramètres cepstraux.
Modélisation - Compensation	➢ Un sous-système GMM-NAP utilisant un PIUBM : - Paramètres ASR. - Un modèle PIUBM[23] : Modèle acoustique à 250k gaussiennes $->$ modèle du monde à 1024 gaussiennes. - Scoring dans l'espace des supervecteurs GMM (produit scalaire).

[22]Un système de base utilisant les paramètres ASR donne de meilleures résultats, en comparaison avec un système de base utilisant les MFCC.

[23]Amélioration des performances (minDCF , EER) par rapport à un système classique.

	➢ Trois sous-systèmes GMM-NAP utilisant un UBM discriminant : - Paramètres ASR ou MFCC. - UBM discriminant[24]augmentant les scores des clients et diminuant ceux des imposteurs. - Scoring dans l'espace des supervecteurs GMM (produit scalaire). ➢ Deux sous-systèmes JFA : - LPCC et MFCC. - UBM à 1024 gaussiennes. - Calcul de scores symétriques, en combinant la vraisemblance des données de test par rapport au modèle client, avec la vraisemblance des données d'apprentissage par rapport à un modèle appris sur les données de test.
Norm. scores	ZT-norm dépendante du canal.
Fusion	Fusion des six systèmes, en utilisant FoCal.

1.10 LIA

Paramétrisation	19 LFCC + 19 dérivées premières + 11 dérivées secondes + dérivée première de l'énergie.
VAD	- Modélisation de l'énergie avec un GMM à 3 gaussiennes. - Des règles morphologiques sont appliquées pour éviter de très courts segments de parole.
Norm. param. acoustiques	Normalisation CMVN (estimation sur l'intégralité de la séquence).
Modélisation - Compensation	➢ 4 sous-systèmes : (a) "GMM-SFA" Supervector Linear kernel[25](512 gaussiennes). (b) GMM-SFA (2048 gaussiennes). (c) Reverse GMM-SFA[26](2048 gaussiennes). (d) GMM-SFA (512 gaussiennes), appris que sur des données à type de canal microphone.
Norm. scores	ZT-norm.
Fusion	Fusion indépendante du canal, en utilisant FoCal.

[24]Un gain en terme de performances par rapport à un système de base.
[25]Un système GSL qui utilise les supervecteurs de modèles GMM compensés.
[26]Apprentissage sur les données de test, et scoring sur les données d'apprentissage.

	- Fusion linéaire des scores des 2 systèmes (a) et (c) (soumission 1). - Fusion linéaire des scores des 4 systèmes (soumission 2). - Fusion linéaire des scores des 2 systèmes (c) et (d) (soumission 3).

2 Analyse latente de facteur

Cette section présente en détail le formalisme SFA, qui a été proposé par le LIA dans [Matrouf et al., 2007], [Fauve et al., 2007].

Soient d et m la dimension de l'espace des données et le nombre de lois gaussiennes des GMM. Dans l'espace des supervecteurs, le supervecteur de la session h du locuteur s $\mathbf{M}_{(h,s)}$ se décompose en une composante indépendante du locuteur et de la session (introduite par l'utilisation de l'UBM), une composante dépendante du locuteur et une composante dépendante de la session [Matrouf et al., 2007] :

$$M_{(h,s)} = M + Dy_s + Ux_{(h,s)}, \tag{1}$$

où

- M est le supervecteur de l'UBM (de taille md).
- y_s est le vecteur du locuteur (de taille md) ; on assume qu'il suit une distribution normale $\mathcal{N}(0, I)$.
- D est une matrice diagonale de taille $md \times md$ où $\mathbf{DD}^T$ représente la matrice de covariance a priori de $\mathbf{y}_s$.
- U est la matrice de la variabilité de la session de rang r (une matrice de taille $md \times r$).
- $x_{(h,s)}$ sont les facteurs du canal (un vecteur de taille r) ; ce vecteur ne dépend pas théoriquement de s, et on assume qu'il suit une distribution normale $\mathcal{N}(0, \mathrm{I})$.

La technique SFA commence par estimer la matrice $\mathbf{U}$ en utilisant un nombre important de locuteurs, ayant chacun plusieurs enregistrements faits dans différentes conditions (sessions).

Notation : Soit $\mathbf{A}$ une matrice de taille $md \times k$ construite en concaténant verticalement m matrices de taille $d \times k$. Notons par $\{\mathbf{A}\}_{[g]}$ la $g^{ème}$ matrice de $\mathbf{A}$ (correspondant à la $g^{ème}$ composante du modèle).

La matrice D_g est calculée par :

$$D_g = \sqrt{\Sigma_g / \tau}, \tag{2}$$

où τ est le facteur de régulation utilisé en la technique d'adaptation MAP et Σ_g est la matrice de covariance de la $g^{ème}$ composante de l'UBM.

2.1 Estimation de la matrice U

2.1.1 Statistiques générales

Les statistiques d'ordre 0 et 1 des données sont calculées pour estimer les variables latentes ($x_{(h,s)}$ et y_s) et la matrice U.

Soient N_s et $N_{(h,s)}$ les vecteurs (de taille m) contenant les statistiques dépendantes du locuteur d'ordre 0 et les statistiques dépendantes de la session d'ordre 0 :

$$N_{s[g]} = \sum_{t \in s} \mathrm{P}(g|o_t) \quad , \quad N_{(h,s)[g]} = \sum_{t \in (h,s)} \mathrm{P}(g|o_t). \tag{3}$$

Les probabilités a postériori $\mathrm{P}(g|o_t) = \frac{w_g \mathcal{N}(o_t|\mu_g, \Sigma_g)}{\sum_{i=1}^{m} w_i \mathcal{N}(o_t|\mu_i, \Sigma_i)}$ sont calculées avec le modèle UBM. La première sommation de l'équation Eq. (3) somme sur tous les échantillons du locuteur s, alors que la deuxième somme uniquement sur les échantillons de la session h du locuteur s.

Soient X_s et $X_{(h,s)}$ les matrices (de taille $m \times d$) contenant les statistiques dépendantes du locuteur d'ordre 1 et les statistiques dépendantes de la session d'ordre 1 :

$$\{X_s\}_{[g]} = \sum_{t \in s} \left(\mathrm{P}(g|o_t).o_t \right) \quad , \quad \left\{X_{(h,s)}\right\}_{[g]} = \sum_{t \in (h,s)} \left(\mathrm{P}(g|o_t).o_t \right). \tag{4}$$

2.1.2 Estimation des variables latentes

Soient $\overline{X}_s$ et $\overline{X}_{(h,s)}$ des statistiques dépendantes du locuteur et des statistiques dépendantes du canal définies par :

$$\begin{aligned} \left\{\overline{X}_s\right\}_{[g]} &= \left\{\overline{X}_s\right\}_{[g]} - \left(\sum_{h \in s} N_{(h,s)[g]}. \left\{M + U x_{(h,s)}\right\}_{[g]} \right), \\ \left\{\overline{X}_{(h,s)}\right\}_{[g]} &= \left\{\overline{X}_{(h,s)}\right\}_{[g]} - \left(\left\{M + D y_s\right\}_{[g]} . \sum_{h \in s} N_{(h,s)[g]} \right). \end{aligned} \tag{5}$$

Comme on le verra dans la section 2.1.3, la matrice U est estimée d'une manière itérative. $\overline{X}_s$ est initialisée par X_s et elle est utilisée pour estimer le vecteur du locuteur en supprimant les effets de sessions, tandis que $\overline{X}_{(h,s)}$ est initialisée par $X_{(h,s)}$ et elle est utilisée pour estimer les facteurs du canal en supprimant les effets du locuteur.

Soient $L_{(h,s)}$ une matrice de taille $r \times r$ et $B_{(h,s)}$ un vecteur de taille r, qui sont définis par :

$$\begin{aligned} L_{(h,s)} &= \mathrm{I} + \sum_{g=1}^{m} \left(N_{(h,s)[g]}. \{U\}_{[g]}^{T} .\Sigma_g^{-1}. \{U\}_{[g]} \right), \\ B_{(h,s)} &= \sum_{g=1}^{m} \left(\{U\}_{[g]}^{T} .\Sigma_g^{-1}. \left\{\overline{X}_{(h,s)}\right\}_{[g]} \right). \end{aligned} \tag{6}$$

En utilisant les variables $L_{(h,s)}$ et $B_{(h,s)}$, les variables latentes sont calculées par les formules :

$$\begin{aligned} x_{(h,s)} &= L_{(h,s)}^{-1}.B_{(h,s)}, \\ \{y_s\}_{[g]} &= \left(\frac{\tau}{\tau+N_{s[g]}}\right).D_g.\Sigma_g^{-1}.\left\{\overline{X}_s\right\}_{[g]}. \end{aligned} \tag{7}$$

2.1.3 Estimation de la matrice de la variabilité de la session

La matrice U peut être estimée ligne par ligne. Notons pas $\{U\}_{[g]}^i$ la $i^{ème}$ ligne de $\{U\}_{[g]}$. $\{U\}_{[g]}^i$ est donnée par :

$$\{U\}_{[g]}^i = (LU_g)^{-1}.(RU_g)^i, \tag{8}$$

où les deux termes de l'équation Eq. (8) sont calculés avec :

$$\begin{aligned} LU_g &= \sum_s \sum_{h \in s} \left(L_{(h,s)}^{-1} + x_{(h,s)} x_{(h,s)}^T\right).N_{(h,s)[g]}, \\ (RU_g)^i &= \sum_s \sum_{h \in s} \left\{\overline{X}_{(h,s)}\right\}_{[g]}[i].x_{(h,s)}. \end{aligned} \tag{9}$$

2.1.4 Algorithme d'estimation de U

L'algorithme suivant présente la stratégie adoptée pour estimer la matrice U. La fonction standard de vraisemblance peut être utilisée pour évaluer la convergence.

Pour chaque locuteur s et session h : $y_s \leftarrow 0$, $x_{(h,s)} \leftarrow 0$
Initialisation aléatoire de U
Calculer les statistiques : N_s, $N_{(h,s)}$, X_s, $X_{(h,s)}$ (Eq. 3,4)
Initialiser $\overline{X}_s$ et $\overline{X}_{(h,s)}$ par respectivement X_s et $X_{(h,s)}$
POUR i de 1 à *nombre_iterations_apprentissage* **FAIRE**
  POUR chaque locuteur s et session h **FAIRE**
    Calculer et inverser $L_{(h,s)}$ (Eq. 6.1)
    Soustraire les statistiques du locuteur : $\overline{X}_{(h,s)}$ (Eq. 5.2)
    Calculer $B_{(h,s)}$ (Eq. 6.2)
    Calculer $x_{(h,s)}$ (Eq. 7.1)
    Soustraire les statistiques du canal : $\overline{X}_s$ (Eq. 5.1)
    Calculer y_s (Eq. 7.2)
  FIN POUR
  Calculer U (Eq. 9,8)
FIN POUR

2.2 Apprentissage du modèle du locuteur

Disposant d'une séquence de parole d'un locuteur s_{tar} (enregistrée lors d'une session h_{tar}) et de la matrice U, on apprend un modèle à ce locuteur en utilisant l'algorithme suivant :

$y_{s_{tar}} \leftarrow 0$, $x_{(h_{tar},s_{tar})} \leftarrow 0$
Calculer les statistiques : $N_{s_{tar}}$, $N_{(h_{tar},s_{tar})}$, $X_{s_{tar}}$, $X_{(h_{tar},s_{tar})}$ (Eq. 3,4)
POUR i de 1 à *nombre_iterations_apprentissage* **FAIRE**
  $\overline{X}_{s_{tar}} \leftarrow X_{s_{tar}}$, $\overline{X}_{(h_{tar},s_{tar})} \leftarrow X_{(h_{tar},s_{tar})}$
  Calculer et inverser $L_{(h_{tar},s_{tar})}$ (Eq. 6.1)
  Soustraire les statistiques du locuteur : $\overline{X}_{(h_{tar},s_{tar})}$ (Eq. 5.2)
  Calculer $B_{(h_{tar},s_{tar})}$ (Eq. 6.2)
  Calculer $x_{(h_{tar},s_{tar})}$ (Eq. 7.1)
  Soustraire les statistiques du canal : $\overline{X}_{s_{tar}}$ (Eq. 5.1)
  Calculer $y_{s_{tar}}$ (Eq. 7.2)
FIN POUR
$M_{(h_{tar},s_{tar})} = M + Dy_{s_{tar}}$
Former un GMM à partir du supervecteur

En pratique, une seule itération est suffisante [Matrouf et al., 2007]. L'apprentissage des modèles compensés des clients se fait en éliminant directement la dissemblance de session dans le domaine du modèle (model domain).

2.3 Phase de test

Dans la phase de test, la compensation se fait directement dans le domaine des caractéristiques (feature domain). Elle peut être considérer comme un post-traitement des données. Soit o un échantillon d'un locuteur s_{test} (enregistré lors d'une session h_{test}). La suppression de l'effet de session fait appel à l'algorithme suivant :

$y_{s_{test}} \leftarrow 0$, $x_{(h_{test},s_{test})} \leftarrow 0$
Calculer les statistiques : $N_{s_{test}}$, $N_{(h_{test},s_{test})}$, $X_{s_{test}}$, $X_{(h_{test},s_{test})}$ (Eq. 3,4)
POUR i de 1 à *nombre_iterations* **FAIRE**
  $\overline{X}_{s_{test}} \leftarrow X_{s_{test}}$, $\overline{X}_{(h_{test},s_{test})} \leftarrow X_{(h_{test},s_{test})}$
  Soustraire les statistiques du locuteur : $\overline{X}_{(h_{test},s_{test})}$ (Eq. 5.2)
  Soustraire les statistiques du canal : $\overline{X}_{s_{test}}$ (Eq. 5.1)
  Calculer et inverser $L_{(h_{test},s_{test})}$ (Eq. 6.1)
  Calculer $B_{(h_{test},s_{test})}$ (Eq. 6.2)
  Calculer $x_{(h_{test},s_{test})}$ (Eq. 7.1)
  Calculer $y_{s_{test}}$ (Eq. 7.2)
FIN POUR

$M_{(h_{test},s_{test})} = M + Dy_{s_{test}} + Ux_{(h_{test},s_{test})}$
Former un GMM à partir du supervecteur
Calculer les probabilités $\mathrm{P}(g|o)$ avec ce GMM
Normaliser l'échantillon de test suivant l'équation Eq. (10)

$$\hat{o} = o - \sum_{g=1}^{m} \left(\mathrm{P}(g|o).\left\{ Ux_{(h_{test},s_{test})} \right\}_{[g]} \right). \tag{10}$$

Bibliographie

[Adami et al., 2003] Adami, A., Mihaescu, R., Reynolds, D., and Godfrey, J. (2003). Modeling prosodic dynamics for speaker recognition. In *Proc. of ICASSP*, volume 4, pages IV–788–IV–791.

[Andrews et al., 2002] Andrews, W., Kohler, M., Campbell, J., Godfrey, J., and Hernández-Cordero, J. (2002). Gender-dependent phonetic refraction for speaker recognition. In *Proc. of ICASSP*, volume 1, pages I–149–I–152.

[Anguera and Bonastre, 2010] Anguera, X. and Bonastre, J.-F. (2010). A novel speaker binary key derived from anchor models. In *Proc. of INTERSPEECH*, pages 2118–2121.

[Archambeau and Verleysen, 2007] Archambeau, C. and Verleysen, M. (2007). Robust Bayesian clustering. *Neural Networks*, 20(1) :129–138.

[Artieres and Gallinari, 1993] Artieres, T. and Gallinari, P. (1993). Neural models for extracting speaker characteristics in speech modelization systems. In *Proc. of EUROSPEECH*, pages 2263–2266.

[Atal, 1972] Atal, B. (1972). Automatic speaker recognition based on pitch contours. *The Journal of the Acoustical Society of America*, 52(6B) :1687–1697.

[Auckenthaler et al., 2000] Auckenthaler, R., Carey, M., and Lloyd-Thomas, H. (2000). Score Normalization for Text-Independent Speaker Verification Systems. *Digital Signal Processing*, 10(1-3) :42–54.

[Avriel, 2003] Avriel, M. (2003). *Nonlinear programming : analysis and methods*. Dover Publications.

[Barras and Gauvain, 2003] Barras, C. and Gauvain, J.-L. (2003). Feature and score normalization for speaker verification of cellular data. In *Proc. of ICASSP*, volume 2, pages II–49–II–52.

[Bartkova et al., 2002] Bartkova, K., Le Gac, D., Charlet, D., and Jouvet, D. (2002). Prosodic parameter for speaker identification. In *Proc. of ICSLP*, pages 1197–1200.

[Bengio and Mariéthoz, 2001] Bengio, S. and Mariéthoz, J. (2001). Learning the decision function for speaker verification. In *Proc. of ICASSP*, volume 1, pages 425–428.

[Bennani et al., 1990] Bennani, Y., Fogelman Soulie, F., and Gallinari, P. (1990). A connectionist approach for automatic speaker identification. In *Proc. of ICASSP*, pages 265–268.

[Bertsekas, 1999] Bertsekas, D. (1999). *Nonlinear programming*. Athena Scientific, Belmont, Massachusetts.

[Bigot, 2011] Bigot, B. (2011). *Recherche du rôle des intervenants et de leurs interactions pour la structuration de documents audiovisuels*. PhD thesis, Université Toulouse 3 Paul Sabatier.

[Bimbot et al., 1992] Bimbot, F., Mathan, L., De Lima, A., and Chollet, G. (1992). Standard and target driven AR-vector models for speech analysis and speaker recognition. In *Proc. of ICASSP*, volume 2, pages 5–8.

[Bishop, 2006] Bishop, C. (2006). *Pattern recognition and machine learning*. Springer Science+Business Media, LLC, New York.

[Bocklet and Shriberg, 2009] Bocklet, T. and Shriberg, E. (2009). Speaker recognition using syllable-based constraints for cepstral frame selection. In *Proc. of ICASSP*, pages 4525–4528.

[Bonastre et al., 2008] Bonastre, J.-F., Scheffer, N., Matrouf, D., Fredouille, C., Larcher, A., Preti, A., Pouchoulin, G., Evans, N., Fauve, B., and Mason, J. (2008). ALIZE/SpkDet : a state-of-the-art open source software for speaker recognition. In *Proc. of Odyssey - The Speaker and Language Recognition Workshop*.

[Brümmer, 2005] Brümmer, N. (2005). *Tools for Fusion and Calibration of automatic speaker detection systems*. Online : http ://www.dsp.sun.ac.za/∼nbrummer/focal/.

[Brümmer and du Preez, 2006] Brümmer, N. and du Preez, J. (2006). Application-independent evaluation of speaker detection. *Computer Speech and Language*, 20(2-3) :230–275.

[Burget et al., 2011] Burget, L., Plchot, O., Cumani, S., Glembek, O., Matějka, P., and Brümmer, N. (2011). Discriminatively trained Probabilistic Linear Discriminant Analysis for speaker verification. In *Proc. of ICASSP*, pages 4832–4835.

[Burton, 1987] Burton, D. (1987). Text-dependent speaker verification using vector quantization source coding. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(2) :133–143.

[Campbell et al., 2003] Campbell, J., Reynolds, D., and Dunn, R. (2003). Fusing high- and low-level features for speaker recognition. In *Proc. of EUROSPEECH*, pages 2665–2668.

[Campbell, 2002] Campbell, W. (2002). Generalized linear discriminant sequence kernels for speaker recognition. In *Proc. of ICASSP*, volume 1, pages I–161–I–164.

[Campbell, 2010] Campbell, W. (2010). Weighted Nuisance Attribute Projection. In *Proc. of Odyssey - The Speaker and Language Recognition Workshop*, pages 97–102.

[Campbell et al., 2004a] Campbell, W., Campbell, J., Reynolds, D., Jones, D., and Leek, T. (2004a). Phonetic Speaker Recognition with Support Vector Machines. In *Advances in Neural Information Processing Systems 16*, pages 1377–1384. MIT Press.

[Campbell et al., 2006a] Campbell, W., Campbell, J., Reynolds, D., Singer, E., and Torres-Carrasquillo, P. (2006a). Support vector machines for speaker and language recognition. *Computer Speech and Language*, 20(2-3) :210–229.

[Campbell et al., 2009] Campbell, W., Karam, Z., and Sturim, D. (2009). Speaker Comparison with Inner Product Discriminant Functions. In *Advances in Neural Information Processing Systems 22*, pages 207–215. MIT Press.

[Campbell et al., 2004b] Campbell, W., Reynolds, D., and Campbell, J. (2004b). Fusing discriminative and generative methods for speaker recognition : experiments on switchboard and NFI/TNO field data. In *Proc. of Odyssey - The Speaker and Language Recognition Workshop*, pages 41–44.

[Campbell et al., 2006b] Campbell, W., Sturim, D., and Reynolds, D. (2006b). Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters*, 13(5) :308–311.

[Campbell et al., 2006c] Campbell, W., Sturim, D., Reynolds, D., and Solomonoff, A. (2006c). Svm based speaker verification using a gmm supervector kernel and nap variability compensation. In *Proc. of ICASSP*, volume 1, pages I–97–I–100.

[Carey et al., 1996] Carey, M., Parris, E., Lloyd-Thomas, H., and Bennett, S. (1996). Robust prosodic features for speaker identification. In *Proc. of ICSLP*, volume 3, pages 1800–1803.

[Chaudhari et al., 2003] Chaudhari, U., Navrátil, J., and Maes, S. (2003). Multigrained modeling with pattern specific maximum likelihood transformations for text-independent speaker recognition. *IEEE Transactions on Speech and Audio Processing*, 11(1) :61–69.

[Chen et al., 1997] Chen, K., Wang, L., and Chi, H. (1997). Methods of combining multiple classifiers with different features and their applications to text-independent speaker identification. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(3) :417–446.

[Chen et al., 2005] Chen, Z., Liao, Y., and Juang, Y. (2005). Prosody modeling and eigenprosody analysis for robust speaker recognition. In *Proc. of ICASSP*, volume 1, pages 185–188.

[Cheng and Leung, 1998] Cheng, Y. and Leung, H.-C. (1998). Speaker verification using fundamental frequency. In *Proc. of ICSLP*, volume 2, pages 161–164.

[Collet et al., 2005] Collet, M., Mami, Y., Charlet, D., and Bimbot, F. (2005). Probabilistic anchor models approach for speaker verification. In *Proc. of INTERSPEECH*, pages 2005–2008.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3) :273–297.

[Daoudi et al., 2011] Daoudi, K., Jourani, R., André-Obrecht, R., and Aboutajdine, D. (2011). Speaker Identification Using Discriminative Learning of Large Margin GMM. In *Neural Information Processing*, volume 7063 of *Lecture Notes in Computer Science*, pages 300–307. Springer Berlin / Heidelberg.

[Davis et al., 2006] Davis, A., Nordholm, S., and Togneri, R. (2006). Statistical voice activity detection using low-variance spectrum estimation and an adaptive threshold. *IEEE Transactions on Audio, Speech and Language Processing*, 14(2) :412–424.

[Davis and Mermelstein, 1980] Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4) :357–366.

[De Cheveigné and Kawahara, 2002] De Cheveigné, A. and Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4) :1917–1930.

[De Veth and Bourlard, 1995] De Veth, J. and Bourlard, H. (1995). Comparison of hidden Markov model techniques for automatic speaker verification in real-world conditions. *Speech Communication*, 17(1-2) :81–90.

[DeGroot, 1970] DeGroot, M. (1970). *Optimal statistical decisions.* McGraw-Hill.

[Dehak and Chollet, 2006] Dehak, N. and Chollet, G. (2006). Support vector GMMs for speaker verification. In *Proc. of IEEE Odyssey - The Speaker and Language Recognition Workshop.*

[Dehak et al., 2009] Dehak, N., Dehak, R., Kenny, P., Brümmer, N., Ouellet, P., and Dumouchel, P. (2009). Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification. In *Proc. of INTERSPEECH*, pages 1559–1562.

[Dehak et al., 2007] Dehak, N., Dumouchel, P., and Kenny, P. (2007). Modeling prosodic features with joint factor analysis for speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 15(7) :2095–2103.

[Dehak et al., 2011a] Dehak, N., Karam, Z., Reynolds, D., Dehak, R., Campbell, W., and Glass, J. (2011a). A channel-blind system for speaker verification. In *Proc. of ICASSP*, pages 4536–4539.

[Dehak et al., 2011b] Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., and Ouellet, P. (2011b). Front-End Factor Analysis for Speaker Verification. *IEEE Transactions on Audio, Speech and Language Processing*, 19(4) :788–798.

[Dempster et al., 1977] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1) :1–38.

[Doddington, 2001] Doddington, G. (2001). Speaker recognition based on idiolectal differences between speakers. In *Proc. of EUROSPEECH*, pages 2521–2524.

[Faltlhauser and Ruske, 2001] Faltlhauser, R. and Ruske, G. (2001). Improving speaker recognition using phonetically structured gaussian mixture models. In *Proc. of EUROSPEECH*, pages 751–754.

[Fant, 1970] Fant, G. (1970). *Acoustic theory of speech production.* Mouton.

[Farrell et al., 1994] Farrell, K., Mammone, R., and Assaleh, K. (1994). Speaker recognition using neural networks and conventional classifiers. *IEEE Transactions on Speech and Audio Processing*, 2(1) :194–205.

[Fauve et al., 2007] Fauve, B., Matrouf, D., Scheffer, N., Bonastre, J.-F., and Mason, J. (2007). State-of-the-Art Performance in Text-Independent Speaker Verification Through Open-Source Software. *IEEE Transactions on Audio, Speech and Language Processing*, 15(7) :1960–1968.

[Ferrer et al., 2008] Ferrer, L., Graciarena, M., Zymnis, A., and Shriberg, E. (2008). System combination using auxiliary information for speaker verification. In *Proc. of ICASSP*, pages 4853–4856.

[Ferrer et al., 2007] Ferrer, L., Shriberg, E., Kajarekar, S., and Sonrnez, K. (2007). Parameterization of prosodic feature distributions for SVM modeling in speaker recognition. In *Proc. of ICASSP*, volume 4, pages IV–233–IV–236.

[Fine et al., 2001a] Fine, S., Navrátil, J., and Gopinath, R. (2001a). A hybrid GMM/SVM approach to speaker identification. In *Proc. of ICASSP*, volume 1, pages 417–420.

[Fine et al., 2001b] Fine, S., Navrátil, J., and Gopinath, R. (2001b). Enhancing GMM scores using SVM "hints". In *Proc. of EUROSPEECH*, pages 1757–1760.

[Fisher, 1925] Fisher, R. (1925). Theory of statistical estimation. In *Proc. of the Cambridge Philosophical Society*, volume 22, pages 700–725.

[Fredouille et al., 2000] Fredouille, C., Bonastre, J.-F., and Merlin, T. (2000). AMIRAL : a block-segmental multirecognizer architecture for automatic speaker recognition. *Digital Signal Processing*, 10(1-3) :172–197.

[Furui, 1981] Furui, S. (1981). Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(2) :254–272.

[Gales, 1998] Gales, M. (1998). Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12(2) :75–98.

[Ganchev et al., 2004] Ganchev, T., Tasoulis, D., Vrahatis, M., and Fakotakis, N. (2004). Locally recurrent probabilistic neural networks with application to speaker verification. *GESTS International Transaction on Speech Science and Engineering*, 1(2) :1–13.

[Garcia-Romero and Espy-Wilson, 2011] Garcia-Romero, D. and Espy-Wilson, C. (2011). Analysis of i-vector Length Normalization in Speaker Recognition Systems. In *Proc. of INTERSPEECH*, pages 249–252.

[Garcia-Romero et al., 2003] Garcia-Romero, D., Fierrez-Aguilar, J., Gonzalez-Rodriguez, J., and Ortega-Garcia, J. (2003). Support vector machine fusion of idiolectal and acoustic speaker information in Spanish conversational speech. In *Proc. of International Conference on Multimedia and Expo*, volume 3, pages 205–208.

[Gauvain and Lee, 1994] Gauvain, J.-L. and Lee, C. (1994). Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2) :291–298.

[Glembek et al., 2009] Glembek, O., Burget, L., Dehak, N., Brümmer, N., and Kenny, P. (2009). Comparison of scoring methods used in speaker recognition with Joint Factor Analysis. In *Proc. of ICASSP*, pages 4057–4060.

[Gravier, 2003] Gravier, G. (2003). *SPro : "Speech Signal Processing Toolkit"*. Online : https://gforge.inria.fr/projects/spro.

[Grenier, 1980] Grenier, Y. (1980). Utilisation de la prédiction linéaire en reconnaissance et adaptation au locuteur. In *Proc. of XIèmes Journées d'Études sur la Parole (JEP)*, pages 163–171.

[Griffin et al., 1994] Griffin, C., Matsui, T., and Furui, S. (1994). Distance measures for text-independent speaker recognition based on MAR model. In *Proc. of ICASSP*, pages 309–312.

[Gudnason and Brookes, 2008] Gudnason, J. and Brookes, M. (2008). Voice source cepstrum coefficients for speaker identification. In *Proc. of ICASSP*, pages 4821–4824.

[Hansen et al., 2004] Hansen, E., Slyh, R., and Anderson, T. (2004). Speaker recognition using phoneme-specific gmms. In *Proc. of Odyssey - The Speaker and Language Recognition Workshop*, pages 179–184.

[Hatch et al., 2006] Hatch, A., Kajarekar, S., and Stolcke, A. (2006). Within-class covariance normalization for SVM-based speaker recognition. In *Proc. of INTERSPEECH*, pages 1471–1474.

[Hatch and Stolcke, 2006] Hatch, A. and Stolcke, A. (2006). Generalized Linear Kernels for One-Versus-All Classification : Application to Speaker Recognition. In *Proc. of ICASSP*, volume 5, pages V–585–V–588.

[Hattori, 1992] Hattori, H. (1992). Text-independent speaker recognition using neural networks. In *Proc. of ICASSP*, volume 2, pages 153–156.

[Hautamäki et al., 2008] Hautamäki, V., Kinnunen, T., Kärkkäinen, I., Saastamoinen, J., Tuononen, M., and Fränti, P. (2008). Maximum a posteriori adaptation of the centroid model for speaker verification. *IEEE Signal Processing Letters*, 15 :162–165.

[He et al., 1999] He, J., Liu, L., and Palm, G. (1999). A discriminative training algorithm for VQ-based speaker identification. *IEEE Transactions on Speech and Audio Processing*, 7(3) :353–356.

[Hebert and Heck, 2003] Hebert, M. and Heck, L. (2003). Phonetic class-based speaker verification. In *Proc. of EUROSPEECH*, pages 1665–1668.

[Heck et al., 2000] Heck, L., Konig, Y., Sönmez, M., and Weintraub, M. (2000). Robustness to telephone handset distortion in speaker recognition by discriminative feature design. *Speech Communication*, 31(2-3) :181–192.

[Hermansky, 1990] Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4) :1738–1752.

[Hermansky and Morgan, 1994] Hermansky, H. and Morgan, N. (1994). RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4) :578–589.

[Ho and Moreno, 2004] Ho, P. and Moreno, P. (2004). SVM kernel adaptation in speaker classification and verification. In *Proc. of INTERSPEECH*, pages 1413–1416.

[Hsu and Lin, 2002] Hsu, C. and Lin, C. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2) :415–425.

[Huenupán et al., 2007] Huenupán, F., Yoma, N., Molina, C., and Garretón, C. (2007). Speaker verification with multiple classifier fusion using Bayes based confidence measure. In *Proc. of INTERSPEECH*, pages 2041–2044.

[Ishizuka and Nakatani, 2006] Ishizuka, K. and Nakatani, T. (2006). Study of noise robust voice activity detection based on periodic component to aperiodic component ratio. In *Proc. of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audition (SAPA)*, pages 65–70.

[Jelinek and Mercer, 1980] Jelinek, F. and Mercer, R. L. (1980). Interpolated estimation of markov source parameters from sparse data. In *Proc. of Workshop Pattern Recognition in Practice*, pages 381–397.

[Jin et al., 2003] Jin, Q., Navrátil, J., Reynolds, D., Campbell, J., Andrews, W., and Abramson, J. (2003). Combining cross-stream and time dimensions in phonetic speaker recognition. In *Proc. of ICASSP*, volume 4, pages IV–800–IV–803.

[Jourani et al., 2010] Jourani, R., Daoudi, K., André-Obrecht, R., and Aboutajdine, D. (2010). Large Margin Gaussian mixture models for speaker identification. In *Proc. of INTERSPEECH*, pages 1441–1444.

[Jourani et al., 2011a] Jourani, R., Daoudi, K., André-Obrecht, R., and Aboutajdine, D. (2011a). Fast training of Large Margin diagonal Gaussian mixture models for speaker identification. In *Proc. of SpeD*, pages 1–4.

[Jourani et al., 2011b] Jourani, R., Daoudi, K., André-Obrecht, R., and Aboutajdine, D. (2011b). Speaker verification using Large Margin GMM discriminative training. In *Proc. of ICMCS*, pages 1–5.

[Jourani et al., irst] Jourani, R., Daoudi, K., André-Obrecht, R., and Aboutajdine, D. (Online First). Discriminative speaker recognition using Large Margin GMM. *Journal of Neural Computing & Applications*, doi :10.1007/s00521-012-1079-y.

[Karam and Campbell, 2007] Karam, Z. and Campbell, W. (2007). A new kernel for SVM MLLR based speaker recognition. In *Proc. of INTERSPEECH*, pages 290–293.

[Kenny, 2010] Kenny, P. (2010). Bayesian Speaker Verification with Heavy-Tailed Priors. In *Proc. of Odyssey - The Speaker and Language Recognition Workshop*.

[Kenny et al., 2005a] Kenny, P., Boulianne, G., and Dumouchel, P. (2005a). Eigenvoice modeling with sparse training data. *IEEE transactions on speech and audio processing*, 13(3) :345–354.

[Kenny et al., 2005b] Kenny, P., Boulianne, G., Ouellet, P., and Dumouchel, P. (2005b). Factor analysis simplified. In *Proc. of ICASSP*, volume 1, pages 637–640.

[Kenny et al., 2006] Kenny, P., Boulianne, G., Ouellet, P., and Dumouchel, P. (2006). Improvements in factor analysis based speaker verification. In *Proc. of ICASSP*, volume 1, pages I–113–I–116.

[Kenny et al., 2007a] Kenny, P., Boulianne, G., Ouellet, P., and Dumouchel, P. (2007a). Joint Factor Analysis Versus Eigenchannels in Speaker Recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4) :1435–1447.

[Kenny et al., 2007b] Kenny, P., Boulianne, G., Ouellet, P., and Dumouchel, P. (2007b). Speaker and session variability in GMM-based speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4) :1448–1460.

[Kenny et al., 2008] Kenny, P., Ouellet, P., Dehak, N., Gupta, V., and Dumouchel, P. (2008). A study of interspeaker variability in speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 16(5) :980–988.

[Keshet and Bengio, 2009] Keshet, J. and Bengio, S. (2009). *Automatic speech and speaker recognition : Large margin and kernel methods*. Wiley.

[Kharroubi et al., 2001] Kharroubi, J., Petrovska-Delacrétaz, D., and Chollet, G. (2001). Combining GMM's with Suport Vector Machines for Text-independent Speaker Verification. In *Proc. of EUROSPEECH*, pages 1761–1764.

[Kinnunen and Alku, 2009] Kinnunen, T. and Alku, P. (2009). On separating glottal source and vocal tract information in telephony speaker verification. In *Proc. of ICASSP*, pages 4545–4548.

[Kinnunen and González-Hautamäki, 2005] Kinnunen, T. and González-Hautamäki, R. (2005). Long-term f0 modeling for text-independent speaker recognition. In *Proc. of SPECOM*, pages 567–570.

[Kinnunen et al., 2004] Kinnunen, T., Hautamäki, V., and Fränti, P. (2004). Fusion of spectral feature sets for accurate speaker identification. In *Proc. of SPECOM*, pages 361–365.

[Kinnunen et al., 2009] Kinnunen, T., Saastamoinen, J., Hautamäki, V., Vinni, M., and Fränti, P. (2009). Comparative evaluation of maximum a posteriori vector quantization and Gaussian mixture models in speaker verification. *Pattern Recognition Letters*, 30(4) :341–347.

[Klusáček et al., 2003] Klusáček, D., Navrátil, J., Reynolds, D., and Campbell, J. (2003). Conditional pronunciation modeling in speaker detection. In *Proc. of ICASSP*, volume 4, pages IV–804–IV–807.

[Krause and Gazit, 2006] Krause, N. and Gazit, R. (2006). SVM-based speaker classification in the GMM models space. In *Proc. of IEEE Odyssey - The Speaker and Language Recognition Workshop*.

[Kuhn et al., 2000] Kuhn, R., Junqua, J.-C., Nguyen, P., and Niedzielski, N. (2000). Rapid speaker adaptation in eigenvoice space. *IEEE Transactions on Speech and Audio Processing*, 8(6) :695–707.

[Lamel et al., 1981] Lamel, L., Rabiner, L., Rosenberg, A., and Wilpon, J. (1981). An improved endpoint detector for isolated word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(4) :777–785.

[Lapidot et al., 2002] Lapidot, I., Guterman, H., and Cohen, A. (2002). Unsupervised speaker recognition based on competition between self-organizing maps. *IEEE Transactions on Neural Networks*, 13(4) :877–887.

[Laskowski and Jin, 2009] Laskowski, K. and Jin, Q. (2009). Modeling instantaneous intonation for speaker identification using the fundamental frequency variation spectrum. In *Proc. of ICASSP*, pages 4541–4544.

[Le and Bengio, 2003] Le, Q. and Bengio, S. (2003). Client Dependent GMM-SVM Models for Speaker Verification. In *Artificial Neural Networks and Neural Information Processing - ICANN/ICONIP*, volume 2714 of *Lecture Notes in Computer Science*, pages 443–451. Springer Berlin / Heidelberg.

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

[Leung et al., 2006] Leung, K., Mak, M., Siu, M., and Kung, S. (2006). Adaptive articulatory feature-based conditional pronunciation modeling for speaker verification. *Speech Communication*, 48(1) :71–84.

[Li et al., 2002] Li, Q., Zheng, J., Tsai, A., and Zhou, Q. (2002). Robust endpoint detection and energy normalization for real-time speech and speaker recognition. *IEEE Transactions on Speech and Audio Processing*, 10(3) :146–157.

[Linde et al., 1980] Linde, Y., Buzo, A., and Gray, R. (1980). An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1) :84–95.

[Liu and Nocedal, 1989] Liu, D. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1) :503–528.

[Liu et al., 2006] Liu, M., Dai, B., Xie, Y., and Yao, Z. (2006). Improved GMM-UBM/SVM for speaker verification. In *Proc. of ICASSP*, volume 1, pages I–925–I–928.

[Louradour, 2007] Louradour, J. (2007). *Noyaux de séquences pour la vérification du locuteur par Machines à Vecteurs de Support*. PhD thesis, Université Toulouse 3 Paul Sabatier.

[Louradour et al., 2007] Louradour, J., Daoudi, K., and Bach, F. (2007). Feature space mahalanobis sequence kernels : Application to svm speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 15(8) :2465–2475.

[Ma et al., 2006] Ma, B., Zhu, D., Tong, R., and Li, H. (2006). Speaker cluster based GMM tokenization for speaker recognition. In *Proc. of INTERSPEECH*, pages 505–508.

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297.

[Magrin-Chagnolleau et al., 1996] Magrin-Chagnolleau, I., Wilke, J., and Bimbot, F. (1996). A further investigation on AR-vector models for text-independent speaker identification. In *Proc. of ICASSP*, volume 1, pages 101–104.

[Mahadeva Prasanna et al., 2006] Mahadeva Prasanna, S., Gupta, C. S., and Yegnanarayana, B. (2006). Extraction of speaker-specific excitation information from linear prediction residual of speech. *Speech Communication*, 48(10) :1243–1261.

[Mak et al., 2006] Mak, M., Hsiao, R., and Mak, B. (2006). A comparison of various adaptation methods for speaker verification with limited enrollment data. In *Proc. of ICASSP*, volume 1, pages I–929–I–932.

[Mami and Charlet, 2006] Mami, Y. and Charlet, D. (2006). Speaker recognition by location in the space of reference speakers. *Speech Communication*, 48(2) :127–141.

[Mariéthoz and Bengio, 2002] Mariéthoz, J. and Bengio, S. (2002). A comparative study of adaptation methods for speaker verification. In *Proc. of ICSLP*, pages 581–584.

[Markel and Gray, 1976] Markel, J. and Gray, A. (1976). *Linear prediction of speech*. Springer-Verlag.

[Markel et al., 1977] Markel, J., Oshika, B., and Gray, A. (1977). Long-term feature averaging for speaker recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 25(4) :330–337.

[Mashao and Skosan, 2006] Mashao, D. and Skosan, M. (2006). Combining classifier decisions for robust speaker identification. *Pattern Recognition*, 39(1) :147–155.

[Matrouf et al., 2007] Matrouf, D., Scheffer, N., Fauve, B., and Bonastre, J.-F. (2007). A straightforward and efficient implementation of the factor analysis model for speaker verification. In *Proc. of INTERSPEECH*, pages 1242–1245.

[Matějka et al., 2011] Matějka, P., Glembek, O., Castaldo, F., Alam, M., Plchot, O., Kenny, P., Burget, L., and Černocký, J. (2011). Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification. In *Proc. of ICASSP*, pages 4828–4831.

[McCulloch and Pitts, 1943] McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(4) :115–133.

[Merlin et al., 1999] Merlin, T., Bonastre, J.-F., and Fredouille, C. (1999). Non directly acoustic process for costless speaker recognition and indexation. In *Proc. of International Workshop on Intelligent Communication Technologies and Applications, with emphasis on Mobile Communications*.

[Montacié and Chollet, 1987] Montacié, C. and Chollet, G. (1987). Systèmes de référence pour l'évaluation d'applications et la caractérisation de bases de données en reconnaissance automatique de la parole. In *Proc. of Journées d'Étude sur la Parole (JEP)*, pages 323–326.

[Montacié and Le Floch, 1993] Montacié, C. and Le Floch, J.-L. (1993). Discriminant AR-Vector Models for Free-Text Speaker Verification. In *Proc. of EUROSPEECH*, pages 161–164.

[Moreno and Ho, 2003] Moreno, P. and Ho, P. (2003). A new SVM approach to speaker identification and verification using probabilistic distance kernels. In *Proc. of EUROSPEECH*, volume 3, pages 2965–2968.

[Murty and Yegnanarayana, 2006] Murty, K. and Yegnanarayana, B. (2006). Combining evidence from residual phase and MFCC features for speaker recognition. *IEEE Signal Processing Letters*, 13(1) :52–55.

[Nadas, 1983] Nadas, A. (1983). A decision theorectic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 31(4) :814–817.

[Naini et al., 2010] Naini, A., Homayounpour, M., and Samani, A. (2010). A real-time trained system for robust speaker verification using relative space of anchor models. *Computer Speech and Language*, 24(4) :545–561.

[Navrátil et al., 2003] Navrátil, J., Jin, Q., Andrews, W., and Campbell, J. (2003). Phonetic speaker recognition using maximum-likelihood binary-decision tree models. In *Proc. of ICASSP*, volume 4, pages IV–796–IV–799.

[Nemer et al., 2001] Nemer, E., Goubran, R., and Mahmoud, S. (2001). Robust voice activity detection using higher-order statistics in the LPC residual domain. *IEEE Transactions on Speech and Audio Processing*, 9(3) :217–231.

[NIST, 2004] NIST (2004). *The NIST Year 2004 Speaker Recognition Evaluation Plan*. Online : http://www.itl.nist.gov/iad/mig/tests/sre/2004/SRE-04_evalplan-v1a.pdf.

[NIST, 2006] NIST (2006). *The NIST Year 2006 Speaker Recognition Evaluation Plan*. Online : http://www.itl.nist.gov/iad/mig/tests/spk/2006/sre-06_evalplan-v9.pdf.

[NIST, 2010] NIST (2010). *The NIST Year 2010 Speaker Recognition Evaluation Plan*. Online : http://www.itl.nist.gov/iad/mig//tests/sre/2010/NIST_SRE10_evalplan.r6.pdf.

[Nocedal and Wright, 1999] Nocedal, J. and Wright, S. (1999). *Numerical optimization*. Springer verlag.

[Oglesby and Mason, 1990] Oglesby, J. and Mason, J. (1990). Optimisation of neural models for speaker identification. In *Proc. of ICASSP*, pages 261–264.

[Oglesby and Mason, 1991] Oglesby, J. and Mason, J. (1991). Radial basis function networks for speaker recognition. In *Proc. of ICASSP*, pages 393–396.

[Omar and Pelecanos, 2010] Omar, M. and Pelecanos, J. (2010). Training Universal Background Models for Speaker Recognition. In *Proc. of Odyssey - The Speaker and Language Recognition Workshop*, pages 52–57.

[Paoloni et al., 1996] Paoloni, A., Ragazzini, S., and Ravaioli, G. (1996). Predictive neural networks in text independent speaker verification : an evaluation on the SIVA database. In *Proc. of ICSLP*, volume 4, pages 2423–2426.

[Park and Hazen, 2002] Park, A. and Hazen, T. (2002). ASR dependent techniques for speaker identification. In *Proc. of ICSLP*, pages 1337–1340.

[Pelecanos and Sridharan, 2001] Pelecanos, J. and Sridharan, S. (2001). Feature warping for robust speaker verification. In *Proc. of Odyssey - The Speaker and Language Recognition Workshop*, pages 213–218.

[Plumpe et al., 1999] Plumpe, M., Quatieri, T., and Reynolds, D. (1999). Modeling of the glottal flow derivative waveform with application to speaker identification. *IEEE Transactions on Speech and Audio Processing*, 7(5) :569–586.

[Prince and Elder, 2007] Prince, S. and Elder, J. (2007). Probabilistic linear discriminant analysis for inferences about identity. In *Proc. of International Conference on Computer Vision*, pages 1–8.

[Przybocki and Martin, 2004] Przybocki, M. and Martin, A. (2004). NIST Speaker Recognition Evaluation Chronicles. In *Proc. of Odyssey - The Speaker and Language Recognition Workshop*, pages 15–22.

[Rabiner, 1989] Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286.

[Rabiner and Juang, 1993] Rabiner, L. and Juang, B. (1993). *Fundamentals of speech recognition*. Prentice Hall.

[Rabiner and Sambur, 1975] Rabiner, L. and Sambur, M. (1975). An algorithm for determining the endpoints of isolated utterances. *The Bell System Technical Journal*, 54(2) :297–315.

[Rabiner and Schafer, 1978] Rabiner, L. and Schafer, R. (1978). *Digital processing of speech signals*. Prentice Hall.

[Ramachandran et al., 2002] Ramachandran, R., Farrell, K., Ramachandran, R., and Mammone, R. (2002). Speaker recognition–general classifier approaches and data fusion methods. *Pattern Recognition*, 35(12) :2801–2821.

[Ramírez et al., 2004] Ramírez, J., Segura, J., Benítez, C., De La Torre, A., and Rubio, A. (2004). Efficient voice activity detection algorithms using long-term speech information. *Speech Communication*, 42(3-4) :271–287.

[Reynolds, 1997] Reynolds, D. (1997). Comparison of background normalization methods for text-independent speaker verification. In *Proc. of EUROSPEECH*, pages 963–966.

[Reynolds, 2003] Reynolds, D. (2003). Channel robust speaker verification via feature mapping. In *Proc. of ICASSP*, volume 2, pages II–53–II–56.

[Reynolds et al., 2003] Reynolds, D., Andrews, W., Campbell, J., Navrátil, J., Peskin, B., Adami, A., Jin, Q., Klusáček, D., Abramson, J., Mihaescu, R., Godfrey, J., Jones, D., and

Xiang, B. (2003). The SuperSID project : Exploiting high-level information for high-accuracy speaker recognition. In *Proc. of ICASSP*, volume 4, pages IV–784–IV–787.

[Reynolds et al., 2000] Reynolds, D., Quatieri, T., and Dunn, R. (2000). Speaker verification using adapted Gaussian mixture models. *Digital signal processing*, 10(1-3) :19–41.

[Reynolds and Rose, 1995] Reynolds, D. and Rose, R. (1995). Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1) :72–83.

[Rodríguez-Fuentes et al., 2012] Rodríguez-Fuentes, L., Varona, A., Diez, M., Penagarikano, M., and Bordel, G. (2012). Evaluation of Spoken Language Recognition Technology Using Broadcast Speech : Performance and Challenges. In *Proc. of Odyssey - The Speaker and Language Recognition Workshop*.

[Rougui, 2008] Rougui, J. (2008). *Indexation de documents audio : Cas des grands volumes de données*. PhD thesis, Université de Nantes / Université Mohammed V-Agdal de Rabat.

[Scheffer and Bonastre, 2006] Scheffer, N. and Bonastre, J.-F. (2006). Fusing generative and discriminative UBM-based systems for speaker verification. In *Proc. of International workshop on MMUA (MultiModal User Authentification)*.

[Schmidt and Gish, 1996] Schmidt, M. and Gish, H. (1996). Speaker identification via support vector classifiers. In *Proc. of ICASSP*, volume 1, pages 105–108.

[Senoussaoui et al., 2010] Senoussaoui, M., Kenny, P., Dehak, N., and Dumouchel, P. (2010). An i-vector Extractor Suitable for Speaker Recognition with both Microphone and Telephone Speech. In *Proc. of Odyssey - The Speaker and Language Recognition Workshop*, pages 28–33.

[Senoussaoui et al., 2011] Senoussaoui, M., Kenny, P., Dumouchel, P., and Castaldo, F. (2011). Well-calibrated heavy tailed Bayesian speaker verification for microphone speech. In *Proc. of ICASSP*, pages 4824–4827.

[Sha, 2007] Sha, F. (2007). *Large margin training of acoustic models for speech recognition*. PhD thesis, University of Pennsylvania.

[Sha and Saul, 2006] Sha, F. and Saul, L. (2006). Large margin Gaussian mixture modeling for phonetic classification and recognition. In *Proc. of ICASSP*, volume 1, pages 265–268.

[Sha and Saul, 2007] Sha, F. and Saul, L. (2007). Large Margin Hidden Markov Models for Automatic Speech Recognition. In *Advances in Neural Information Processing Systems 19*, pages 1249–1256. MIT Press.

[Shen et al., 1998] Shen, J.-L., Hung, J.-W., and Lee, L.-S. (1998). Robust entropy-based endpoint detection for speech recognition in noisy environments. In *Proc. of ICSLP*.

[Shimodaira et al., 2001] Shimodaira, H., Noma, K., Nakai, M., and Sagayama, S. (2001). Support vector machine with dynamic time-alignment kernel for speech recognition. In *Proc. of EUROSPEECH*, pages 1841–1844.

[Shriberg et al., 2005] Shriberg, E., Ferrer, L., Kajarekar, S., Venkataraman, A., and Stolcke, A. (2005). Modeling prosodic feature sequences for speaker recognition. *Speech Communication*, 46(3-4) :455–472.

[Solewicz and Koppel, 2007] Solewicz, Y. and Koppel, M. (2007). Using post-classifiers to enhance fusion of low-and high-level speaker recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 15(7) :2063–2071.

[Solomonoff et al., 2005] Solomonoff, A., Campbell, W., and Boardman, I. (2005). Advances in channel compensation for SVM speaker recognition. In *Proc. of ICASSP*, volume 1, pages 629–632.

[Sönmez et al., 1997] Sönmez, M., Heck, L., Weintraub, M., and Shriberg, E. (1997). A lognormal tied mixture model of pitch for prosody based speaker recognition. In *Proc. of EUROSPEECH*, pages 1391–1394.

[Sönmez et al., 1998] Sönmez, M., Shriberg, E., Heck, L., and Weintraub, M. (1998). Modeling dynamic prosodic variation for speaker verification. In *Proc. of ICSLP*, volume 7, pages 3189–3192.

[Soong and Rosenberg, 1988] Soong, F. and Rosenberg, A. (1988). On the use of instantaneous and transitional spectral information in speaker recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(6) :871–879.

[Soong et al., 1985] Soong, F., Rosenberg, A., Rabiner, L., and Juang, B. (1985). A vector quantization approach to speaker recognition. In *Proc. of ICASSP*, volume 10, pages 387–390.

[Staroniewicz and Majewski, 2004] Staroniewicz, P. and Majewski, W. (2004). SVM based text-dependent speaker identification for large set of voices. In *Proc. of EUSIPCO*, pages 333–336.

[Stolcke et al., 2005] Stolcke, A., Ferrer, L., Kajarekar, S., Shriberg, E., and Venkataraman, A. (2005). MLLR transforms as features in speaker recognition. In *Proc. of INTERSPEECH*, pages 2425–2428.

[Stolcke et al., 2007] Stolcke, A., Kajarekar, S., Ferrer, L., and Shrinberg, E. (2007). Speaker recognition with session variability normalization based on MLLR adaptation transforms. *IEEE Transactions on Audio, Speech and Language Processing*, 15(7) :1987–1998.

[Sturim and Reynolds, 2005] Sturim, D. and Reynolds, D. (2005). Speaker Adaptive Cohort Selection for Tnorm in Text-Independent Speaker Verification. In *Proc. of ICASSP*, volume 1, pages 741–744.

[Svensén and Bishop, 2005] Svensén, M. and Bishop, C. (2005). Robust Bayesian mixture modelling. *Neurocomputing*, 64 :235–252.

[Thévenaz and Hügli, 1995] Thévenaz, P. and Hügli, H. (1995). Usefulness of the LPC-residue in text-independent speaker verification. *Speech Communication*, 17(1-2) :145–157.

[Tucker, 1992] Tucker, R. (1992). Voice activity detection using a periodicity measure. *IEE Proceedings I Communications, Speech and Vision*, 139(4) :377–380.

[Valtchev et al., 1997] Valtchev, V., Odell, J., Woodland, P., and Young, S. (1997). MMIE training of large vocabulary recognition systems. *Speech Communication*, 22(4) :303–314.

[Vandenberghe and Boyd, 1996] Vandenberghe, L. and Boyd, S. (1996). Semidefinite Programming. *SIAM Review*, 38(1) :49–95.

[Vapnik, 1998] Vapnik, V. (1998). *Statistical Learning Theory*. Wiley-Interscience, New York.

[Viikki and Laurila, 1998] Viikki, O. and Laurila, K. (1998). Cepstral domain segmental feature vector normalization for noise robust speech recognition. *Speech Communication*, 25(1-3) :133–147.

[Vogt et al., 2008] Vogt, R., Kajarekar, S., and Sridharan, S. (2008). Discriminant NAP for SVM speaker recognition. In *Proc. of Odyssey - The Speaker and Language Recognition Workshop*.

[Wan and Campbell, 2000] Wan, V. and Campbell, W. (2000). Support vector machines for speaker verification and identification. In *Proc. of Neural Networks for Signal Processing X*, volume 2, pages 775–784.

[Wan and Carmichael, 2005] Wan, V. and Carmichael, J. (2005). Polynomial dynamic time warping kernel support vector machines for dysarthric speech recognition with sparse training data. In *Proc. of INTERSPEECH*, pages 3321–3324.

[Wan and Renals, 2002] Wan, V. and Renals, S. (2002). Evaluation of kernel methods for speaker verification and identification. In *Proc. of ICASSP*, volume 1, pages I–669–I–672.

[Wan and Renals, 2005] Wan, V. and Renals, S. (2005). Speaker verification using sequence discriminant support vector machines. *IEEE Transactions on Speech and Audio Processing*, 13(2) :203–210.

[Xiang, 2003] Xiang, B. (2003). Text-independent speaker verification with dynamic trajectory model. *IEEE Signal Processing Letters*, 10(5) :141–143.

[Xiang et al., 2002] Xiang, B., Chaudhari, U., Navrátil, J., Ramaswamy, G., and Gopinath, R. (2002). Short-time gaussianization for robust speaker verification. In *Proc. of ICASSP*, volume 1, pages I–681–I–684.

[Yegnanarayana and Kishore, 2002] Yegnanarayana, B. and Kishore, S. (2002). AANN : an alternative to GMM for pattern recognition. *Neural Networks*, 15(3) :459–469.

[Ying et al., 2011] Ying, D., Yan, Y., Dang, J., and Soong, F. (2011). Voice Activity Detection Based on an Unsupervised Learning Framework. *IEEE Transactions on Audio, Speech and Language Processing*, 19(8) :2624–2633.

[Zheng et al., 2007] Zheng, N., Lee, T., and Ching, P. (2007). Integration of complementary acoustic features for speaker recognition. *IEEE Signal Processing Letters*, 14(3) :181–184.

[Zhu et al., 2008] Zhu, D., Ma, B., and Li, H. (2008). Using MAP estimation of feature transformation for speaker recognition. In *Proc. of INTERSPEECH*, pages 849–852.

[Zhu et al., 2009] Zhu, D., Ma, B., and Li, H. (2009). Joint map adaptation of feature transformation and Gaussian Mixture Model for speaker recognition. In *Proc. of ICASSP*, pages 4045–4048.

Résumé

Depuis plusieurs dizaines d'années, la reconnaissance automatique du locuteur (RAL) fait l'objet de travaux de recherche entrepris par de nombreuses équipes dans le monde. La majorité des systèmes actuels sont basés sur l'utilisation des Modèles de Mélange de lois Gaussiennes (GMM) et/ou des modèles discriminants SVM, i.e., les machines à vecteurs de support. Nos travaux ont pour objectif général la proposition d'utiliser de nouveaux modèles GMM à grande marge pour la RAL qui soient une alternative aux modèles GMM génératifs classiques et à l'approche discriminante état de l'art GMM-SVM. Nous appelons ces modèles LM-dGMM pour Large Margin diagonal GMM. Nos modèles reposent sur une récente technique discriminante pour la séparation multi-classes, qui a été appliquée en reconnaissance de la parole. Exploitant les propriétés des systèmes GMM utilisés en RAL, nous présentons dans cette thèse des variantes d'algorithmes d'apprentissage discriminant des GMM minimisant une fonction de perte à grande marge. Des tests effectués sur les tâches de reconnaissance du locuteur de la campagne d'évaluation NIST-SRE 2006 démontrent l'intérêt de ces modèles en reconnaissance.

Mots-clés: Apprentissage discriminant, Modèles de Mélange de lois Gaussiennes, maximisation de la marge, reconnaissance du locuteur, compensation de la variabilité inter-sessions.

Extended Abstract

1 Introduction

Most of state-of-the-art speaker recognition systems rely on the generative training of Gaussian Mixture Models (GMM) using maximum likelihood estimation and maximum a posteriori estimation (MAP) [Reynolds et al., 2000]. A speaker independent world model or Universal Background Model (UBM) is first trained with the Expectation-Maximization algorithm from hundreds of hours of speech data. The parameters of that model are then MAP adapted to the feature distribution of a target speaker. In speaker recognition applications, the mismatch between the training and testing conditions can decrease considerably the performances. The session variability remains the most challenging problem to solve. The Factor Analysis techniques [Kenny et al., 2005a], [Kenny et al., 2007b], e.g., Symmetrical Factor Analysis (SFA) [Matrouf et al., 2007], [Fauve et al., 2007], were proposed to address that problem in GMM based systems, by compensating for speaker and channel variability in the GMM supervector space.

The generative training of the GMM does not however directly optimize the classification performance. It was therefore of interest to develop alternative discriminative approaches that address directly the classification problem [Keshet and Bengio, 2009], [Louradour et al., 2007], and they lead generally to better performances than generative methods. For instance, Support Vector Machines (SVM) combined with GMM supervectors are among the state-of-the-art approaches in speaker verification [Campbell et al., 2006b] , [Campbell et al., 2006c]. The Nuisance Attribute Projection (NAP) [Solomonoff et al., 2005] compensation technique is designed for the SVM based systems. NAP is a pre-processing method that aims to remove the directions of undesired sessions variability, before the SVM training.

Recently a new discriminative approach for multiway classification has been proposed, the Large Margin Gaussian mixture models (LM-GMM) [Sha and Saul, 2006]. As in SVM, the parameters of LM-GMM are trained by solving a convex optimization problem. However they differ from SVM by using ellipsoids to model the classes directly in the input space, instead of half-spaces in an extended high-dimensional space, thus no kernel trick/matrix is required. While LM-GMM have been used in speech recognition, they have not been used in speaker recognition (to the best of our knowledge).

In this thesis, we propose simplified, fast and more efficient versions of LM-GMM which exploit the properties and characteristics of speaker recognition applications and systems, the LM-dGMM models.

2 Overview on Large Margin GMM training

This section gives an overview on Large Margin training by first recalling the original Large Margin training algorithm developed in [Sha and Saul, 2006], [Sha, 2007], and then recalling its simplified version that we propose. Some experimental results are finally reported in the end of the section.

2.1 Large Margin GMM

In Large Margin GMM [Sha and Saul, 2006], [Sha, 2007], each class is modeled by a mixture of ellipsoids in the feature space. For each class c, the m^{th} ellipsoid is parameterized by a centroid vector μ_{cm}, a positive semidefinite matrix Ψ_{cm} defining its orientation and a nonnegative scalar offset θ_{cm}, all then collected into a single enlarged matrix Φ_{cm} :

$$\Phi_{cm} = \begin{pmatrix} \Psi_{cm} & -\Psi_{cm}\mu_{cm} \\ -\mu_{cm}^T\Psi_{cm} & \mu_{cm}^T\Psi_{cm}\mu_{cm} + \theta_{cm} \end{pmatrix}. \tag{1}$$

Considering a set of labeled training examples $\{(x_{n,t}, y_n)\}_{n=1}^N$ where $x_{n,t} \in \mathcal{R}^D$ is the T_n feature vectors of the n^{th} segment (i.e. n^{th} speaker training data) and $y_n \in \{1, 2, ..., C\}$ is the associated class (C is the total number of classes), the goal of LM-GMM training is to find matrices Φ_{cm} such that "all" examples are correctly classified by at least one margin unit. To do so, a GMM is first fit to each class using maximum likelihood estimation. Second, an index $m_{n,t}$ is associated to each example $x_{n,t}$; this label corresponds to the GMM mixture component with the highest posterior probability and is called proxy label.

Given the joint labels (y_n,$m_{n,t}$) for each learning example, the LM-GMM criterion is given as :

$$\forall c \neq y_n, \qquad -\log \sum_{m=1}^{M} e^{-z_{n,t}^T \Phi_{cm} z_{n,t}} \; - \; z_{n,t}^T \Phi_{y_n m_{n,t}} z_{n,t} \; \geq \; 1, \tag{2}$$

where $z_{n,t} = \begin{bmatrix} x_{n,t} \\ 1 \end{bmatrix}$. Because of the softmax inequality : $\min_m a_m \geq -\log \sum_m e^{-a_m}$, equation Eq. (2) states that for each competing class $c \neq y_n$ the match (in term of Mahalanobis distance) of any centroid in class c is worse than the target centroid by a margin of at least one unit. In a segmental training scheme, the loss function is thus given by :

$$\begin{aligned} \mathbf{L} \;\; = \;\; & \sum_{n=1}^{N} \sum_{c \neq y_n} max\left(0 \;,\; 1 + \frac{1}{T_n} \sum_{t=1}^{T_n} \left(z_{n,t}^T \Phi_{y_n m_{n,t}} z_{n,t} + \log \sum_{m=1}^{M} e^{-z_{n,t}^T \Phi_{cm} z_{n,t}} \right)\right) \\ & + \alpha \sum_{c=1}^{C} \sum_{m=1}^{M} trace(\Psi_{cm}), \end{aligned} \tag{3}$$

where the second term penalizes large trace Mahalanobis metrics. The hyperparameter α is set by cross-validation on development data. Finally, the decision rule used for classification is :

$$y \;\; = \;\; \underset{c}{argmin} \left\{ \sum_{t=1}^{T} -\log \sum_{m=1}^{M} e^{-z_t^T \Phi_{cm} z_t} \right\}. \tag{4}$$

As opposed to other discriminative training algorithms such as conditional log-likelihood learning, this loss function has the great advantage to be convex. For a complete description of the LM-GMM and their extension to LM-HMM, we refer to [Sha and Saul, 2006], [Sha, 2007], [Sha and Saul, 2007].

2.2 Large Margin GMM with diagonal covariances (LM-dGMM)

In speaker recognition, most of state-of-the-art systems use diagonal-covariances GMM. In these GMM based speaker recognition systems, a world model or Universal Background Model (UBM) representing speaker-independent distribution of the feature vectors is first trained with the EM algorithm [Bishop, 2006] from tens or hundreds of hours of speech data gathered from a large number of speakers. When enrolling a new speaker to the system, the parameters of the UBM are adapted to the feature distribution of the new speaker using the maximum a posteriori (MAP) algorithm [Reynolds et al., 2000]. The adapted model is then used as the model of that target speaker. Traditionally, in the GMM-UBM approach, only the mean vectors are adapted. The (diagonal) covariances and the weights remain unchanged.

Following the same philosophy of traditional GMM, we propose to neglect the orientation of the Ψ_{cm} matrices in training. That is, in our Large Margin diagonal GMM (LM-dGMM), each class (speaker) c is initially modeled by a GMM with M diagonal mixtures trained by MAP adaptation of the UBM. For each class c, the m^{th} Gaussian is parameterized by a mean vector μ_{cm}, a diagonal covariance matrix $\Sigma_m = diag(\sigma^2_{m1}, ..., \sigma^2_{mD})$, and the scalar factor $\theta_m = \frac{1}{2}\big(D\log(2\pi) + \log|\Sigma_m|\big) - \log(w_m)$ which corresponds to the weight of the Gaussian.

With this relaxation on the matrices Ψ_{cm}, for each example $x_{n,t}$, the goal of the training algorithm is now to force the log-likelihood of its proxy label Gaussian $m_{n,t}$ to be at least one unit greater than the log-likelihood of each Gaussian component of all competing classes. That is, given the set of training examples $\{(x_{n,t}, y_n, m_{n,t})\}_{n=1}^N$, we seek mean vectors μ_{cm} which satisfy the LM-dGMM criterion :

$$\forall c \neq y_n, \forall m, \qquad d(x_{n,t}, \mu_{cm}) + \theta_m \;\geq\; 1 \;+\; d(x_{n,t}, \mu_{y_n m_{n,t}}) + \theta_{m_{n,t}}, \tag{5}$$

where

$$d(x_{n,t}, \mu_{cm}) \;=\; \sum_{i=1}^{D} \frac{(x_{n,ti} - \mu_{cmi})^2}{2\sigma^2_{mi}}. \tag{6}$$

Afterward, these M constraints are fold into a single one using the softmax inequality. In a segmental training scheme, the LM-dGMM criterion becomes thus :

$$\forall c \neq y_n, \quad \frac{1}{T_n}\sum_{t=1}^{T_n} -\log \sum_{m=1}^{M} \exp\Big(-d(x_{n,t}, \mu_{cm}) - \theta_m\Big) \;\geq\; 1 + \frac{1}{T_n}\sum_{t=1}^{T_n} d(x_{n,t}, \mu_{y_n m_{n,t}}) + \theta_{m_{n,t}}. \tag{7}$$

The loss function to minimize for LM-dGMM is then given by :

$$\begin{aligned} \mathbf{L} \;=\; \sum_{n=1}^{N}\sum_{c\neq y_n} \max\Bigg(0 \;,\; 1 + \frac{1}{T_n}\sum_{t=1}^{T_n}\Bigg(& d(x_{n,t}, \mu_{y_n m_{n,t}}) + \theta_{m_{n,t}} \\ & + \log \sum_{m=1}^{M} \exp\Big(-d(x_{n,t}, \mu_{cm}) - \theta_m\Big)\Bigg)\Bigg). \end{aligned} \tag{8}$$

2.3 Experimental results

We perform experiments on the NIST-SRE 2006 [NIST, 2006] speaker identification task and compare the performances of the baseline GMM, the original LM-GMM and our modified version. Performances are measured in term of the speaker identification rate.

The feature extraction is carried out by the filter-bank based cepstral analysis tool Spro [Gravier, 2003]. Bandwidth is limited to the 300-3400Hz range. 24 filter bank coefficients are first computed over 20ms Hamming windowed frames at a 10ms frame rate and transformed into Linear Frequency Cepstral Coefficients (LFCC) [Davis and Mermelstein, 1980]. Consequently, the feature vector is composed of 50 coefficients including 19 LFCC, their first derivatives, their 11 first second derivatives and the delta-energy. The LFCCs are preprocessed by cepstral mean subtraction and variance normalization [Viikki and Laurila, 1998]. We applied an energy-based voice activity detection to remove silence frames, hence keeping only the most informative frames. Finally, the remaining parameter vectors are normalized to fit a zero mean and unit variance distribution.

We use the state-of-the-art open source software ALIZE/Spkdet [Fauve et al., 2007], [Bonastre et al., 2008] for GMM modeling. The code for the original LM-GMM modeling was kindly given to us by Fei SHA.

A male-dependent UBM is trained using all the telephone data from the NIST-SRE 2004. Then we train a MAP adapted GMM for each speaker of **50 male target speakers** belonged to the **NIST-SRE 2006 primary task (1conv4w-1conv4w)**. A corresponding list of 11600 trials involving **232 test segment** is used for testing. Session variability modeling and score normalization techniques are not used in these experiments. The so MAP adapted GMM are used to define the traditional GMM system. They are used as initialization for the two Large Margin systems (original and simplified).

We underline the fact that we have used no development data for the Large Margin modeling[27], we thus give directly the performances of the best models on the test set.

Table Tab. 1 shows the speaker identification accuracy scores of the three systems. For each one, we study two configurations with 16 and 32 Gaussian components.

TAB. 1: Speaker identification rates with GMM and the original and simplified Large Margin Training algorithms.

System	16 Gaussians	32 Gaussians
GMM	61.6%	68.1%
LM-GMM	60.8%	72.0%
LM-dGMM	67.7%	71.6%

The results of Table Tab. 1 show that our simplified LM-dGMM algorithm yields significantly better scores than the GMM system and the improvement is higher when less Gaussians are used. This is consistent with the well known fact that discriminative models perform better when the "true" distribution is not correctly captured by the generative model.

The results of Table Tab. 1 show also that our algorithm performs as well as the original one with 32 Gaussians, and significantly outperforms it with 16 Gaussians. These results suggest that the strategy of discriminating only the mean vectors is worth, and that the use the Φ_{cm} matrices could even degrade the performances while they are computationally (relatively) demanding.

[27] There is only one speaker utterance to train models in the 1conv4w-1conv4w condition.

Indeed, computing gradients of the loss function with respect to the enlarged matrices Φ_{cm} is much more time and memory consuming than with respect to only the mean vectors μ_{cm}. Moreover, our algorithm requires less iterations than the original one to converge.

Another potentially major advantage of our simplified Large Margin models is that we still have normalized GMM after the discriminative training, which is not the case with the original algorithm. This can be extremely useful for post-processing.

3 LM-dGMM training with k-best Gaussians

3.1 Description of the training algorithm

Despite the fact that our LM-dGMM is computationally much faster than the original LM-GMM of [Sha and Saul, 2006], [Sha, 2007], we still encountered efficiency problems when dealing with high number of Gaussian mixtures. Indeed, even for the easy previous 50 speakers identification task, we could not run the training in a relatively short time with our current implementation. This would imply that large scale applications such as NIST-SRE, where hundreds or thousands of target speakers are available, would be infeasible in reasonable time (for instance, 5460 target speakers are included in the NIST-SRE 2010 core condition, with 610748 trials to process involving 13325 test segments [NIST, 2010]).

In order to develop a fast training algorithm which could be used in large scale applications, we propose to drastically reduce the number of constraints to satisfy in equation Eq. (7). By doing so, we would drastically reduce the computational complexity of the loss function and its gradient, which are the quantities responsible for most of the computational time. To achieve this goal we propose to use another property of state-of-the-art GMM systems, that is, decision is not made upon all mixture components but only using the k-best scoring Gaussians. In other words, for each x_n and each class c, instead of summing over the M mixtures in the left side of equation Eq. (7), we would sum only over the k Gaussians with the highest posterior probabilities selected using the GMM of class c.

In order to further improve efficiency and reduce memory requirement, we exploit the property reported in [Reynolds et al., 2000] about correspondence between MAP adapted GMM mixtures and UBM mixtures. We use the UBM to select one unique set $S_{n,t}$ of k-best Gaussian components per frame $x_{n,t}$, instead of $(C-1)$ sets. This leads to a $(C-1)$ times faster and less memory consuming selection. Thus, the higher the number of target speakers is, the greater computation and memory saving is. More precisely, we now seek mean vectors μ_{cm} that satisfy the Large Margin constraints in equation Eq. (9) :

$$\forall c \neq y_n,$$
$$\frac{1}{T_n}\sum_{t=1}^{T_n}\left(-\log\sum_{m\in S_{n,t}}\exp(-d(x_{n,t},\mu_{cm})-\theta_m)\right) \;\geq\; 1 \;+\; \frac{1}{T_n}\sum_{t=1}^{T_n} d(x_{n,t},\mu_{y_n m_{n,t}})+\theta_{m_{n,t}}. \tag{9}$$

The loss function becomes :

$$\begin{aligned}\mathbf{L} \quad = \quad & \sum_{n=1}^{N} \sum_{c \neq y_n} max\Bigg(0 \ , \ 1 + \frac{1}{T_n} \sum_{t=1}^{T_n} \Bigg(d(x_{n,t}, \mu_{y_n m_{n,t}}) \\ & + \theta_{m_{n,t}} + \log \sum_{m \in S_{n,t}} \exp(-d(x_{n,t}, \mu_{cm}) - \theta_m) \Bigg) \Bigg).\end{aligned} \tag{10}$$

This loss function remains convex and can still be solved using dynamic programming.

3.2 Handling of outliers

We adopt the strategy of [Sha and Saul, 2006] to detect outliers and reduce their negative effect on learning. Outliers are detected using the initial GMM models. We compute the accumulated hinge loss incurred by violations of the Large Margin constraints in equation Eq. (9) :

$$\begin{aligned}h_n \quad = \quad & \sum_{c \neq y_n} max\Bigg(0 \ , \ 1 + \frac{1}{T_n} \sum_{t=1}^{T_n} \Bigg(d(x_{n,t}, \mu_{y_n m_{n,t}}) \\ & + \theta_{m_{n,t}} + log \sum_{m \in S_{n,t}} exp(-d(x_{n,t}, \mu_{cm}) - \theta_m) \Bigg) \Bigg).\end{aligned} \tag{11}$$

$h_n \geq 0$ measures the decrease in the loss function when an initially misclassified segment is corrected during the course of learning. We associate outliers with large values of h_n, i.e., with losses greater than 1. We then re-weight[28] the hinge loss terms in equation Eq. (10) by using segment weights $sw_n \ = \ min\Big(1, \frac{1}{h_n}\Big)$:

$$\mathbf{L} \ = \ \sum_{n=1}^{N} sw_n h_n. \tag{12}$$

This re-weighting equalizes the losses incurred by all initially misclassified examples, thus reducing the malicious effect of outliers. We solve this unconstrained non-linear optimization problem using the second order optimizer LBFGS [Nocedal and Wright, 1999].

In summary, our new and fast training algorithm of LM-dGMM is the following :

- For each class (speaker), initialize with the GMM trained by MAP of the UBM.
- Select Proxy labels using these GMM.
- Select the set of k-best UBM Gaussian components for each training frame.
- Compute the segment weights.
- Using the LBFGS algorithm, solve the unconstrained non-linear optimization problem according to equation Eq. (12)

$$min \quad \mathbf{L}. \tag{13}$$

[28] Note that by setting the segment weights to one, i.e., no handling of outliers is done, the experiments show that the performances degrade.

3.3 Evaluation phase

During test, we use the same principle as in the training to achieve fast scoring. Given a test segment of T frames, for each test frame x_t we use the UBM to select the set E_t of k-best scoring proxy labels.

In an identification task, we compute the LM-dGMM likelihoods using only these k labels. The decision rule is thus given as :

$$y = \underset{c}{argmin} \left\{ \sum_{t=1}^{T} -\log \sum_{m \in E_t} \exp(-d(x_t, \mu_{cm}) - \theta_m) \right\}. \tag{14}$$

In a verification task, we compute a match score depending on both the target model $\{\mu_{cm}, \Sigma_m, \theta_m\}$ and the UBM $\{\mu_{Um}, \Sigma_m, \theta_m\}$ for the test hypothesis (trial). The average log likelihood ratio is calculated using only the k labels :

$$\begin{aligned} LLR_{avg} &= \frac{1}{T} \sum_{t=1}^{T} \left(log \sum_{m \in E_t} \exp(-d(x_t, \mu_{cm}) - \theta_m) \right. \\ &\quad \left. - log \sum_{m \in E_t} \exp(-d(x_t, \mu_{Um}) - \theta_m) \right). \end{aligned} \tag{15}$$

This quantity provides a score for the test segment to be uttered by the target model/speaker c.

3.4 Experimental results

We perform experiments on NIST-SRE 2006 speaker recognition tasks and compare the performances of the baseline GMM, the LM-dGMM (with k-best Gaussians) and the SVM systems, with and without using channel compensation techniques. The comparisons are made on the **male part of the NIST-SRE 2006 core condition (1conv4w-1conv4w)**. In the verification task, performances are measured in terms of equal error rate (EER) and minimum of detection cost function (minDCF) which is calculated following NIST criteria [Przybocki and Martin, 2004].

For front-end processing, we follow the same procedure as in the section 2.3. We use ALIZE/Spkdet for GMM, SFA [Matrouf et al., 2007], GSL [Campbell et al., 2006b] and GSL-NAP [Campbell et al., 2006c] modeling. We train a MAP adapted GMM for the **349 target speakers** belonging to the primary task using the male-dependent UBM trained on NIST-SRE 2004 data. The **identification is made on a list of trials involving 1546 test segments**, whereas the **verification task uses a list of 22123 trials** (involving 1601 test segments) for test. Score normalization techniques are not used in our experiments. The so MAP adapted GMM define the baseline GMM system, and are used as initialization for the LM-dGMM one. The GSL system uses a list of 200 impostor speakers from the NIST-SRE 2004, on the SVM training. The LM-dGMM-SFA system is initialized by model domain compensated GMM, which are then discriminated using feature domain compensated data. The session variability matrix $\mathbf{U}$ of SFA and the channel matrix $\mathbf{S}$ of NAP, both of rank $R = 40$, are estimated on NIST-SRE 2004 data using 2934 utterances of 124 different male speakers.

Table Tab. 2 presents the speaker identification accuracy scores of the various systems. Table Tab. 3 presents the speaker verification scores (EER and minDCF). We show performances using

GMMs with 256 and 512 Gaussian components ($M = 256, 512$). All the scores are obtained with the 10 best proxy labels selected using the UBM, $k = 10$. Experiments done with different values of k show that selecting k > 10 does not however improve the performances. The selection is thus restricted to the 10 best Gaussians.

Tab. 2: Speaker identification rates with GMM, Large Margin diagonal GMM and GSL models, with and without channel compensation

System	Speaker identification rate	
	256 Gaussians	512 Gaussians
GMM	75.87%	77.88%
LM-dGMM	77.62%	78.40%
GSL	81.50%	82.21%
GSL-NAP	87.26%	87.77%
GMM-SFA	89.26%	90.75%
LM-dGMM-SFA	89.65%	**91.27%**

Tab. 3: EERs(%) and minDCFs(x100) of GMM, Large Margin diagonal GMM and GSL systems with and without channel compensation

System	256 Gaussians		512 Gaussians	
	EER	minDCF(x100)	EER	minDCF(x100)
GMM	9.43%	4.26	9.74%	4.18
LM-dGMM	8.97%	3.97	9.66%	4.12
GSL	7.39%	3.41	7.23%	3.44
GSL-NAP	6.40%	2.72	5.90%	2.73
GMM-SFA	6.15%	2.41	5.53%	2.18
LM-dGMM-SFA	5.58%	2.29	**5.02%**	**2.18**

The results of Table Tab. 2 and Table Tab. 3 show that, without SFA channel compensation, the LM-dGMM system outperforms the classical generative GMM one, however it does yield worse performances than the discriminative approach GSL. Nonetheless, when applying channel compensation techniques, compensated models outperform the non-compensated ones as expected, but the LM-dGMM-SFA system significantly outperforms the GSL-NAP and GMM-SFA ones in the two tasks. Our best system achieves 91.27% speaker identification rate, while the best GSL-NAP achieves 87.77%. This leads to a 3.5% improvement. In verification, the LM-dGMM-SFA and GSL-NAP achieve respectively 5.02% and 5.90% equal error rates, and $2.18 * 10^{-2}$ and $2.73 * 10^{-2}$ minDCF values. This shows that LM-dGMM-SFA yields relative reductions of EER and minDCF of about 14.92% and 20.15% over the GSL-NAP system. Moreover, The performances of the GMM-SFA system show that LM-dGMM-SFA yields relative reductions of speaker identification rate and EER of about 0.57% and 9.22% over this system.

4 Improving large margin modeling

This section explores some lines of work to improve our Large Margin modeling, by studying the impact of the minimal margin to satisfy on the performances and the combination of Large Margin and SVM modelings, and by developing a new strategy to detect outliers and reduce their negative effect in training.

4.1 Feature vectors weighting

The previous strategy for detecting outliers and reducing their negative effect on learning consisted on re-weighting the hinge loss terms in equation Eq. (10) by using segment weights. We propose in this section a novel and better strategy that outperforms the previous one.

We keep the global Large Margin constraints segmental, but we will apply now a *frame* (feature vectors) weighting scheme. For each feature vector $x_{n,t}$, we calculate $(C-1)$ weights $s^{\mathbf{c}}_{n,t}$ relative to each class $c \neq y_n$. For each $x_{n,t}$ and each competing class c, we compute the loss incurred by violations of the Large Margin constraints :

$$h^{\mathbf{c}}_{n,t} = \frac{1 + d(x_{n,t}, \mu_{y_n m_{n,t}}) + \theta_{m_{n,t}} + \log \sum_{m \in S_{n,t}} \exp\Big(- d(x_{n,t}, \mu_{cm}) - \theta_m \Big)}{T_n}. \tag{16}$$

$h^{\mathbf{c}}_{n,t}$ measures the decrease in the loss function when an initially misclassified feature vector is corrected during the course of learning. We associate outliers with values of $h^{\mathbf{c}}_{n,t} > 1$, and in this case we multiply this term by the frame weight $s^{\mathbf{c}}_{n,t} = \frac{1}{h^{\mathbf{c}}_{n,t}}$. The new loss function becomes thus :

$$\mathbf{L} = \sum_{n=1}^{N} \sum_{c \neq y_n} \max\left(0 \ , \ \sum_{t=1}^{T_n} s^{\mathbf{c}}_{n,t} h^{\mathbf{c}}_{n,t}\right). \tag{17}$$

This unconstrained non-linear optimization problem could be solved using the second order optimizer LBFGS.

We evaluate this feature vectors weighting strategy in the verification task. Table Tab. 4 gives the EER scores of LM-dGMM and LM-dGMM-SFA systems using the two weighting strategies, for models with 512 Gaussian components and $k = 10$.

Tab. 4: Segmental weighting strategy vs frame weighting strategy

System		EER
Segmental weighting	LM-dGMM	9.66%
	LM-dGMM-SFA	5.02%
Frame weighting	LM-dGMM	**9.47%**
	LM-dGMM-SFA	**4.89%**

One can see that the frame weighting approach further improves the LM-dGMM (+SFA) performance.

4.2 Complementarity between the Large Margin and SVM modelings

We realize a speaker recognition system that combines the Large Margin and SVM modelings. We form supervectors by stacking the mean vectors of the compensated system LM-dGMM-SFA. As in the traditional GSL system [Campbell et al., 2006b], we use the UBM weight and variance parameters to normalize the supervectors before feeding them into a linear kernel SVM training. This system is referred to as (LM-dGMM-SFA) – SVM.

In verification, the (LM-dGMM-SFA) – SVM models with 512 Gaussian components ($M = 512$) and $k = 10$ achieve **4.39%** equal error rate, improving thus the performances of both single systems (modelings). This result show that Large Margin and SVM modelings are complementary. Moreover, the combination accelerates the scoring procedure during the evaluation phase. This study should be further investigated in the futur.

4.3 Margin selection

We train Large Margin models subject to non-unit margin constraints. Table Tab. 5 shows the EER scores of LM-dGMM-SFA models with different minimal margin values to satisfy.

Tab. 5: EER(%) performances for LM-dGMM systems with different margins.

Margin	1	1.125	1.5	2	3	5	9
EER	5.02	4.85	5.53	5.53	5.52	5.53	5.53

Like in SVM, one can see that an improved margin selection can improve the LM-dGMM (+SFA) performance (an ERR of 4.82% here instead of 5.02% with a unit margin). One of our futur works will consist in improving the margin selection.

5 Conclusion

We proposed new algorithms for discriminative learning of diagonal GMM under a Large Margin criterion. Our algorithm is highly efficient which makes it well suited to process large scale databases such as in NIST-SRE campaigns. We carried out experiments on full speaker recognition tasks under the NIST-SRE 2006 core condition. Combined with the SFA channel compensation technique, the resulting algorithm significantly outperforms the state-of-the-art speaker recognition discriminative approach GSL-NAP. We emphasize that while we have applied our algorithm to speaker recognition, it can be actually applied in many other classification applications such as language identification.

Keywords: Discriminative learning, Gaussian mixture models, large margin training, speaker recognition, session variability modeling.

Publications Personnelles

- Jourani, R., Daoudi, K., André-Obrecht, R., and Aboutajdine, D. (Online First). Discriminative speaker recognition using Large Margin GMM. *Journal of Neural Computing & Applications*, doi :10.1007/s00521-012-1079-y.
- Jourani, R., Daoudi, K., André-Obrecht, R., and Aboutajdine, D.. Fast training of large margin diagonal gaussian mixture models for speaker identification. Submitted as Book Chapter.
- Daoudi, K., Jourani, R., André-Obrecht, R., and Aboutajdine, D. (2011). Speaker Identification Using Discriminative Learning of Large Margin GMM. In *Neural Information Processing*, volume 7063 of *Lecture Notes in Computer Science*, pages 300–307. Springer Berlin / Heidelberg.
- Jourani, R., Daoudi, K., André-Obrecht, R., and Aboutajdine, D. (2011). Apprentissage discriminant des GMM à grande marge pour la vérification automatique du locuteur. Dans *les Actes du XXIIIe Colloque GRETSI - Traitement du Signal et des Images*.
- Jourani, R., Daoudi, K., André-Obrecht, R., and Aboutajdine, D. (2011). Speaker verification using Large Margin GMM discriminative training. In *Proc. of ICMCS*, pages 1–5.
- Jourani, R., Daoudi, K., André-Obrecht, R., and Aboutajdine, D. (2011). Fast training of Large Margin diagonal Gaussian mixture models for speaker identification. In *Proc. of SpeD*, pages 1–4.
- Jourani, R., Daoudi, K., André-Obrecht, R., and Aboutajdine, D. (2010). Large Margin Gaussian mixture models for speaker identification. In *Proc. of INTERSPEECH*, pages 1441–1444.
- Jourani, R., Langlois, D., Smaïli, K., Daoudi, K., and Aboutajdine, D. (2010). Cleaning Statistical Language Models. In *Proc. of SIIE*.
- Jourani, R., Smaïli, K., Aboutajdine, D., and Daoudi, K. (2008). Building Arabic textual corpus from the Web. In *Proc. of SIIE*.

Par (nom, prénom) : JOURANI Reda
TITRE . : Reconnaissance automatique du locuteur par des GMM ú grande marge
Directeurs de Recherche : ANDRŁ-OBRECHT Rýgine & ABOUTAJDINE Driss
Laboratoire d'accueil : Institut de Recherche en Informatique de Toulouse

RESUME : Depuis plusieurs dizaines d'années, la reconnaissance automatique du locuteur (RAL) fait l'objet de travaux de recherche entrepris par de nombreuses équipes dans le monde. La majorité des systèmes actuels sont basés sur l'utilisation des Modèles de Mélange de lois Gaussiennes (GMM) et/ou des modèles discriminants SVM, i.e., les machines à vecteurs de support. Nos travaux ont pour objectif général la proposition d'utiliser de nouveaux modèles GMM à grande marge pour la RAL qui soient une alternative aux modèles GMM génératifs classiques et à l'approche discriminante état de l'art GMM-SVM. Nous appelons ces modèles LM-dGMM pour Large Margin diagonal GMM. Nos modèles reposent sur une récente technique discriminante pour la séparation multi-classes, qui a été appliquée en reconnaissance de la parole. Exploitant les propriétés des systèmes GMM utilisés en RAL, nous présentons dans cette thèse des variantes d'algorithmes d'apprentissage discriminant des GMM minimisant une fonction de perte à grande marge. Des tests effectués sur les tâches de reconnaissance du locuteur de la campagne d'évaluation NIST-SRE 2006 démontrent l'intérêt de ces modèles en reconnaissance.

MOTS-CLES : Apprentissage discriminant, ModÃles de Mýlange de lois Gaussiennes, maximisation de la marge, reconnaissance du locuteur, compensation de la variabilitý inter-sessions.

ABSTRACT : Most of state-of-the-art speaker recognition systems are based on Gaussian Mixture Models (GMM), trained using maximum likelihood estimation and maximum a posteriori (MAP) estimation. The generative training of the GMM does not however directly optimize the classification performance. For this reason, discriminative models, e.g., Support Vector Machines (SVM), have been an interesting alternative since they address directly the classification problem, and they lead to good performances. Recently a new discriminative approach for multiway classification has been proposed, the Large Margin Gaussian mixture models (LM-GMM). As in SVM, the parameters of LM-GMM are trained by solving a convex optimization problem. However they differ from SVM by using ellipsoids to model the classes directly in the input space, instead of half-spaces in an extended high-dimensional space. While LM-GMM have been used in speech recognition, they have not been used in speaker recognition (to the best of our knowledge). In this thesis, we propose simplified, fast and more efficient versions of LM-GMM which exploit the properties and characteristics of speaker recognition applications and systems, the LM-dGMM models. In our LM-dGMM modeling, each class is initially modeled by a GMM trained by MAP adaptation of a Universal Background Model (UBM) or directly initialized by the UBM. The models mean vectors are then re-estimated under some Large Margin constraints. We carried out experiments on full speaker recognition tasks under the NIST-SRE 2006 core condition. The experimental results are very satisfactory and show that our Large Margin modeling approach is very promising.

KEYWORDS : Discriminative learning, Gaussian mixture models, large margin training, speaker recognition, session variability modeling.