Reconnaissance du locuteur en milieux difficiles

Waad Ben Kheder

ACADÉMIE D'AIX-MARSEILLE
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE

THÈSE

présentée à l'Université d'Avignon et des Pays de Vaucluse
pour obtenir le diplôme de DOCTORAT

SPÉCIALITÉ : Informatique

École Doctorale 380 « Sciences et Agronomie »
Laboratoire d'Informatique (EA 4128)

Reconnaissance du locuteur en milieux difficiles

par
Waad BEN KHEDER

Soutenue publiquement le 18 Juillet 2017 devant un jury composé de :

M.	Denis Jouvet	Professeur, LORIA - INRIA Nancy	Rapporteur
M.	Claude Barras	MCF-HDR, Université Paris Sud, LIMSI	Rapporteur
M.	Thomas Pellegrini	MCF, Université de Toulouse, IRIT	Examinateur
M.	Romain Serizel	MCF, Université de Lorraine, LORIA	Examinateur
M.	Jean-François Bonastre	Professeur, Université d'Avignon, LIA	Examinateur
M.	Fabrice Lefèvre	Professeur, Université d'Avignon, LIA	Examinateur
M.	Rachid El-Azouzi	Professeur, Université d'Avignon, LIA	Examinateur
M.	Driss Matrouf	MCF-HDR, Université d'Avignon, LIA	Directeur de thèse

Laboratoire d'Informatique d'Avignon

À mes très chers parents et à mon frère,
à tous mes amis,

et au prof de M.G. dans mon école préparatoire
qui m'a dit que faire de l'informatique allait être
la pire erreur de ma vie ..
Ce jeune docteur en ***informatique*** *n'a aucun regret, monsieur.*

Acronymes

DCF Decision Cost Function
DCT Discrete Cosine Transform
DET Detection Error Tradeoff
DNN Deep Neural Network

EER Equal Error Rate
EFR Eigen Factor Radial

GMM Gaussian Mixtures Model

HMM Hidden Markov Model

JFA Joint Factor Analysis

LDA Linear Discriminant Analysis

MAP Maximum a Posteriori
MFCC Mel Frequency Cepstral Coefficients
MMSE Minimum Mean Square Error

NAP Nuisance Attribute Projection
NIST National Institute of Standards and Technology
NMF Non-negative Matrix Factorization

PCA Principal Component Analysis
PLDA Probabilistic Linear Discriminant Analysis

RAL Automatic speaker recognition
RAP Automatic speech recognition

SITW Speakers In The Wild
SVM Support Vector Machine

UBM Universal Background Model

VAD Voice Activity Detection

VQ Vector Quantization

WCCN Within Class Covariance Normalization

Remerciements

A l'issue de la rédaction de ce rapport, je suis convaincu que cette thèse est loin d'être un travail solitaire. En effet, je n'aurais jamais pu réaliser ce travail sans le soutien d'un grand nombre de personnes dont la générosité, la bonne humeur et l'intérêt manifestés à l'égard de ma recherche m'ont permis de progresser dans cette phase délicate de « l'apprenti-chercheur ».

En premier lieu, je tiens à remercier mon directeur de thèse, monsieur Driss Matrouf, pour la confiance qu'il m'a accordée en acceptant d'encadrer ce travail doctoral, pour ses multiples conseils et pour toutes les heures qu'il a consacrées à diriger cette recherche. J'aimerais également lui dire à quel point j'ai apprécié sa grande disponibilité et son respect sans faille des délais serrés de relecture des documents que je lui ai adressés. Enfin, j'ai été extrêmement sensible à ses qualités humaines d'écoute, de compréhension et de support tout au long de cette période de thèse.

Mes remerciements vont également à tous les membres de mon jury pour leur présence et leur participation à la soutenance de cette thèse : monsieur Fabrice Lefèvre, qui a accepté de présider ce jury, ainsi que messieurs Thomas Pelligrini, Romain Serizel, Jean-François Bonastre et Rachi El Azzouzi pour avoir fait partie de mon jury en qualité d'examinateurs. Un grand merci également à messieurs Denis Jouvet et Claude Barras qui ont de plus accepté la charge d'être rapporteurs de ce travail.

Merci aux RALeurs au laboratoire ; Pierre-Michel, Mikael, Driss et Jean-François dont j'ai beaucoup appris et plus particulièrement Moez, le thésard avec qui j'ai eu le plaisir de partager des centaines de cafés et de discuter pendant des heures, voir des semaines, de tout ce qui touche la RAL de près ou de loin.

Je n'oublierai pas de remercier mes deux colocataires de bureau Mohamed et Titouan qui ont su faire du bureau RC11 un coin de travail, de bonne humeur et de régal. Merci à eux pour leur aide et leur amitié. Ce fut aussi un plaisir de connaître et de travailler avec les autres membres du LIA qui ont fait de ce laboratoire une deuxième maison plus qu'un lieu de travail. Merci à Imed, Mohamed, Zak, Greg, Manu, Xavier, Cedric, Nejat, Imran, Louis, Elvis, Killian, Elvys, Etienne, Mathieu, Adrien, Abdelillah, Imen, Olfa, Majed, Oussama, Fen, Min, Yonatan, Mayeul, Cyril, Mathias, Teva, Tania, Stephane, Simone, Yann, Corinne, Bassam, Richard, George, Philou et j'en oublie.

Enfin, merci à Alexandra Elbakyan, la "Robin Hood" des sciences qui a rendu ma vie de chercheur plus facile. Merci à Joshua Bell et à Maxim Vengerov dont les mélodies ont rendues plus agréables les longues nuits de recherche, de débuggage et de rédaction.

Résumé

Le domaine de la reconnaissance automatique du locuteur (RAL) a vu des avancées considérables dans la dernière décennie permettant d'atteindre des taux d'erreurs très faibles dans des conditions contrôlées. Cependant, l'implémentation de cette technologie dans des applications réelles est entravée par la grande dégradation des performances en présence de nuisances acoustiques en phase d'utilisation. Un grand effort a été investi par la communauté de recherche en RAL dans la conception de techniques de compensation des nuisances acoustiques. Ces techniques opèrent à différents niveaux : signal, paramètres acoustiques, modèles ou scores. Avec le développement du paradigme de "variabilité totale", de nouvelles possibilités peuvent être explorées profitant des propriété statistiques simples de l'espace des i-vecteurs.

Notre travail de thèse s'inscrit dans ce cadre et propose des techniques de compensation des nuisances acoustiques qui opèrent directement dans le domaine des i-vecteurs. Ces algorithmes utilisent des relations simples entre les i-vecteurs corrompus et leurs versions propres et font abstraction de l'effet réel des nuisances dans cet espace. Afin de mettre en œuvre cette méthodologie, des exemples de données propres / corrompues sont générés artificiellement et utilisés pour construire des algorithmes de compensation des nuisances acoustiques. Ce procédé permet d'éviter les dérivations qui peuvent être complexes, voire très approximatives. Les techniques développées dans cette thèse se divisent en deux classes :

La première classe de techniques se base sur un modèle de distorsion dans le domaine des i-vecteurs. Une relation entre la version propre et la version corrompue d'un i-vecteur est posée et un estimateur permettant de transformer un i-vecteur de test corrompu en sa version propre est construit. Deux stratégies ont été développées dans ce contexte. La première se base sur l'algorithme de Kabsch et modélise l'effet de la corruption acoustique dans l'espace des i-vecteurs sous forme d'une translation suivie d'une rotation. Cet algorithme est testé en présence de bruit additif ; une amélioration relative de 40% a été observée sur les données de l'évaluation NIST SRE 2008 bruitées artificiellement.

Le deuxième algorithme, appelé I-MAP, utilise le critère du maximum à posteriori. A la différence du premier algorithme, I-MAP considère la différence entre la version propre et la version corrompue d'un i-vecteur comme étant un bruit additif dans le domaine des i-vecteurs. La version propre correspondant à un i-vecteur de test corrompu est obtenue en utilisant le critère du maximum à posteriori (MAP) tout en supposant

une distribution Gaussienne pour les i-vecteurs propres ainsi que pour le bruit dans l'espace des i-vecteurs. Cet algorithme a permis d'obtenir des gains pouvant atteindre 60% d'amélioration relative en termes d'EER sur les données de l'évaluation NIST SRE 2008 bruitées artificiellement et de 50% sur les données de SITW bruités naturellement.

Une combinaison itérative des deux algorithmes s'est avérée efficace permettant d'atteindre 80% d'amélioration relative en EER sur les données de l'évaluation de NIST SRE 2008 bruitées artificiellement.

Mis à part ces deux techniques, une méthodologie de construction automatique de techniques de compensation des nuisances a été mise en place. Cette approche modélise les algorithmes de compensation des nuisances sous forme d'arbre et utilise un algorithme de programmation génétique pour générer de nouvelles techniques de compensation des nuisances acoustiques. Un processus itératif qui s'inspire de l'évolution Darwinienne est par la suite adopté. Cette approche sélectionne les meilleures arbres solution générées par l'algorithme génétique et les utilise pour créer de nouvelles solutions. Cette méthode a été testée dans le contexte du bruit additif et s'est avérée efficace pour la construction de techniques de compensation adaptées à un bruit et niveau SNR donnés.

La deuxième classe de techniques n'utilise aucun modèle de distorsion dans le domaine des i-vecteurs. Elle permet de tenir compte à la fois de la distribution des i-vecteurs propres, corrompus ainsi que la distribution jointe. Des améliorations significatives des performances atteignant des gains relatifs de 80% en termes d'EER sont observées sur les données de l'évaluation NIST SRE 2008 bruitées artificiellement et 66% sur les données de SITW bruitées naturellement. Cette approche a aussi été évaluée dans le contexte des segments de courtes durées en se basant sur la distribution jointe entre les i-vecteurs construits sur de longues sessions et leurs versions de courtes durées. Dans ce contexte, on a observé des gains relatifs en termes d'EER qui peuvent atteindre 40% sur les données de l'évaluation NIST SRE 2008 et un gain de 35% sur les segments de courte durée de la base SITW (seuls les segments propres sont utilisés).

Abstract

Speaker recognition witnessed considerable progress in the last decade, achieving very low error rates in controlled conditions. However, the implementation of this technology in real applications is hampered by the great degradation of performances in presence of acoustic nuisances. A lot of effort has been invested by the research community in the design of nuisance compensation techniques in the past years. These algorithms operate at different levels : signal, acoustic parameters, models or scores. With the development of the "total variability" paradigm, new possibilities can be explored due to the simple statistical properties of the i-vector space.

Our work falls within this framework and presents new compensation techniques which operate directly in the i-vector space. These algorithms use simple relationships between corrupted i-vectors and the corresponding clean versions and ignore the real effect of nuisances in this domain. In order to implement this methodology, pairs of clean and corrupted data are artificially generated then used to develop nuisance compensation algorithms. This method avoids making complex derivations and approximations. The techniques developed in this thesis are divided into two classes :

The first class of techniques is based on a distortion model in the i-vector space. A relationships between the clean version of an i-vector and its corrupted version is set and an estimator is built to transform a corrupted test i-vector to its clean counterpart. Two strategies have been developed in this context. The first one is based on the Kabsch algorithm and models the effects of acoustic corruption in the i-vector space by a translation followed by a rotation. This algorithm is tested in the presence of additive noise ; A relative improvement of 40% was observed on the artificially corrupted NIST SRE 2008 data.

The second algorithm, called I-MAP, uses the maximum a posterior criterion. Unlike the first algorithm, I-MAP models the difference between the clean and the corrupted version of an i-vector as an additive noise. The clean version corresponding to a corrupted i-vector is obtained by using the maximum a posteriori criterion (MAP) while assuming a Gaussian distribution for the clean i-vectors as well as the noise distribution in the i-vector space. This algorithm yielded gains of up to 60% relative EER improvement on the artificially corrupted NIST SRE 2008 data and 50% on noisy SITW data.

An iterative combination of the two algorithms was proven effective achieving 80% of relative EER improvement on the artificially corrupted NIST SRE 2008 data.

Besides these two techniques, a methodology for automatic construction of nuisance compensation techniques has been put in place. This approach models the nuisance compensation algorithms in the form of a syntactic tree and uses a genetic programming algorithm to generate new compensation techniques. An iterative process inspired by Darwinian evolution is subsequently adopted. This approach selects the best solution trees generated by the genetic algorithm and uses them to create new solutions. This method has been tested in the context of additive noise and was proven effective for the construction of compensation techniques adapted to a given noise and SNR level.

The second class of techniques does not use any distortion model in the i-vectors domain. It takes into account both the distribution of the clean, corrupt i-vectors as well as the joint distribution. Significant improvements in performance reaching relative gains of 80% in EER are observed on artificially corrupted NIST SRE 2008 data and 66% on noisy SITW data. This approach has also been evaluated in the context of short-duration segments based on the joint distribution between i-vectors built using long sessions and the corresponding short versions. In this context, relative EER gains were observed, reaching 40% on the NIST SRE 2008 data and 35% on the short segments of the SITW database (only the clean segments are used).

Table des matières

Introduction

"If my brain can tell the difference between noise and signal, my heart cannot."

Nassim Nicholas Taleb
Fooled by Randomness

La reconnaissance automatique du locuteur (RAL) a atteint aujourd'hui une maturité suffisante pour être considérée comme solution viable dans des applications réelles. L'authentification basée sur la voix devient de plus en plus courante dans les systèmes de sécurité modernes et a aussi trouvé des applications dans le domaine judiciaire. Dans la dernière décennie, le domaine de la RAL a vécu une révolution avec le développement du paradigme i-vecteur. Ceci a permis d'avoir une représentation compacte qui résume l'information acoustique contenue dans un segment de parole sous forme d'un vecteur de dimension réduite. Les propriétés statistiques du domaine des i-vecteurs (dimension faible et distribution Gaussienne) ont ouvert de nouvelles possibilités et ont permis de tester des techniques qui étaient difficiles à implémenter avec les représentations à base de GMM ou de supervecteurs. Couplés avec un système de scoring PLDA (*Probabilistic Linear Discriminant Analysis*), les systèmes de RAL à base d'i-vecteurs sont devenus le standard et ont permis d'atteindre de très bonnes performances dans des conditions propres.

En pratique, de nombreux facteurs liés aux conditions d'enregistrement peuvent dégrader significativement les performances des systèmes de RAL. Ces facteurs peuvent être en relation avec l'environnement (bruit additif, réverbération, etc), le dispositif d'enregistrement (distorsions de canal) ou le locuteur lui même (état psychologique, effort vocal, changement de voix, etc). Souvent, ces facteurs ne peuvent pas être connus à l'avance, ce qui représente un défi pour les applications réelles. Un intérêt croissant a été accordé par la communauté de recherche en RAL à la compensation de ces nuisances au cours des trente dernières années. Cependant, une attention particulière a été portée à la construction d'extracteurs d'i-vecteurs plus robustes depuis l'introduction du paradigme i-vecteur. Les algorithmes développés dans ce contexte (exp : algorithmes à base de séries de Taylor) propagent l'effet des nuisances acoustiques du domaine temporel vers les paramètres du modèle acoustique. Ce genre de techniques résulte en des dérivations complexes et utilise souvent des approximations pour rendre les calculs plus faciles. En dépit de la structure simple de l'espace des i-vecteurs, peu de travaux se sont

attaqués au problème de traitement de nuisances directement dans ce domaine.

Notre travail de recherche est un premier pas dans cette direction. C'est une invitation à considérer la possibilité de modéliser et de compenser les nuisances acoustiques directement dans le domaine des i-vecteurs. Cette approche permet d'éviter la complexité induite par les techniques de compensation de modèles et de tirer parti de la structure simple du domaine des i-vecteurs. En effet, il est possible de modéliser directement l'effet des nuisances acoustiques dans l'espace des i-vecteurs en se basant sur des données qui reflètent leurs effets dans ce domaine. Différentes relations entre la version propre et la version corrompue d'une même session peuvent donc être considérés et utilisés pour construire des techniques de compensation des nuisances acoustiques. Ceci est rendu possible par la propriété de reproductibilité de certaines nuisances acoustiques dans le domaine temporel (exp : le bruit additif) qui nous permet de générer artificiellement des versions corrompues et permet d'avoir deux versions (propre/corrompue) d'un segment de parole. Une fois générés, ces couples d'i-vecteurs peuvent être utilisés pour construire un algorithme de compensation des nuisances acoustiques.

Dans un premier temps, on présente un algorithme basé sur l'algorithme de Kabsch qui modélise la corruption dans le domaine des i-vecteurs sous forme de translation suivie par une rotation. Dans ce cadre, les couples d'i-vecteurs propres et d'i-vecteurs corrompus sont utilisés pour estimer le meilleur vecteur de translation et la meilleure matrice de rotation qui transforment un i-vecteur corrompu en sa version propre. Cette approche est testée en présence de bruit additif et a permis d'avoir des gains significatifs atteignant un gain de 40% en EER sur les données de l'évaluation NIST SRE 2008 bruitées artificiellement. Vu que cette approche n'intègre pas de connaissances à priori sur la distribution des i-vecteurs propres et peut produire des estimations distordues, un deuxième algorithme bayésien portant le nom de I-MAP est développé. Cette technique modélise la corruption dans le domaine des i-vecteurs sous forme de bruit additif. Par la suite, un estimateur basé sur le critère du maximum à posteriori (MAP) est développé pour obtenir la version propre étant donnée la version bruitée correspondante. Cet algorithme se base sur une modélisation Gaussienne de la distribution des i-vecteurs propres ainsi que celle du bruit. Cette approche s'est avérée efficace dans le contexte du bruit additif donnant un gain relatif de 60% en termes d'EER sur les données de l'évaluation NIST SRE 2008 bruitées artificiellement et de 50% sur les données de SITW bruités naturellement. En pratique, l'implémentation de I-MAP nécessite l'estimation de la distribution de bruit pour chaque session. Cette procédure est coûteuse en temps de calcul. Afin d'accélérer cette opération, une solution qui construit une base de distributions d'i-vecteurs de bruits en utilisant un ensemble de bruits et de niveaux de SNR a été proposée. En phase de test, l'i-vecteur de test est utilisé pour sélectionner la meilleure distribution de bruit correspondant aux conditions acoustiques de test. Cette approche permet d'accélérer significativement les performances de l'algorithme tout en gardant des gains importants en EER.

Vu que l'algorithme I-MAP suppose l'indépendance entre les deux distributions des i-vecteurs propres et i-vecteurs corrompus, une autre approche est développée pour tenir compte de l'information jointe entre les deux représentations. Cet algorithme mo-

délise directement la distribution jointe entre les i-vecteurs propres et les i-vecteurs corrompus. Des améliorations significatives des performances atteignant des gains relatifs de 80% en termes d'EER sont observées sur les données de l'évaluation NIST SRE 2008 bruitées artificiellement et 66% sur les données de SITW bruitées naturellement. Cette approche a aussi été évaluée dans le contexte des segments de courtes durées en se basant sur la distribution jointe entre les i-vecteurs correspondant à des segments de longues durées et leurs versions de courtes durées. Cette technique a donné des gains relatifs en termes d'EER qui peuvent atteindre 40% sur les données de l'évaluation NIST SRE 2008 et un gain de 35% sur les segments de courte durée de la base SITW.

Organisation du document

La partie I de ce document présente un état de l'art de la reconnaissance automatique du locuteur dans les conditions difficiles :

Le chapitre 1 présente la problématique de la reconnaissance automatique du locuteur. Par la suite, une description schématique des composantes d'un système de vérification du locuteur est faite. Les étapes de paramétrisation, de modélisation et de décision sont par la suite reprises en détails en mettant l'accent sur les systèmes de RAL à base d'i-vecteurs.

Le chapitre 2 introduit les variabilités et nuisances acoustiques qui peuvent rendre la tâche de reconnaissance du locuteur difficile à accomplir. Les défis rencontrés en RAL sont présentés ainsi que les enjeux dans les applications réelles. Par la suite, des exemples de nuisances acoustiques sont détaillés (bruit additif, distorsion canal, réverbération) ainsi que leurs effets sur le signal de parole.

Le chapitre 3 présente un panorama des techniques de compensation des nuisances acoustiques développées en RAL. Ce chapitre couvre les techniques d'extraction de paramètres robustes, les techniques d'amélioration de signal et de compensation de paramètres, les approches de compensation de modèles ainsi que les modèles de scoring robustes.

La partie II de ce document présente les contributions et travaux conduits dans cette thèse :

Le chapitre 4 introduit les jeux de données utilisées tout au long de cette thèse. Cette partie couvre les données utilisées pour entraîner les systèmes ainsi que les données d'évaluation. Les systèmes de RAL qui seront utilisés dans la partie expérimentale sont détaillés. Enfin, plus d'attention est accordée à deux types de nuisances ; le bruit additif et la variabilité des durées. Les effets de ces nuisances, sur les performances du système de RAL de base utilisé dans cette thèse, sont analysés.

Le chapitre 5 introduit deux algorithmes de compensation des nuisances dans le domaine des i-vecteurs et évalue leurs performances dans des conditions bruitées. Le premier algorithme se base sur l'algorithme de Kabsch et modélise la corruption dans l'espace des i-vecteurs sous forme d'une translation suivie d'une rotation. Le deuxième

algorithme utilise le critère de maximum à posteriori (MAP) et modélise la corruption sous forme de bruit additif dans l'espace des i-vecteurs. Enfin, une combinaison itérative de ces deux algorithmes est évaluée.

Le chapitre 6 explore une méthode de génération automatique de techniques de compensation des nuisances basée sur la programmation génétique.

Jusqu'alors, les approches proposées supposent l'indépendance entres les distributions des i-vecteurs propres et des i-vecteurs corrompus. Elle utilisent aussi une relation spécifique entre les i-vecteurs propres et leurs versions corrompues. Le chapitre 7 présente une méthode générique de traitement de nuisances dans le domaine des i-vecteurs qui modélise la distribution jointe entre les i-vecteurs propres et leurs versions corrompues. Cette approche est évaluée dans le contexte du bruit additif et des segments de courte durée.

Nous concluons ce document par une analyse des contributions réalisées tout au long de cette thèse ainsi qu'un panorama des perspectives ouvertes par notre travail.

Première partie

Reconnaissance du locuteur dans les milieux difficiles, un état de l'art

Chapitre 1

Reconnaissance automatique du locuteur : du signal aux i-vecteurs

1.1 Fondements de la reconnaissance automatique du locuteur

La reconnaissance automatique du locuteur (RAL) est généralement décrite comme l'identification d'une personne par sa voix. Il est possible de classifier les systèmes de reconnaissance automatique du locuteur selon leur dépendance au texte prononcé, la tâche ciblée ou leur domaine d'application. On présente dans les sous-sections suivantes, des classifications possibles selon ces critères et on finit par décrire la structure d'un système de vérification du locuteur.

1.1.1 Dépendance et indépendance au texte

Il est possible de diviser les systèmes de RAL en deux types de systèmes :

1. *Systèmes dépendants du texte :* le texte prononcé par le locuteur en phase d'utilisation est le même que celui qui a été prononcé lors de la phase d'apprentissage.
2. *Systèmes indépendants du texte :* le locuteur peut prononcer n'importe quelle phrase pour être reconnu.

Néanmoins, il existe plusieurs niveaux de dépendance du texte suivant les applications (Bimbot et al., 1994), on les liste selon le degré croissant de dépendance au texte :

— Systèmes à texte libre (*free-text*) : le locuteur prononce ce qu'il veut.
— Systèmes à texte suggéré (*text-prompted*) : un texte, différent à chaque session et pour chaque personne, est imposé au locuteur et affiché à l'écran par la machine.
— Systèmes dépendants de traits phonétiques (*speech event dependent*) : certains traits phonétiques spécifiques sont imposés dans le texte que le locuteur doit prononcer.
— Systèmes dépendants du vocabulaire (*vocabulary dependent*) : le locuteur prononce une séquence de mots issus d'un vocabulaire limité (exp. : séquence de digits).
— Systèmes personnalisés dépendants du texte (*user-specific text dependent*) : chaque locuteur a son propre mot de passe.

Bien qu'il soit difficile de faire des comparaisons directes entre les systèmes dépendants et indépendants du texte, l'opinion acceptée au sein de la communauté de RAL stipule que les performances des systèmes dépendants du texte sont généralement supérieures à leurs analogues indépendants du texte (Campbell, 1997). Cet écart est une conséquence de la variabilité due au contenu linguistique de la phrase prononcée ainsi que la variabilité de durée des enregistrements utilisés dans le cas indépendant du texte.

1.1.2 Différentes tâches en RAL et applications

Un nombre de tâches centrées sur l'identité des locuteurs ont été étudiées dans la dernière trentaine d'années. Ces tâches répondent à des besoins applicatifs et permettent de tirer parti de l'identité du locuteur de différentes manières dépendant de l'application ciblée. On introduit les principales tâches utilisées dans les systèmes de

RAL et on fait le lien avec les applications dans lesquelles ces tâches sont implémentées.

Vérification vs. identification du locuteur

Dés la lancée des premiers travaux pendant les années 70, la vérification automatique et l'identification automatique du locuteur étaient, et restent jusqu'à ce jour, deux tâches pionnières dans le domaine de la reconnaissance automatique du locuteur (Atal, 1976; Doddington, 1985, 1998; Furui, 1996, 1997; Naik, 1994; Rosenberg, 1992; Douglas, 1986; Hansen et Hasan, 2015) :

— ***La vérification*** consiste à accepter ou refuser l'identité proclamée par un locuteur, en se basant sur un modèle qui lui est associé. Elle traite le scénario où le système prend en entrée un énoncé de test ainsi qu'une identité proclamée. La tâche consiste alors à prendre une décision binaire qui va confirmer ou infirmer le fait que l'enregistrement de test est effectivement prononcé par le locuteur proclamé.
— ***L'identification*** consiste en la reconnaissance d'un locuteur particulier parmi un ensemble fini de locuteurs possibles. Cette tâche peut à nouveau être de deux types : *in-set* et *out-of-set*. Dans le scénario *in-set*, le système automatisé suppose que le locuteur de test (le locuteur demandant l'authentification) doit être parmi les locuteurs connus, tandis que dans le scénario *out-of-set*, le système peut décider que le locuteur de test ne figure pas parmi les locuteurs reconnus.

D'autres tâches ont aussi été développées pour répondre à des besoins applicatifs, à savoir l'indexation par locuteur de flux audio (Johnson, 1999; Delacourt, 2000), le suivi des locuteurs (*speaker tracking*) (Rosenberg et al., 1998; Sönmez et al., 1999; Bonastre et al., 2000; Delacourt et al., 2000; Martin et Przybocki, 2000) ainsi que la détection d'un locuteur dans une conversation (Martin et Przybocki, 2000; Przybocki et Martin, 1999). On présente dans ce qui suit les domaines d'application dans lesquels interviennent ces tâches.

Applications des technologies de RAL

Un intérêt croissant est accordé aux technologies basées sur l'identification vocale dans les domaines public et industriel. En effet, la RAL intervient de nos jours dans un grand nombre d'applications :

— **Applications de sécurité :** contrôle d'accès (complément d'un code ou d'un badge) ; accès physique (pour des banques, voitures, entreprises) ou accès distant (consultation de comptes bancaires par téléphone).
— **Police criminelle / identification de suspects :** filtrage de voix suspectes (+ validation humaine).
— **Indexation multimédia :** indexation par locuteur, adaptation automatique des

modèles acoustiques à la voix du locuteur pour les applications de reconnaissance de la parole.

En pratique, les tâches de RAL indépendantes du texte sont plus difficiles en raison de la variabilité phonétique présente en phase de test. Ce cadre est plus intéressant en recherche vu qu'il impose moins de restrictions sur les données utilisées et reflète un contexte d'évaluation plus général. Cet intérêt est, en effet, visible dans les évaluations de RAL indépendantes du texte NIST SRE financées par DARPA, une agence du département de la défense des États-Unis (DOD). Pour toutes ces raisons, on se met dans le contexte de la RAL indépendante du texte tout au long de cette thèse.

1.1.3 Structure d'un système de RAL

Les systèmes de vérification du locuteur sont composés de trois modules comme le montre la figure 1.1 :

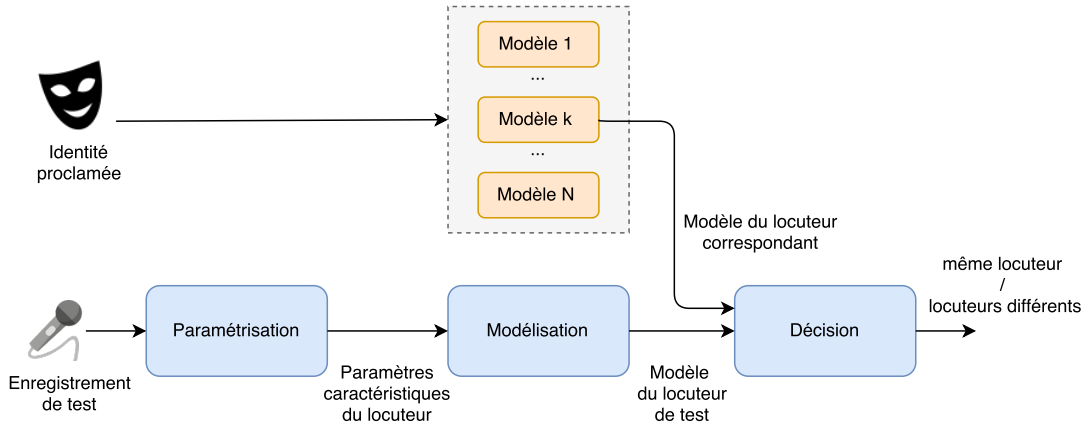

FIGURE *1.1* – *Structure d'un système de vérification du locuteur.*

— **Paramétrisation :** Cette étape vise à capturer des paramètres caractéristiques de la parole d'une personne donnée. Suite à de nombreux travaux de recherche (Sambur, 1975; Bonastre et Meloni, 1992), il s'est avéré que les paramètres basés sur la représentation spectrale de la parole sont les plus pertinents pour la RAL. Ces paramètres sont corrélés à la forme du conduit vocal et sont les plus utilisés dans les systèmes de RAL modernes. Cependant, les paramètres prosodiques qui décrivent le style de parole du locuteur sont aussi utilisés en pratique.
— **Modélisation :** Les paramètres acoustiques extraits d'un enregistrement donné sont utilisés pour construire un modèle qui résume l'information acoustique correspondante.
— **Décision :** La phase de décision désigne l'identité du locuteur reconnu. Dans le cas de la vérification, cette décision est binaire et consiste à confirmer ou infirmer la correspondance de la session de test à une identité proclamée.
Vu qu'il est impossible d'avoir une similarité de 100% entre le signal du locuteur de test et celui des locuteurs clients, les modèles sont conçus de telle sorte qu'une

telle comparaison fournisse un score (une valeur scalaire) indiquant si les deux énoncés correspondent au même locuteur. Si ce score est supérieur (inférieur) à un seuil prédéfini, le système accepte (rejette) le locuteur de test.

1.2 Production du signal de parole

Les techniques de paramétrisation de la parole utilisées en RAL se basent généralement sur deux types de modèles :

— *Un modèle de production :* les paramètres extraits visent à caractériser l'appareil de production de la parole.

— *Un modèle de perception :* la conception de paramètres s'inspire de la perception humaine de la parole. Les travaux conduits en psychoacoustique [1] sont souvent utilisés dans ce contexte.

Afin de mieux comprendre les principes directeurs derrière les techniques de paramétrisation utilisées en RAL, on commence par présenter le mécanisme de production de la parole. Par la suite, on fait le lien avec les paramètres acoustiques utilisées.

1.2.1 Mécanisme de production de la parole

La production de la parole est un processus de nature linguistique (message à transmettre) qui évolue vers une exécution motrice (séquence de contractions musculaires) mettant en jeu plusieurs composantes de l'anatomie humaine et résultant en un signal de parole. Ce processus peut être décomposé en trois étapes (Brown et Hagoort, 2000; Blank et al., 2002) :

1. **La conceptualisation (ou préparation conceptuelle) :** dans cette étape, l'intention de créer la parole génère les concepts désirés correspondant au message à transmettre.
2. **La formulation :** dans cette étape, la forme linguistique requise pour l'expression du message désiré est créée. La formulation comprend le codage grammatical (sélectionner les mots et la forme syntaxique appropriée), le codage morpho-phonologique (découper les mots en syllabes), la syllabification et l'encodage phonétique.
3. **L'articulation et exécution motrice de la parole :** qui consiste à l'exécution de la séquence articulatoire correspondant au message. Dans cette étape, le locuteur exécute une série de signaux neuromusculaires qui servent de commandes et permettent de contrôler les cordes vocales, les lèvres, la mâchoire, la langue et le vélum (voile du palais), produisant ainsi la séquence sonore voulue en sortie (Levelt, 1993). Les principales composantes responsables de la production de la parole humaine sont illustrées dans la figure 1.2.

1. La psychoacoustique est une branche de la psychophysique qui étudie la perception des sons.

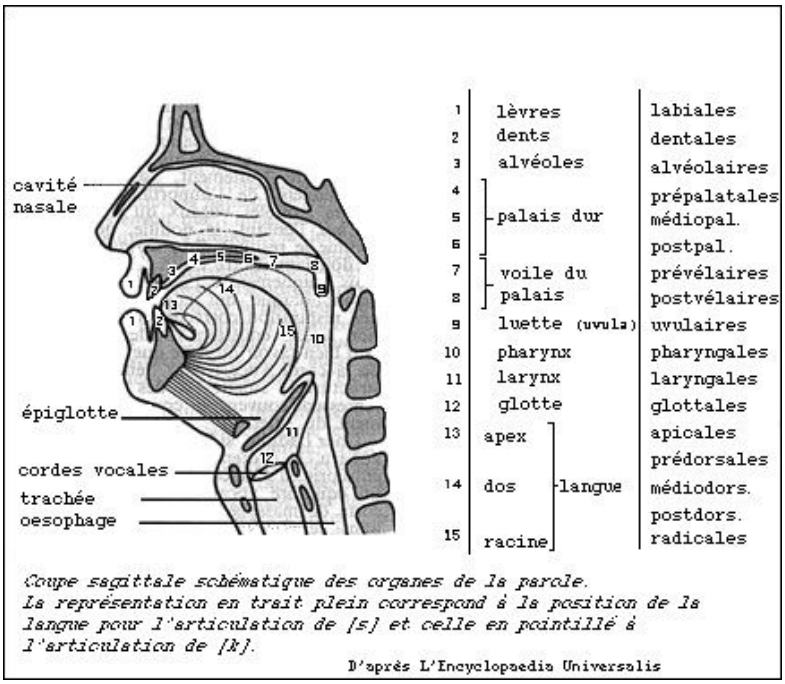

FIGURE 1.2 – *Anatomie des organes de production de la parole humaine (source : Encyclopédie Universalis).*

1.2.2 Propriétés acoustiques du conduit vocal

Dans la littérature de traitement de la parole, le terme **conduit vocal** (ou tractus vocal) fait référence à la totalité de la cavité remplie d'air qui se trouve entre la glotte (entrée du larynx) et les lèvres. Cette cavité est plastique et dynamique, capable de prendre un nombre considérable de configurations et de changer très rapidement de forme. Ces modifications sont faites par le mouvement d'articulateurs (comme la langue). En essence, la parole produite correspond à une variation de la pression d'air suite à la modulation de l'air sortant du larynx par l'activité des cordes vocales et des cavités dans le conduit vocal, produisant différents sons phonétiques. Le contenu en fréquence du signal acoustique est modifié par les propriétés de résonance des différentes cavités le long du trajet. Ces fréquences de résonance sont connues sous le nom de **formants** et sont généralement numérotées en allant des basses fréquences vers les hautes fréquences (F_1, F_2, F_3, ..). La fréquence F_0 est appelée **fréquence fondamentale** et correspond à la fréquence de vibration des cordes vocales [2].

En se basant sur ces propriétés acoustiques ainsi que sur la réponse fréquentielle de l'appareil phonatoire, des modèles dits **source-filtre** ont été établis pour décrire les mécanismes et les propriétés de production des sons et modéliser le couple {cordes vocales, cavités supra-glottiques} en {source, filtre}. Ce genre de modèles permet d'assimiler la

2. Les propriétés du son de parole sont généralement divisées en propriétés physiques (la valeur mesurée d'une entité) et la valeur perçue. Le terme **pitch** est généralement utilisé pour qualifier la propriété perceptuelle corrélée à la fréquence F_0 (une fréquence plus élevée correspond à un **pitch** plus aigu).

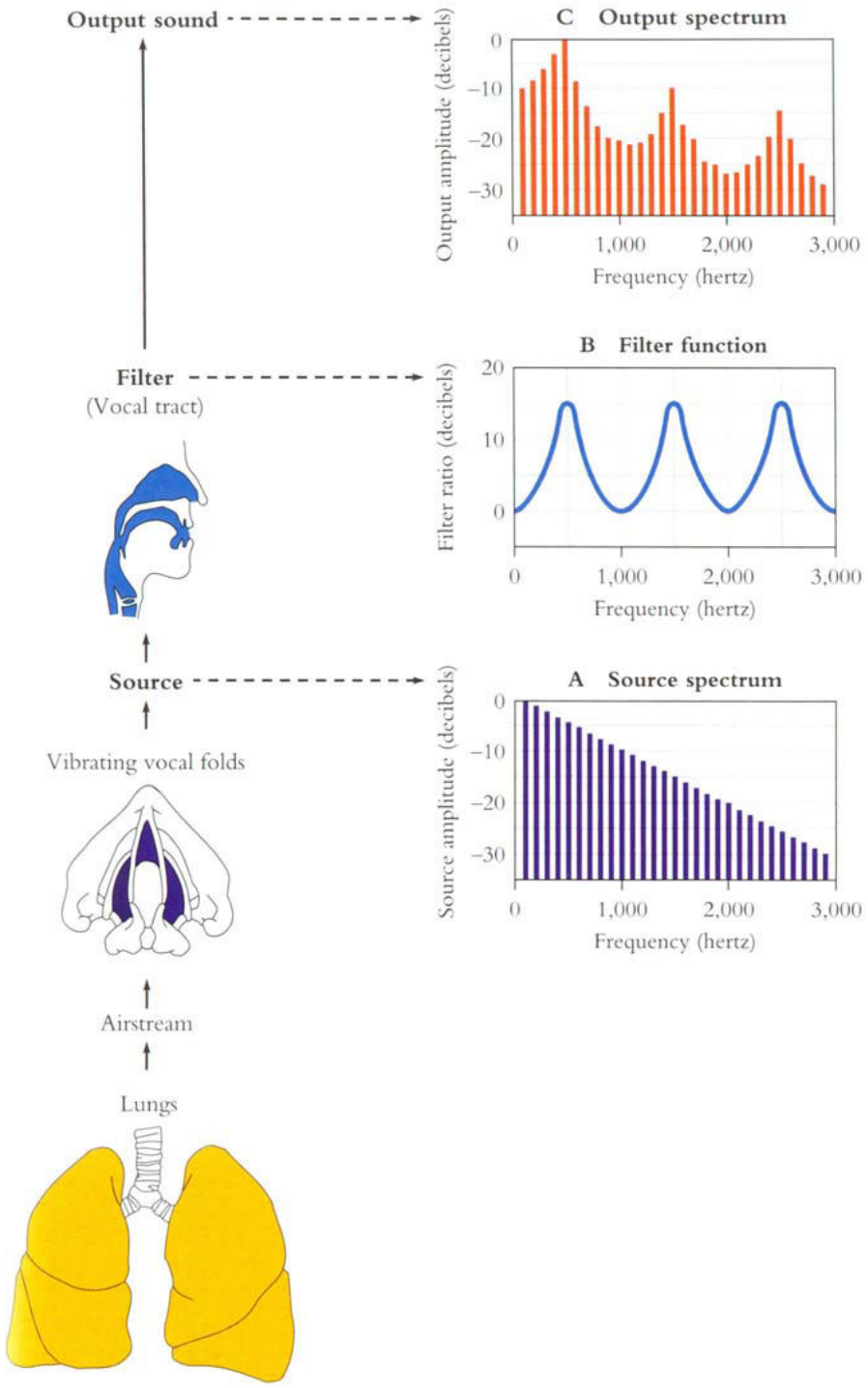

FIGURE 1.3 – *Aperçu détaillé du modèle source-filtre (source : http://www.ling.upenn.edu/courses/Spring_2001/ling001/phonetics.html).*

réponse de l'appareil phonatoire (plus précisément celle des cavités supra-glottiques) à celle d'un filtre qu'il est possible de modéliser et comparer entre différents locuteurs.

Dans ce modèle (détaillé dans la figure 1.3), le spectre de la source glottique est harmonique du fait de sa périodicité (le son contient de l'énergie à la fréquence fondamentale de vibration des cordes vocales F_0 ainsi qu'aux fréquences $2 \times F_0$, $3 \times F_0$,... $n \times F_0$). L'énergie diminue toutefois avec la fréquence (figure 1.3 (**A**)). L'effet de filtrage des cavités buccale et nasale sur la source glottique est représenté par une fonction de transfert (la figure 1.3 (**B**) donne un exemple qui contient trois fréquences de résonance). Suite au passage de l'air par le conduit vocal, il en résulte un spectre (toujours harmonique) dont l'énergie varie avec la fréquence (figure 1.3 (**C**)).

Cette modélisation permet de réduire la forme complexe du signal de la parole à un vecteur de paramètres (les coefficients du filtre). Ces paramètres permettent de décrire la réponse fréquentielle du conduit vocal et peuvent être utilisées pour la caractérisation du locuteur.

1.2.3 Caractérisation du locuteur

La forme du conduit vocal est caractéristique de la voix d'un locuteur donné et dépend de l'emplacement exact de chaque organe tout au long de la cavité du conduit vocal. Cependant, la nature non-stationnaire du signal de la parole et sa grande variabilité fait en sorte que les motifs acoustiques générés pour un message donné varient avec le temps et le contexte (contenu linguistique, condition psychologique, maladie, âge, ..). De plus, il convient de noter que l'information d'identité d'un locuteur est une information non-linguistique incorporée dans le signal de parole et, par conséquent, il est peu probable qu'une mesure de paramètres simple caractérise de façon unique un locuteur en tout temps. On cite (Doddington, 1985) dans ce contexte :

> The secondary speech messages, including speaker discriminants, are encoded as nonlinguistic articulatory variations of the basic linguistic message. Thus the information useful for identifying the speaker is carried indirectly in the speech signal, a side effect of the articulatory process, and the speaker information may be viewed as "noise" applied to the basic linguistic message. Thus the problem with speaker recognition is that there are no known speech features or feature transformations which are dedicated solely to carrying speaker-discriminating information, and further that the speaker-discriminating information is a second-order effect in the speech features.

Dans ce passage, Doddington met l'accent sur le caractère non-linguistique de l'information locuteur et sur la non-existence de caractéristiques de parole simples à extraire qui contiennent exclusivement les informations discriminantes correspondant au locuteur.

Cependant, certains outils d'analyse spectrale, à savoir les spectrogrammes, se sont avérés utiles pour l'analyse phonétique et ont aussi été utilisés avec succès pour la différenciation des locuteurs (Bolt et al., 1970). Cette approche a été validée par plusieurs travaux qui ont montré la pertinence des paramètres spectraux pour les tâches

de reconnaissance automatique du locuteur (Sambur, 1975; Bonastre et Meloni, 1992). Motivés par les observations mentionnées ci-dessus, les systèmes de RAL utilisent principalement les paramètres spectraux de la forme d'onde du signal de la parole. Cette analyse est faite sur des segments de courte durée (typiquement entre 10 et 30ms) où le signal devient quasi-stationnaire et les propriétés acoustiques, potentiellement uniques au locuteur, sont saisies.

On discute dans la section suivante le processus de paramétrisation du signal de parole et on explore les caractéristiques acoustiques à court et à long terme ainsi que leurs avantages et points faibles. Par la suite, on se concentre sur les paramètres les plus utilisées en RAL (les paramètres court terme).

1.3 Paramétrisation du signal de parole

Le signal de parole est, par nature, complexe et redondant et possède une grande variabilité ce qui le rend difficile à utiliser d'une manière directe par les systèmes de RAL. Cette complexité provient de la combinaison de plusieurs facteurs ; la grande variabilité inter- et intra-locuteur, les effets de la coarticulation en parole continue, les conditions d'enregistrement, etc. De ce fait, il devient nécessaire de procéder à une étape de paramétrisation qui a pour but d'extraire une représentation plus compacte de cette information acoustique qui réduit les redondances et permet d'accentuer les propriétés spécifiques au locuteur.

Wolf a résumé dans (Wolf, 1972), les caractéristiques des paramètres acoustiques idéaux pour la reconnaissance du locuteur en six points :

1. Ils se produisent naturellement et fréquemment dans la parole normale.
2. Ils sont facilement mesurables.
3. Ils varient autant que possible entre les différents locuteurs, mais sont aussi consistants que possible pour chaque locuteur.
4. Ils ne changent pas avec le temps et ne sont pas affectés par la santé du locuteur.
5. Ils ne sont pas affectés par un bruit de fond d'intensité raisonnable et ne dépendent pas du moyen de transmission.
6. Ils ne sont pas modifiables par l'effort conscient du locuteur, ou, au moins, peu susceptibles d'être affectés par des tentatives de déguisement de la voix.

Il est clair que satisfaire simultanément toutes ces conditions est difficile à accomplir en pratique. Cependant, ces critères peuvent être considérés comme des objectifs de conception idéalistes pour la caractérisation de la parole en RAL.

Caractéristiques acoustiques à long/court terme

La différence entre locuteurs se manifeste sur plusieurs niveaux et couvre :

— La différence entre les appareils phonatoires : différence des propriétés de raisonnance de chaque conduit vocal.
— La différence de phonétisation : réalisation différente des mouvements phonétiques par chaque locuteur.
— La différence d'articulation au niveau syllabique.
— La différence entre les traits phonétiques supra-segmentaux (prosodie et style de parole).
— La différence de dialecte, d'accent et utilisation distinguée de la grammaire.

Ces caractéristiques peuvent être divisés en *paramètres de bas niveau* (paramètres acoustiques décrivant l'appareil phonatoire sur de courts intervalles de temps dans différents contextes) et *paramètres de haut niveau* (paramètres extraits au niveau phrase ou mot relatifs à la prosodie et au style de parole). Suite à un grand effort de recherche, il a été établi que les paramètres de bas niveau sont plus efficaces pour des tâches de RAL. Cette préférence vient du fait que les paramètres acoustiques de haut niveau sont plus difficiles à extraire et plus vulnérables au mimétisme (Reynolds et al., 2003; Shriberg et al., 2005). Cependant, ces deux classes de paramètres pourraient être combinées lors de la mise en œuvre d'un système de RAL. En effet, les paramètres de haut niveau peuvent s'avérer bénéfiques et complémentaires aux paramètres de bas niveau. On s'intéresse dans ce qui suit aux paramètres à court terme vu qu'ils sont les plus utilisés dans les applications de RAL en pratique.

Paramètres acoustiques à court terme utilisées en RAL

De nombreux paramètres à court terme ont été développés au fil des années pour des applications de reconnaissance de la parole puis utilisés en reconnaissance du locuteur, on en cite quelques uns :

— **L'analyse prédictive linéaire cepstrale (LPCC : *Linear Predictive Cepstral Coefficients*) :** est une méthode d'analyse cepstrale paramétrique qui repose sur un modèle **source-filtre** (Makhoul, 1975; Markel et Gray, 1982). Dans cette approche, la réponse fréquentielle du conduit vocal est représentée par un filtre et chaque échantillon de parole est exprimé comme une combinaison linéaire d'échantillons passés (d'où le nom *analyse prédictive linéaire*). Une fois estimés, les paramètres du filtre sont convertis en coefficients cepstraux. Les coefficients cepstraux résultants sont utilisés comme paramétrisation à court terme du signal de parole.
— **L'analyse perceptive linéaire (PLP : *Perceptual Linear Prediction*)** : est aussi une méthode paramétrique qui se base sur un modèle **source-filtre**. La paramétrisation PLP est identique à l'analyse LPC, sauf que les caractéristiques spectrales sont transformées pour correspondre aux caractéristiques du système auditif humain. En effet, l'analyse PLP implémente trois propriétés perceptuelles : l'intégration des bandes critiques, la pré-accentuation par la courbe d'isosonie et l'implémentation de la loi de Stevens (Hermansky, 1990).
— **L'analyse en banc de filtres (MFCC : *Mel Frequency Cepstral Coefficients*)** : est une technique d'analyse cepstrale non-paramétrique qui utilise l'échelle de Mel

pour refléter la perception non-linéaire des fréquences par l'oreille humaine (Davis et Mermelstein, 1980).

Généralement, les paramètres de vitesse (Δ) et d'accélération ($\Delta\Delta$) calculés sur plusieurs trames de parole, et représentant les dérivées de premier et second ordre, sont ajoutés aux coefficients cepstraux statiques. Ces paramètres représentent les propriétés dynamiques des coefficients cepstraux à court terme et leur utilité a été prouvée pour les tâches de RAL (Soong et Rosenberg, 1988).

Malgré les efforts qui ont été investis dans la conception de paramètres acoustiques plus pertinents pour la reconnaissance automatique du locuteur et plus robustes aux distorsions acoustiques, les paramètres MFCC et PLP restent jusqu'à ce jour les deux méthodes de paramétrisation prépondérantes en RAL. Pour cette raison, nous optons pour la paramétrisation MFCC dans le cadre de cette thèse.

1.3.1 Détection d'activité vocale (VAD)

Il est évident que les paramètres acoustiques utilisés pour la reconnaissance du locuteur doivent correspondre à des zones de parole et non à des zones de silence (ou de bruit). Pour cette raison, un algorithme de détection d'activité vocale (VAD ; *Voice Activity Detection*)[3] est utilisé avant de procéder à la phase de reconnaissance du locuteur. La détection des segments de parole devient encore plus critique lorsque des conditions acoustiques très bruitées ou dégradées sont considérées. Les approches de détection d'activité vocale les plus populaires en RAL se basent sur la distribution d'énergie des trames et utilisent un seuil de décision pour détecter les zones de parole. Dans ce contexte, la distribution du logarithme de l'énergie est modélisée sous forme d'un modèle GMM à 3 composantes. Par la suite, les trames appartenant à la Gaussienne ayant la plus grande moyenne (les trames les plus énergétiques) sont marquées comme parole (Larcher et al., 2013).

Un exemple de sortie d'un VAD est illustré dans la figure 1.4 (a), où la présence et l'absence de la parole sont indiquées par un signal binaire superposé sur les échantillons de parole. Le spectrogramme correspondant est également montré dans la figure 1.4 (b).

1.3.2 Normalisation des paramètres

Comme indiqué précédemment dans les critères de Wolf (Wolf, 1972), la robustesse à la dégradation et aux nuisances acoustiques fait partie des critères caractérisant les paramètres acoustiques idéales pour la RAL. En réalité, il n'est pas possible de concevoir des paramètres acoustiques qui restent inchangés quelles que soient les conditions d'enregistrement. Cependant, ces changements peuvent être atténués de diverses manières en utilisant des techniques de normalisation.

3. Le terme SAD (*Speech Activity Detection*) est aussi utilisé dans la littérature.

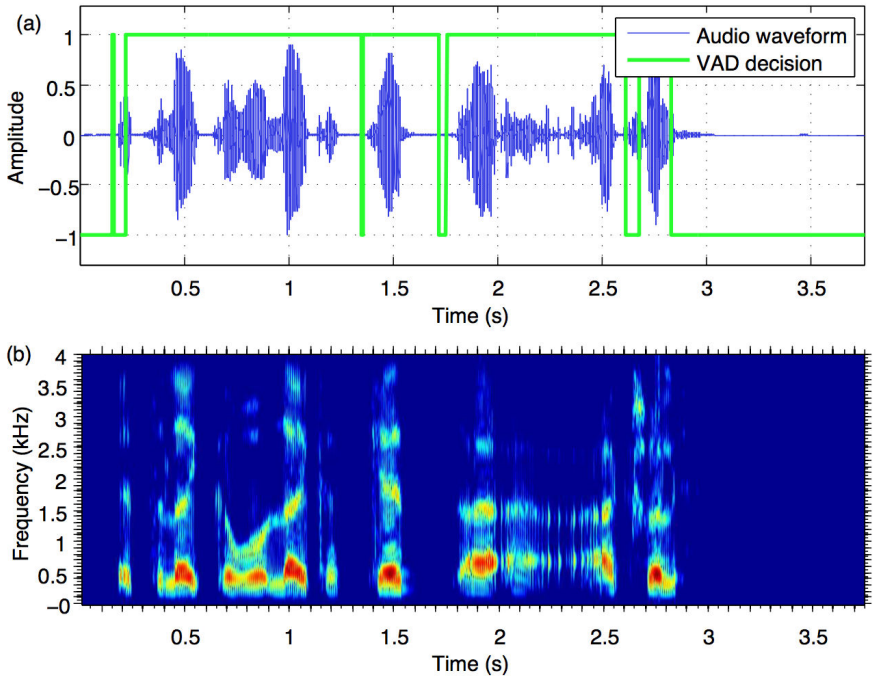

FIGURE 1.4 – *(a) La forme d'onde d'un signal de parole avec la VAD superposée. Une valeur de 1 et -1 indique respectivement la parole et la non-parole (silence). (b) Spectrogramme du signal de parole (source : (Hansen et Hasan, 2015)).*

La soustraction du cepstre moyen : Une méthode efficace pour traiter la variabilité du canal est la soustraction du cepstre moyen (CMN ; *Cepstral Mean Normalization* aussi appelée CMS ; *Cepstral Mean Substraction*) (Schwartz et al., 1993). Cette technique compense le bruit convolutif en supprimant le biais au niveau du cepstre résultant des composantes fixes ou à variation lente. La transformation appliquée peut être écrite sous la forme :

$$\hat{x}_t = x_t - \mu_{cmn} \quad \text{avec} \quad \mu_{cmn} = \frac{1}{T}\sum_{t=1}^{T} x_t \tag{1.1}$$

Où x_t représente un vecteur de paramètres cepstraux à normaliser. La moyenne μ_{cmn} est calculée en utilisant les trames de parole correspondant à un locuteur donné sur une même session [4].

La normalisation CMVN : Une extension de CMN est la normalisation de la moyenne et de la variance cepstrale (CMVN : *Cepstral Mean and Variance Normalization*) où la variance cible des vecteurs paramètres est fixée à l'unité. Cette technique permet d'avoir le même ordre de grandeur sur les trames provenant de différents segments de parole et peut être écrite sous la forme :

4. La moyenne peut être calculée sur toute une session, un segment de parole ou sur une fenêtre glissante de taille fixe.

$$\hat{x}_{t,d} = \frac{x_{t,d} - \mu_{cvn,d}}{\sqrt{\sigma^2_{cvn,d}}} \tag{1.2}$$

$x_{t,d}$ représente la $d$ème composante du vecteur de paramètres cepstraux x_t à normaliser. La moyenne $\mu_{cvn,d}$ et écart-type $\sigma_{cvn,d}$ sont calculés en utilisant les trames de parole correspondant à un locuteur donné sur une même session [5].

La soustraction du cepstre moyen ainsi que la normalisation CMNV restent jusqu'à ce jour les deux techniques de normalisation de paramètres acoustiques les plus utilisées en raison de leur simplicité et de leur amélioration consistante des performances en RAL.

La normalisation par *feature warping* : La méthode de normalisation *feature warping* est aussi une approche populaire au sein de la communauté de RAL. Le principe étant de faire une transformation non linéaire de la distribution des coefficients cepstraux afin de la faire correspondre à une distribution Gaussienne (comme illustré par la figure 1.5). Cette transformation est appliquée sur une fenêtre glissante à durée courte (3 secondes).

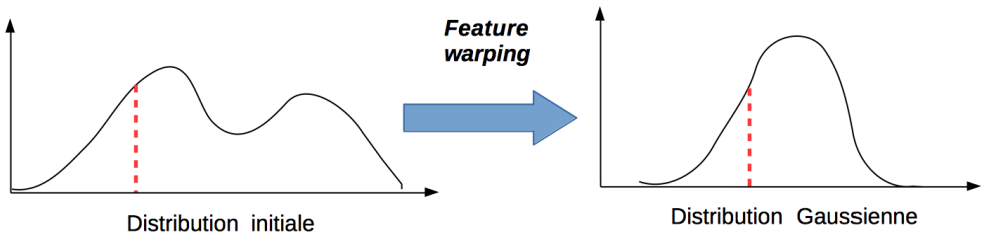

FIGURE 1.5 – *Principe de la normalisation feature warping.*

1.4 Modélisation de locuteurs

Les arguments de Doddington présentées dans la sous-section 1.2.3 stipulent qu'il est difficile d'extraire des paramètres qui caractérisent exclusivement l'identité du locuteur. Il en découle que les modèles construits en utilisant les paramètres acoustiques extraits du signal de parole contiendront inévitablement des informations supplémentaires qui ne sont pas nécessairement pertinentes pour la reconnaissance du locuteur (à savoir la langue, le canal, etc). Dans ce cadre, les modèles produits représentent l'intégralité du segment couvrant l'identité du locuteur ainsi que des informations sur le canal et la session. En conséquence, des techniques de compensation de variabilité session/canal sont développées conjointement à ces modèles pour mieux cibler l'information locuteur et atteindre de bonnes performances.

5. La moyenne et l'écart type peuvent être calculés sur toute une session, un segment de parole ou sur une fenêtre glissante de taille fixe.

Note :

Le terme **empreinte vocale** (*voiceprint*) est jugé trompeur au sein de la communauté de RAL (spécialement dans les applications légales (Bonastre et al., 2003)) vu qu'il insinue l'existence d'une représentation de la parole capable de caractériser d'une manière unique et fiable un locuteur donné (similairement au typage ADN).

Le tableau 1.1 présente une chronologie des modèles utilisées en RAL et qui seront le sujet des sous-sections suivantes.

TABLE 1.1 – *Chronologie des modèles de locuteurs utilisés en RAL.*

1987	VQ (Soong et al., 1987).
2000	GMM-UBM (Reynolds et al., 2000).
2003	Modèle Eigenchannel (Kenny et al., 2003).
2005	Modèle Eigenvoice (Kenny et al., 2005).
2006	GMM-SVM (Campbell et al., 2006a).
2007	Modèle JFA (Kenny et al., 2007).
2011	i-vecteur (Dehak et al., 2011).

1.4.1 Modélisation à base de mélange de Gaussiennes

La quantification vectorielle (VQ), introduite à la fin des années 1980 (Li et Wrench, 1983; Soong et al., 1987; Burton, 1987; Markel et al., 1977), a été l'une des formes les plus primitives de modèles utilisés en RAL. En se basant sur le principe du partitionnement en K-moyennes (Bishop, 2006), l'ensemble des vecteurs de paramètres extraits d'un enregistrement correspondant à un locuteur donné sont partitionnés en un certain nombre de *clusters* disjoints. Les modèles de locuteurs individuels sont représentés par la pile de centroïdes des *clusters* auxquels ils contribuent et sont souvent appelés *codebook*. La classification d'un segment de test est basée sur la minimisation d'une mesure de distorsion donnée par la distance euclidienne moyenne des vecteurs aux *codebook*. La figure 1.6 montre un exemple de partitionnement de paramètres acoustiques en utilisant la VQ.

Cette modélisation s'est avérée restrictive dans sa capacité à représenter les enregistrements à grande variabilité acoustique. Par conséquent, elle a été abandonnée au profit d'une modélisation plus flexible qui intègre un aspect probabiliste basée sur les mélanges de Gaussiennes (GMM : *Gaussian Mixture Model*) (Reynolds, 1992; Reynolds et Rose, 1995). Cette nouvelle modélisation a l'avantage d'utiliser la densité des para-

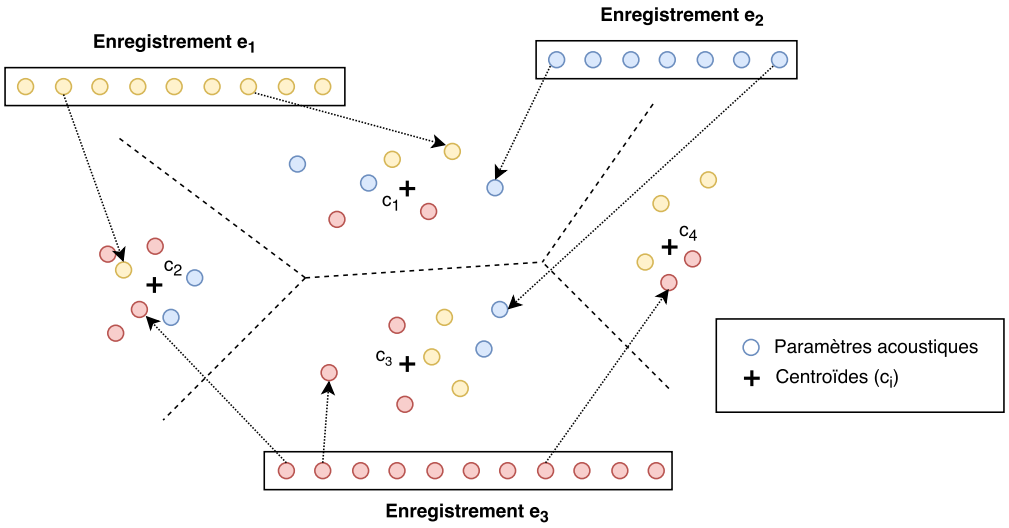

FIGURE 1.6 – *Diagramme de Voronoï montrant un exemple de quantification vectorielle sur les paramètres acoustiques de trois enregistrements.*

mètres acoustiques pour représenter les locuteurs (au lieu d'utiliser des vecteurs prototype dans le cas de la VQ) et permet de prendre en compte la variance des données ainsi que la contribution des paramètres acoustiques à chaque zone de l'espace acoustique sous forme de probabilité. Dans (Reynolds, 1992; Reynolds et Rose, 1995), Reynolds a proposé d'imposer une contrainte de diagonalité sur les matrices de covariance permettant ainsi de réduire le nombre de paramètres du modèle GMM à apprendre [6]. Cette modélisation a permis d'atteindre de bonnes performances et reste utilisée jusqu'à ce jour par la majorité des systèmes de RAL. La figure 1.7 montre la structure de l'espace acoustique en utilisant cette modélisation.

Afin de faire face aux problèmes d'insuffisance de données qui pourraient être posés dans le cas des sessions courtes où les modèles GMM construits deviennent peu fiables, l'utilisation d'un modèle générique appelé UBM (*Universal Background Model*) a été proposée dans (Reynolds et al., 2000). Ce modèle fut un point central dans le développement de la RAL et a permis de construire efficacement des modèles spécifiques aux locuteurs à partir d'un seul modèle générique. Dans ce contexte, le modèle UBM est appris sur un grand nombre de locuteurs et les modèles des locuteurs d'apprentissage et de test sont dérivés par adaptation MAP (*Maximum a posteriori*) (Gauvain et Lee, 1994).

Cette modélisation, généralement appelée GMM-UBM dans la littérature, a permis d'avoir des modèles de locuteurs plus fiables que les GMM appris directement sur les données correspondant à chaque locuteur. En pratique, seules les moyennes du GMM

6. Deux autres types de matrices de covariance ont aussi été testées dans (Reynolds et Rose, 1995) ; (1) ***Grand covariance :*** Une seule matrice de covariance est partagée entre toutes les Gaussiennes d'un modèle de locuteur. (2) ***Global covariance :*** Une seule matrice de covariance partagée entre tous les locuteurs. Cependant, l'utilisation de matrices de covariances nodales (une matrice de covariance différente pour chaque Gaussiennes) a donné les meilleurs résultats.

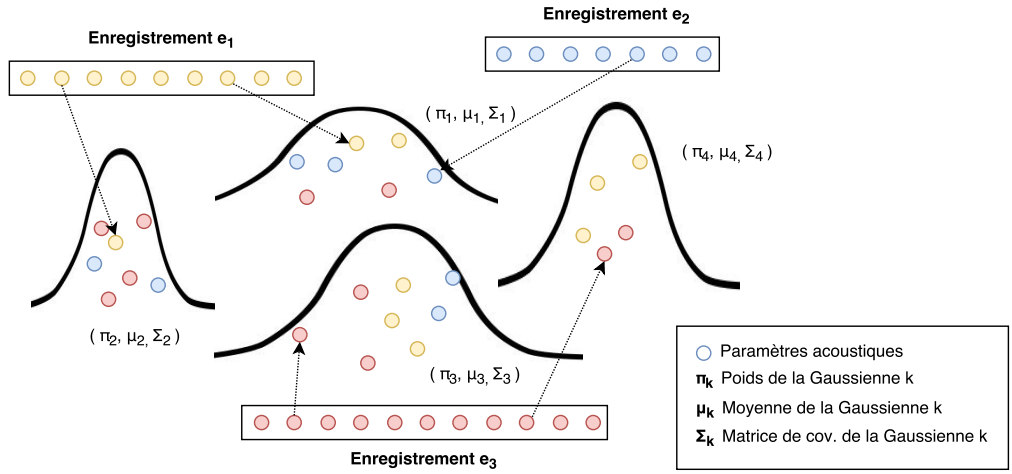

FIGURE 1.7 – *Un mélange de Gaussiennes (GMM) construit en utilisant des paramètres acoustiques issues de plusieurs enregistrements.*

sont adaptées pour la construction de nouveaux modèles. Ce choix est motivé par des résultats empiriques qui indiquent que l'adaptation de poids ou de matrices de covariance peut dégrader les performances (Reynolds et al., 2000; Barras et Gauvain, 2003).

Il convient de noter deux propriétés intéressantes de l'adaptation MAP utilisée dans ce framework :

— À mesure que la quantité de données d'adaptation augmente, le modèle produit par l'adaptation MAP est garanti d'être asymptotiquement équivalent à un GMM entraîné en se basant seulement sur les données spécifiques à un locuteur donné.

— En présence de très peu de données d'adaptation, le modèle produit par l'adaptation MAP est équivalent au modèle UBM, et peut donc toujours être considéré (théoriquement) comme un modèle de locuteur (mais peu fiable).

Plus tard, les méthodes GMM-SVM ont été développées (Campbell et al., 2006a,b) où le GMM correspondant à chaque session est appris via adaptation MAP. Par la suite, les moyennes du GMM résultant sont concaténées pour former un **super-vecteur**. Enfin, un classifieur SVM est utilisé en phase de décision. La figure 1.8 montre le processus de construction de super-vecteurs en utilisant des paramètres acoustiques et en appliquant une adaptation de moyennes sur l'UBM.

Afin de compenser la variabilité session dans le cadre GMM-SVM, un ensemble d'algorithmes ont été développés dont la projection NAP (*Nuisance Attribute Projection*) (Campbell et al., 2006c; Vogt et al., 2008). Cette technique est inspirée des premiers modèles d'analyse factorielle développées par Kenny (Kenny et al., 2003) qui décomposent les super-vecteurs en une composante dépendante du canal et une deuxième composante dépendante du locuteur. L'algorithme NAP construit une matrice de projection qui projette un super-vecteur dans un sous-espace qui réduit la composante nuisible (d'où l'appellation *Nuisance Attribute Projection*). Cette approche permet d'améliorer l'EER de 10% à 16% en termes de gain relatif.

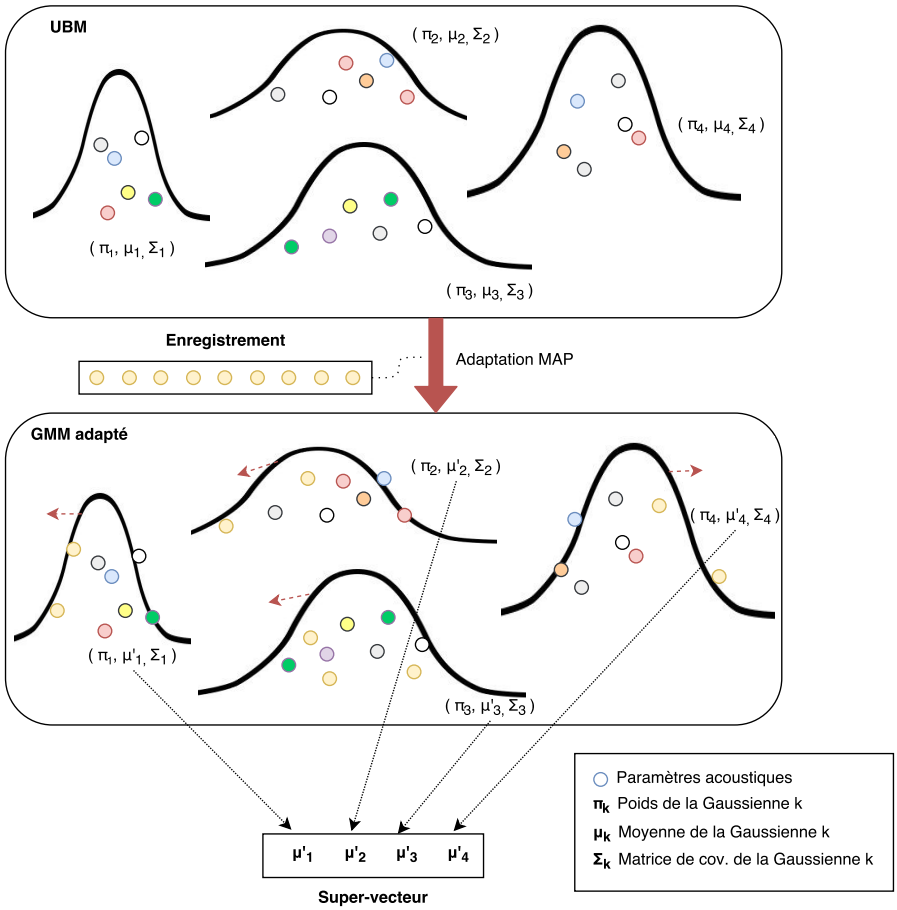

FIGURE 1.8 – *Processus de construction de super-vecteurs en utilisant des paramètres acoustiques d'un enregistrement. L'adaptation des moyennes est utilisée en se basant sur le modèle UBM.*

Une deuxième approche de compensation de la variabilité session développée dans ce contexte est la normalisation WCCN (*Within-class covariance normalization*) (Hatch et al., 2006). Dans cette normalisation, la compensation de la variabilité session est faite par projection dans un espace de même dimension ou de dimension inférieure (en combinant WCCN avec une analyse en composantes principales) visant à minimiser la dispersion intra-locuteurs. Cette normalisation apporte un gain relatif variant entre 10% et 30% en termes d'EER.

1.4.2 Modèles d'analyse factorielle : de l'adaptation MAP à la JFA

Des méthodes à base d'analyse factorielle ont commencé à être développées parallèlement au framework GMM-SVM et furent un point d'avancement majeur dans le domaine de la RAL permettant d'améliorer significativement les performances des systèmes comparés aux modèles GMM-UBM et GMM-SVM (Kenny et Dumouchel, 2004).

Ces modèles traitent le problème de variabilité canal en supposant que le super-

vecteur correspondant à une session peut être décomposé en une composante qui caractérise le locuteur et une autre qui caractérise le canal. Ces deux variabilités résident dans des sous-espaces de dimension faible dans l'espace des super-vecteurs et de nombreux modèles ont été explorés pour la modélisation de chacune de ces variabilités.

Le modèle d'analyse factorielle latente (*Eigen-channel*) (Kenny et al., 2003) a été l'un des modèles d'analyse factorielle les plus réussis vu qu'il a permis de donner des améliorations significatives en termes de performances (une amélioration d'EER par un facteur de deux). Le modèle correspondant s'écrit sous la forme :

$$m_{s,h} = m_0 + Dy_s + Ux_{s,h} \tag{1.3}$$

où :
— m_0 est le super-vecteur issu de l'UBM par concaténation de moyennes (considéré comme moyenne de l'espace acoustique).
— $m_{s,h}$ est le super-vecteur dépendant du canal et de la session.
— U est une matrice de projection de rang faible représentant le sous-espace canal/session.
— $x_{s,h}$ est un vecteur représentant les facteurs canal/session.
— D est une matrice diagonale.
— y_s est un vecteur représentant les facteurs locuteur.

Ce modèle est aussi appelé dans la littérature *Eigen-Channel MAP* vu qu'il étend le modèle MAP ($m_0 + Dy_s$) en rajoutant un composante qui modélise la variabilité session ($Ux_{s,h}$).

Il convient de noter que d'autres modèles ont aussi été développés, à savoir le modèle *Eigen-voice* (Kenny et al., 2005) qui s'intéresse à modéliser l'information correspondant aux locuteurs et le modèle *Joint Factor Analysis (JFA)* (Kenny et al., 2007) qui représente à la fois la composante canal et locuteur en utilisant deux sous-espaces différents.

1.4.3 L'approche i-vecteur

Le paradigme i-vecteur a été proposé dans (Dehak et al., 2011) comme extension des modèles d'analyse factorielle et propose d'apprendre un seul sous espace latent qui contient à la fois la variabilité locuteur et session. Cet espace, appelé espace de **variabilité totale**, a permis d'avoir une représentation à faible dimension qui capture l'ensemble des variabilités acoustiques existant dans un enregistrement donné. Le modèle équivalent peut être représenté par :

$$m_{s,h} = m_0 + T\, w_{s,h}. \tag{1.4}$$

où m_0 représente la moyenne de l'espace acoustique et correspond à l'empilement des moyennes de l'UBM, **T** est une matrice de projection de rang faible et les facteurs correspondants à la variable latente $w_{s,h} \sim \mathcal{N}(0, I)$ sont appelés **facteurs totaux** (*total factors*). Les vecteurs produits par ce modèle sont appelés **i-vecteurs** (pour vecteurs d'**i**dentité)

et cette modélisation peut être interprétée comme une compression appliquée sur la représentation d'une session dans l'espace des super-vecteurs.

Ce modèle suppose que la distribution de $m_{s,h}$ est Gaussienne suivant $m_{s,h} \sim \mathcal{N}(m_0, TT^T)$ et l'entraînement de la matrice **T** peut être effectué en utilisant l'algorithme EM en adoptant la même procédure que le modèle *Eigen-voice* (Kenny et al., 2005) vu qu'une modélisation similaire est utilisée. Le processus d'entraînement de l'espace de variabilité totale et d'estimation d'i-vecteurs sont détaillés dans l'annexe A.

L'introduction de ce paradigme a permis de différer le problème de compensation de la variabilité canal/session à la phase de scoring et a permis de profiter de la structure simple de l'espace de variabilité totale; un espace à dimension réduite et à distribution simple (une distribution Gaussienne (Dehak et al., 2011)). Ceci a ouvert de nouvelles possibilités pour le calcul de similarité entre les modèles et a permis d'exploiter des techniques de compensation des variabilités nuisibles qui étaient difficiles à implémenter dans les paradigmes GMM-UBM ou GMM-SVM. De plus, un gain significatif de performances a été observé comparé à un système basé sur le modèle JFA donnant un gain absolu de 4% en EER (près de 30% de gain relatif en performance) (Dehak et al., 2011).

1.5 Systèmes de RAL à base d'i-vecteurs

Les systèmes de RAL basés sur les i-vecteurs sont devenus le standard au sein de la communauté dans la dernière dizaine d'années. Dans cette section, on s'intéresse exclusivement à ces systèmes. On commence par présenter les principales techniques de normalisation et de compensation de la variabilité session dans cet espace. Par la suite, on donne un aperçu des modèles de scoring développés dans ce framework.

1.5.1 Normalisation d'i-vecteurs

Normalisation de la longueur des i-vecteurs

Une analyse de la longueur[7] des i-vecteurs dans (Garcia-Romero et Espy-Wilson, 2011) a révélé un décalage significatif entre les distributions de longueurs des i-vecteurs d'apprentissage et de test. Afin de corriger ce décalage, un prétraitement couramment utilisée en RAL consiste à normaliser la longueur des i-vecteurs. Cette transformation non-linéaire permet de Gaussianiser la distribution des i-vecteurs et améliorer les performances .

Ce processus divise simplement chaque i-vecteur par sa norme L^2. La forme normalisée d'un i-vecteur w est donnée par :

$$w_{norm} = \frac{1}{||w||} \times w \tag{1.5}$$

7. Le terme *longueur* fait référence à la norme euclidienne L^2.

Normalisation EFR

Dans (Bousquet et al., 2011), Bousquet a proposé d'appliquer une transformation sur les i-vecteurs d'entraînement et de test qui fait intervenir à la fois la normalisation par la longueur et la standardisation de la distribution des i-vecteurs appelée normalisation [8] EFR (*Eigen-Factor Radial*). La première étape est de calculer la moyenne empirique $\overline{w}$ et la matrice de covariance Σ des i-vecteurs d'entraînement. La matrice de covariance Σ est par la suite décomposée en utilisant la diagonalisation :

$$\Sigma = \mathbf{PDP}^T \tag{1.6}$$

où $\mathbf{P}$ est une matrice qui contient les vecteurs propres de Σ, et $\mathbf{D}$ est la version diagonale de Σ. Un i-vecteur d'entraînement w est par la suite transformé en w_{EFR} suivant :

$$w_{EFR} = \frac{\mathbf{D}^{-\frac{1}{2}}\mathbf{P}^T(w-\mu)}{\sqrt{(w-\mu)^T\Sigma^{-1}(w-\mu)}} \tag{1.7}$$

Le numérateur est équivalent à une rotation $\Sigma^{-\frac{1}{2}}(w-\mu)$ et la norme Euclidienne de w_{EFR} est égale à 1. La même transformation est appliquée aux i-vecteurs de test en utilisant les paramètres calculés sur l'ensemble d'entraînement (moyenne μ et matrice de covariance Σ). La figure 1.9 montre les étapes de la transformation. La figure 1.9-(a) représente les données d'entraînement d'origine ; La figure 1.9-(b) montre la rotation appliquée sur l'ensemble d'entraînement initial autours des axes principaux de l'espace de variabilité totale quand $\mathbf{P}^T$ est appliqué ; La figure 1.9-(c) montre l'étape de standardisation des i-vecteurs quand $\mathbf{D}^{-\frac{1}{2}}$ est appliqué ; et finalement, la figure 1.9-(d) montre la projection des i-vecteurs w_{EFR} sur la surface de l'hypersphère unitaire suite à la division par $\sqrt{(w-\mu)^T\Sigma^{-1}(w-\mu)}$.

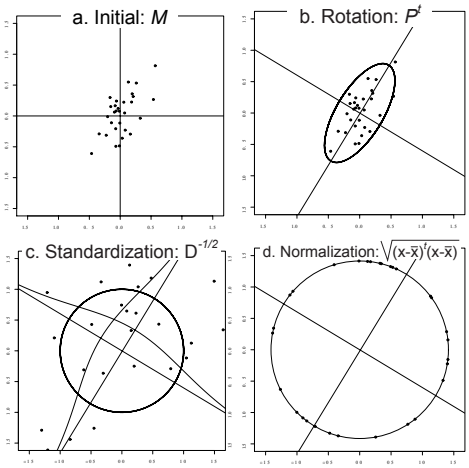

FIGURE 1.9 – *L'effet du conditionnement EFR sur les données (source (Bousquet et al., 2011)).*

8. Le terme *conditionnement* est aussi utilisé dans la littérature.

Une variante de cet algorithme appelée *Spherical Normalization* [9] a aussi été développée. Dans cette technique, la matrice de covariance globale Σ est remplacée par la matrice de covariance intra-locuteur $\mathbf{W}$ (définie dans l'équation 1.8) dans l'étape de diagonalisation (équation 1.6) et lors de la transformation des i-vecteurs (équation 1.7).

$$\mathbf{W} = \frac{1}{n}\sum_{s=1}^{S}\sum_{i=1}^{n_s}(w_{s,i} - \mu_s)(w_{s,i} - \mu_s)^T \tag{1.8}$$

où n_s est le nombre de sessions pour le locuteur s, n est le nombre total de sessions, $w_{s,i}$ sont les i-vecteurs de session du locuteur s, μ_s est la moyenne de tous les i-vecteurs du locuteur s.

1.5.2 Compensation de la variabilité session dans l'espace des i-vecteurs

En substance, l'approche i-vecteur ne fournit qu'une représentation à dimension réduite ($\sim$ 400-800) d'un enregistrement donné et n'effectue pas de compensation de la variabilité session/canal. Ainsi, les méthodes discriminatives qui étaient peu pratiques dans le paradigme GMM-UBM et GMM-SVM (comme l'analyse discriminante linéaire) pourraient être appliquées dans cet espace à dimension réduite. Certaines techniques de normalisation (comme la WCCN et la projection NAP) avaient des racines dans la modélisation super-vecteur et seront discutés dans ce qui suit dans le contexte des i-vecteurs.

Normalisation WCCN

La technique WCCN (*Within-Class Covariance normalization*) a été proposée dans (Hatch et al., 2006) pour améliorer la robustesse dans le cadre de la reconnaissance du locuteur basée sur les SVM. La projection WCCN vise à améliorer les performances en minimisant le taux de fausses alertes (FA) lors de l'apprentissage des SVM. L'espace de projection construit est défini par la racine carrée de l'inverse de la matrice $\mathbf{W}_{WCCN}$ définie par :

$$\mathbf{W}_{WCCN} = \frac{1}{S}S_W = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{n_s}\sum_{i=1}^{n_s}(w_{s,i} - \mu_s)(w_{s,i} - \mu_s)^T \tag{1.9}$$

La matrice de projection $\mathbf{Q}$ est trouvée par la décomposition de Cholesky de l'inverse de la matrice $\mathbf{W}_{WCCN}$ définie par :

$$\mathbf{W}_{WCCN}^{-1} = \mathbf{Q}\mathbf{Q}^T \tag{1.10}$$

9. Les noms **SphNorm** ou normalisation **LW** sont aussi utilisés dans la littérature.

Bien qu'utilisée pour des application de RAL à base de SVM au début, cette méthode a été exploitée pour la compensation de la variabilité session par (Dehak et al., 2011) en calculant la matrice $\mathbf{W}_{WCCN}$ en utilisant les i-vecteurs correspondant à différents locuteurs et en appliquant la transformation $w_{WCCN} = \mathbf{Q} \times w$.

Analyse discriminante linéaire LDA

L'analyse discriminante linéaire (*Linear Discriminant Analysis*) est une technique couramment utilisée en RAL et vise à projeter les données sur un ensemble d'axes orthogonaux qui minimisent la variabilité inter-locuteur et maximisent la variabilité intra-locuteur.

En notant respectivement par S_W et S_B les matrices de dispersion intra- et inter-classe, les valeurs $w^T S_W w$ et $w^T S_B w$ mesurent la dispersion intra- et inter-classe de la version projetée de l'i-vecteur w. L'analyse discriminante linéaire est définie sous forme d'un problème d'optimisation à deux critères et peut être réduit à la maximisation du critère de Rayleigh :

$$\underset{w}{\operatorname{argmax}}\, J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{1.11}$$

En utilisant une grande quantité de données d'entraînement correspondant à des locuteurs enregistrés dans différentes sessions en utilisant différents canaux, la projection LDA peut être utilisée pour la compensation de la variabilité session/canal.

Compensation Radial-NAP

Présenté dans (Campbell et al., 2006c), la technique NAP compense l'effet de la session en soustrayant de chaque vecteur de dimension p sa projection sur le sous-espace principal des r premiers vecteurs propres de la matrice de covariance intra-locuteur $\mathbf{W}$ (équation 1.8), avec $r < p$.

Cet algorithme a été utilisé de la même manière dans l'espace des i-vecteurs sous le nom de *Radial-NAP* (Bousquet et al., 2011). Si on note par $p_{W_r}(w)$ la projection d'un i-vecteur w sur un espace de rang r de la matrice $\mathbf{W}$, un i-vecteur est transformé avec NAP en w_{NAP} suivant :

$$w_{NAP} = \frac{w - p_{W_r}(w)}{||w - p_{W_r}(w)||} \tag{1.12}$$

1.5.3 Scoring dans l'espace des i-vecteurs

Suite à l'introduction du paradigme de la variabilité totale, de nombreuses méthodes ont été proposées pour comparer d'une manière efficace deux i-vecteurs correspondant à deux enregistrements donnés.

La comparaison d'i-vecteurs calcule un quotient de vraisemblances qui teste la validité de deux hypothèses (les hypothèses client et imposteur). Étant donnés deux i-vecteurs w_1 et w_2 à comparer, l'opération de calcul de scores est définie par :

$$score = log \frac{P(w_1, w_2|H_{client})}{P(w_1, w_2|H_{impost})} \tag{1.13}$$

L'hypothèse H_{client} indique que les vecteurs w_1 et w_2 sont issus du même locuteurs et l'hypothèse H_{impost} indique qu'ils correspondent à deux locuteurs différents.

On présente dans ce qui suit les principales distances et modèles discriminatifs de *scoring* développées en RAL.

Distance de cosinus

La mesure de similarité de cosinus (*Cosine Distance*) a été utilisée dans le papier qui était à l'origine du paradigme de variabilité totale (Dehak et al., 2011). Pour cette mesure, le score entre i-vecteur de client w_1 et un i-vecteur de test w_2 est calculé en tant que produit scalaire normalisé :

$$score_{CD}(w_1, w_2) = \frac{w_1 \cdot w_2}{||w_1|| \times ||w_2||} \tag{1.14}$$

Le "." fait référence au produit scalaire entre deux vecteurs.

Cette approche s'est révélée très efficace en RAL, en dépit de sa structure simple, vu qu'elle n'intègre aucune information à priori sur les locuteurs d'apprentissage. Cette distance est généralement utilisée dans la littérature conjointement avec un algorithme de compensation de la variabilité canal/session comme la normalisation WCCN et/ou l'analyse discriminante linéaire.

Distance de Mahalanobis

Cette distance a été introduite dans contexte de la RAL basée sur les i-vecteurs dans (Bousquet et al., 2011). Elle est définie par :

$$score_{Mahal}(w_1, w_2) = -\frac{1}{2}||w_1 - w_2||^2_{\mathbf{W}^{-1}} = -\frac{1}{2}(w_1 - w_2)^T \mathbf{W}^{-1} (w_1 - w_2) \tag{1.15}$$

$\mathbf{W}$ correspond à la matrice de covariance intra-locuteur (définie dans l'équation 1.8).

La distance de Mahalanobis se base sur un argument probabiliste qui stipule que sous l'hypothèse d'homoscédasticité (égalité des covariances de classes locuteur) et la Gaussiannité de la densité $P(\text{locuteur } \mathbf{S}|w_{test})$, le locuteur $\mathbf{S}$ le plus probable peut être obtenu par la solution optimale de Bayes qui minimise $||w_{\mathbf{S}} - w_{test}||^2_{\mathbf{W}^{-1}}$.

Note :

Il est important de ne pas confondre la matrice de covariance intra-locuteur $\mathbf{W}$ utilisée dans la distance de Mahalanobis avec la matrice $\mathbf{W}_{WCCN}$ utilisée dans la normalisation $\mathbf{W}_{WCCN}$.
Si on note par $\mathbf{W}_s$ la matrice de covariance relative à un locuteur s :

$$\mathbf{W}_s = \frac{1}{n_s}\sum_{i=1}^{n_s}(w_{s,i} - \mu_s)(w_{s,i} - \mu_s)^T \tag{1.16}$$

— La matrice $\mathbf{W}_{WCCN}$ est une moyenne sur l'ensemble des locuteurs des matrices de dispersion intra-locuteur S_W :

$$\mathbf{W}_{WCCN} = \frac{1}{S}S_W = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{n_s}\sum_{i=1}^{n_s}(w_{s,i} - \mu_s)(w_{s,i} - \mu_s)^T \tag{1.17}$$

— La matrice $\mathbf{W}$ est une somme des matrices de covariances locuteurs $\mathbf{W}_s$ pondérée par leurs effectifs respectifs et normalisée par le le nombre total de sessions n :

$$\mathbf{W} = \frac{1}{n}\sum_{s=1}^{S} n_s\mathbf{W}_s = \frac{1}{n}\sum_{s=1}^{S}\sum_{i=1}^{n_s}(w_{s,i} - \mu_s)(w_{s,i} - \mu_s)^T \tag{1.18}$$

Modèle deux covariances

Le modèle deux covariances (Two-cov ; *Two-covariance model*) (Brümmer et De Villiers, 2010) est un cas particulier de l'analyse discriminante linéaire probabiliste (PLDA) décrite dans (Prince et Elder, 2007).

Two-cov est un simple modèle linéaire génératif et Gaussien dans lequel un i-vecteur w correspondant à un locuteur **s** peut être décomposé en :

$$w = y_s + \varepsilon \tag{1.19}$$

où le modèle locuteur y_s est un vecteur de même taille que l'i-vecteur w et ε est un bruit Gaussien avec :

$$P(y_s) = \mathcal{N}(\mu, \mathbf{B}) \tag{1.20}$$

$$P(w|y_s) = \mathcal{N}(y_s, \mathbf{W}) \tag{1.21}$$

$\mathcal{N}$ dénote la distribution normale, μ représente la moyenne globale de l'ensemble d'entraînement, $\mathbf{B}$ et $\mathbf{W}$ représentent respectivement les matrices de covariance inter- et intra-locuteur définis par :

$$\mathbf{B} = \sum_{s=1}^{S} \frac{n_s}{n} (\mu_s - \mu)(\mu_s - \mu)^T \tag{1.22}$$

$$\mathbf{W} = \frac{1}{n} \sum_{s=1}^{S} \sum_{i=1}^{n_s} (w_{s,i} - \mu_s)(w_{s,i} - \mu_s)^T \tag{1.23}$$

n_s est le nombre de sessions pour le locuteur **s**, n est le nombre total de sessions, $w_{s,i}$ sont les i-vecteurs de session du locuteur **s**, y_s est la moyenne de tous les i-vecteurs du locuteur **s** et μ représente la moyenne globale de l'ensemble de données d'entraînement. En se basant sur les définitions 1.22 et 1.23, le score correspondant au modèle Two-cov peut être exprimé comme suit :

$$score_{two-cov}(w_1, w_2) = \frac{\int \mathcal{N}(w_1|y, \mathbf{W})\mathcal{N}(w_2|y, \mathbf{W})\mathcal{N}(y|\mu, \mathbf{B})dy}{\prod_{i=1,2} \int \mathcal{N}(w_i|y, \mathbf{W})\mathcal{N}(y|\mu, \mathbf{B})dy} \tag{1.24}$$

où w_1 et w_2 correspondent respectivement à w_{client} et w_{test}. La solution explicite de 1.24 est donnée dans (Brümmer et De Villiers, 2010).

Il convient de citer une extension de ce modèle, appelée modèle 4-covariances, qui a été proposée dans (Bousquet, 2014). Ce modèle suppose que les i-vecteurs clients et les i-vecteurs de test ont des matrices de covariance intra- et inter-locuteurs différents (**W** et **B**) et intègre à la place quatre matrices de covariance intra- et inter-locuteurs ($\mathbf{W}_1$, $\mathbf{W}_2$ et $\mathbf{B}_1$, $\mathbf{B}_2$) dans le calcul des scores.

Analyse Discriminante Linéaire Probabiliste (PLDA)

La PLDA a été introduite dans (Prince et Elder, 2007) pour des applications de reconnaissance faciale. Dans ce contexte, la PLDA se base sur un modèle discriminatif à caractère probabiliste visant à minimiser la variabilité intra-individu et maximiser la variabilité inter-individu.

Dans le contexte de la reconnaissance du locuteur, la PLDA décompose chaque i-vecteur $w_{s,h}$ en une composante dépendante du locuteur, une composante dépendante de la session et une composante résiduelle. La PLDA suppose que les composantes locuteur et session se trouvent dans un sous-espace de rang faible et le modèle correspondant est exprimé sous la forme :

$$w_{s,h} = \mu + \Phi\beta_s + \Gamma\alpha_h + \epsilon_{s,h} \tag{1.25}$$

avec :

— μ un vecteur de dimension N qui correspond à une moyenne globale sur l'espace des i-vecteurs et représente la composante indépendante du locuteur et du canal.
— Φ est une matrice de rang faible $N \times R_{loc}$ ($R_{loc} \ll N$) représentant une base sur le sous-espace des locuteurs (*eigen-voices*).

— Γ est une matrice de rang faible $N \times R_{can}$ ($R_{can} \ll N$) représentant une base sur le sous-espace canal (*eigen-channel*).
— $\beta_s \sim \mathcal{N}(0, I)$ est une variable latente de taille $R_{loc} \times 1$ qui représente la composante locuteur.
— $\alpha_h \sim \mathcal{N}(0, I)$ est une variable latente de taille $R_{can} \times 1$ qui représente la composante canal.
— $\epsilon_{s,h} \sim \mathcal{N}(0, \Sigma)$ est une variable aléatoire de dimension R représentant le bruit résiduel (avec Σ une matrice de covariance pleine).

La version couramment utilisée en RAL de ce modèle (Kenny, 2010) utilise un sous-espace canal à rang plein $R_{can} = N$. Cette contrainte permet de simplifier le modèle 1.25 sans causer une perte significative de performances. L'équation 1.25 devient alors :

$$w_{s,h} = \mu + \Phi\beta_s + \epsilon_{s,h} \tag{1.26}$$

La formulation originale du modèle PLDA repose sur des hypothèses de Gaussianité pour les distributions des facteurs latents et du bruit résiduel (β_s, α_h et $\epsilon_{s,h}$) et porte aussi le nom G-PLDA (*Gaussian PLDA*) dans la littérature (Garcia-Romero et Espy-Wilson, 2011). Dans ce contexte, le score correspondant peut être écrit sous la forme :

$$score_{(PLDA)}(w_1, w_2) = -\begin{bmatrix} w_1 - \mu \\ w_2 - \mu \end{bmatrix}^T \left(N_{client}^{-1} - N_{impost}^{-1} \right) \begin{bmatrix} w_1 - \mu \\ w_2 - \mu \end{bmatrix} \tag{1.27}$$

avec :

$$N_{client} = \begin{bmatrix} \Phi\Phi^T + \Sigma & \Phi\Phi^T \\ \Phi\Phi^T & \Phi\Phi^T + \Sigma \end{bmatrix} \tag{1.28}$$

$$N_{impost} = \begin{bmatrix} \Phi\Phi^T + \Sigma & 0 \\ 0 & \Phi\Phi^T + \Sigma \end{bmatrix} \tag{1.29}$$

Les méta-paramètres de la PLDA (Φ, Σ et μ) utilisés dans ces équations sont estimés par l'algorithme itératif EM (*Expectation Maximization*) décrit dans (Prince et Elder, 2007).

Analyse Discriminante Linéaire Probabiliste HT-PLDA Bien que l'utilisation des hypothèses de Gaussianité pour les distributions des facteurs latents et du bruit résiduel (β_s, α_h et $\epsilon_{s,h}$) fournit une commodité en termes de calcul et de dérivation des équations de scores (Prince et Elder, 2007), Kenny a proposé dans (Kenny, 2010) une alternative à ce modèle qui adopte des hypothèses plus adéquates aux distributions réelles en supposant que les facteurs latents et le bruit résiduel du modèle PLDA suivent une loi t de Student. Ce modèle est appelé *Heavy-Tailed PLDA* ou PLDA à queue lourde, faisant référence aux queues fortes des distributions t de Student permettant une meilleure représentation des valeurs extrêmes.

Plus tard, il a été démontré que lorsque les i-vecteurs sont normalisés par la longueur, le modèle PLDA Gaussien (G-PLDA) devient équivalent à la version *Heavy-Tailed* (Garcia-Romero et Espy-Wilson, 2011). En conséquence, le modèle, *G-PLDA* est

plus couramment utilisé en pratique vu qu'il offre plus de simplicité en termes de calcul par rapport au modèle HT-PLDA.

La phase de *scoring* fournit une valeur unique pour chaque comparaison et ne permet pas d'évaluer les performances globales des systèmes de RAL. Il serait aussi utile de pouvoir comparer, en pratique, les performances de différents systèmes de RAL. On présente dans ce qui suit les mesures et métriques utilisées pour une telle évaluation et on discute les contraintes et conditions nécessaire pour rendre possible la comparaison fiable de systèmes de RAL.

1.6 Évaluation des performances en RAL

L'évaluation des performances des systèmes de RAL est un processus délicat qui dépend de la tâche ciblée (identification, vérification, etc.) et d'un nombre de paramètres qui peuvent influencer sa qualité ou fiabilité. Ces facteurs ont été détaillés dans (Bimbot et Chollet, 1997) qui s'inscrivait dans le cadre du projet européen EAGLES [10] :

— **La qualité du signal de parole :** environnement calme ou bruité en entraînement et en test, enregistrements en studio ou via un canal téléphonique.

— **La variation temporelle *(temporal drift)* :** la voix d'un locuteur varie selon son état physique et émotionnel (variabilité intra-locuteur). Le comportement d'un locuteur peut aussi changer lorsqu'il s'habitue au système.

— **Quantité et variété de la parole :** le niveau des performances d'un système augmente généralement avec la quantité de parole disponible en phase d'entraînement et en test, mais se stabilise après une certaine quantité. En pratique, la collecte d'une grande quantité de données peut être problématique et un compromis devrait alors être trouvé pour assurer des performances acceptables.
Un autre facteur lié à la quantité de parole est la *variété* de la parole : pour une quantité de parole donnée, il est généralement plus efficace de couvrir un large éventail de phénomènes linguistiques. En absence d'une mesure quantitative universelle de la couverture linguistique, une description qualitative du matériel linguistique est la seule façon de préciser cet aspect.

— **Taille de la population de la base de locuteurs et leurs typologie :** en identification du locuteur, la taille de la population a une influence directe sur les performances ; la qualité de la population (proportion hommes/femmes, bonne répartition géographique des locuteurs parlant une même langue) est également un facteur à intégrer.

— **Intention des locuteurs :** la distinction est faite entre les locuteurs coopératifs (qui veulent être reconnus par le système) et les locuteurs non-coopératifs qui modifient leur voix pour ne pas être reconnus, aussi appelés imposteurs (cas de certaines applications judiciaires par exemple). Il est important de faire référence

10. Le projet EAGLES *(Expert Advisory Group on Language Engineering Standards)* (Blasband et al., 1999) visait à proposer des normes, des directives et des recommandations de bonnes pratiques dans les domaines du traitement de langage et de la parole, à savoir la construction de corpus et les méthodes d'évaluation.

à un problème courant dans les systèmes de RAL où l'imposture est représentée par des locuteurs différents du client et existant dans la base de référence. Bien qu'utilisée couramment dans les systèmes modernes, cette modélisation n'est pas réaliste vu qu'un imposteur réel qui tentera d'imiter la voix du locuteur pour lequel il voudra être reconnu, n'existera pas forcément dans la base de référence.

Afin de contrôler certains de ces facteurs, des campagnes d'évaluation ont été lancées visant à produire des jeux de données normalisés (d'entraînement et de test) et fixer les protocoles d'évaluation afin de rendre possible la comparaison objective de différents systèmes de RAL. Les campagnes NIST SRE (*NIST Speaker Recognition Evaluation*) s'inscrivent dans ce contexte et fournissent un jeu de données divisé en un ensemble de données d'entraînement et un autre ensemble de test et fournissent un nombre de tâches correspondant à des conditions différentes en termes de genre, canal, langue ou autre pour permettre de tester les performances des systèmes dans différentes configurations et correspondre à des scénarios réalistes (comparaison homme / homme, homme/femme, téléphone/microphone, court/long, ..). Cependant, un nombre suffisant de comparaisons doit être fourni pour une mesure d'évaluation statistiquement significative (Doddington et al., 2000). Pour un jeu de données et une tâche donnés, les systèmes évalués à l'aide d'un coût spécifique ou d'un critère d'erreur peuvent être comparés.

On présente dans ce qui suit les types d'erreurs rencontrées dans les systèmes de vérification du locuteur avant de présenter les mesures de performance communes utilisées dans les campagnes d'évaluation NIST SRE.

1.6.1 Types d'erreurs

Dans le cas d'un système de vérification du locuteur, deux types d'erreurs peuvent être observées :

— **Fausses acceptations** (FA *False acceptance*) : le cas où le système accepte le locuteur alors que celui-ci n'est pas la personne qu'il prétend être.
— **Faux rejets** (FR *False rejects*) : le cas où le système refuse l'accès à un locuteur alors qu'il correspond bien à l'identité proclamée.

Les taux d'erreurs correspondants ; FAR (taux de fausses acceptations) et FRR (taux de faux rejets) sont définis comme suit :

$$\text{FAR} = \frac{\#\ \text{FA}}{\#\ \text{comparaisons imposteurs}} \tag{1.30}$$

$$\text{FRR} = \frac{\#\ \text{FR}}{\#\ \text{comparaisons clients}} \tag{1.31}$$

Il est possible de faire le lien entre cette terminologie et celle d'autres types génériques d'erreurs rencontrées dans les problèmes de reconnaissance de formes à deux

classes. Si la classe client est supposée *positive* et la classe imposteur est supposée *négative*, on obtient une matrice de confusion comprenant quatre éléments en considérant les classes prédites et réelles. Le tableau 1.2 donne un résumé de ces erreurs.

TABLE 1.2 – *Matrice de confusion dans le cas d'un classifieur binaire.*

		Classe réelle	
		Positive	*Négative*
Classe prédite	*Positive*	Vrai positif (VP)	Fausse acceptation (FA)
	Négative	Faux rejet (FR)	Vrai négatif (VN)

Les systèmes de vérification des locuteurs génèrent un score sous forme d'un scalaire qui décrit la similarité entre deux modèles de locuteurs. Dans ce contexte, une valeur élevée de score reflète une grande similarité entre les deux modèles et une valeur faible indique une grande différence entre les deux modèles. Pour pouvoir prendre une décision, un seuil de décision τ doit être fixé comme illustré dans la figure 1.10. Un seuil de valeur faible résulte en une augmentation des fausses acceptations (FA) alors qu'une valeur élevée donnera beaucoup de faux rejets (FR). Dans un contexte pratique, le réglage de ce seuil dépend de l'application ciblée et du niveau de sécurité désiré. Pour les applications de haute sécurité, un seuil élevé devrait être fixé de façon à minimiser les erreurs FA. Cependant, pour une grande commodité, le seuil devrait avoir une valeur faible.

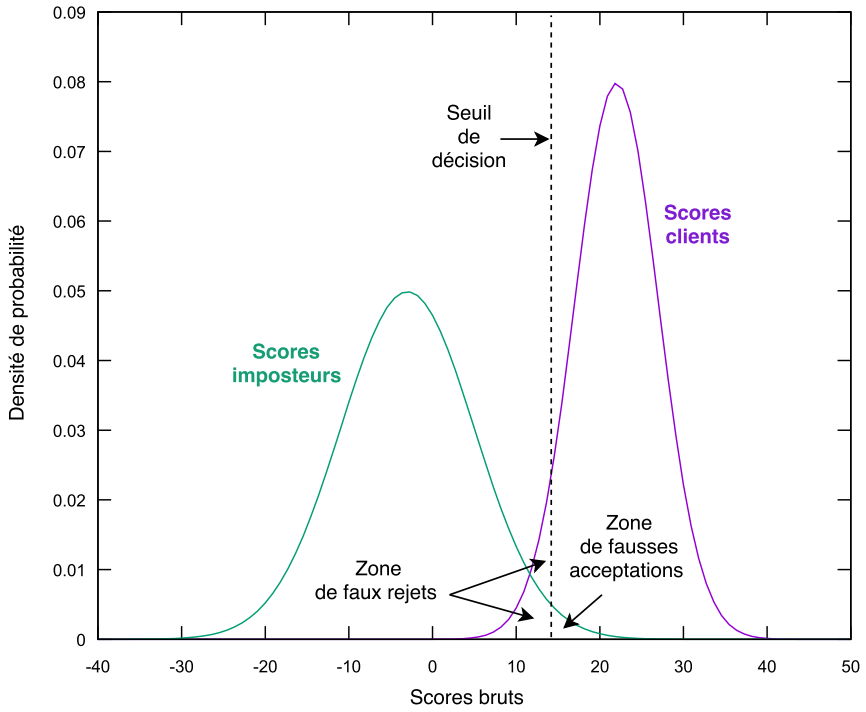

FIGURE 1.10 – *Processus de décision à base de seuil sur les scores.*

1.6.2 Taux d'égale erreur (EER)

Le taux d'égale erreur (EER ; *Equal Error Rate*) est l'une des mesures les plus populaires en vérification de locuteurs vu qu'elle permet de comparer deux systèmes en se basant sur une seule mesure. Elle est définie comme le point opératoire où les valeurs FAR et FRR deviennent presque égales. Cette configuration est atteinte en faisant varier le seuil de décision τ jusqu'à ce que les deux zones correspondant aux fausses acceptations et aux faux rejets dans la figure 1.10 deviennent égales.

Il convient de noter que l'EER ne constitue pas nécessairement un point opératoire optimal dans les applications réelles qui peuvent nécessiter des niveaux de sécurité élevés.

1.6.3 Fonction de coût de détection (DCF : *Detection Cost Function*)

Le taux d'égale erreur (EER) ne fait pas de distinction entre les fausses acceptations (FA) et les faux rejets (FR) dans le sens où aucune mesure de coût n'est introduite pour pénaliser les erreurs de type FA ou FR ce qui en fait une mesure de performance pas très réaliste. La mesure DCF a été introduite par NIST dans (Martin et Przybocki, 2000) introduisant des pénalités sur les erreurs FA et FR. La probabilité à priori de rencontrer un locuteur client est également fournie. Le DCF est calculé sur toute la gamme des seuils de décision comme suit :

$$DCF(\tau) = C_{FR} P_{FR}(\tau) P_{client} + C_{FA} P_{FA}(\tau)(1 - P_{client}). \tag{1.32}$$

avec :

— C_{FR} = Coût d'un faux rejet (erreur FR)
— C_{FA} = Coût d'une fausse acceptation (erreur FA)
— $P_{FR}(\tau) = Pr(\text{Erreur } FR \mid \text{locuteur ciblé, seuil} = \tau)$
— $P_{FA}(\tau) = Pr(\text{Erreur FA} \mid \text{locuteur non-cible, seuil} = \tau)$
— P_{client} = Probabilité préalable d'observer un locuteur cible.

Une mesure communément utilisée est le MinDCF qui définit la valeur minimale de DCF qui peut être obtenue en changeant le seuil τ :

$$MinDCF = \min_{\tau} [C_{FR} P_{FR}(\tau) P_{client} + C_{FA} P_{FA}(\tau)(1 - P_{client})] \tag{1.33}$$

Il est important de noter que la MinDCF n'est pas un taux d'erreur au sens propre du terme et son interprétation n'est pas simple. Plus la valeur de MinDCF est faible, meilleure sont les performances du système. Cependant, la valeur exacte du MinDCF ne peut être utilisée que pour comparer d'autres systèmes évalués en utilisant les mêmes comparaisons et calculés en utilisant les mêmes coûts. Généralement, la tendance du EER d'un système de RAL suit celle du DCF. Une discussion approfondie sur la relation entre EER et DCF peut être trouvée dans (Brümmer, 2010).

1.6.4 Courbe DET (Detection Error Tradeoff)

Lorsqu'il existe un compromis entre différents types d'erreurs (FA/FR), l'utilisation d'une seule mesure de performances peut s'avérer insuffisante pour représenter les capacités d'un système. En effet, les performances d'un système de vérification de locuteur peuvent être étudiées sur plusieurs points opératoires (*operating points*) et seraient mieux représentées par une courbe de performances. Traditionnellement, la courbe ROC (*Receiver operating characteristic* ou caractéristique de fonctionnement du récepteur) a été utilisée à cette fin pour les problèmes de décision binaire et le taux de fausses acceptations (*false alarms rate*) est tracé sur l'axe horizontal, tandis que le taux de détections correctes (*correct detection rate*) est tracé sur l'axe vertical (Egan, 1975; Swets, 1964).

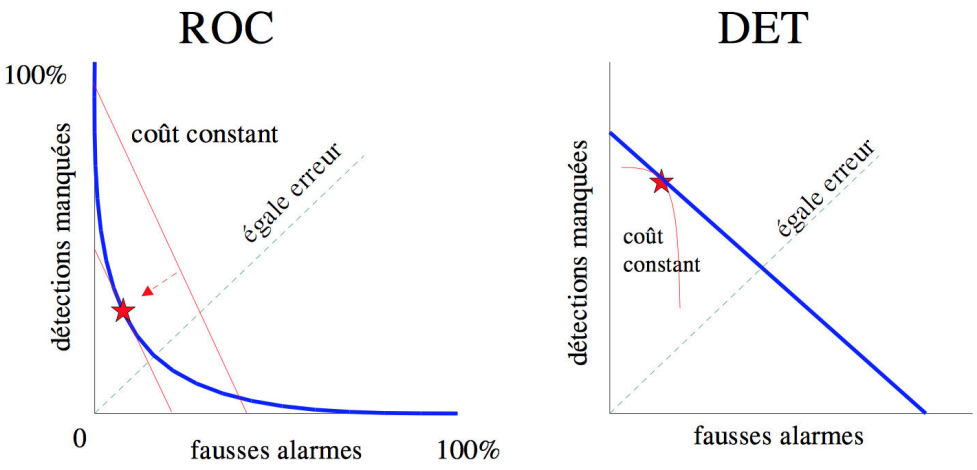

FIGURE 1.11 – *Courbes ROC et DET pour une tâche de vérification du locuteur (source : (Barras, 2016)).*

Il a été établi qu'une variante de ces courbes, appelée courbe DET (*Detection Error Tradeoff* ou courbe de détection de compromis) serait plus adaptée aux applications de vérification de locuteur (Martin et al., 1997) affichant deux types d'erreurs sur les deux axes. Dans ces courbes, le taux de fausses acceptations (*False positive rate*) est tracé sur l'axe horizontal trandis que le taux de faux rejets (*miss detection date*) est tracé sur l'axe vertical. En comparant la figure 1.11 (a) et la figure 1.11 (b), la relation entre une courbe ROC et une courbe DET devient évidente. Dans ces figures, l'emplacement du EER et du MinDCF tels que définis dans NIST SRE 2008 (minDCF'08) sont affichés.

Conclusion

Ce chapitre a un rôle introductif au domaine de la reconnaissance automatique du locuteur. Les tâches et applications liées à la RAL ont d'abord été introduites et la structure d'un système de RAL indépendant du texte a été présentée. Les trois composantes

qui constituent ces systèmes ont par la suite été développées ; un aperçu a été fait des paramètres acoustiques utilisées en RAL. L'évolution de la modélisation de locuteurs en RAL a été détaillée et l'accent a été mis sur le framework i-vecteur. Après, les algorithmes de normalisation et de compensation de la variabilité session utilisés dans le framework i-vecteur ont été cités. Enfin, les principaux modèles de scoring ont été introduits ainsi que les procédures d'évaluation utilisées dans les tâches de RAL dans les évaluations NIST SRE.

Chapitre 2

Variabilités et nuisances en RAL

Sommaire

Introduction

Ce chapitre introduit les variabilités et nuisances qui peuvent rendre la tâche de reconnaissance du locuteur difficile à accomplir. On commence par présenter les défis rencontrés en RAL ainsi que les enjeux dans les applications réelles. Par la suite, on se concentre sur les variabilités nuisibles et on détaille les sources de variabilité qui peuvent affecter les performances du système, à savoir les variabilités liées aux locuteurs, les variabilités de haut niveau reliées à l'interlocuteur (homme ou machine) et les variabilités liées au support technologique et aux perturbations externes. Après, on s'intéresse à la modélisation de trois types de nuisances acoustiques ; les bruits additif et convolutif ainsi que la réverbération. Enfin, on aborde les problèmes de reproductibilité et de modélisation des variabilités nuisibles.

2.1 Les défis en reconnaissance du locuteur

Les premiers travaux en télécommunication ont mis en évidence l'effet que peuvent avoir les perturbations relatives à l'environnement ainsi que le canal de transmission sur un signal analogique. Les technologies vocales ne sortent pas de ce contexte vu qu'elles ont trouvé leurs racines dans les applications ciblant les réseaux téléphoniques. Depuis le début des années 70, ces technologies ont attiré l'attention d'un nombre d'acteurs ; à savoir les projets de nature militaire comme ARPA (Klatt, 1977) ainsi que les industriels opérant dans les domaines de téléphonie mobile, de systèmes embarqués dans l'automobile ainsi que les systèmes de sécurité (exp. Siemens (Vollert, 1992), Texas Instruments (Netsch et Doddington, 1992), NEC (Yamada et Hattori, 1999), etc). Ce cadre a donné lieu à un effort considérable d'étude et d'analyse pour prendre en compte les contraintes réelles qui peuvent affecter les performances d'un système de RAL et assurer des taux de reconnaissance et de fiabilité acceptables dépendant de l'application ciblée. Ce genre d'étude nécessite la collecte de données d'apprentissage et de test qui reflètent des scénarios d'utilisation réalistes et qui permettent d'avoir une évaluation des performances aussi objective que possible des systèmes de RAL et des résultats statistiquement significatifs. On discute ces points dans ce qui suit et on présente certains défis relatifs aux données utilisées pour la construction et l'évaluation d'un système de RAL.

2.1.1 Défis technologiques

Bien que les débuts des réseaux téléphoniques étaient confinés aux téléphones à cadran rotatif dans les résidences et lieux publics, la technologie des téléphones portables / smartphones a dominé le marché mondial des télécommunications dans les vingt dernières années et la diversité des scénarios de téléphonie s'est considérablement développée rendant le problème de la variabilité canal encore plus difficile à aborder (cependant, il est important de préciser que la qualité du signal de la parole s'est améliorée). De plus, pratiquement tous les téléphones portables ont une option "mains libres"

qui permet l'interaction vocale à distance du microphone et rend le signal de parole encore plus vulnérable aux perturbations externes (bruit de fond et réverbération) et aux mouvement du microphone et sa distance au locuteur. De plus, les systèmes de reconnaissance automatique du locuteur interviennent dans des domaines commerciaux où le bruit de fond est difficile, voir impossible, à contrôler (exp : systèmes d'authentification vocale via téléphone pour des applications bancaires, etc).

2.1.2 Défis relatifs au locuteur

Vu que la parole est une biométrie comportementale [1], la nature caractéristique du signal de parole invite à considérer d'autres facteurs relatifs à la performance du locuteur (état psychologique, maladie, etc) et aux conditions d'enregistrement (distance au microphone, etc). Le niveau de coopération des locuteurs peut aussi venir à l'encontre de l'obtention de résultats fiables. Ceci constitue un grand centre d'intérêt dans le contexte des applications judiciaires (support de témoignage et identification judiciaire) où les intentions du locuteur devraient être prises en considération (déguisement et changement de voix) (Bonastre et al., 2003).

2.1.3 Défis de collecte de données pour l'analyse des variabilités

La RAL, une tâche dirigée par les données :

Comme c'est le cas de tout système de reconnaissance basé sur l'apprentissage automatique, la conception des systèmes de RAL se base sur un ensemble de données d'entraînement [2]. Idéalement, utiliser un jeu de données qui correspond aux conditions d'évaluation permettrait de construire un système adapté aux conditions d'intérêt et à assurer de bonnes performances. Ceci n'est généralement pas le cas dans un contexte réel vu la variété des conditions d'évaluation et la présence/absence de certaines variabilités acoustiques dépendant du cadre d'utilisation. Afin de concevoir un système de RAL qui couvre un large éventail de conditions acoustiques, une grande quantité de données d'entraînement devrait (théoriquement) être utilisées ce qui n'est pas faisable en réalité. Pour parer au problème de la quantité de données d'entraînement, il est possible d'analyser l'effet individuel des nuisances acoustiques qui peuvent affecter un système de RAL et implémenter des stratégies de compensation qui permettraient d'améliorer, globalement, sa performance. Dans cette optique, la modélisation correcte de nuisances et l'exploitation des données disponibles deviennent décisifs pour l'amélioration des performances du système. Il est important de préciser que les approches d'augmentation de données (*multi-style training*) ont aussi pris du succès dans ce contexte, et plus particulièrement quand les DNN (*Deep Neural Network*) sont utilisés.

1. La biométrie vocale se base sur la parole, un signal généré consciemment par une personne donnée, alors que d'autres types de biométries (comme les empreintes digitales et l'ADN) se basent sur des mesures directes de caractéristiques physiologiques.
2. Le terme *data-driven* est couramment utilisé pour qualifier ces systèmes.

Un problème d'interprétation et de généralisation de résultats est aussi posé en pratique et un niveau de significativité doit être garanti pour permettre de faire des analyses correctes.

Significativité des résultats en RAL :

La conception de bases d'évaluation qui reflètent les performances réelles d'un système de RAL est une tâche difficile. Une considération primordiale dans la conception de telles bases est la significativité statistique des résultats correspondants. En substance, si on veut concevoir un système utile pour des fins de recherche ou pour des fins applicatives, la signification statistique doit être à l'égard de multiples et diverses conditions choisies. Par exemple, si un seul locuteur cible est utilisé, des résultats statistiquement significatifs peuvent être obtenus, mais malheureusement, ils ne seront valables que pour ce locuteur en particulier. De toute évidence, si la signification statistique d'une population en général est requise, un nombre suffisant de locuteurs doit être utilisé. Cependant, peu importe le nombre de locuteurs échantillonnés, il faut faire preuve de prudence pour s'assurer que la population d'échantillon représente la population d'intérêt. La même problématique se pose dans le cadre des nuisances acoustiques où l'existence de différentes conditions d'apprentissage (présence du bruit additif, réverbération et différents types de canaux) permettrait de concevoir des systèmes plus robustes qui couvrent une multitude de conditions acoustiques et une connaissance à priori des conditions d'évaluation et du contexte d'utilisation pourrait aider significativement lors de la conception d'un système de RAL.

La "règle des 30" a été proposée par Doddington dans le cadre de la significativité des résultats données par un système de RAL (Doddington et al., 2000), c'est une règle générale concernant le nombre de comparaisons requises pour avoir une estimation d'erreur qui corresponde à un intervalle de confiance élevé. On cite un passage de (Doddington et al., 2000) :

> ***The rule of 30 :*** In determining the required size of a corpus, a helpful rule is what might be called "the rule of 30". This comes directly from the binomial distribution, assuming independent trials. Here is the rule :
>
> To be 90% confident that the true error rate is within ±30% of the observed error rate, there must be at least 30 errors.
>
> This confidence interval and proportional bound on error rate are reasonable values that yield reasonable requirements for the size of an evaluation corpus. The rule may be applied by considering the performance goals or expectations for the evaluation. For example, suppose that the performance goals are 1% miss and 0.1% false alarm. Thirty errors at 1% miss implies a total of 3000 true speaker trials and 30 errors at 0.1% false alarm implies a total of 30,000 impostor trials.

La transposition de ce principe est difficile en réalité vu qu'une grande quantité de données est nécessaire pour tenir compte de la variété de locuteurs, langues et conditions d'enregistrement. Ce besoin a été clairement souligné par l'AFCP[3] dans le communiqué du 3 Décembre 2002 concernant "l'identification des individus par leur voix" (afcp2002, 2002) :

> L'évaluation scientifique de la fiabilité des empreintes digitales et génétiques repose notamment sur l'existence de bases de données expérimentales de dimension très importante. Dans le domaine vocal, les bases de données disponibles actuellement ne comportent pas un nombre suffisant de locuteurs, de langues, de conditions d'enregistrement pour évaluer la fiabilité des méthodes existantes dans un contexte d'authentification vocale.

Il est important de rappeler que deux problèmes sont posés en pratique concernant la quantité de données : les données d'entraînement d'un coté, et les données de test d'un autre. Ces deux contraintes doivent être prises en compte afin de construire un système de reconnaissance du locuteur qui tient compte d'une variété de conditions acoustiques et qui fournit une décision significative en sortie.

Données réelles vs. données artificielles :

Afin de pouvoir faire de telles études et de pouvoir tester les méthodes de compensation de variabilités développées, deux choix de bases de données se présentent :

1. ***L'utilisation de données collectées dans diverses conditions acoustiques :*** dans ce contexte, les données utilisées reflètent des conditions d'enregistrement différentes (différents microphones, différents bruits de fond, etc). Certains types de données, appelés **bases stéréophoniques** dans la littérature, proposent deux versions de la même session enregistrées en utilisant deux microphones et permettent d'avoir à la fois une version propre du signal (capturée par un microphone rapproché de la source) et une version distordue (capturée par un microphone distant de la source). En utilisant ces bases, il est possible d'étudier l'effet des nuisances acoustiques sur un nombre donné de locuteurs et de généraliser les résultats trouvés sur le reste de la population.
2. ***L'utilisation de données artificielles :*** dans ce contexte, seules les données de bonnes qualité sont utilisées et la distorsion de signaux est faite artificiellement pour la génération de données corrompues. Cette approche exploite la reproductibilité des nuisances acoustiques au domaine temporel. Ceci permet d'éviter une dépendance des conditions d'apprentissage et de générer des segments correspondants à diverses conditions. Cependant, cette approche peut perdre en justesse ce qu'elle gagne en praticité vu qu'elle peut donner une simulation non-réaliste de la nuisance. Un exemple connu est l'effet Lombard qui est généralement ignoré en pratique lors du bruitage artificiel de données.

3. AFCP : Association Francophone de la Communication Parlée

Une discussion détaillée de ces deux types de données ainsi que leurs implications peut être trouvée dans (Ribas et al., 2016).

Bases de données utilisées en RAL :

Historiquement, les premiers projets qui s'intéressaient aux technologies vocales ciblaient des applications de reconnaissance automatique de la parole et la collecte d'un grand nombre de bases de données a été mise en place (exp. SWITCHBOARD (Godfrey et al., 1992), etc) pour étudier la variation des canaux de communication et dispositifs d'acquisition (réseau téléphonique/microphone/..) sur la qualité d'un signal de parole et sur les performances des systèmes de reconnaissance. Des variantes de ces bases ont été mises en place pour les tâche de vérification de locuteurs et de nombreuses campagnes d'évaluation NIST SRE ont mis l'accent sur ces problématiques poussant la communauté à étudier les variabilités dues au canal, à la session ainsi qu'au bruit additif qui a été rajouté lors de l'évaluation SRE 2012 (Martin et Przybocki, 2000; nist2008eval, 2008; nist2010eval, 2010; nist2012eval, 2012). Cependant, la dernière évaluation NIST SRE 2016 (nist2016eval, 2016) était principalement focalisée sur le problème du *mismatch* de la langue entre les conditions d'entraînement et de test.

La collection de données RedDots a aussi été récemment publiée (Lee et al., 2015). Cette base s'intéresse aux segments à contenu phonétique variable et fournit des segments de test de courtes durées.

En reconnaissance du locuteur dépendante du texte, la base RSR2015 a été publiée (Larcher et al., 2012) permettant d'étudier l'influence du contenu lexical dans des segments de courtes durées.

Un autre exemple est la base SITW (*Speakers In The Wild*) (McLaren et al., 2016) qui propose une multitude de conditions d'enregistrements (différents bruits de fond, codecs, conditions de réverbération, etc) et qui sera utilisée dans la partie expérimentale de notre travail de thèse.

2.1.4 Risques et enjeux

La présence de variabilités nuisibles peut dégrader significativement les performances des systèmes de RAL et amener à des résultats non-crédibles (Bonastre et al., 2003). Un exemple récent d'utilisation abusive de la reconnaissance automatique des locuteurs a été observé lors de l'affaire judiciaire Américaine portant sur George Zimmerman, qui a été accusé d'avoir tiré sur Trayvon Martin lors d'une dispute (Colby et al., 2012). Dans cette affaire, un appel d'urgence 911 a capturé un cri d'aide entendu en arrière-plan. L'équipe de défense a prétendu que c'était Zimmerman qui hurlait alors qu'il était, soi-disant, attaqué par Trayvon Martin (qui a été tué). Alternativement, les procureurs ont soutenu que c'était la victime non armée qui criait. Les parents des deux parties ont témoigné que la voix entendue sur l'appel 911 appartenait à leur propre fils.

Certains experts en identification judiciaire ont tenté d'utiliser des techniques semi-automatiques pour comparer le cri original et un cri simulé obtenu de Zimmerman. Les experts du bureau fédéral d'investigation (FBI) et de l'Institut national des normes et de la technologie (NIST) ont affirmé que ces méthodes ne sont pas fiables. Une brève analyse de la technologie du cri et de la reconnaissance des locuteurs a confirmé les limites de la technologie actuelle (Hansen et Shokouhi, 2013).

Les applications judiciaires ne sont pas les seules à souffrir en cas d'utilisation abusive ou mal-guidée des systèmes de RAL. En effet, les applications industrielles de contrôle d'authentification peuvent aussi souffrir de ces problèmes. Dans la littérature, le terme *spoofing* est utilisé pour faire référence à une tentative d'abuser le système. Ceci peut se faire avec un déguisement de voix (imitation) (Lau et al., 2004) ou bien avec une transformation artificielle de voix (Matrouf et al., 2006; Jin et al., 2008). Des approches d'anti-*spoofing* ont été mises en place par la communauté pour arriver à détecter ces attaques (Wu et al., 2012; Alegre et al., 2013; Wu et al., 2015). Des solutions alternatives sont aussi utilisées en pratique telles que la limitation du nombre d'essais d'authentification afin de renforcer le protocole de sécurité mis en place.

2.2 Une classification des variabilités et nuisances en reconnaissance automatique du locuteur

Afin de mieux mettre en place les techniques qui permettent d'améliorer les performances des systèmes de RAL, on s'intéresse dans ce qui suit aux variabilités nuisibles qui peuvent influencer négativement les performances d'un système de RAL. Il est possible de faire une classification de ces variabilités suivant leur source (le locuteur lui même, l'environnement externe, etc) et leurs niveau (niveau acoustique, linguistique, etc). La classification présentée dans (Hansen et Hasan, 2015) utilise ces critères et sera adoptée dans cette section pour mettre en évidence l'effet que peut avoir chaque nuisance sur la qualité d'un signal de parole et sur les variabilités qui y résident. Ceux-ci peuvent être répartis sur la base de trois grandes classes comme affiché dans la figure 2.1. Dans cette classification, les variabilités nuisibles sont divisés suivant leurs source et une distinction est faite entre les variabilités qui proviennent du locuteur, les variabilités de haut niveau en relation avec la langue et les variabilités qui proviennent de la technologie d'acquisition de transmission et de l'environnement externe. On détaille dans ce qui suit ces trois classes et on pointe vers les efforts de recherches en relation avec ces variabilités.

2.2.1 Variabilités reliées aux locuteurs

L'information relative aux locuteurs est le pilier principal des systèmes de RAL et leurs principal centre d'intérêt vu qu'elle fait office d'outil caractéristique d'un coté (pour capturer les propriétés acoustiques spécifiques de chaque locuteur) et d'un outil discriminatif d'un autre (pour pouvoir distinguer différents locuteurs). Ceci rend

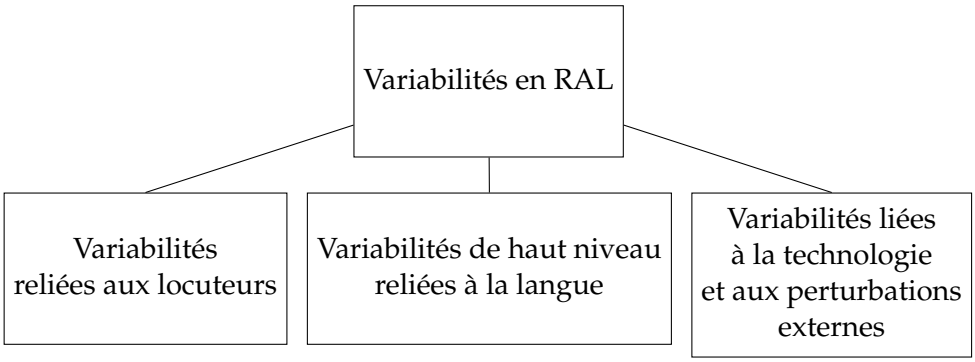

FIGURE 2.1 – *Classification des variabilités et nuisances en RAL.*

la capture des paramètres acoustiques et des paramètres qui caractérisent le style de parole à court ou à long terme décisive pour assurer de bonnes performances. Cependant, de nombreux travaux de recherche ont montré que la même personne ne dit pas exactement les mêmes mots exactement de la même façon à chaque fois ; ce qui est connu comme changement de style ou variabilité intra-locuteur (Eckert et Rickford, 2001). Cette variabilité couvre les modifications de parole causées par différents styles de parole : qualité de la voix, taux d'articulation, stress, pitch etc. En effet, une dégradation croissante des performances a été observée au fur et à mesure que le temps qui sépare la session d'apprentissage de la session de test augmente (Furui, 1974).

Mis à part ces effets, le locuteur peut aussi être influencé par le bruit de fond présent lors de l'enregistrement qui va affecter sa prononciation à mesure que le niveau de bruit augmente. Cette déformation, appelée généralement *effet Lombard* se manifeste à travers une hyper-articulation, causant une accentuation des voyelles et une déformation des consonnes (Junqua et Anglade, 1990; Junqua, 1993). Des travaux de recherche ont montré que cet effet peut affecter significativement les performances des systèmes de reconnaissance du locuteur (Hanson et Applebaum, 1990; Hansen et Varadarajan, 2009).

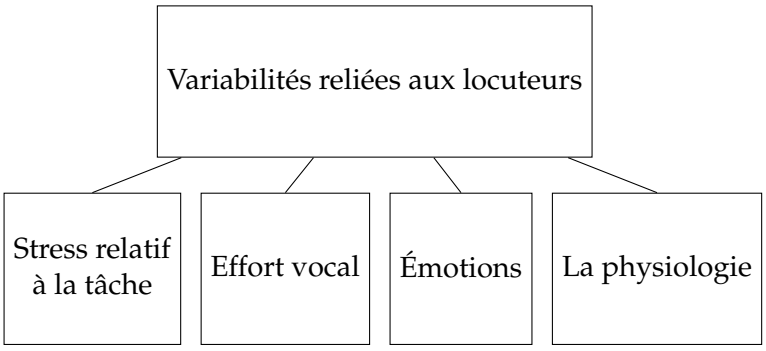

FIGURE 2.2 – *Classification des variabilités relatives aux locuteurs.*

Ces variabilités peuvent être divisées en 4 sous-classes :

1. ***Le stress relatif à la tâche :*** un changement de style de parole peut être observé

en cas d'exécution d'une tâche en parlant ; comme le fait de conduire une voiture ou l'utilisation une entrée vocale à mains libres (réglage d'usine, intervention d'urgence / pompiers, etc.). Ces effets peuvent inclure le stress de la tâche cognitive aussi bien que physique (Hansen, 1996).

2. ***Effort vocal :*** il se peut qu'une personne altère sa production de la parole par rapport à la phonation normale, ce qui provoque un discours chuchoté (Fan et Hansen, 2011; Zhang et Hansen, 2011), doux, fort ou crié (Zhang et Hansen, 2007; Hansen et al., 2017). Un changement dans la mécanique de production de la parole est aussi observé en présence de bruit, permettant au locuteur de parler plus efficacement (effet Lombard) (Hansen et Varadarajan, 2009). Ou la personne chante au lieu de parler (Mehrabani et Hansen, 2013).
3. ***Émotion :*** la personne communique son état émotionnel en parlant (exp. Colère, tristesse, bonheur, etc.) (Hansen et al., 2000).
4. ***Physiologique :*** le sujet a une maladie ou est intoxiqué ou sous l'influence d'un médicament. Cela peut aussi inclure le vieillissement (Lanitis, 2009).

2.2.2 Variabilités de haut niveau reliées à l'interlocuteur

Il est connu que le style de parole d'une personne peut changer suivant son interlocuteur. Ces variabilités sont dites "de haut-niveau" et reflètent différents scénarios concernant l'interaction vocale avec une autre personne ou un système technologique. Elles peuvent aussi être issues de différences par rapport à la langue ou au dialecte parlés. De cette définition, il est possible de diviser ces variabilités en deux classes comme affiché dans la figure 2.3 :

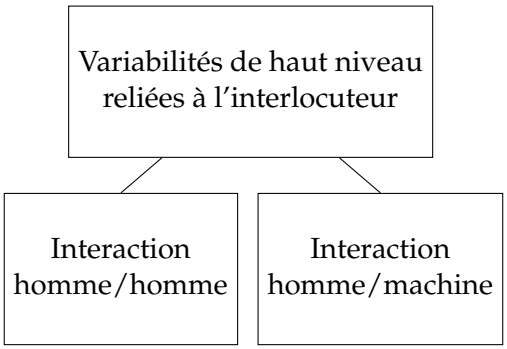

FIGURE 2.3 – *Classification des variabilités de haut niveau relatives à l'interlocuteur.*

1. ***Interaction homme/homme :*** Le style de parole d'une personne diffère selon son interlocuteur ; une ou plusieurs personnes en interaction ou une personne parlant et s'adressant à une audience. Ces changement peuvent se manifester sous forme d'un changement en langue ou en dialecte selon l'interlocteur. Le type de parole et la taille du public peuvent aussi être considérés ; lue / conversation (par affichage visuel ou par écouteur) / parole déguisée ou spontanée.

2. ***Interaction homme/machine :*** le style de parole d'une personne peut changer quand elle dirige son discours vers un téléphone portable, smartphone, téléphone fixe ou ordinateur ; interaction avec un système vocal / entrée vocale sur un ordinateur / interaction avec un système de traitement automatique de parole ou un système de dialogues).

2.2.3 Variabilités liées à la technologie et aux perturbations externes

Il est possible d'identifier une troisième source de variabilités qui dépend de la technologie utilisée lors de l'acquisition du signal de parole ainsi que de l'environnement externe. Cette classe peut être divisée en trois sous-classes :

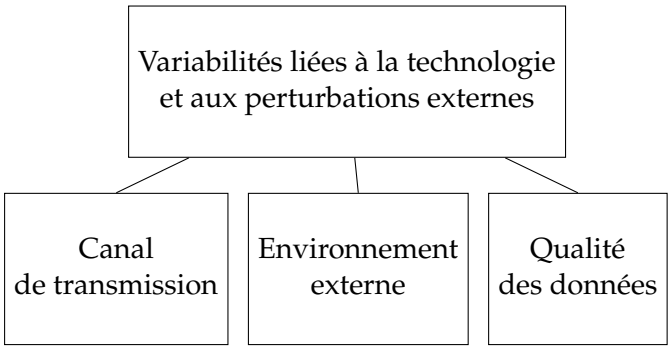

FIGURE 2.4 – *Classification des variabilités liées à la technologie et aux perturbations externes.*

Les sources de variabilité de la technologie ou de l'externalité : elles incluent comment et où l'audio est capté :

1. ***Canal de transmission :*** électromécanique, combiné (cellulaire, sans fil et ligne fixe) (Reynolds et al., 1995; Kenny et al., 2007; Auckenthaler et al., 2000) microphone.
2. ***Environnement externe :*** bruit de fond (Rose et al., 1994) (stationnaire, impulsif, variable dans le temps, etc.), acoustique de pièce (Jin et al., 2007), réverbération (Greenberg et al., 2010) et microphone éloigné.
3. ***Qualité des données :*** la durée de l'enregistrement, le taux d'échantillonnage, le codec audio utilisé, la méthode de compression. (Kuitert et Boves, 1997; Besacier et al., 2000) ont étudié l'effet du codage de la parole utilisé dans le réseau téléphonique mobile GSM sur les performances de vérification du locuteur.

De nombreuses méthodes ont aussi été mises en place pour atténuer l'effet des variabilités liées aux perturbations externes et au canal de transmission en se basant sur des modèles qui reflètent leurs effets sur le signal, les paramètres ou sur les paramètres du systèmes.

Dans ce qui suit, on s'intéresse à cette classe de variabilités. On présente les modèles utilisées pour trois types de nuisances ; le bruit additif, la distorsion canal ainsi que la réverbération. L'effet de ces nuisances sur le signal de parole est aussi discuté.

2.3 Les bruits convolutifs et additifs

2.3.1 Définition et sources

La parole est l'information la plus véhiculée dans les systèmes de télécommunication et un grand effort de recherche en traitement automatique de la parole a ciblé les réseaux téléphoniques. Une dégradation en qualité et en intelligibilité peut être observée dans de tels réseaux en cause de différentes sources de distorsion telles que la limitation de la bande utile, les bruits de fond, couramment supposés comme des *bruits additifs*, le bruit causé par le combiné téléphonique et le canal de transmission, couramment supposé comme des *bruits convolutifs*. En effet, le réseau téléphonique commuté (PSTN : *public switched telephone network*) est limité à une bande de fréquences variant entre $300Hz \sim 3300Hz$ suivant les spécifications G.711 [4] (G.711specs, 2017), appelée bande vocale (*voiceband*), qui est beaucoup plus restrictive que la plage auditive humaine (de $20Hz \sim 20kHz$). Par conséquent, la modélisation et la compréhension des effets de tels bruits pourrait s'avérer critique pour améliorer les performances des systèmes de RAL. Il est important de comprendre les difficultés que les bruits convolutifs et additifs présentent aux algorithmes de paramétrisation et de modélisation actuels avant d'aborder le problème de la reconnaissance automatique du locuteur dans les milieux bruités. En pratique, l'effet de ces deux bruits peut être caractérisé d'une manière approximative en utilisant un modèle d'environnement acoustique.

2.3.2 Modélisation

Le modèle d'environnement acoustique a été présenté dans (Acero, 1990; Hansen, 1996) et englobe les perturbations relatives au locuteur, à son environnement et au canal de transmission qui peuvent affecter le signal de parole (figure 2.5).

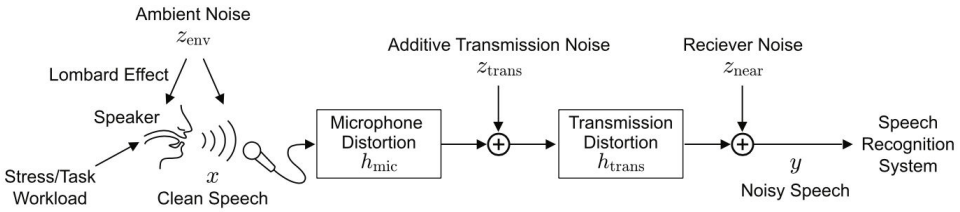

FIGURE 2.5 – *Sources de bruits additifs et convolutifs pouvant affecter la parole (source : (Hansen, 1996)).*

En effet, le signal de parole $x(t)$ peut être influencé par l'état psychologique du locuteur (stress, émotions, ..), le bruit ambiant (bruit de fond) ainsi que les distorsions

4. Les spécifications G.711 présentent des normes de codage et de compression utilisés en téléphonie (les lois de quantification μ et A, le codage PMC, etc.). Ils ont été initialement présentés par Bell Systems dans les années 1970 et ont été officiellement normalisés par l'Union internationale des télécommunications (UIT) en 1988. Aujourd'hui, G.711 est couramment utilisé dans Voice over Internet Protocol (VoIP), également connu sous le nom de téléphonie Internet.

de canal (soit en raison du microphone ou du réseau avec un bruit de canal ajouté). Un deuxième bruit est aussi possible à l'extrémité proche du système de reconnaissance (appelé bruit de réception).

Ce modèle peut être décomposé en trois parties :

— ***Le bruit additif :*** noté $z_{env}(\tau)$ dans la figure 2.1, causé par l'environnement d'enregistrement (exp. bruit de circulation, bruit de pluie, etc).

— ***Le locuteur :*** qui est à la fois générateur du signal vocal et sujet d'un nombre de variabilités (variabilité intra-locuteur).

— ***Le microphone / canal de transmission :*** Le signal combiné de parole et de bruit est capturé et filtré par la réponse impulsionnelle du microphone (notée $h_{mic}(\tau)$) qui peut être une autre grande source de distorsion. La transmission peut également ajouter du bruit, représenté par $z_{trans}(\tau)$ et $h_{trans}(\tau)$, bien que l'on s'attend à ce que sa valeur soit petite. On s'attend également à ce que le bruit au niveau du récepteur $z_{near}(\tau)$ soit minimal.

De ce fait, l'effet ce cet environnement acoustique peut être résumé par l'équation 2.1 :

$$y(\tau) = \left[\left(\left\{ \left[x(\tau) \middle| \begin{array}{c} \text{Effet Lombard} \\ \text{Stress/Émotions} \\ (...) \end{array} \right]_{z_{env}(\tau)} + z_{env}(\tau) \right\} * h_{mic}(\tau) + z_{canal}(\tau) \right) * h_{canal}(\tau) \right] + z_{proche}(\tau) \tag{2.1}$$

où :

— $y(\tau)$: représente la parole bruitée reçue par le système de reconnaissance du locuteur

— $x(\tau)$: représente la parole originale "propre"

— $z_{env}(\tau)$: représente le bruit additif causé par l'environnement

— $h_{mic}(\tau)$: représente l'effet convolutif du microphone

— $z_{canal}(\tau)$: représente un bruit additif pouvant exister dans le canal de transmission

— $h_{canal}(\tau)$: représente l'effet convolutif du canal de transmission

— $z_{proche}(\tau)$: représente le bruit additif existant du coté proche du système de reconnaissance du locuteur

Vu que les valeurs de $z_{trans}(\tau)$, $h_{trans}(\tau)$ et $z_{proche}(\tau)$ sont supposés être faibles, l'équation 2.1 peut être simplifiée en combinant les différentes sources de bruit additif et convolutif en un seul bruit additif $z(\tau)$ et un bruit de canal convolutif $h(\tau)$. En faisant ces simplifications, un nouveau modèle standard de l'environnement acoustique bruité, souvent utilisé dans la littérature (Acero, 1990; Moreno, 1996; Gong, 1995) surgit et peut être écrit dans le domaine temporel comme :

$$y(\tau) = x(\tau) * h(\tau) + z(\tau) \tag{2.2}$$

où :

— $y(\tau)$: représente la parole bruitée

— $x(\tau)$: représente la parole propre

— $h(\tau)$: représente l'effet du canal de transmission

Il est important de noter que $z(\tau)$ est une version filtrée du bruit ambiant réel $z_{env}(\tau)$ par le microphone et le canal (et donc dépendant de $h(\tau)$). Mais pour des raisons de simplicité, ils sont généralement supposés indépendants dans la littérature.

Caractérisation du bruit additif : Le rapport signal sur bruit SNR_{dB} est utilisé pour caractériser l'intensité du bruit et défini par :

$$SNR_{dB} = 10\ log_{10}\left(\frac{P_{signal}}{P_{bruit}}\right) \tag{2.3}$$

où :

— P_{signal} représente la puissance moyenne du signal.

— P_{noise} représente la puissance moyenne du bruit.

Plusieurs algorithmes ont été proposés dans la littérature pour le calcul de SNR_{dB} (Hirsch, 1993; Nemer et al., 1999; Plapous et al., 2006).

2.3.3 Effet sur le signal et les paramètres acoustiques

L'effet du bruit d'environnement est additif dans le domaine temporel et spectral et il est possible de voir son effet sur un signal de parole dans la figure 2.6. Vu la grande variabilité de la nature des bruits possibles, son effet peut varier significativement entre les différentes bandes de fréquences et une partie importante du signal d'origine peut se trouver masquée, voir détruite, dans le cas des niveaux de SNR faibles. Le bruit additif affecte aussi d'une manière non-linéaire la distribution des paramètres acoustiques (Moreno et al., 1998) et peut détériorer d'une manière significative les performances des systèmes de RAL.

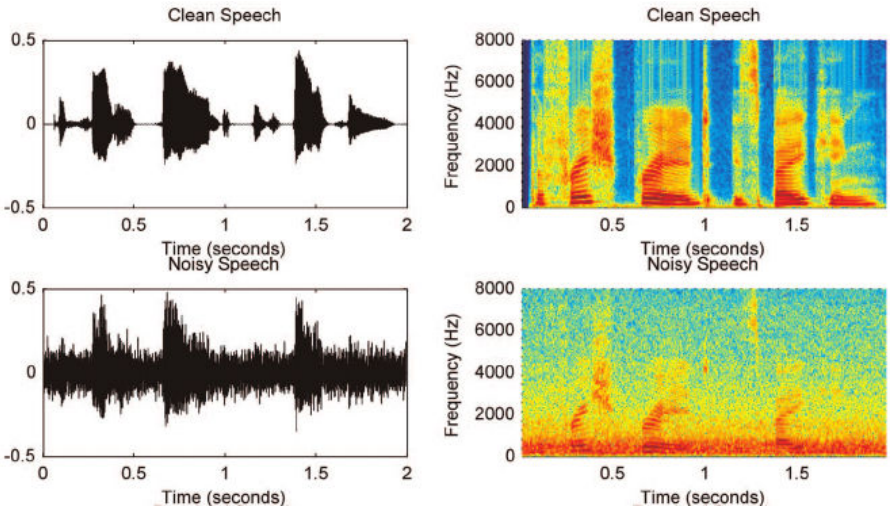

FIGURE 2.6 – *Signal et spectrogramme d'un signal de parole propre (en haut) et le signal et spectrogramme de la version bruitée correspondante à 0dB (en bas) (source : (Ye et al., 2013)).*

2.4 La réverbération

2.4.1 Définition et sources

Lorsque la parole ou tout autre signal acoustique est produit dans une pièce, il suit plusieurs chemins de la source au récepteur. Les réflexions causées de ce signal sont appelés réverbération. Une partie de l'énergie du signal qui atteint le récepteur est transmise directement dans l'air, tandis que le reste est réfléchi par une ou plusieurs surfaces dans la pièce avant la réception (figure 2.7).

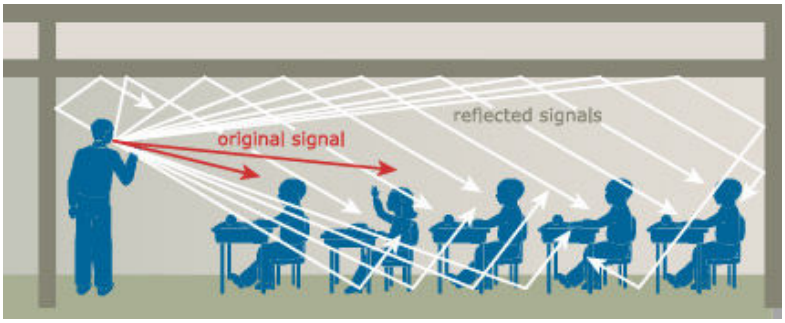

FIGURE 2.7 – *Exemple de réverbération.*

2.4.2 Modélisation

Le processus de réverbération peut être modélisé comme une convolution du signal de parole par la réponse impulsionnelle d'une pièce :

$$y(t) = \sum_{\tau=0}^{T} r(\tau)x(t-\tau) = r(t) * x(t) \tag{2.4}$$

où :

— $x(t)$ représente un signal de parole.
— $r(t)$ représente la réponse impulsionnelle d'une pièce.
— $y(t)$ représente un signal réverbéré.

Ce modèle néglige cependant de nombreux effets. Il ignore le fait que les caractéristiques de la transmission du son de la source au récepteur peuvent changer de façon significative car les positions et les orientations de la source et du récepteur varient (Mourjopoulos, 1985) (exp : ouverture ou fermeture des portes, déplacement des gens, etc). A cause de ces mouvements, r peut varier au cours du temps rendant le problème plus complexe. Malgré ces inconvénients, le modèle convolutif est suffisamment précis pour être utile pour simuler une grande partie des effets de réverbération de pièces et reste un modèle très utilisé dans des applications de dé-réverbération.

2.4.3 Effet sur le signal et les paramètres acoustiques

La figure 2.8 illustre la structure d'une réponse impulsionnelle en salle typique. Les caractéristiques importantes de la réponse impulsionnelle sont la réponse directe initiale, les échos précoces discrets et la queue réverbérante, qui est semblable à un bruit décroissant exponentiellement. Le caractère de bruit de la queue est une conséquence de la sommation d'un grand nombre de chemins de transmission ayant des magnitudes et des phases différentes. La queue décroît de façon exponentielle car, à chaque réflexion, une partie de l'énergie acoustique est absorbée par la surface réfléchissante.

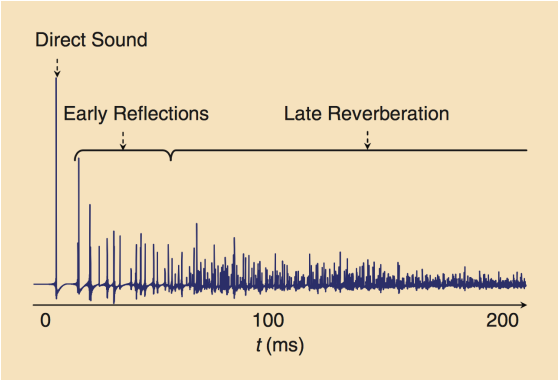

FIGURE 2.8 – *Un exemple de réponse impulsionnelle d'une pièce en présence de réverbération (source (Yoshioka et al., 2012)).*

Dans une représentation temps-fréquence, l'effet de réverbération est apparenté à un étalement du signal le long de la dimension temporelle, comme illustré dans la figure 2.9.

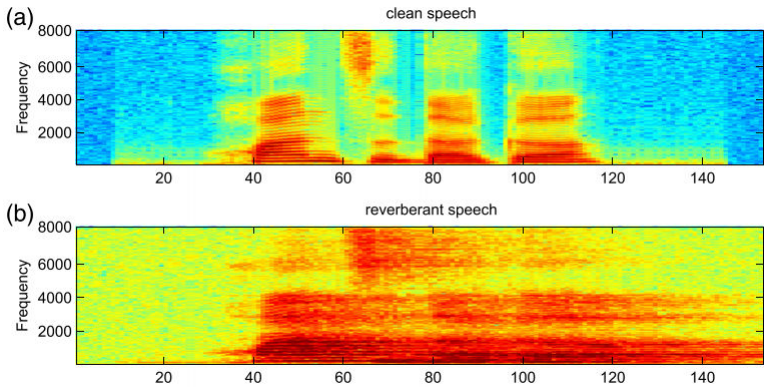

FIGURE 2.9 – *(A) Spectrogramme correspondant à un segment de parole propre; (B) Spectrogramme correspondant à un segment de parole en présence de réverbération avec $T_{60} = 0.61s$ à une distance $d = 3m$ (source : (Gao et al., 2016)).*

2.4.4 Caractérisation de la réverbération

Quelques paramètres sont utilisés pour caractériser la réverbération, à savoir la géométrie de la pièce, son volume, le coefficient d'absorption acoustique des matériaux présents, etc. Une mesure très utilisée est aussi *le temps de réverbération*, notée généralement T_{60}, définit comme étant le temps nécessaire pour que l'énergie du signal d'origine diminue de 60dB (à un millième de sa valeur initiale) (Ratnam et al., 2003; Gaubitch et al., 2012).

Il est aussi important de faire la distinction entre les notions de réverbération et d'écho dans la pratique. En effet, la réverbération est perçue lorsque l'onde sonore réfléchie atteint l'oreille en moins de 0.1 seconde après la génération de l'onde sonore originale alors que l'écho qualifie le cas inverse, pour lequel la distinction entre le signal d'origine et sa version répétée devient possible (l'écho est aussi définit en pratique comme le bouclage sur la source dans les systèmes à mains libres).

2.5 Modélisation et reproductibilité de variabilités nuisibles

2.5.1 Modélisation de nuisances acoustiques

Pour certaines nuisances acoustiques tels que le bruit additif et la réverbération, il est possible de formuler l'effet des nuisances sous forme mathématique dans le domaine temporel. Cette propriété peut s'avérer utile en pratique pour la conception de techniques de compensation de variabilités adaptées à la nature de la nuisance ciblée. Ceci a été validé dans de nombreux travaux comme les techniques de compensation de modèles à base de VTS (*Vector Taylor Series*) qui arrivent à intégrer ces modèles efficacement dans un système d'extraction d'i-vecteurs, le rendant plus robuste en présence de bruit (Lei et al., 2013).

Cependant, une telle modélisation ne peut pas être envisageable pour d'autres types de nuisances tel que la variabilité des durées qui se réduit à une présence/absence de données. L'effet Lombard est aussi un exemple de variabilités difficiles à analyser et à modéliser, étant une distorsion non-linéaire qui dépend du locuteur, du type de bruit et de son niveau SNR (Hansen et Varadarajan, 2009). Cependant, des modèles empiriques ont été proposés dans la littérature pour décrire son effet dans le domaine cepstral sous forme de termes additifs ou multiplicatifs (Chen, 1987; Hansen et Cairns, 1995; Paul, 1987).

2.5.2 Reproductibilité de nuisances acoustiques

La reproductibilité est une deuxième propriété importante à identifier lors du traitement des nuisances acoustiques et fait référence à la possibilité de générer des données affectées par une nuisance donnée en se basant sur des données propres.

La reproductibilité peut se faire en se basant sur un modèle de distorsion dans le domaine temporel, comme c'est le cas du bruit additif et de la réverbération, mais peut aussi se faire en se basant seulement sur l'effet de la nuisance ciblée sur les données ; la variabilité des durées est un exemple ce ces nuisances où la reproduction se réduit à une suppression de données.

L'injection de nuisances acoustiques dans des données propres est une procédure très populaire dans le contexte de la RAL, et a même été utilisée dans l'évaluation NIST SRE 2012 (nist2012eval, 2012), vu qu'elle permet de mieux étudier leurs effets au niveau des paramètres, des modèles et des scores et de mieux les compenser en pratique. Dans la dernière décennie, cette procédure a beaucoup été instrumentalisée dans les architectures basées sur les réseaux de neurones profonds (DNN) où le but est d'apprendre une régression entre les versions corrompues et propres du signal (Kolbœk et al., 2016) ou des paramètres acoustiques (Du et al., 2015; Tan et al., 2016). Dans ce contexte, il devient difficile de collecter une quantité suffisante de données stéréophoniques (propres/bruitées) pour pouvoir apprendre les modèles DNN et la génération artificielle de données corrompues devient la seule solution possible pour surmonter ce problème.

Conclusion

Dans ce chapitre, on a commencé par discuter les défis rencontrés en reconnaissance automatique du locuteur. Par la suite, on a exposé les variabilités et sources de nuisances qui peuvent rendre difficile la tâche de reconnaissance. Après, on s'est concentré sur les nuisances reproductibles et on a présenté les modèles mathématiques utilisés dans le cas des bruits additifs et convolutifs et celui de la réverbération dans le domaine temporel. On a aussi illustré l'effet des nuisances sur le signal ainsi que sur le spectrogramme du signal de parole.

Chapitre 3

Traitement de nuisances en RAL

Sommaire

Introduction

La robustesse des systèmes de RAL est liée à leurs capacité de maintenir de bonnes performances quand la qualité de la parole en entrée est dégradée (réverbération, bruit additif, distorsion canal, etc). Afin de faire face à ces nuisances acoustiques, de nombreux algorithmes ont été développés durant la dernière trentaine d'années opérant sur différents niveaux : au niveau du signal, des paramètres acoustiques, des modèles ou en phase de scoring. Avec le développement des modèles d'analyse factorielle, qui ont permis de réduire significativement les taux d'erreurs dans des conditions propres, de plus en plus d'attention a été portée par la communauté de RAL à la compensation des nuisances au niveau des modèles et à l'intégration des effets de ces nuisances en phase de scoring. On présente dans ce chapitre un aperçu des approches de compensation de variabilités nuisibles en RAL tout en se concentrant sur les algorithmes en relation directe avec le framework i-vecteur.

3.1 Extraction robuste de paramètres

Les paramètres MFCC et PLP sont généralement considérés comme un standard en traitement automatique de la parole et ont fait leurs preuves dans plusieurs domaines dont la reconnaissance automatique du locuteur. Bien que ces paramètres, couplés avec un système i-vecteur/PLDA, permettent d'atteindre des taux d'erreurs faibles dans des conditions propres, ces performances peuvent être considérablement affectées en présence de distorsions acoustiques tel que le bruit additif, la réverbération ou les distorsions causées par le canal. Ceci a motivé la mise en place de techniques de paramétrisation plus sophistiquées afin de mieux capturer les informations pertinentes à la reconnaissance du locuteur dans des conditions difficiles.

3.1.1 Paramètres MHEC

Les paramètres MHEC (*Mean Hilbert Envelope Coefficients*) ont été présentés dans le contexte de la RAL dans (Sadjadi et al., 2012). Ces paramètres s'inspirent du mécanisme de perception humaine et utilisent un banc de filtres Gammatone couplé avec un estimateur de l'enveloppe d'Hilbert. Le diagramme de flux de données des paramètres MHEC est détaillé dans la figure 3.1.

Dans cette paramétrisation, le banc de filtres Gammatone permet de simuler d'une manière précise l'effet du filtrage auditif, qui se déroule le long de la membrane basilaire dans la cochlée (Patterson et al., 1992). La transformée de Hilbert constitue une bonne mesure du premier front d'onde, comme suggéré par un modèle de calcul pour *l'effet de précédence* (Martin, 1997) [1]. Les paramètres MHEC permettent d'estimer d'une

1. **L'effet de précédence :** Lorsqu'un son est suivi d'un autre son séparé par un délai suffisamment court (sous le seuil d'écho de l'auditeur), les auditeurs perçoivent une seule image auditive fusionnée ; Son emplacement spatial perçu est dominé par l'emplacement du premier son arrivant (le premier front

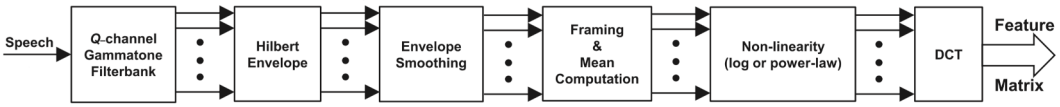

FIGURE 3.1 – *Flux de données pour la paramétrisation MHEC (source : (Sadjadi et Hansen, 2015)). **(1)** Utilisation d'un banc de filtres Gammatone **(2)** Estimation de l'enveloppe de Hilbert **(3)** Lissage d'enveloppe **(4)** Création de trames et calcul de moyenne **(5)** Non-linéarité (log ou loi de puissance) **(6)** DCT.*

manière plus robuste le spectre de la parole dans des conditions bruitées et surpassent d'une manière consistante les performances données par les paramètres MFCC dans des conditions bruitées. Un gain de 17% à 39% en termes de EER est observé en utilisant cette paramétrisation comparée à un système à base de paramètres MFCC sur les données NIST SRE 2010 (Sadjadi et al., 2012).

3.1.2 Paramètres PNCC

Les paramètres PNCC (*Power-Normalized Cepstral Coefficients*) ont été proposés pour la reconnaissance robuste du locuteur dans le contexte du bruit additif et de la réverbération (Ambikairajah et al., 2012; McLaren et al., 2013). Ces paramètres sont basés sur un modèle auditif et se distinguent de la paramétrisation MFCC en quelques points :

— L'utilisation d'un banc de filtres Gammatone.
— L'utilisation d'un traitement "à moyen-terme" d'une durée de 50 à 120 ms pour analyser la dégradation causée par l'environnement.
— L'utilisation d'un algorithme de suppression du bruit basé sur le filtrage asymétrique qui supprime l'excitation de fond pour chaque trame et composante de fréquence.
— Utilisation d'un module qui réalise le masquage temporel.
— La non-linéarité basée sur le logarithme est remplacée par une loi de puissance $(\bullet)^{1/15}$.

Le diagramme de flux de données des paramètres PNCC est détaillé dans la figure 3.2. Malgré la complexité de cette technique de paramétrisation, de faibles gains ont été observés dans (Ambikairajah et al., 2012) sur les données de NIST SRE 2010 ne dépassant pas les 5% d'amélioration relative en EER.

3.1.3 Paramétrisation basée sur la factorisation spectrale

Les techniques de factorisation matricielle sont couramment utilisées en traitement du signal, à savoir la décomposition en valeurs propres et en valeurs singulières, per-

d'onde).

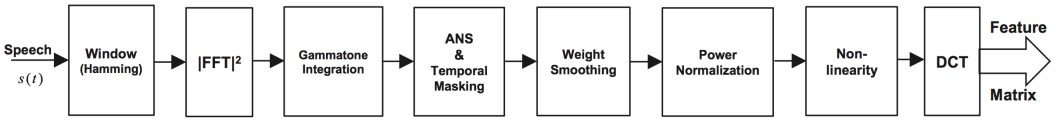

FIGURE 3.2 – *Flux de données pour la paramétrisation PNCC (source : (Sadjadi et Hansen, 2015)). **(1)** Application de la fenêtre de Hamming **(2)** Calcul de $|FFT|^2$ **(3)** Utilisation d'un banc de filtres Gammatone **(4)** Suppression asymétrique de bruit (utilisation de la soustraction cepstrale) et masquage temporel **(5)** Lissage de poids **(6)** Normalisation de puissance **(7)** Non-linéarité (loi de puissance) **(8)** DCT.*

mettent de représenter efficacement les données observées en utilisant un nombre limité d'atomes[2].

Ce principe est implémenté dans les techniques de factorisation spectrale. Le spectrogramme d'un signal de parole est utilisé comme matrice de données et un ensemble de dictionnaires est appris en utilisant un algorithme de décomposition matricielle sur une fenêtre glissante ($\sim$ 25 trames consécutives). Les dictionnaires appris pour un ensemble de locuteurs de référence sont par la suite utilisés comme extracteurs de paramètres en phase d'évaluation. L'algorithme NMF (*Non-negative matrix factorization*) a pris du succès dans ce contexte, permettant d'écrire un spectrogramme $\mathbf{D}$ de taille $(M \times N)$ sous la forme (figure 3.3) :

$$\mathbf{D} \approx \mathbf{FG} \tag{3.1}$$

$\mathbf{F}$ et $\mathbf{G}$ sont des matrices non-négatives de taille $(M \times K)$ et $(K \times N)$. La matrice $\mathbf{D}$ contient N vecteurs réels de dimension M, la matrice d'activations $\mathbf{G}$ contient les vecteurs correspondants dans un espace de dimension $K < M$ et la matrice de passage $\mathbf{F}$ contient les vecteurs de base.

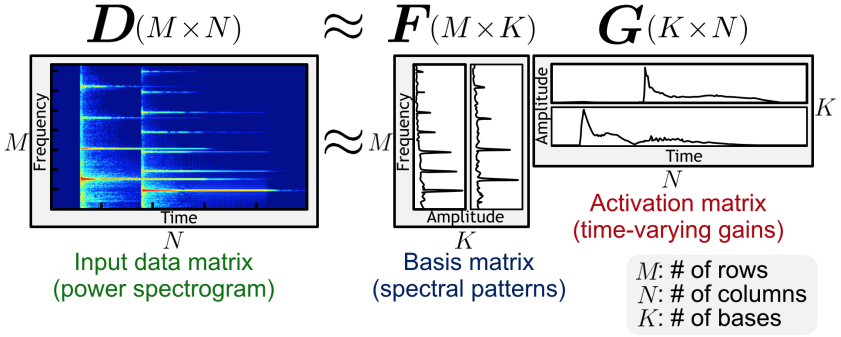

FIGURE 3.3 – *Exemple d'une décomposition d'un spectrogramme en utilisant la factorisation NMF (source : (Kitamura et Ono, 2016)).*

Dans le contexte de la factorisation spectrale, la divergence de Kullback-Leibler (D_{KL}) est généralement utilisée dans la fonction objectif et le problème d'estimation

2. Le terme *atome* est généralement utilisé en factorisation matricielle pour faire référence aux éléments du dictionnaire ou de la base apprise.

des matrices non-négatives **F** et **G** s'écrit sous la forme :

$$\underset{\mathbf{F},\mathbf{G}}{\operatorname{argmin}} D_{KL}(\mathbf{D}|\mathbf{FG}) \tag{3.2}$$

Cette décomposition s'est avérée utile dans le contexte de la parole bruitée permettant de modéliser les composantes additives du signal (parole, bruit) et à capturer les informations caractéristiques aux locuteurs (Saeidi et al., 2012). Ceci a permis de mettre en œuvre un ensemble de méthodes d'extraction de paramètres pour la RAL basées sur la décomposition NMF. On en cite quelques exemples dans ce qui suit.

Représentations basées sur le *sparse coding*

Deux approches nommées *Exemplar-based sparse representation* (Saeidi et al., 2012) et *Convolutive Sparse Coding* (Hurmalainen et al., 2015) ont été explorées pour la reconnaissance du locuteur en milieux bruités. Ces algorithmes utilisent une factorisation NMF convolutive et se basent sur une approche de *sparse coding* [3].

En phase d'entraînement, la NMF est utilisée pour créer des dictionnaires de parole correspondant à différents locuteurs ainsi que des dictionnaires de bruit pour capturer l'information relative à l'environnement. En phase de test, les activations de parole et de bruit sont extraites et utilisées comme modèles de locuteurs. Cette paramétrisation a été testée avec différentes stratégies de scoring et a permis d'améliorer significativement les performances du système apportant un gain de 50% en termes de précision sur les données du challenge CHiME 2 comparé à un système i-vecteur standard (Saeidi et al., 2012).

Group Non-negative matrix factorization

Cette approche, proposée dans (Serizel et al., 2016), s'inspire des méthodes d'analyse factorielle développées par Kenny (Kenny et al., 2007) et vise à capturer l'information locuteur au niveau des spectrogrammes tout en tenant compte de la variabilité session. Pour le faire, les dictionnaires $F^{h,s}$ construits par l'algorithme NMF pour chaque session h et locuteur s sont divisés en sous-dictionnaires qui représentent respectivement le locuteur, la session et une composante résiduelle (similairement aux modèles d'analyse factorielle) :

$$F^{(h,s)} = \left[F^{(h,s)}_{loc} \quad | \quad F^{(h,s)}_{sess} \quad | \quad F^{(h,s)}_{res} \right] \tag{3.3}$$

3. Ce terme fait référence à l'utilisation de méthodes de construction de dictionnaires sous la contrainte de "rareté" des activations. Ce codage s'inspire de l'activité neuronale dans le cerveau et vise à représenter des données en utilisant un sous-ensemble réduit d'atomes à activation forte. Dans ce contexte, les activations nulles correspondent à des atomes désactivés alors que les activations non-nulles correspondent à des atomes actifs.

De plus, l'information locuteur et session est intégrée dans les contraintes de la décomposition NMF :

— Maximiser la similarité des dictionnaires $F_{loc}^{(h,s)}$ issus de différentes sessions et correspondant à un même locuteur.
— Maximiser la similarité des dictionnaires de session $F_{sess}^{(h,s)}$ correspondant à différents locuteurs dans une même session.

Cette méthode a été testée dans (Serizel et al., 2016) sur un ensemble réduit de données de la base ESTER (Galliano et al., 2005) et un gain de 5% a été observé en termes de F1-score par rapport à un système i-vecteur standard basé sur les paramètres MFCC.

3.1.4 Paramètres *bottleneck*

Avec l'émergence des réseaux de neurones profonds (DNN) dans la dernière décennie, des techniques d'extraction de paramètres à base de DNN ont vu le jour visant à extraire des paramètres plus robustes et pertinents pour la RAL. Dans ce contexte, une concaténation des paramètres acoustiques [4] de N trames consécutives est donnée au DNN comme entrée et le réseau est entraîné pour faire la classification de la trame centrale. Les classes généralement utilisées dans ces systèmes correspondent aux états du modèle HMM entraîné pour une tâche de reconnaissance de la parole.

Afin de fournir une version compressée des paramètres données en entrée, une couche cachée de taille réduite, dite *bottleneck* [5] est utilisée (Yamada et al., 2013; Matvejka et al., 2016) et le vecteur d'activations correspondant constitue un vecteur de paramètres *bottleneck*. La figure 3.4 donne une illustration de cette architecture.

Une fois extraits, les paramètres *bottleneck* peuvent être utilisés pour :

1. l'estimation robuste de statistiques, en assimilant les classes du DNN à des composantes d'un nouvel UBM (Matvejka et al., 2016; Tan et al., 2016). Les moyennes, matrices de covariance et poids de chaque composante sont calculés en utilisant directement les trames qui appartiennent à chaque classe.
2. la construction d'un nouveau système GMM-UBM/i-vecteur en utilisant les paramètres *bottleneck* (Yaman et al., 2012; Sarkar et al., 2014; Richardson et al., 2015).

Ces approches permettent d'améliorer significativement les performances des systèmes de RAL et apportent des gains qui varient entre 10% et 30% en termes de EER comparés à un système à base de GMM-UBM/i-vecteur (Richardson et al., 2015; Matvejka et al., 2016).

Une extension de cette paramétrisation a été proposée dans (Matvejka et al., 2016) sous le nom de *Stacked Bottleneck Features* basée sur un empilement de paramètres *bottleneck* extraits en utilisant deux réseaux de neurones profonds. Un premier DNN est

4. La sortie du banc de filtres de Mel ou les paramètres MFCC sont généralement utilisés.
5. Le terme *bottleneck* peut être traduit en "col de bouteille" et fait référence à la réduction significative de taille entre la couche d'entrée et cette couche cachée (similairement à la variation du rayon d'une bouteille près du col).

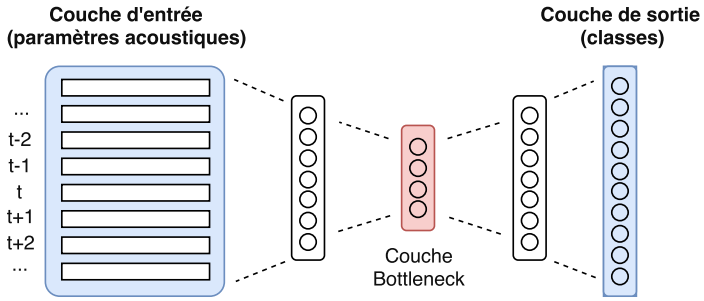

FIGURE 3.4 – *Structure d'un DNN utilisé pour l'extraction des paramètres bottleneck. Une concaténation des vecteurs paramètres de N trames consécutives est donnée en entrée et la classe correspondante à la trame centrale est donnée en sortie. En phase d'entraînement, un seul neurone est mis à 1 pour chaque entrée et le reste des neurones sont mis à 0. En phase de test, les valeurs des neurones de sortie correspondent à la probabilité d'appartenance à posteriori de la trame centrale à chacune des classes. Les activations de la couche bottleneck fournissent une nouvelle paramétrisation des données d'entrée.*

utilisé pour extraire des paramètres *bottleneck*. Par la suite, les paramètres *bottleneck* correspondant à des trames espacées de 5 trames sont empilées et utilisées comme entrée pour un deuxième DNN (5 vecteurs sont empilés, correspondant aux instants $t-10$, $t-5$, t, $t+5$ et $t+10$). Le contexte final correspond donc à 11 trames $\times$ 5 = 55 trames. Cette approche permet de capturer un contexte acoustique plus large qui modélise des unités phonétiques et capture éventuellement l'information locuteur d'une manière plus robuste. Ce système apporte un gain de 20% en termes de performance en EER par rapport aux systèmes *bottleneck* qui utilisent un seul DNN.

3.2 Amélioration de signal et compensation de paramètres

Depuis le début des années 90, une grande gamme de techniques d'amélioration de signal (*speech enhancement*) et de compensation de paramètres ont été proposées à savoir la soustraction spectrale, l'égalisation spectrale (Acero, 1990), le filtrage Wiener (Paliwal et Basu, 1987) ainsi que des méthodes de filtrage plus robustes tel que le filtrage RASTA (Hermansky et Morgan, 1994). Dans la dernière décennie, de nouvelles approches plus efficaces ont été proposées pour la RAL. Certains exemples de ces techniques sont présentés dans ce qui suit.

3.2.1 Techniques à base de réseaux de neurones profonds

Amélioration de signal

Les réseaux de neurones profonds ont été utilisés pour l'amélioration du signal dans le contexte de la RAL (Kolbœk et al., 2016). Cette approche utilise une architecture à

base de LSTM (*Long Short-Term Memory*)[6] et transforme la version bruitée d'un signal donné en sa version propre. Dans ce système, le module des coefficients FFT est utilisés comme entrée pour le modèle LSTM avec un contexte qui comporte les 15 trames précédentes et les 15 trames suivantes (un total de $N_{FFT} \times 31 = 257 \times 31 = 7967$ coefficients). Cette approche a été testée sur les données RSR2015 et a permis des gains significatifs qui peuvent atteindre jusqu'à 40% en termes d'EER.

Compensation de paramètres

Les réseaux de neurones profonds (DNN) ont aussi été utilisés pour la compensation de paramètres en présence de bruit additif. Dans ces approches un réseau de neurones profond est utilisé pour transformer la version bruitée des paramètres acoustiques vers la version propre correspondante. L'entraînement est fait sur un contexte de N trames comme le montre la figure 3.5 (généralement 1 trame + les 5 trames précédentes + les 5 trames suivantes = 11 trames).

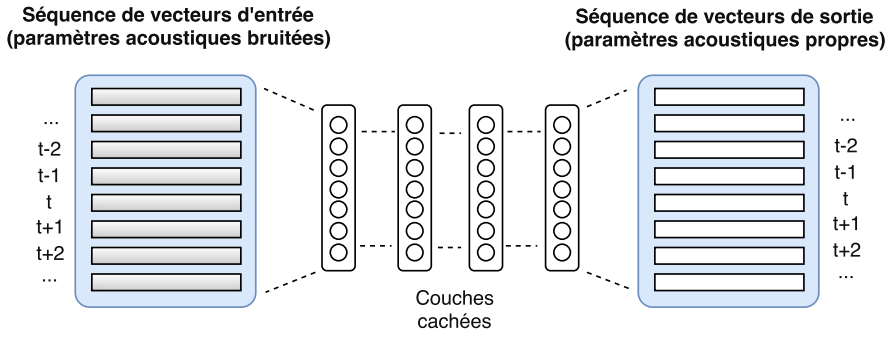

FIGURE 3.5 – *Exemple de DNN pour la compensation de paramètres bruitées.*

Dans ces systèmes, le DNN est d'abord entraîné en utilisant un DAE (*Denoising Autoencoder*)[7] (Vincent et al., 2008) ou un empilement de machines de Boltzman (Salakhutdinov et Hinton, 2009) de manière à minimiser l'erreur quadratique entre les paramètres propres et les paramètres transformés. Ces approches apportent des gains qui peuvent atteindre jusqu'à 26% d'amélioration relative en EER sur les données NIST SRE 2010.

6. Les LSTM sont une classe de réseaux de neurones récurrents qui permettent de gérer les données séquentielles à taille variable et de faire les transformations séquence-vers-séquence. Ils implémentent le concept de "mémoire" qui permet d'extraire des paramètres caractérisant un long contexte de données.

7. Un autoencodeur est un réseau de neurones profond qui apprend la fonction identité $f(X) = X$ en minimisant l'erreur de reconstruction quadratique (les mêmes vecteurs sont utilisés en entrée et en sortie). Un DAE est une version d'autoencodeurs qui vise à apprendre une version robuste des vecteurs de données en rajoutant artificiellement un bruit Gaussien aux entrées ($f(X_{\text{bruité}}) = X$).

3.2.2 Compensation stochastique de paramètres

Une classe d'algorithmes de compensation stochastique de paramètres a été évaluée pour la RAL dans (Sarkar et Sreenivasa Rao, 2014). Ces approches se basent sur la distribution des paramètres acoustiques propres et celle des paramètres acoustiques bruités et visent à débruiter les paramètres acoustiques en phase de test. Étant donné un vecteur de paramètres de test bruité y_t, le critère de l'erreur quadratique moyenne (MMSE : *Minimum mean square error*) est utilisé pour estimer le vecteur paramètre propre correspondant $\hat{x}_t$ comme suit :

$$\hat{x}_t = E[x|y_t] = \int_X P(x|y_t)\, x\, dx \tag{3.4}$$

x est une variable aléatoire représentant le vecteur de paramètres propres et $p(x|y_t)$ représente la distribution de probabilité conditionnelle de x sachant y_t.

Différents algorithmes ont été proposés pour l'estimation de $P(x|y_t)$; Les algorithmes RATZ (Moreno et al., 1998) (*Multivariate Gaussian-based cepstral normalization*) et SPLICE (*Stereo-based Piecewise LInear CompEnsation for Environments*) (Deng et al., 2001) modélisent le décalage entre la distribution des paramètres acoustiques propres et bruitées sous forme de composantes additives au niveau des moyennes et des matrices de covariance. Cette relation est par la suite intégrée dans le calcul de $P(x|y_t)$ dans l'équation 3.4. Grâce au caractère bayésien de cette estimation, ces algorithmes permettent d'intégrer des connaissances à priori sur la distribution des paramètres acoustiques propres $P(x)$. Ceci permet de fournir des estimations de $\hat{x}_t$ qui sont consistantes avec la distribution des paramètres ciblés. RATZ et SPLICE permettent d'atteindre une amélioration relative en EER variant entre 30% et 50% respectivement par rapport à un système de base propre (Sarkar et Sreenivasa Rao, 2014).

Malgré les bonnes performances données par ces algorithmes, la compensation de paramètres peut causer des inconsistances lors de l'utilisation des coefficients dynamiques (Δ et $\Delta\Delta$). Vu que la compensation est faite trame par trame, la cohérence des composantes dynamiques des trames successives débruitées ne peut pas être garantie. Ce problème a été traité dans (Zen et al., 2009) avec l'algorithme TRAJMAP (*TRAJectory Mapping*) (Zen et al., 2009) où la distribution jointe de trames successives sur une fenêtre de taille fixe est prise en compte dans la modélisation, garantissant plus de consistance au niveau des paramètres prédits et donnant des résultats largement supérieurs aux algorithmes précédents. Cette méthode vient confirmer l'importance de la composante dynamique en présence de bruit additif et a permis d'atteindre 70% d'amélioration relative en termes d'EER par rapport à un système de base propre (Sarkar et Sreenivasa Rao, 2014).

Un autre algorithme basé sur les modèles joints a aussi été développé dans (Afify et al., 2009) sous le nom de SSM (*Stereo Stochastic Mapping*). Au lieu d'estimer les distributions des paramètres propres et bruitées, cet algorithme estime la distribution jointe entre les deux versions $z_t = [x_t^T, y_t^T]^T$. SSM s'est avéré largement supérieur à RATZ

et SPLICE en termes de performances permettant d'atteindre 70% d'amélioration relative en EER par rapport à un système de base propre (Sarkar et Sreenivasa Rao, 2014). Ces performances sont dues à l'information supplémentaire intégrée dans l'estimateur MMSE (par rapport à RATZ et SPLICE) et qui décrit la composante jointe entre la distribution des paramètres propres et bruitées.

Malgré l'amélioration significative de performances en utilisant ces algorithmes, ces approches ne sont pas pratiques dans des applications réelles vu qu'elles supposent une connaissance à priori des conditions acoustiques de test. Une base de données stéréophonique est aussi requise pour pouvoir entraîner tous les modèles citées dans cette sous-section.

3.3 Compensation de modèles

Malgré les avancées qui ont été achevées en compensation de paramètres, l'utilisation de telles techniques avec un système de RAL appris sur des données propres peut produire des estimations biaisées d'i-vecteurs. Ceci est dû à la nature des ces algorithmes qui fournissent une estimation des paramètres débruitées mais ne permettent pas de prendre en compte ni l'incertitude relative à cette procédure ni la fiabilité des paramètres acoustiques utilisées lors du calcul d'i-vecteurs.

Pour faire face à ce problème des approches de compensation de modèles qui permettent d'intégrer l'effet des nuisances acoustiques au niveau des modèles ou de concevoir des extracteurs d'i-vecteurs plus robustes ont été développées.

3.3.1 Entraînement *multi-style*

Les approches d'entraînement *multi-style* [8] ont pris du succès en traitement de la parole permettant de construire des modèles acoustiques plus robustes en se basant seulement sur les données. Cette méthode vise à modéliser une grande variété de conditions acoustiques et à construire des modèles (UBM, matrice **T**, PLDA) plus robustes en utilisant à la fois des données d'apprentissage correspondant à de la parole propre et bruitée (Ribas et al., 2015a). Ce régime d'apprentissage est appelé *full multi-style training* dans la littérature vu qu'il injecte l'information de bruit dans toutes les composantes du système i-vecteur, une alternative qui utilise les seulement les données bruitées lors de l'apprentissage du modèle PLDA est appelée *partial multi-style training*.

Cette méthode d'apprentissage est populaire en pratique en raison de sa simplicité et permet d'améliorer d'une manière consistante les performances des systèmes de RAL en présence de bruit additif ou de réverbération atteignant des gains de 20% en termes d'EER par rapport à un système appris sur des données propres. Cependant, les connaissances à priori sur les conditions acoustiques de test peut permettre d'atteindre des gains plus importants ($\sim$ 45% d'amélioration relative) (Ribas et al., 2015a).

8. Appelé aussi *multi-condition training* dans la littérature.

3.3.2 Utilisation de méthodes de décodage d'incertitude

Étant donné un ensemble de paramètres acoustiques X_s correspondant à un segment propre s, le processus l'extraction de l'i-vecteur correspondant w_s se base sur le calcul de l'espérance (Dehak et al., 2011) :

$$w_s = E[P(w|X_s)] \tag{3.5}$$

En présence d'une distorsion acoustique (bruit additif, réverbération, etc), les paramètres X_s utilisées pour estimer la distribution conditionnelle $P(w|X_s)$ deviennent peu fiables. Les techniques de propagation d'incertitude visent à rendre le processus d'extraction d'i-vecteurs plus robuste en intégrant l'incertitude liée aux distorsions acoustiques dans les calculs et permettent à l'extracteur d'i-vecteurs de se concentrer sur les paramètres acoustiques fiables ou efficacement compensés.

Dans (Yu et al., 2014), le décodage d'incertitude est fait en utilisant la distribution des paramètres acoustiques propres et celle des paramètres acoustiques corrompues (*SPLICE Uncertainty Estimation*) ou la distribution jointe entre les deux représentations (*Joint Uncertainty Estimation*) pour permettre une estimation plus robuste du terme $P(w|X)$. Ces algorithmes montrent des améliorations relatives plus élevées à mesure que le niveau SNR augmente. Ceci est dû au fait que la qualité de l'incertitude estimée diminue à mesure que le SNR diminue. Le calcul de statistiques a aussi été modifié dans (Ribas et al., 2015b) pour permettre le calcul non-biaisé de statistiques. Les gains données par cette classe d'algorithmes peuvent atteindre 30% d'amélioration relative en termes d'EER.

3.3.3 Utilisation des séries de Taylor

Un ensemble d'algorithmes basés sur les séries de Taylor (VTS : *Vector Taylor Series*) ont été proposés dans (Lei et al., 2013, 2014a) pour la reconnaissance robuste du locuteur basée sur les i-vecteurs en présence du bruit additif ou de la réverbération. En substance, ces algorithmes propagent l'effet du bruit du domaine temporel jusqu'aux paramètres du modèle acoustique (UBM). Cette approche utilise le développement en séries de Taylor pour approximer la fonction de corruption au voisinage des moyennes de l'UBM et permet de calculer d'une manière plus robuste les i-vecteurs correspondants. Un autre algorithme a été proposé dans (Martınez et al., 2014) qui utilise une méthode d'approximation de fonction non-linéaires appelée la transformée UT (*Unscented Transform*) et a donné de meilleures performances dans le contexte de la RAL.

Des gains significatifs sont atteints par ces systèmes donnant entre 70% et 80% de gains en fausses acceptations pour une probabilité FA aux alentours de 10% et améliorant le EER d'un facteur de deux par rapport à un système propre. Il est important de préciser que les algorithmes basés sur les séries de Taylor sont dépendant de la nature de la nuisance ciblée et de la paramétrisation utilisée. En effet, la transposition de ces

techniques dans un système qui utilise des paramètres acoustiques ou une procédure de normalisation différente pourrait impliquer la re-dérivation de l'algorithme.

3.3.4 Modélisation robuste à base de DNN

Avec la montée des réseaux de neurones profonds, deux approches ont été développées pour l'estimation robuste de modèles de locuteurs. La première approche se focalise sur la génération de versions plus robustes d'i-vecteurs en entraînant un classifieur de trames et en l'utilisant pour l'estimation robuste des statistiques. La deuxième utilise un réseau de neurones pour apprendre une représentation vectorielle plus représentative que les i-vecteurs.

Calcul robuste de statistiques

Le processus d'extraction d'i-vecteurs se base sur des statistiques calculées sur les données par rapport à un modèle acoustique générique (UBM) (Dehak et al., 2011). Dans ce contexte, deux types de statistiques sont calculées ; les statistiques d'ordre zéro qui correspondent à l'accumulation des probabilités à posteriori des vecteurs de paramètres pour chaque composante de l'UBM et les statistiques du premier ordre qui correspondent à l'accumulation des vecteurs de paramètres pondérées par leurs probabilités à posteriori pour chaque composante de l'UBM.

Ceci rend la phase d'estimation des probabilités à posteriori par rapport aux paramètres acoustique cruciale pour l'estimation des i-vecteurs et pour avoir de bonnes performances. Dans cette optique, des approches basées sur les réseaux de neurones profonds ont été proposées afin de servir d'estimateurs robustes de probabilités à posteriori (Lei et al., 2014b; Garcia-Romero et al., 2014). Dans ces travaux, un contexte de N trames consécutives est donné en entée au DNN et les états d'un HMM entraîné pour la reconnaissance de la parole sont utilisées comme classes de sortie. Le DNN est alors entraîné comme classifieur pour la trame centrale en mettant à 1 le neurone de sortie correspondant à l'état HMM qui le génère et à 0 le reste des neurones. Afin de pouvoir intégrer ce DNN dans un système de RAL à base d'i-vecteurs, un nouvel UBM "supervisé" est construit en calculant les poids, moyennes et matrices de covariance sur les données correspondant à chaque état du HMM (Lei et al., 2014b; Tan et al., 2016). Enfin, le nouvel UBM supervisé est utilisé pour estimer une nouvelle matrice de variabilité totale $\mathbf{T}$ et le DNN est utilisé pour le calcul des probabilités à posteriori en entraînement et en test.

Cette approche permet de tirer parti de la puissance de classification des réseaux de neurones tout en restant dans le cadre de la RAL à base d'i-vecteurs et devient le nouveau standard en termes de modélisation robuste dans le domaine permettant d'avoir des gains relatifs en termes d'EER variant entre 10% et 35% sur les données de NIST SRE 2012 (Lei et al., 2014b; Tan et al., 2016) par rapport à un système i-vecteur de base.

Modélisation du locuteur à base de DNN

Certains travaux ont aussi utilisé les DNN pour la modélisation de locuteurs cherchant des représentations plus robustes que les i-vecteurs. Le modèle "d-vector" a été proposé dans (Variani et al., 2014) où un DNN est appris pour la tâche de classification phonétique de trames (basée sur la sortie d'un HMM). En phase d'évaluation, les activations de l'avant-dernière couche sont calculées sur les trames d'un enregistrement donné et leur moyenne est utilisée comme modèle de session remplaçant les i-vecteurs.

Un autre modèle appelé *speaker embedding* a aussi été proposé dans (Rouvier et al., 2015) en prenant les super-vecteurs associés à chaque enregistrement en entrée et en faisant correspondre la classe locuteur en sortie. Dans ce contexte, l'activation d'une couche intermédiaire du DNN est utilisée comme modèle de session.

En pratique, ces modèles ont apporté de faibles gains comparés aux i-vecteurs. Ceci est dû au processus d'extraction utilisé qui n'impose pas de contraintes sur la distribution des modèles calculés et peu générer des données qui s'éloignent de la distribution théorique désirée (distribution à queue lourde au lieu d'une distribution Gaussienne (Variani et al., 2014)).

3.4 Comparaison robuste des modèles de locuteurs

Outre les techniques de compensation du bruit au niveau paramètres et modèle, des techniques ont été développées pour la comparaison robuste de locuteurs. Ces approches peuvent être divisées en deux classes :

— **Des techniques de projection** : ces algorithmes permettent de projeter les i-vecteurs dans un nouvel espace qui réduit les variabilités nuisibles. Ces techniques ont l'avantage de pouvoir être combinées indépendamment de l'approche de scoring utilisée.

— **Des techniques de scoring** : ces techniques se basent sur des modèles d'analyse factorielle et étendent le modèle PLDA standard pour tenir compte des variabilités nuisibles durant la phase de scoring.

La plupart des techniques de projection dans l'espace des i-vecteurs s'intéressent au problème de la variabilité session, on en présente des exemples dans ce qui suit et on développe par la suite les modèles PLDA alternatifs qui ont été proposés de manière à tenir compte de l'effet des nuisances acoustiques pendant le scoring.

3.4.1 Compensation de la variabilité canal et session

Radial NAP

La projection d'attributs de nuisance (NAP : *Nuisance Attribute Projection*) a été proposée dans le contexte de la RAL dans le framework GMM-SVM (Solomonoff et al., 2004). Une version appelée *Radial-NAP* qui se base sur cet algorithme a par la suite été

adaptée dans le contexte des i-vecteurs appelée *Radial-NAP*. Cette version a aussi été proposée dans (Bousquet et al., 2011) pour compenser la variabilité canal en supprimant les dimensions qui correspondent aux nuisances dans cet espace.

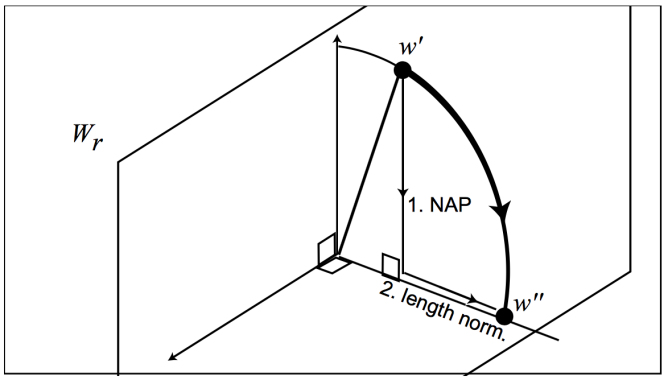

FIGURE 3.6 – *Effet de l'algorithme Radial-NAP (source : (Bousquet et al., 2011)).*

Présenté dans (Campbell et al., 2006c), la technique NAP estime la variabilité de la session en tant que sous-espace de rang intermédiaire obtenu en utilisant les axes principaux (vecteurs propres ayant les plus grandes valeurs propres) de la matrice de covariance intra-classe **W** et projette les i-vecteurs dans le sous-espace complémentaire orthogonal, supposé être l'espace du locuteur.

La figure 3.6 montre l'effet de cette projection ; la projection NAP compense l'effet de la variabilité session et la division par la norme projette l'i-vecteur sur l'hypersphère unitaire. L'implémentation de cet algorithme est détaillée dans la sous-section 1.5.2.

Normalisation WCCN

La normalisation WCCN (*Within Class Covariance Normalization*) a été introduite dans le contexte du framework GMM-SVM (Ferrer et al., 2007; Campbell et al., 2003, 2006a) pour la compensation de la variabilité session et a été introduite dans le contexte des i-vecteurs par Dehak (Dehak et al., 2011). Cette normalisation est détaillée dans la sous-section 1.5.2.

Normalisation de source

L'utilisation de la normalisation WCCN devient peu efficace en présence de plusieurs sources de variation dans les conditions de test (McLaren et Van Leeuwen, 2012). Afin de construire une méthode de normalisation qui représente adéquatement les directions de la variation au sein des locuteurs, la normalisation de source a été proposée dans (McLaren et Van Leeuwen, 2012; McLaren et al., 2012) comme une extension de la normalisation WCCN fournissant une meilleure estimation des matrices de dispersion intra-locuteur.

Dans cette approche, une source de variation est définie comme étant un facteur tel que la langue ou le type du canal (microphone ou téléphone) qui cause une différence entre les i-vecteurs correspondant à un même locuteur et qui cause en conséquence une dégradation des performances du système de RAL. La normalisation de source utilise le théorème de décomposition des variances par classes :

$$S_T = S_W + S_B \tag{3.6}$$

Les matrices S_T, S_W et S_B représentent respectivement la matrice de dispersion globale, la matrice de dispersion intra-locuteur et la matrice de dispersion inter-locuteur. Une nouvelle méthode d'estimation de la matrice de dispersion $\hat{S}_B$ est utilisée pour mieux estimer la matrice de dispersion inter-locuteur.

$$\hat{S}_W = S_T - \hat{S}_B \tag{3.7}$$

La matrice $\hat{S}_B$ représente une accumulation de matrices de dispersion dépendantes de la source :

$$\hat{S}_B = \sum_{src} S_B^{src} \tag{3.8}$$

où :

$$S_B^{src} = \sum_{S_{src}} N_s^{src} (\mu_s^{src} - \mu_{src})(\mu_s^{src} - \mu_{src})^T \tag{3.9}$$

S_{src} représente le nombre de locuteurs qui ont des i-vecteurs dans la source *src* dans les données d'entraînement et μ_s^{src} est la moyenne des N_s^{src} i-vecteurs du locuteur s et la source *src*. Le centre de l'espace des i-vecteurs est fixé, dans cette approche, au centres des sources $\mu_{src} = \frac{1}{N_{src}} \sum_{n=1}^{N_{src}} \bar{\mathbf{w}}_n^{src}$ où N_{src} est le nombre d'i-vecteurs de la source *src*.

La nouvelle matrice WCCN normalisée par la source est calculée en utilisant :

$$W_{SN-WCCN} = \frac{1}{S} \hat{S}_W \tag{3.10}$$

Cette normalisation surpasse d'une manière consistante la normalisation WCCN donnant des gains qui varient entre 10% et 30% en présence de plusieurs sources de variation dans les conditions de test (différents canaux / langues).

3.4.2 Comparaison robuste au bruit additif et convolutif

L'analyse NDA

Dans (Sadjadi et al., 2014), une analyse discriminante non paramétrique qui utilise la règle des voisins les plus proches pour estimer les matrices de dispersion intra- et

inter-locuteur portant le nom de NDA (*Neighbor Discriminant Analysis*) a été proposée et s'est avérée plus robuste que la LDA (*Linear Discriminant Analysis*) en présence du bruit additif et de la distorsion canal.

Cet algorithme essaie de corriger trois défauts de l'analyse discriminante linéaire (LDA) :

1. L'hypothèse de Gaussianité de la distribution de chaque classe qui devient non-valide en présence de nuisances acoustiques ou lors de l'utilisation de données hors-domaine [9]. L'hypothèse de mono-modalité au sein de classes n'est aussi pas toujours garantie.
2. La dimension fournie par la LDA est limitée par le rang de la matrice de dispersion inter-classe S_B. Cette contrainte peut devenir difficile à gérer si le nombre de classes d'entraînement est plus petit que la dimension des i-vecteurs.
3. Vu que seuls les centroïdes de classe sont pris en compte pour le calcul de la matrice de dispersion inter-classe S_B, la LDA ne peut pas capturer efficacement la structure de limite entre les classes adjacentes, ce qui est essentiel pour la classification (Fukunaga, 2013). La figure 3.7 montre la différence entre la dispersion paramétrique (utilisée par la LDA) vs. non paramétrique (utilisée par la NDA) entre deux classes.

Pour surmonter les limitations de la LDA mentionnées ci-dessus, l'algorithme NDA est utilisé. Cette approche mesure à la fois les dispersions intra- et inter-classes sur une base locale à l'aide d'une règle des voisins les plus proches :

$$S_B^{kNN} = \sum_{i=1}^{C} \sum_{\substack{j=1 \\ j \neq i}}^{C} \sum_{l=1}^{N_i} w_l^{ij} (x_l^i - \mathcal{M}_l^{ij})(x_l^i - \mathcal{M}_l^{ij})^T \tag{3.11}$$

où x_l^i dénote le $l^{\text{ème}}$ élément de la classe i et $\mathcal{M}^{ij}$ est une moyenne locale des k plus proches voisins pour x_l^i de la classe j qui est calculée suivant :

$$\mathcal{M}^{ij} = \frac{1}{K} \sum_{k=1}^{k} NN_k(x_l^i, j) \tag{3.12}$$

où $NN_k(x_l^i, j)$ est le $k^{\text{ème}}$ le plus proche voisin de x_l^i dans la classe j, , K est le nombre des plus proches voisins à considérer et w_l^{ij} est une fonction de pondération définie par :

$$w_l^{ij} = \frac{min\{d^\alpha(x_l^i, NN_K(x_l^i, i)), d^\alpha(x_l^i, NN_K(x_l^i, j))\}}{d^\alpha(x_l^i, NN_K(x_l^i, i)), d^\alpha(x_l^i, NN_K(x_l^i, j))} \tag{3.13}$$

$\alpha \in \mathbb{R}$ est une constante et $d(.)$ dénote la distance Euclidienne.

9. Le terme hors-domaine (*out-of-domain*) fait référence a des données qui ne ressemblent pas aux données utilisées pour l'apprentissage de l'UBM et de la matrice **T**. Les i-vecteurs qui en résultent peuvent ne pas obéir aux caractéristiques de l'espace des i-vecteurs en termes de distribution (Gaussianité / mono-modalité).

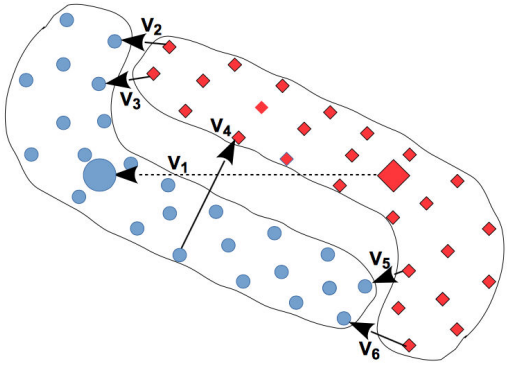

FIGURE 3.7 – *Exemple illustrant la dispersion paramétrique vs. non paramétrique entre deux classes. v_1 représente le gradient global des centroïdes de classe. Les vecteurs $v_2, , v_6$ représentent les gradients locaux (source : (Sadjadi et al., 2016)).*

Les expérimentations conduits dans (Sadjadi et al., 2014) ont montré que la NDA surpasse la LDA dans toutes les conditions en présence de bruit additif ou de distorsion de canal.

Le modèle *SNR-invariant PLDA*

Un modèle appelé *SNR-invariant PLDA* a été proposé dans (Li et Mak, 2015) pour la reconnaissance du locuteur robuste en présence de bruit additif. Dans ce modèle, les i-vecteurs correspondant à des segments qui ont des SNR proches sont supposés partager des informations communes. Cette hypothèse est implémentée sous la forme d'une composante supplémentaire dans le modèle PLDA standard. Dans ce nouveau modèle, trois facteurs sont estimés représentant l'information locuteur, canal et SNR. Le modèle correspondant est écrit sous la forme :

$$w_{s,h}^{k} = \underbrace{\mu + \Phi\beta_s + \epsilon_{s,h}}_{\text{Modèle PLDA standard}} + \overbrace{\mathbf{U}\mathbf{w}_k}^{\text{Composante SNR}} \tag{3.14}$$

Pour pouvoir estimer la matrice $\mathbf{U}$, les données d'entraînement sont partitionnées en K sous-groupes suivant leurs niveaux de SNR et les segments appartenant au même groupe sont supposés avoir des niveaux de SNR proches. Plusieurs valeurs de K ont été testés dans (Li et Mak, 2015) et un gain relatif pouvant atteindre 15% est reporté sur les données bruitées de NIST SRE 2012 par rapport au modèle PLDA standard.

3.4.3 Entraînement PLDA *multi-style*

Afin de construire un modèle de scoring robuste, une méthode d'entraînement *multi-style* [10] qui consiste à utiliser des donner d'entraînement correspondant à des données propres et bruitées pour la construction du modèle PLDA. Ces modèles permettent de capturer plus de variabilités acoustiques et d'augmenter la robustesse du modèle utilisé (Lei et al., 2012; Garcia-Romero et al., 2012).

3.4.4 Comparaison robuste à la variabilité des durées

Modèle basé sur le décodage d'incertitude

Le modèle PLDA standard considère que toutes les estimations ponctuelles des i-vecteurs sont également fiables et ne gère pas l'incertitude causée par les segments de courte durées (Kenny, 2010). Ceci a motivé la conception d'une variante du modèle PLDA qui remédie à ce défaut en modélisant l'incertitude due aux courtes durées par une composante additive dans le modèle PLDA (Kenny et al., 2013).

Dans ce modèle, les facteurs de canal sont remplacés par une composante qui reflète l'incertitude (le bruit d'observation) du processus d'extraction des i-vecteurs et l'information de durées

$$w_{s,h} = \underbrace{\mu + \Phi\beta_s + \epsilon_{s,h}}_{\text{Modèle PLDA standard}} + \overbrace{\mathbf{U}_s x_s}^{\text{Bruit d'observation}} \tag{3.15}$$

Ce modèle permet d'améliorer les performances en présence de variabilité des durées (*mismatch* de durées entre les données d'entraînement et de test) par rapport au modèle PLDA standard rapportant un gain relatif de 10% en termes de EER. Il convient de noter qu'un travail indépendant a été conduit dans (Cumani et al., 2014; Cumani, 2015) sous le nom de *Full Posterior Distribution PLDA* (FP-PLDA) où les procédures d'entraînement du modèle et de calcul de score sont simplifiées.

Entraînement PLDA *multi-durées*

La méthode d'apprentissage *multi-durées* a aussi été adaptée dans (Hasan et al., 2013) au contexte de la variabilité des durées en utilisant des sessions de durées différentes (courtes et longues) pour l'apprentissage du modèle PLDA. Ce modèle vise à réduire le *mismatch* de durées entre les conditions d'entraînement et de test qui peut affecter significativement les performances des systèmes de RAL (Sarkar et al., 2012).

10. Contrairement à (Ribas et al., 2015a) qui utilise un apprentissage *full multi-style training* et consiste à injecter le bruit dans toutes les composantes du système de RAL (UBM, matrice **T** et PLDA), ces méthodes peuvent être interprétées comme des versions "partielles" de l'entraînement *multi-style* où les données bruités sont seulement utilisés au niveau du modèle PLDA.

Ce régime d'entraînement permet d'avoir des gains variant entre 5% et 25% en termes d'EER par rapport au modèle PLDA estimé sur des segments de longue durée.

3.5 Compensation de scores

De nombreux travaux se sont intéressés à la compensation des variabilités nuisibles au niveau des scores. Bien que ces méthodes ne s'intéressent pas directement à l'effet des nuisances, elles permettent de mettre en place des méthodes d'optimisation globales de performance. Trois types de techniques peuvent être cités dans ce contexte : la normalisation, la fusion et la calibration robuste de scores.

3.5.1 La normalisation de scores

Ces méthodes appliquent généralement une transformation dépendante du locuteur et du canal aux scores données par le système de RAL et visent à normaliser les distributions de scores. Une population de référence est utilisée pour normaliser la distribution des scores par la moyenne et/ou la variance et les approches les plus utilisés dans le contexte de la RAL sont ; la normalisations H-norm (*Handset Nomalization*) (Reynolds, 1996), T-norm (*Test Normalization*), Z-norm (*Zero Normalization*) (Li et Porter, 1988), C-norm (*Cellular Normalization*) (Reynolds, 2003). Ces techniques ont été utiles dans la RAL basée sur les GMM (GMM-UBM, GMM-SVM) ainsi que les méthodes d'analyse factorielle qui ont précédé l'approche i-vecteur (Eigenchannels, Eigenvoices, etc) mais ont perdus leurs intérêt dans le contexte du paradigme de variabilité totale.

3.5.2 La fusion de scores

Dans ce contexte, les scores issus de deux (ou plusieurs) systèmes sont fusionnés afin d'avoir une décision plus robuste. La régression logistique est l'une des approches les plus utilisées dans ce contexte (Pigeon et al., 2000; Brummer et al., 2007). Cette technique a été utilisée face à un nombre de nuisances comme le bruit d'environnement et la distorsion canal (McLaren et al., 2013) ainsi que la variabilité de durée en utilisant un ensemble de projecteurs i-vecteur correspondant à différentes durées et fusionner les scores données par tous les sous-systèmes (Sarkar et al., 2012).

3.5.3 La calibration basée sur les mesures de qualité (QMF)

La calibration (Brümmer et Du Preez, 2006; Ramos-Castro et al., 2006) est une transformation appliquée sur la distribution des scores afin qu'ils puissent être interprétés comme des LR (*Likelihood Ratio*). Cette notion est utile dans le contexte judiciaire et aide le juge à prendre sa décision (plus grande (resp. petite) est la valeur de lu LR, plus l'hypothèse du procureur (resp. de la défense) est soutenue).

Dans ce contexte, des fonctions de mesures de qualité (QMF *Quality measure functions*) ont été proposées dans (Mandasari et al., 2013; Nautsch et al., 2016; Mandasari et al., 2015) afin d'améliorer la robustesse de la calibration des systèmes de RAL dans le contexte du bruit additif et de la variabilité des durées. Dans ces travaux, des mesures dérivées de la durée et du SNR des signaux de parole sont utilisées pour estimer les paramètres de la transformation de calibration.

Cette calibration utilise les paramètres des segments de référence et de test :

$$x = w_0 + w_1 s + Q(\lambda_{ref}, \lambda_{test}) \tag{3.16}$$

s représente le score brut et x le score calibré. La fonction $Q()$ utilise des paramètres des segments de référence et de test. Dans (Nautsch et al., 2016), les durées d_{ref} et d_{test} des deux sessions à comparer ainsi que leurs niveaux de SNR ; SNR_{ref} et SNR_{test} sont intégrées dans la fonction et plusieurs fonctions sont évaluées.

$$x = w_0 + w_1 s + Q(d_{ref}, d_{test}, SNR_{ref}, SNR_{test}) \tag{3.17}$$

Bien que cette approche permette d'améliorer la qualité de la calibration des scores en pratique, elle a montré de faibles gains en termes de performance.

Conclusion

Ce chapitre a présenté un panorama des techniques de traitement de nuisances acoustiques pour les applications de RAL. Les approches d'amélioration de signal et de compensation de paramètres ont d'abord été abordés. L'efficacité de ces algorithmes varie suivant la technique utilisée et peut apporter des gains significatifs en termes de performance quand la connaissance à priori des conditions de test est supposée (compensation stochastique de paramètres). Par la suite, des exemples d'approches de compensation de modèles ont été présentées. Ces techniques donnent des taux d'amélioration globalement supérieurs aux approches de compensation de paramètres permettant de capturer l'incertitude causée par les variabilités nuisibles aux niveau des paramètres acoustiques. Enfin, des exemples d'approches de compensation de variabilités opérant dans le domaine des i-vecteurs ainsi que les modèles de scoring robustes ont été donnés. Ces techniques tirent parti des propriétés statistiques de l'espace des i-vecteurs et permettent de calculer les scores d'une manière plus fiable. Des stratégies de fusion et de calibration robuste de scores ont aussi été présentées dans la dernière partie.

Deuxième partie

Compensation des nuisances acoustiques dans l'espace des i-vecteurs

Chapitre 4

Jeux de données et systèmes utilisés

Sommaire

Introduction

Notre travail de thèse se focalise sur le développement de techniques de compensation des nuisances pour les applications de RAL. Pour permettre ce genre d'études, un choix adéquat de jeux de données doit être fait. Dans nos expériences, on utilise principalement la base NIST SRE 2008 ainsi que les données SITW récemment publiées. On donne dans ce qui suit un aperçu de ces deux bases de données et on décrit les tâches utilisées dans notre partie expérimentale. Enfin, on donne une description des systèmes développés.

4.1 Campagnes d'évaluation NIST SRE

NIST (*National Institute of Standards and Technology*) est un organisme qui fait partie du département du Commerce des États-Unis et vise à développer des technologies et des standards en coopération avec les académiques et l'industrie. Depuis 1996, NIST organise une série de campagnes d'évaluation sous le nom de NIST SRE [1] (*Speaker recognition Evaluation*) qui s'appuient sur des données collectées par le LDC [2], se focalisent sur le problème de la vérification du locuteur et visent à développer de nouvelles technologies dans ce domaine en collaboration avec les académiques et les industriels.

Les données fournies au sein des évaluations NIST SRE portent sur des personnes des deux sexes, d'âge varié, parlant en une multitude de langues (en anglais natif ou pas, avec une vingtaine d'autres langues). Les données correspondent à des segments enregistrés à l'aide de différents microphones dans des conditions peu bruitées ou à des conversations téléphoniques qui mettent en relation deux locuteurs. Les données sont de durée variable (quelques secondes à plusieurs minutes), dans des cadres d'enregistrement différents (niveaux de bruit ambiant, effort vocal, caractéristiques du microphone et du canal de transmission, processus de codage du signal).

Cette variété résulte en un ensemble de corpus de taille considérable (des centaines d'heures d'enregistrements) qui offrent aux organismes de recherche plein de possibilités en termes d'analyse de performance et de comparaison de systèmes. C'est ce qui fait du jeu de données NIST SRE une référence au sein de la communauté de RAL.

4.1.1 Données d'entraînement utilisées

Pour la construction de nos systèmes de RAL, on utilise des données d'entraînement provenant des bases NIST SRE 2004, 2005, 2006 et Switchboard qui correspondent à 15660 sessions correspondant à 1147 locuteurs hommes (345 heures de parole). Les figures 4.1 et 4.2 montrent respectivement la distribution des durées et des rapports signal sur bruit (SNR) des segments utilisés.

1. NIST SRE : http ://www.itl.nist.gov/iad/mig/tests/sre/
2. LDC : The Linguistic Data Consortium http ://www.ldc.upenn.edu/

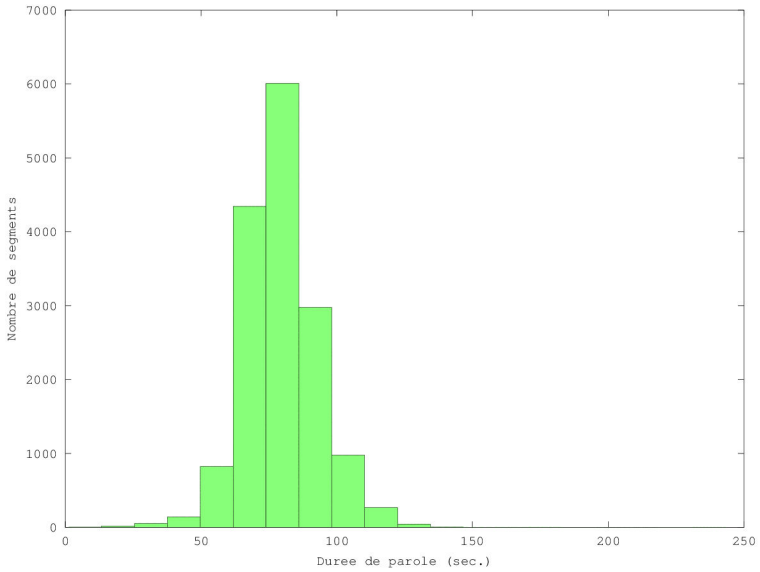

FIGURE 4.1 – *Histogramme des durées des segments d'entraînement.*

La durée des segments varie entre 30 secondes et 2 minutes et la distribution des rapports signal sur bruit (SNR) montre que les données utilisés sont peu bruités (97% des segments ont un SNR $> 15dB$). Cette base sera utile dans notre partie expérimentale pour les raisons suivantes :

— Cette base sera utilisée pour construire un système de RAL (UBM, matrice T, PLDA) basé sur des données propres et de longues durées. Ces conditions seront considérées comme "idéales"[3] dans notre contexte et seront utilisées pour évaluer la réponse du système de RAL en présence de nuisances acoustiques.
— Les performances données par ce système sur des données de test propres seront utilisées comme référence, étant les meilleurs performances possibles (borne inférieure en termes de EER). Elles permettront d'évaluer l'efficacité des algorithmes de compensation des nuisances développées dans notre partie expérimentale.
— Ces données seront utilisées pour générer artificiellement des segments bruités et des segments de courte durée. Ces données seront utiles lors du développement des algorithmes de compensation de nuisances acoustiques dans la partie contribution.

3. Il est important de noter que la distorsion canal et la variabilité session restent deux problèmes à gérer dans ces bases en phase de scoring et que le terme "idéal" fait ici référence à l'absence de bruit additif, de réverbération et de segments de courte durée.

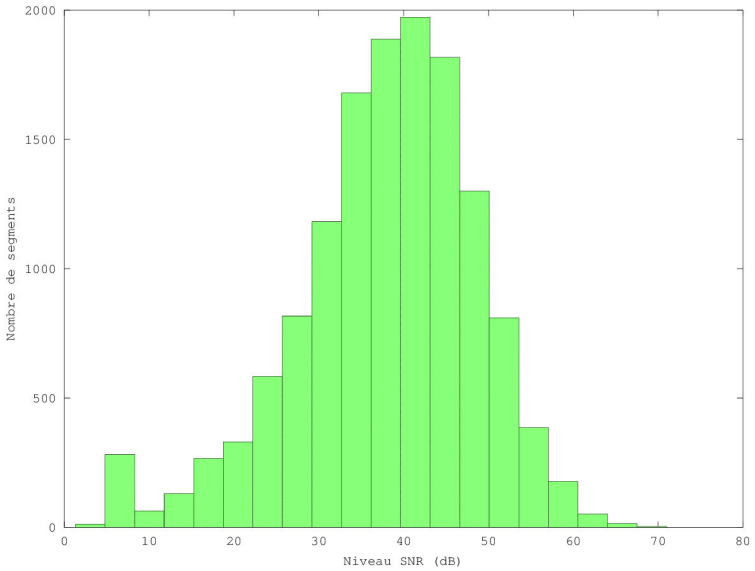

FIGURE 4.2 – *Histogramme des niveaux de SNR des segments d'entraînement.*

4.1.2 Évaluation sur les données NIST SRE 2008

En phase d'évaluation, on utilise la condition ***short2-short3***[4] de 2008 NIST SRE (nist2008eval, 2008) qui fournit des données d'apprentissage et de test correspondant à des conversations téléphoniques à deux canaux d'une durée de 2 à 3 minutes. Cette condition contient huit tâches d'évaluation qui mettent en correspondance des données d'apprentissage et de test issus de différents canaux et langues. Ces tâches sont notées de *det1* à *det8* et correspondent à :

- *det1 :* Toutes les comparaisons impliquant uniquement la parole interview dans l'apprentissage et le test.
- *det2 :* Toutes les comparaisons portant sur la parole interview à partir du même type de microphone dans l'apprentissage et de test.
- *det3 :* Toutes les comparaisons portant sur de la parole interview à partir de différents types de microphones dans l'apprentissage et de test.
- *det4 :* Toutes les comparaisons relatives à de la parole interview en apprentissage et de la parole de test téléphonique.
- *det5 :* Toutes les comparaisons relatives à des segments de type téléphonique en apprentissage et des segments en qualité microphone en test.
- *det6 :* Toutes les comparaisons impliquent uniquement la parole téléphonique en apprentissage et en test.
- *det7 :* Toutes les comparaisons impliquent uniquement la parole téléphonique en anglais dans l'apprentissage et de test.
- *det8 :* Toutes les comparaisons impliquent uniquement la parole téléphonique en

4. Condition short2-short3 : enregistrements correspondant à une conversation téléphonique ou à un interview ayant une durée moyenne de 2.5 minutes.

anglais natif américain dans l'apprentissage et le test.

Le tableau 4.1 donne le nombre de comparaisons client (*target*) et imposteurs (*non-target*) dans chaque condition.

TABLE 4.1 – *Nombre de comparaisons client et imposteur dans les conditions short2-short3 de NIST SRE 2008.*

		# comparaisons client	# comparaisons imposteur
Short2-Short3	**det1**	4901	9504
	det2	248	483
	det3	4653	9021
	det4	439	4609
	det5	640	3406
	det6	874	11637
	det7	439	6176
	det8	228	3028

Dans notre partie expérimentale, on utilisera la condition "det7" vu qu'elle permet d'avoir des conditions de test contrôlées en termes de variabilité canal et langue tout en fournissant une quantité suffisante de comparaisons client/imposteur. Ce cadre nous permettra de nous focaliser sur d'autres types de nuisances dans nos expériences, à savoir le bruit additif et la variabilité des durées, tout en gardant un niveau acceptable de significativité statistique.

4.2 Les données SITW

La base de données SITW (*Speakers In The Wild*) a été publiée par SRI[5] pendant Interspeech 2016 et vise à évaluer la technologie de reconnaissance du locuteur indépendante du texte dans des conditions difficiles et non-contraintes. Ce challenge vise à étudier la réponse des systèmes dans les conditions difficiles en fournissant une grande variété de nuisances acoustiques (bruit réel, réverbération, variabilité intra-locuteur, courtes durées et artefacts de compression). Ces facteurs font de SITW une base de données intéressante dans le cadre de notre thèse et sera utilisée pour évaluer les performances des algorithmes développés dans des conditions réelles.

4.2.1 Description des données

La base de données contient des échantillons de parole annotés manuellement à partir de sources ouvertes contenant des enregistrements d'un seul ou plusieurs locuteurs enregistrés dans des conditions réelles. La base de données se compose d'enregistrements de 299 locuteurs, avec une moyenne de 8 sessions différentes par personne. Les

5. SRI International : http ://https ://www.sri.com

données sont annotées par locuteur et par le type de distorsion acoustique présente dans les segments.

4.2.2 Tâches d'évaluation

La base SITW offre deux tâches principales de test ; la condition ***Core*** qui correspond aux fichiers audio contenant la parole d'un seul locuteur et la condition ***Multi*** qui à son tour contient des fichiers audio contenant un ou plusieurs locuteurs. On s'intéresse dans notre travail à la condition ***Core*** pour des raisons de simplicité. En effet, il serait nécessaire d'utiliser un système de segmentation en locuteurs pour évaluer nos systèmes sur la condition *Multi* et l'erreur relative à l'étape de segmentation devrait être prise en compte lors de l'évaluation du système.

4.2.3 Données de test

Vu que les données fournis dans la base SITW utilisent un échantillonnage à 16KHz, toutes les données issus de cette base sont tout d'abord filtrés à $300Hz \sim 3400Hz$ et sous-échantillonnés à 8KHz [6]. Ce traitement est nécessaire pour pouvoir utiliser les données de test SITW avec les données d'entraînement NIST SRE qui correspondent à des données téléphoniques.

Pour nos expériences sur les données de SITW, on prend deux sous-ensembles des données de test de la tâche ***Core*** :

1. ***Ensemble I*** : Une partie qui correspond à des données bruitées de longues durées. Dans ce sous-ensemble, les données d'apprentissage et de test correspondant à des segments de durée de parole supérieure à 30s et à des SNR inférieurs à 10dB. La figure 4.3 montre la distribution des SNR des segments d'apprentissage et de test utilisées dans cet ensemble.
2. ***Ensemble II*** : Une partie qui correspond à des données propres de courtes durées. Dans ce sous-ensemble, les données d'apprentissage et de test correspondent à des segments de durée de parole inférieure à 15s et à des SNR supérieurs à 20dB. La figure 4.4 montre la distribution des durées (en secondes) des segments d'apprentissage et de test utilisées dans cet ensemble.

4.3 Données générées artificiellement

Mis à part les données fournies par les bases NIST SRE et la base SITW, on génère artificiellement des données corrompues afin d'avoir plus de contrôle sur les conditions

6. Cette opération vient au prix d'une perte de performances. Le lecteur peut se référer à l'analyse de (Novotnỳ et al., 2016) pour voir la différence entre les systèmes qui utilisent les données échantillonnées à 16kHz et à 8kHz.

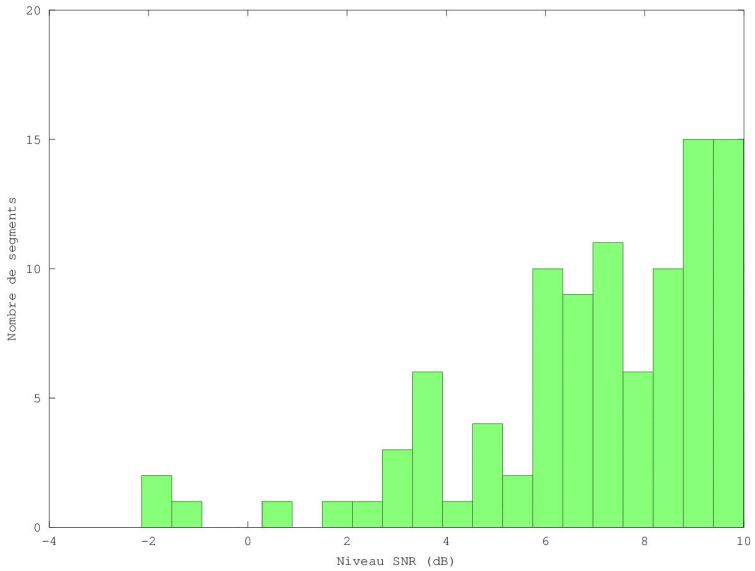

FIGURE 4.3 – *Histogramme des niveaux de SNR des segments de l'ensemble I.*

d'entraînement et de test dans nos expériences. Ces données seront aussi utilisées pour construire des modèles de scoring PLDA multi-style.

On traite dans nos expériences deux types de nuisances ; le bruit additif et la variabilité des durées. Dans ce qui suit, on présente le protocole de génération des données pour ces deux nuisances acoustiques.

4.3.1 Données bruitées

Nous utilisons 10 types de bruit issus du site FreeSound.org (Freesound.org, 2017) pour bruiter les données d'entraînement, d'apprentissage et de test : {bruit de climatiseur, bruit de voiture, bruit de foule, bruit de nature, bruit de pluie, applaudissements, sonnerie, bruit de fond d'une station de bus, bruit de vagues et bruit de tempêtes }. Les bruits ont été ajoutés aux signaux originaux générant de nouveaux segments bruités pour les niveaux de SNR suivants : {0dB, 5dB, 10dB, 15dB}.

4.3.2 Données courtes

Des versions courtes des 15660 segments d'entraînement (données NIST SRE 2004, 2005, 2006 et Switchboard) ainsi que les données de test de NIST SRE 2008 ont été générées pour différentes durées : 5s, 10s, 15s, 20s et 30s. Ce découpage est effectué dans le domaine temporel en sélectionnant au hasard un segment continu du signal original contenant de la parole. La sortie du système VAD est utilisée pour éviter la génération

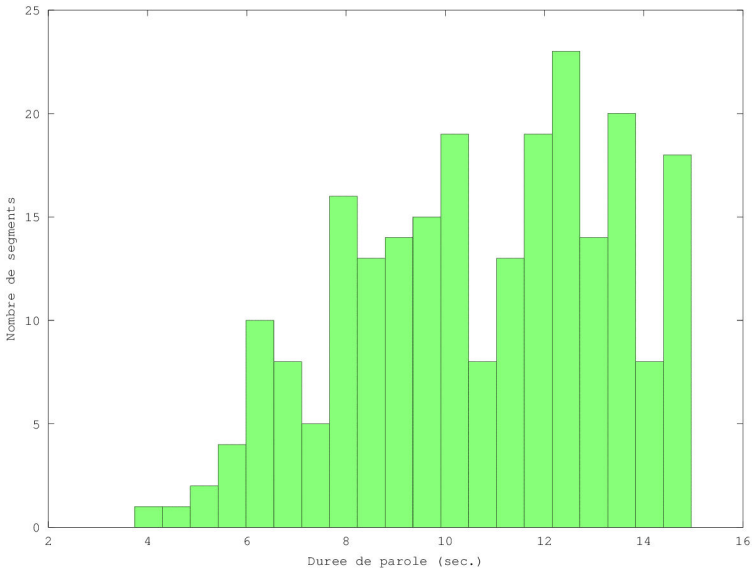

FIGURE 4.4 – *Histogramme des durées des segments de l'ensemble II.*

de segments qui contiennent exclusivement du silence. Le système VAD ainsi que la paramétrisation utilisée dans toutes nos expériences sont décrits dans la section suivante.

4.4 Système VAD et paramétrisation

Dans cette section, on présente le système de détection d'activité vocale utilisé dans l'ensemble de nos expériences ainsi que la procédure de paramétrisation et de normalisation adoptés.

4.4.1 Détection d'activité vocale

Les systèmes VAD les plus couramment utilisés en reconnaissance du locuteur se basent sur l'énergie des trames (Mak et Yu, 2014; Sahidullah et Saha, 2012). Le système VAD utilisé dans cette thèse est décrit dans (Sahidullah et Saha, 2012; Larcher et al., 2013). Il est basé sur la distribution des log-énergie des trames.

Tout d'abord, les log-énergies de chacune des trames d'un enregistrement donné sont calculés. Puis, en utilisant l'algorithme EM, la distribution des coefficients log-énergie est estimée en utilisant un modèle de mélange de gaussiennes à 3 composantes. Les trames qui correspondent à la Gaussienne ayant la plus grande moyenne (trames de haute énergie) sont ensuite utilisées comme trames de parole tandis que les trames de faible énergie, correspondant principalement au silence et au bruit, sont rejetées. Un seuil est calculé pour déterminer le seuil à partir duquel une trame est considérée comme étant de la parole ou du silence. ce seuil est défini dans l'équation (4.1) :

$$\tau_{thr} = \mu_3 - \alpha\,\sigma_3 \tag{4.1}$$

où μ_3 et σ_3 correspondent à la moyenne et l'écart type de la gaussienne correspondant aux trames de haute énergie, et α est une variable de contrôle de sélectivité. L'augmentation de la valeur du coefficient α permet de prendre plus de trames de haute énergie en compte. La sélection des trames est par la suite lissée en utilisant une fenêtre morphologique (Larcher et al., 2013). La figure 4.5 montre un exemple d'une distribution de log-énergie à 3-composantes ainsi que le seuil utilisé pour sélectionner des trames de parole/non-parole. La figure 4.6 donne un exemple réel sous forme d'un histogramme de log-énergies ainsi que le seuil de décision parole/non-parole.

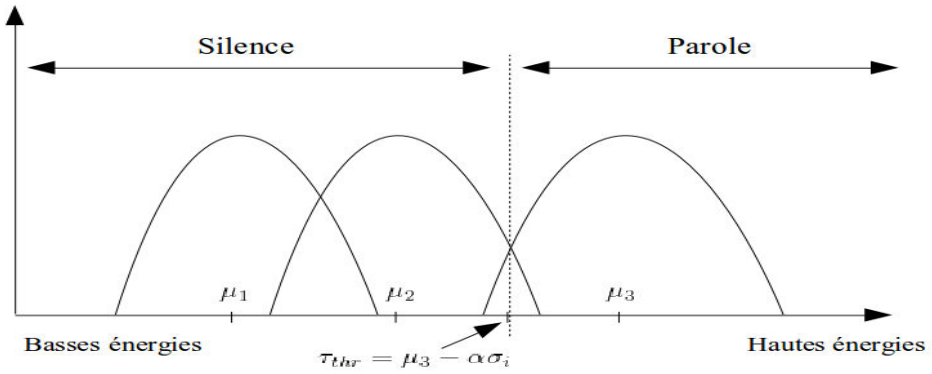

FIGURE 4.5 – *La distribution des Log-énergies et le seuil de détection de parole.*

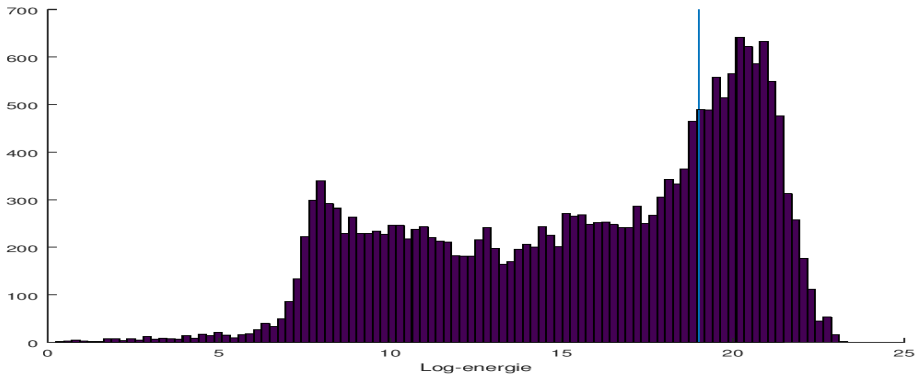

FIGURE 4.6 – *Histogramme des Log-énergies correspondant à une session de durée de 300 secondes. La ligne verticale correspond au seuil de décision parole/non-parole (seuil = 19) ; la durée de parole sélectionnée est de 95 secondes.*

Dans toutes nos expériences, nous avons utilisé $\alpha = 0$ lors du processus de détection d'activité vocale afin d'avoir une sélection stricte des trames de parole et de minimiser le risque d'erreur de sélection entre les trames de parole/non-parole qui pourrait se produire dans des conditions de SNR faibles (près de 0dB). Cette configuration correspond à une moyenne de 30% des trames sélectionnées comme trames de parole pour chaque segment de parole.

4.4.2 Paramétrisation et normalisation

On utilise la paramétrisation MFCC à 19 coefficients (+ énergie) augmentée avec 19 dérivées premières (Δ) et 11 dérivées secondes ($\Delta\Delta$)[7]. La sortie du VAD est utilisée pour effectuer une normalisation CMNV ; la moyenne et la variance sont calculées et utilisées pour normaliser les trames retenues (qui correspondent à de la parole).

4.4.3 Outils et bibliothèques utilisés

Les outils suivants sont utilisés pour la construction des systèmes :
— **Paramétrisation :** La boîte à outils open-source SPRO4 (Gravier, 2003).
— **Système VAD, estimation de l'UBM et de la matrice T :** Le paquet LIA_SpkDet de la boîte à outils LIA_RAL/ALIZE (Larcher et al., 2013)
— **Bruitage de données :** La boîte à outils open-source Fant (Hirsch, 2017).
— **Programmation génétique :** La bibliothèque python Pyevolve (Perone, 2009) est utilisée pour implémenter tous les algorithmes de programmation génétique.

4.5 Systèmes de RAL développés

Quatre systèmes de référence ont été développés dans le cadre de cette thèse. Ces systèmes partagent la même partie front-end (paramétrisation, modèle UBM et matrice **T**) mais utilisent des modèles de scoring différents. Ce choix est motivé par la simplicité de la procédure d'entraînement de ces systèmes et du fait qu'ils utilisent un UBM et une matrice **T** propres. Ce cadre nous permettra de comparer nos techniques de compensation des nuisances qui opèrent exclusivement dans le domaine des i-vecteurs.

4.5.1 Systèmes de base

Un systèmes de RAL dépendant du genre est construit en utilisant 15660 sessions correspondant à 1147 locuteurs hommes issus des bases NIST SRE 2004, 2005, 2006 et Switchboard. Ces données sont utilisées pour l'apprentissage d'un UBM à 512 composantes (et à matrices de covariance diagonales) et une matrice de variabilité totale de rang égal à 400.

Avant la phase de scoring, une normalisation EFR (Bousquet et al., 2011) à 2 itérations est effectuée sur tous les i-vecteurs[8] (i-vecteurs d'entraînement, d'apprentissage et de test).

La figure 4.7 résume la chaîne de traitements effectuées pour l'extraction et la normalisation d'un i-vecteur correspondant une session donnée.

7. Les 11 premières composantes des $\Delta\Delta$ sont utilisées.
8. Les paramètres de normalisation (moyenne et matrice de covariance) sont calculées sur l'ensemble de données d'entraînement.

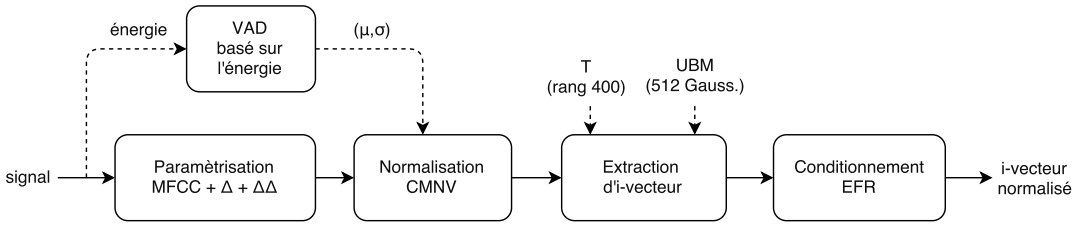

FIGURE 4.7 – *Processus d'extraction d'i-vecteur.*

Finalement, les données d'entraînement sont utilisées pour construire un modèle de scoring PLDA à 1147 classes locuteurs avec une matrice de canaux propres (*eigen-channel*) à rang plein (400) et une matrice de voix propres (*eigen-voice*) à rang égal à 100.

4.5.2 Système PLDA bruité

Le système PLDA bruité utilise le même UBM et matrice **T** que le système de base (appris sur des données propres). Par contre, le modèle PLDA utilisé est appris sur des données bruitées.

Dans ce système, les 15660 sessions d'entraînement sont bruitées en utilisant les mêmes bruits et niveaux de SNR existant dans les conditions d'apprentissage et de test. La même quantité d'i-vecteurs est utilisée pour chaque bruit / niveau SNR.

4.5.3 Système *multi-style*

Le système *multi-style* correspond à "l'apprentissage *multi-style* partiel" dans le sens où le modèle UBM et la matrice **T** sont appris avec des données propres alors que le modèle PLDA est construit en utilisant des données bruitées.

Ce système de distingue de la PLDA bruité au niveau des bruits et niveaux de SNR utilisées. Alors que la PLDA bruité utilise les mêmes bruits et niveaux de SNR présents dans les conditions de test, la PLDA multi-style est entraîné en utilisant des bruit différents "inconnus" à la base de test. Les 15660 sessions d'entraînement sont bruités aléatoirement en utilisant 5 bruits différents (applaudissements, sonnerie, bruit de fond d'une station de bus, bruit de vagues et bruit de pluie) à différents niveaux de SNR variant entre 0dB à 25dB (chaque i-vecteur d'entraînement bruité correspond à un seul bruit / niveau SNR).

4.5.4 Système *multi-durées*

Ce système reprend l'UBM et la matrice **T** du système de base qui correspondent à des données propres de longues durée. Un modèle PLDA *multi-durées* est construit en utilisant les 15660 sessions d'entraînement découpées aléatoirement de façon à générer

des segments de durées 5s, 10s, 15s, 20s, 30s. Des sessions de longues durées sont aussi utilisées dans l'apprentissage de ce système et le même nombre d'i-vecteurs est utilisé pour chaque durée.

Conclusion

Dans ce chapitre, on a donné un aperçu des jeux de données utilisés pour la construction du système de RAL ainsi que les données de test. En entraînement, les bases NIST SRE sont utilisées comme données "propres", fournissant des segments de longue durée et de niveau de SNR élevé. En test, deux bases sont utilisées ; la base NIST SRE 2008 d'un coté, et la base SITW d'un autre. La base NIST SRE 2008 fournit une mesure de performance du système de RAL sur des données propres et sera utilisées pour générer artificiellement des conditions de test corrompues. D'un autre coté, la base SITW fournit des données de test naturellement corrompues et sera utilisée dans la partie contribution pour valider les performances des algorithmes développés dans des conditions réelles.

Chapitre 5

Traitement des variabilités nuisibles dans l'espace des i-vecteurs

Sommaire

5.1 Introduction générale et motivations

Un grand intérêt a été accordé par la communauté de la RAL à la compensation des variabilités nuisibles au cours de la dernière trentaine d'années. Une grande par-

tie de ces techniques s'est concentrée sur la compensation des nuisances au niveau du signal ou des paramètres acoustiques permettant d'améliorer globalement les performances des systèmes dans les conditions difficiles. Cependant, les gains apportés par ces approches sont généralement faibles et une connaissance à priori sur les conditions acoustiques de test est parfois requise pour atteindre des gains plus importants (exp : méthodes de compensation stochastiques de paramètres).

Depuis le développement des modèles basés sur l'analyse factorielle, plus d'attention a été accordée au traitement des nuisances au niveau des modèles. Cet intérêt est principalement motivé par l'amélioration significative des performances obtenues par ces modèles comparés aux modèles GMM-UBM et GMM-SVM. En effet, les performances obtenues par l'approche i-vecteur dans des conditions propres ont encouragé la construction d'extracteurs d'i-vecteurs plus robustes. Les approches basées sur les séries de Taylor (Lei et al., 2013) sont un exemple de tels algorithmes où l'effet des nuisances acoustiques dans le domaine temporel est propagé vers les paramètres du modèle acoustique. Malgré leurs bonnes performances en pratique, ces procédures ont quelques inconvénients :

— Ils sont dépendants des paramètres acoustiques utilisés : utiliser des paramètres acoustiques plus sophistiqués que les MFCC pourrait rendre les calculs plus difficiles (exp : les paramètres MHEC ou PNCC).
— Ils sont dépendants de la méthode de normalisation de paramètres utilisée.
— Ils doivent être re-dérivés pour chaque type de nuisance acoustique dépendant de son effet dans le domaine temporel.

Une autre approche a aussi été proposée dans le contexte des i-vecteurs où le traitement des nuisances acoustiques est pris en compte lors de la phase de scoring. Ces méthodes intègrent l'information relative aux nuisances acoustiques de deux manières différentes :

1. l'utilisation de données d'entraînement propres et corrompues ; entraînement *multi-style* (Garcia-Romero et al., 2012).
2. la modélisation de l'effet des nuisances acoustiques sous forme de sous-espace dans le modèle d'analyse factorielle de la PLDA ; PLDA dépendante du SNR (Li et Mak, 2015), décodage d'incertitude au niveau de la PLDA (Kenny et al., 2013).

Malgré l'existence de ces techniques, peu de travaux ont été fait pour la compensation directe des nuisances acoustiques dans le domaine des i-vecteurs. En effet, dériver directement l'effet des nuisances acoustiques dans cet espace peut s'avérer difficile en raison de la complexité de leurs effets dans ce domaine.

Dans notre travail, on vise à tirer parti des propriétés statistiques simples de l'espace des i-vecteurs (espace vectoriel de faible dimension avec une distribution théorique Gaussienne) pour le développement de techniques de compensation des nuisances acoustiques. L'objectif est de s'éloigner des approches classiques qui modélisent les nuisances acoustiques sous forme de sous-espaces dans les modèles PLDA et de s'attaquer directement au problème au niveau des i-vecteurs. Pour le faire, on propose un ensemble d'algorithmes qui se basent sur des ensembles appariés d'i-vecteurs (i-vecteurs corrompus et les versions propres correspondantes) pour apprendre un estimateur de l'i-vecteur propre correspondant à un i-vecteur de test corrompu donné. Ceci peut être

accompli en explorant différentes hypothèses concernant la relation entre un i-vecteur propre et sa version corrompue. Cette méthodologie offre plusieurs avantages :

— les méthodes de compensation des nuisances acoustiques peuvent être conçues sous forme d'algorithmes de transformation de données et peuvent donc être combinées avec d'autres techniques. La LDA et la normalisation WCCN sont deux algorithmes de compensation de la variabilité session qui sont implémentés sous forme de multiplications matricielles et qui peuvent être combinés de manière séquentielle. Une telle combinaison permet de bénéficier des avantages de chaque technique et d'améliorer les performances.
— les données produites par ces techniques sont supposées être "débruitées" (dans le sens large du terme bruit). Il devient donc possible d'utiliser un système entraîné avec des données propres avec les données générés par de tels algorithmes.
— les meilleurs performances des systèmes de RAL sont obtenues dans le cas { système propre / données propres } et la compensation de variabilités acoustiques permettrait (théoriquement) de nous ramener à ces conditions "idéales".

5.2 Effet des variabilités nuisibles sur les systèmes de RAL basés sur les i-vecteurs

Il est connu que les performances des systèmes de RAL se dégradent significativement en présence de nuisances acoustiques. Cette dégradation peut être due à la distorsion de l'information acoustique contenue dans les segments de parole (bruit additif, réverbération, etc) ou à un manque de données (segments de courte durée).

Dans ce qui suit, on étudie l'impact de deux types de variabilités nuisibles sur les performances des systèmes de RAL ; le bruit additif et la variabilité des durées.

5.2.1 Effet du bruit additif sur les performances d'un système de RAL

Le modèle UBM est une composante centrale dans les systèmes de RAL à base d'i-vecteurs vu qu'il permet ; (1) de structurer l'espace des paramètres acoustiques ; (2) d'estimer les probabilités à posteriori des paramètres acoustiques qui sont nécessaires à l'estimation des i-vecteurs. En présence de nuisances acoustiques, les paramètres extraits peuvent être distordus de différentes manières (dépendant de la nature de la distorsion et de son intensité (Openshaw et Masan, 1994)). La distribution des paramètres acoustique peut aussi être affectée, ce qui change considérablement la qualité de l'i-vecteur estimé et cause une détérioration significative des performances des systèmes de RAL.

Afin d'étudier l'effet de la distorsion des probabilités à posteriori des trames due au bruit additif, nous utilisons le système de base décrit dans la section 4.5 (GMM-UBM propre et modèle de scoring propre). Pour chaque combinaison de bruit / niveau de SNR, les i-vecteurs de test sont extraits de données bruitées et le taux d'erreur correspondant est calculé. Deux résultats sont présentés pour chaque configuration en utilisant :

— **Probabilités postérieures bruitées :** (noté "Prob. à post. bruitées") où des trames de test bruitées sont utilisées pour calculer les probabilités à posteriori par rapport au GMM-UBM propre.
— **Probabilités postérieures propres :** (noté "Prob. à post. propres") où pour chaque trame de test bruitée, la trame propre correspondante est utilisée pour calculer les probabilités à posteriori par rapport au GMM-UBM propre.

Le tableau 5.1 présente les performancess du système dans ces deux conditions. Il est clair que les estimations à posteriori bruitées affectent gravement les performances du système de base et contribuent fortement au taux d'erreur observé. Par rapport à la condition "à post. propre", l'augmentation relative du EER varie entre 9% et 23% et les performances décroissent rapidement près de 0dB. Ceci est dû au grand déplacement de la positions des trames dans l'espace acoustique lorsque les sessions de test sont affectées par un bruit à faible niveau de SNR.

TABLE 5.1 – *Performances du système de base en utilisant des probabilités a posteriori propres et bruitées avec des données de test bruitées et des données d'apprentissage propres.*

		EER du système de base		**Dégradation relative (%)**
		Probabilités a post. propres	**Probabilités a post. bruitées**	
Bruit de climatiseur	**0dB**	**20.79**	26.85	-22.54
	5dB	**12.48**	15.21	-17.89
	10dB	**8.31**	9.51	-12.54
	15dB	**4.66**	5.41	-13.68
Bruit de voiture	**0dB**	**20.00**	25.54	-21.68
	5dB	**12.08**	14.54	-16.87
	10dB	**7.33**	8.32	-11.85
	15dB	**4.35**	4.82	-9.67

5.2.2 Effet de la variabilité des durées sur les performances d'un système de RAL

Il est connu que l'utilisation d'i-vecteurs correspondant à des sessions de longues durées en entraînement et en test fournit une meilleure performance par rapport aux sessions courtes puisqu'ils sont plus riches en information locuteur. Dans le cas de sessions courtes en test et en apprentissage, l'utilisation d'un modèle PLDA appris en utilisant de longues sessions peut être sous-optimal (Sarkar et al., 2012).

Pour des durées différentes (5s, 10s, 15s, 20s, 30s et durée complète), les données de test sont scorés à l'aide de 7 modèles PLDA différents (chacun correspond à une durée spécifique et le modèle *multi-condition* utilise des données d'entraînement appartenant à toutes les durées). Le tableau 5.2 montre que les meilleurs performances sont données en utilisant le modèle PLDA correspondant à la même durée présente dans le test (bien que les segments longs contiennent plus de données spécifiques au locuteur).

TABLE 5.2 – *Effet de la durée d'entraînement et de test sur les performances du système de RAL.*

		Durée de parole d'entraînement						
		Complet	**30s**	**20s**	**15s**	**10s**	**5s**	***Multi-condition***
Durée de parole de test	**Complet**	***1.59***	2.05	2.49	2.73	3.18	4.56	2.63
	30s	3.59	3.18	***2.96***	3.18	3.87	5.21	3.41
	20s	5.26	4.32	***3.87***	3.87	4.78	5.69	4.55
	15s	7.28	5.92	5.89	***5.50***	5.72	6.54	6.37
	10s	11.84	8.65	7.99	***7.28***	7.75	8.43	9.11
	5s	21.83	17.31	15.91	15.26	13.62	***13.21***	16.40

Ces résultats montrent que :

1. Le modèle PLDA doit être adapté aux durées des segments de test pour une performance optimale.
2. La meilleure performance est obtenue dans la condition où les données de test et d'entraînement correspondent à des sessions de longue durée.

Sur la base de ces deux observations, l'utilisation d'i-vecteurs longs (à la fois en apprentissage et en test) peut être considérée comme une condition "idéale" et développer une transformation qui transforme tout i-vecteur correspondant à une session courte en sa version longue améliorerait (théoriquement) les performances du système de RAL.

5.3 Une approche géométrique pour le débruitage d'i-vecteurs

Dans plusieurs problèmes de reconnaissance de formes, il est utile d'apprendre une transformation de données d'une première représentation (bruitée, corrompue, non-normalisée) à une deuxième représentation "idéale" (au sens large du terme) ou plus adaptée à la tâche ciblée. Ce principe est récurent en reconnaissance de formes et plusieurs algorithmes de normalisation, d'adaptation de domaine et de régression implémentent ce principe. L'analyse procustéenne est une approche populaire dans ce contexte qui se concentre sur la modélisation géométrique d'une telle transformation en combinant trois types de transformations ; rotation, translation et mise à échelle.

Étant donnés deux ensembles de données $\{x_i\}_{i=\{1,..,N\}}$ et $\{y_i\}_{i=\{1,..,N\}}$ ($x_i, y_i \in \mathbb{R}^M$) qui correspondent respectivement une représentation idéale et une représentation corrompue d'un ensemble de données, ces données peuvent être écrites sous forme matricielle ($X, Y \in \mathbb{R}^{M\times N}$) et le problème de Procrustes peut être formulé comme :

$$X = bEY + C \tag{5.1}$$

où, b est un facteur de mise à échelle, E est une matrice de rotation orthogonale et C est une matrice de translation (avec des valeurs constantes dans chaque colonne).

Dans (Ben Kheder et al., 2016c), on a présenté une implémentation de ce principe dans l'espace des i-vecteurs pour le traitement des variabilités nuisibles. Dans ce

contexte, on apprend une transformation géométrique qui représente une combinaison de translation suivie par une rotation entre une représentation corrompue d'un i-vecteur vers sa représentation propre. Cette méthode est motivée par plusieurs raisons :

— On vise à mettre en place une technique de compensation des nuisances qui repose exclusivement sur des couples d'i-vecteurs (corrompus/propres) et fait abstraction de l'effet réel des nuisances dans l'espace de variabilité totale.
— La mise en place d'une technique de transformation d'i-vecteurs de leurs version corrompues vers leurs versions propres permet d'utiliser un backend propre. Cette procédure permet (théoriquement) d'avoir les meilleurs résultats vu qu'elle rapporte les i-vecteurs de test au domaine des i-vecteurs propres qui donne les meilleurs résultats.
— Créer une transformation capable de compenser les variabilités acoustiques nuisibles présentes dans les i-vecteurs d'évaluation quelle que soit la source de nuisance permettrait d'utiliser un seul modèle de scoring appris en utilisant des données propres et de bonne qualité et peut garantir plus d'efficacité en terme de performance par rapport à un modèle de scoring *multi-style* qui tente de prendre en compte d'une manière "aveugle" les conditions acoustiques des données de test.

5.3.1 L'algorithme de Kabsch

Étant donné deux ensembles appariés de points de $\{x_i\}_{i=1..N}$ et $\{y_i\}_{i=1..N}$ définis dans un espace de dimension M où chaque point x_i dans le premier ensemble correspond un point unique y_i du deuxième (d'où l'appariement des deux ensembles), il est possible d'écrire les coordonnées correspondantes sous format matriciel :

$$P_X = \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,M} \\ x_{2,1} & x_{2,2} & \dots & x_{2,M} \\ \vdots & \vdots & & \vdots \\ x_{N,1} & x_{N,2} & \dots & x_{N,M} \end{pmatrix} \tag{5.2}$$

$$P_Y = \begin{pmatrix} y_{1,1} & y_{1,2} & \dots & y_{1,M} \\ y_{2,1} & y_{2,2} & \dots & y_{2,M} \\ \vdots & \vdots & & \vdots \\ y_{N,1} & y_{N,2} & \dots & y_{N,M} \end{pmatrix} \tag{5.3}$$

Étant données P_X et P_Y, le problème orthogonal de Procuste est un problème d'approximation de matrice qui vise à trouver la meilleure matrice R orthogonale qui transforme P_Y en P_X :

$$R = \underset{R}{\operatorname{argmin}} \|RP_Y - P_X\|_F \tag{5.4}$$

où : $R^T R = I_M$ et $\|.\|_F$ représente la norme de Frobenius.

Il est possible de contraindre ce problème en autorisant seulement les matrices de rotation (à savoir des matrices orthogonales de déterminant égal à 1). Dans ce contexte, la solution peut être trouvée en utilisant l'algorithme Kabsch détaillé dans (Kabsch, 1976). Cet algorithme permet d'estimer la meilleure matrice de rotation R qui transforme l'ensemble $\{x_i\}_{i=1..N}$ (P_X) en $\{y_i\}_{i=1..N}$ (P_Y) en se basant sur le critère de minimisation de la déviation quadratique (RMSD).

Note :

Afin de simplifier la terminologie utilisée tout au long de ce chapitre, on utilisera le terme "bruité" pour faire référence à une donnée affectée par n'importe quel type de nuisance acoustique et le terme "propre" pour faire référence à une donnée de bonne qualité.

Dans le contexte de reconnaissance du locuteur, nous allons utiliser l'algorithme Kabsch pour trouver, pour un certain bruit de test, la meilleure matrice de rotation R entre un ensemble d'i-vecteurs affectés par un bruit donné et leurs versions propres. Ce faisant, il sera possible d'appliquer la transformation résultante aux i-vecteurs de test bruités et récupérer une version "débruitèe".

L'algorithme de Kabsch permet de trouver la meilleure matrice de rotation qui transforme P_Y sur P_X et suit trois étapes :

1. Les deux ensembles de points (P_X et P_Y) sont transformés de sorte que leur barycentres coïncident avec l'origine du système de coordonnées.
2. La matrice de rotation est estimée en utilisant les deux matrices centrées (en utilisant une décomposition SVD).
3. Pendant la phase de test, la matrice de rotation est appliquée aux i-vecteurs de test bruités puis le barycentre des données propres est additionné.

Étape 1 : Translation des deux ensembles de points :

1. Calcul des centroïdes des deux ensembles d'i-vecteurs propres et i-vecteurs bruités :
   — $\overline{P_X} = centroïde(P_X)$
   — $\overline{P_Y} = centroïde(P_Y)$
2. Centrage de tous les points de P_X et P_Y autours de l'origine du système de coordonnées :
   — $\widetilde{P}_{X_i} = P_{Xi} - \overline{P_X}$ pour chaque ligne P_{Xi} de P_X.
   — $\widetilde{P}_{Y_i} = P_{Yi} - \overline{P_Y}$ pour chaque ligne P_{Yi} de P_Y.

Étape 2 : Estimation de la matrice de rotation :

1. Estimation de la matrice A de dimension $M \times M$ définie par : $A = \widetilde{P}_X^T \widetilde{P}_Y$
2. Décomposition SVD de la matrice A : $A = VSW^T$
3. Calcul de $d = signe(det(WV^T))$

4. Estimation de la matrice de rotation R :

$$R = W \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & & 0 \\ \vdots & & 1 & \vdots \\ 0 & \dots & 0 & d \end{pmatrix} V^T \tag{5.5}$$

Étape 3 : Application de la rotation sur les données de test :

Étant donné un ensemble d'i-vecteurs de test bruités $\{t_i\}_{i\in\{1,..,N\}}$:

1. Centrage des i-vecteurs de test :
   $\tilde{t_i} = t_i - \overline{P_Y}$ pour tous les i dans $i \in \{1, .., N\}$.
2. Rotation des i-vecteurs de test :
   $\hat{t_i} = R\tilde{t_i} + \overline{P_X}$ pour tous les i dans $i \in \{1, .., N\}$.

Vu que les i-vecteurs résultants $\hat{t_i}$ sont sensés être propres, il devient possible d'utiliser un modèle de scoring appris en utilisant des données propres. Il est important de noter que le centroïde de $\overline{P_X}$ peut être calculé une seule fois sur un grand ensemble d'i-vecteurs propres dans une étape *off-line* avant la phase d'évaluation vu qu'il est indépendant du bruit. Aussi, afin d'avoir une bonne estimation de la matrice de covariance A, nous allons travailler dans une configuration où $N > M$.

5.3.2 Performances en utilisant l'algorithme Kabsch

On évalue les performances de l'algorithme de Kabsch en présence du bruit additif et on le compare aux systèmes *multi-style* et PLDA bruité. On rajoute aux données de test trois bruits (bruit de climatiseur, bruit de voiture et bruit de foule) à 4 niveaux de SNR différents : 0dB, 5dB, 10dB et 15dB.

Dans ce contexte, les systèmes utilisés sont :
— **Système de base :** i-vecteurs bruités utilisés avec une PLDA propre (une description détaillée du système est donnée dans la section 4.5).
— **Système *multi-style* :** i-vecteurs bruités utilisés avec un modèle PLDA entraîné avec des données bruitées. Les données d'entraînement sont bruitées avec les bruits { applaudissements, sonnerie, bruit de fond d'une station de bus, bruit de vagues et bruit de pluie } à des niveaux de SNR variant entre 0dB à 25dB (une description détaillée du système est donnée dans la section 4.5).
— **Système PLDA bruité :** i-vecteurs bruités utilisés avec un modèle PLDA entraîné avec des données bruitées avec les bruits/SNRs des données de test (une description détaillée du système est donnée dans la section 4.5).
— **Algorithme Kabsch :** pour chaque bruit n / niveau SNR s [1] :

1. Le bruit et le niveau SNR sont connus d'avance vu que les données d'entraînement et de test bruitées sont générées artificiellement.

1. Un ensemble de segments d'entraînement propres sont affectés par le bruit n au niveau SNR s dB dans le domaine temporel.
2. Les i-vecteurs bruités correspondants aux segments résultants $\{y_i\}_{i=1..N}$ et leurs homologues propres $\{x_i\}_{i=1..N}$ sont extraits.
3. Les étapes 1 et 2 de l'algorithme Kabsch sont appliquées à $\{x_i\}_{i=1..N}$ et $\{y_i\}_{i=1..N}$ et le vecteur de translation $\overline{P_Y}$ ainsi que la matrice de rotation R sont estimés.
4. L'étape 3 de l'algorithme Kabsch est appliquée sur les i-vecteurs de test bruités en utilisant R et $\overline{P_Y}$.

Tout d'abord, nous présentons les performances du système en utilisant les données d'apprentissage propres et de test bruitées, par la suite, on l'évalue dans un contexte hétérogène (plusieurs bruits et niveaux SNR sont utilisés).

Performances des systèmes en utilisant des données de test bruitées

Pour trois bruits de test différents (bruit de climatiseur, bruit de foule et bruit de voiture), les données de test propres sont corrompues dans le domaine temporel et les i-vecteurs correspondants sont évalués avant et après l'application de l'algorithme de Kabsch. Le tableau 5.3 donne les performances des systèmes pour différents bruits de test.

TABLE 5.3 – *Performances dans différentes conditions de test en utilisant des données d'apprentissage propres et des données de test bruitées.*

Condition de test		EER			
		Système de base	**Système Multi-style**	**Système PLDA bruité**	**Kabsch**
Bruit de climatiseur	**0dB**	26.85	23.53	22.01	**17.18**
	5dB	15.21	12.21	12.92	**10.34**
	10dB	9.51	8.62	7.32	**5.70**
	15dB	5.41	4.72	4.65	**3.40**
Bruit de voiture	**0dB**	25.54	22.85	22.21	**15.83**
	5dB	14.54	10.54	11.63	**9.30**
	10dB	8.32	7.24	6.40	**5.15**
	15dB	4.82	4.20	4.14	**3.22**
Bruit de foule	**0dB**	24.24	22.03	20.60	**16.48**
	5dB	13.94	10.01	10.73	**9.20**
	10dB	7.77	5.97	6.75	**5.20**
	15dB	4.01	3.82	3.12	**2.52**

Lorsque l'algorithme Kabsch est utilisé, une d'amélioration relative qui varie entre 33% et 40% est observée par rapport au système de base dépassant les performances du système générique *multi-style* ainsi que celle du système PLDA bruité. L'amélioration des performances par rapport au système *multi-style* était attendue vu que ce dernier

n'est pas adapté aux conditions acoustiques d'apprentissage/test contrairement à l'algorithme Kabsch. Des gains ont aussi été observés observés par rapport au système PLDA bruité (qui est adapté aux conditions acoustiques d'apprentissage/test). Ceci valide notre hypothèse qui dit qu'il serait plus intéressant de supprimer les nuisances acoustiques d'un i-vecteur donné plutôt que d'intégrer les conditions acoustiques associées dans le modèle PLDA.

Performances des systèmes dans un contexte hétérogène

Nous avons effectué une autre expérience pour tester la validité de notre technique dans une situation où le niveau de bruit varie de façon aléatoire entre les segments d'apprentissage / test. Dans cette expérience, toutes les sessions d'apprentissage et le test sont corrompues par un bruit choisi aléatoirement parmi les bruits suivants {bruit de climatiseur, bruit de voiture et bruit de foule} avec un niveau de SNR choisi au hasard entre 0dB à 20dB. En conséquence, chaque session bruitée est affectée par un bruit unique, à un niveau SNR fixe. Le tableau 5.4 montre les résultats obtenus avec les quatre systèmes.

TABLE 5.4 – *Comparaison des performances dans une configuration hétérogène. Toutes les sessions d'apprentissage et le test sont corrompues par un bruit choisi aléatoirement parmi les bruits suivants {bruit de climatiseur, bruit de voiture et bruit de foule} avec un niveau de SNR choisi au hasard entre 0dB à 20dB.*

	EER
Système de base	29.65
Système *multi-style*	23.12
Système PLDA bruité	20.72
Kabsch	**18.78**

Il est facile de voir que l'algorithme Kabsch améliore les performances de 36% dans une situation de données hétérogènes et surpasse les systèmes *multi-style* et PLDA bruité.

Note :

Nos expériences ont montré que cette approche est très sensible au niveau de SNR et au bruit utilisé. En effet, l'utilisation d'un seul vecteur de translation et une matrice de rotation pour la compensation de plusieurs bruits ou niveaux de SNR réduit significativement les gains précédemment obtenus. Ceci constitue un défaut qui pourrait être corrigé avec les approches proposées dans les chapitres suivants.

5.4 Un algorithme bayésien pour la compensation de variabilités nuisibles dans l'espace des i-vecteurs

Malgré le potentiel montré par l'approche géométrique présentée dans la section précédente, une telle méthode peut donner des estimations distordues des i-vecteurs de test. Ceci est dû partiellement à l'absence de connaissances à priori sur la distribution des i-vecteurs propres dans la formulation du modèle. En effet, l'intégration de cette information permettrait d'éviter les estimations distordues et permettrait de produire des i-vecteurs plus consistants avec la distribution réelle. Les approches bayésiennes semblent être un choix naturel dans ce contexte. Dans cette partie, on propose une approche probabiliste de débruitage d'i-vecteurs qui reprend la structure présentée auparavant dans le sens où l'on vise à transformer un ensemble d'i-vecteurs bruités en leurs versions propres. Cette approche est motivée par les raisons suivantes :

— Généralement, les techniques bayésiennes surpassent les algorithmes basés sur un critère MMSE.
— L'intégration d'informations à priori sur la distribution des i-vecteurs propres dans la procédure de débruitage évite d'avoir des estimations distordues des i-vecteurs corrompus (ce qui n'était pas le cas dans l'algorithme Kabsch).
— Les i-vecteurs débruités peuvent être utilisés avec un système de scoring entraîné en utilisant des données propres.
— Cette approche reste valide en cas de mismatch de bruit entre les sessions d'apprentissage et de test vu que la sortie de l'algorithme est supposée être propre.

5.4.1 Formulation de l'algorithme

Cet algorithme, que nous avons appelé I-MAP, a été introduit dans (Kheder et al., 2014) et son intégration dans un système de RAL a été développé dans (Ben Kheder et al., 2015; Matrouf et al., 2015).

Formellement, étant donné un i-vecteur bruité Y_0, notre objectif est d'estimer la version propre correspondante X_0. Nous allons définir deux variables aléatoires X et Y correspondant respectivement aux i-vecteurs propres et bruités. Soit N une troisième variable aléatoire représentant le bruit et définie par :

$$N = Y - X \tag{5.6}$$

Note :

Il est important de souligner le fait que ce qu'on appelle "bruit" dans ce modèle ne représente pas le bruit additif mais plutôt le décalage entre une version corrompue d'un i-vecteur et sa version d'origine. Si cette corruption correspond à un bruit additif, le terme N engloberait l'effet du bruit sur les trames ainsi que toutes autres distorsions de statistiques (la distorsion des probabilités à posteriori des trames calculées par rapport à l'UBM). Ce modèle vise donc de "corriger" toutes

les distorsions causées par la nuisance acoustique ciblée.

Bien que le processus d'extraction d'i-vecteurs proposé par Dehak suppose que les i-vecteurs résultants suivent une loi Gaussienne (Dehak et al., 2011), les travaux de Kenny ont montré que la distribution empirique est à queue lourde (Kenny, 2010). Dans les applications de RAL, il est commun d'utiliser une procédure de normalisation d'i-vecteurs (division par la longueur) afin de Gaussianiser la distribution des i-vecteurs (Garcia-Romero et Espy-Wilson, 2011). Dans le cadre de l'algorithme présenté dans cette section, tous les i-vecteurs bruités et propres utilisés sont d'abord normalisés par la longueur (une modélisation Gaussienne de la distribution des i-vecteurs peut alors être utilisée).

Supposons que les i-vecteurs propres X sont normalement distribués et supposons que le bruit (N) peut aussi être représenté par une distribution normale dans l'espace des i-vecteurs. Nous pouvons alors définir les fonctions de distribution de probabilité correspondantes $P(X)$ et $P(N)$ comme :

$$P(X) = \mathcal{N}(\mu_X, \Sigma_X) \qquad (5.7) \qquad\qquad P(N) = \mathcal{N}(\mu_N, \Sigma_N) \qquad (5.8)$$

$\mathcal{N}(\mu, \Sigma)$ représente une distribution normale avec une moyenne μ et une matrice de covariance pleine Σ.

Se référant à (5.6), (5.7) et (5.8), nous pouvons exprimer $P(Y_0|X)$ pour un i-vecteur de test bruité donné Y_0 par :

$$P(Y_0|X) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma_N|^{\frac{1}{2}}} exp^{-\frac{1}{2}(Y_0-X-\mu_N)^T\Sigma_N^{-1}(Y_0-X-\mu_N)} \qquad (5.9)$$

En se basant sur la définition (5.6) et les deux distributions précédemment définies, on peut estimer, pour un i-vecteur bruité donnée Y_0, la version propre correspondante $\hat{X}_0$ en utilisant le critère de maximum à posteriori (MAP) :

$$\hat{X}_0 = \underset{X}{\text{argmax}}\{\ln P(X|Y_0)\} \qquad (5.10)$$

En utilisant la loi de Bayes, on peut écrire $P(X|Y_0)$ comme :

$$P(X|Y_0) = \frac{P(Y_0|X)P(X)}{P(Y_0)} \qquad (5.11)$$

En combinant (5.10) et (5.11) :

$$\hat{X}_0 = \underset{X}{\text{argmax}}\{\ln P(Y_0|X)P(X)\} \qquad (5.12)$$

Trouver $\hat{X}_0$ devient équivalent à la résolution de :

$$\frac{\partial}{\partial X}\{\ln P(Y_0|X) + \ln P(X)\} = 0 \qquad (5.13)$$

En développant (5.13) en utilisant (5.9), obtient l'expression :

$$\frac{\partial}{\partial X}\{(Y_0 - X - \mu_N)^T \Sigma_N^{-1}(Y_0 - X - \mu_N)\} + \frac{\partial}{\partial X}\{(X - \mu_X)^T \Sigma_X^{-1}(X - \mu_X)\} = 0 \quad (5.14)$$

Après la dérivation, l'expression finale de l'i-vecteur débruité $\hat{X}_0$, étant donné sa version bruitée Y_0 ainsi que les deux distributions de X et N, est :

$$\hat{X}_0 = (\Sigma_N^{-1} + \Sigma_X^{-1})^{-1}(\Sigma_N^{-1}(Y_0 - \mu_N) + \Sigma_X^{-1}\mu_X) \quad (5.15)$$

5.4.2 Estimation de $P(X)$ et $P(N)$

Dans cette section, nous travaillons avec un total de six configurations : deux bruits différents (bruit de foule et bruit de climatiseur) et trois niveaux de SNR (10dB, 5dB et 0dB) en utilisant 3000 segments d'entraînement propres ($SNR > 25dB$).

La distribution d'i-vecteurs propres $P(X)$ et la distribution de bruit $P(N)$ sont les deux éléments les plus importants dans cette procédure de débruitage et leur estimation est critique vis à vis des performances du système. La distribution $P(X)$ a l'avantage d'être indépendante du bruit. Elle pourrait donc être estimée une seule fois pour toutes sur un grand ensemble d'i-vecteurs propres dans une étape hors ligne initialement avant de procéder à l'étape de débruitage.

D'autre part, $P(N)$ rend le système capable de s'adapter au bruit présent dans le signal et ainsi compenser son effet plus efficacement. Cette distribution est estimée pour chaque bruit de test différent et nécessite l'existence d'un ensemble d'i-vecteurs propres ainsi que les versions bruitées correspondantes aux mêmes segments. Tout d'abord, pour la partie propre et une fois que les fichiers d'entraînement sont fixés, les i-vecteurs propres correspondants (X) sont calculés. Puis, pour un segment de test bruité donné, le bruit est extrait du signal (en utilisant un détecteur de parole et la sélection des trames de faible énergie ; cette procédure sera détaillée dans les sous-sections suivantes), puis ajouté aux segments d'entraînement propres. Enfin, les i-vecteurs bruités correspondants (Y) sont estimés et l'équation (5.6) est utilisée pour calculer N, puis $P(N)$.

Dans ce qui suit, nous nous concentrons sur la réduction du nombre de segments d'entraînement utilisés pour construire $P(N)$ et nous fixons leurs critères de sélection.

Nombre d'i-vecteurs nécessaires pour estimer $P(N)$

Dans un contexte de données d'apprentissage propres et de données de test bruitées et pour chacune des six configurations de bruit décrites précédemment, l'EER est évalué en utilisant un nombre différent d'i-vecteurs d'entraînement pour estimer la distribution $P(N)$ allant de 400 à 3000. A chaque fois, $N = Y - X$ est utilisé avant la phase de scoring utilisant les i-vecteurs sélectionnés pour estimer μ_N et Σ_N. Pour chaque quantité d'i-vecteurs, la figure 5.1 montre l'EER obtenu sur 10 sous-ensembles différents choisis aléatoirement à partir de l'ensemble des i-vecteurs d'entraînement.

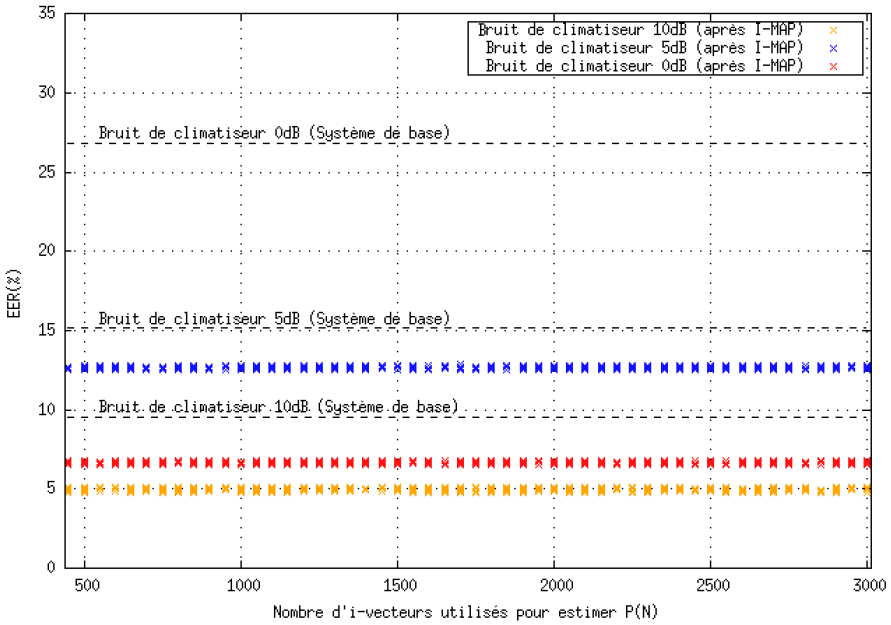

FIGURE 5.1 – *Variation du EER avec la quantité d'i-vecteurs utilisés pour estimer la distribution de bruit $P(N)$ pour le "bruit de climatiseur" à 0dB, 5dB et 10 dB (10 sous-ensembles pour chaque quantité d'i-vecteurs).*

L'axe des abscisses de la figure 5.1 commence à 450 puisque la dimension de l'espace des i-vecteurs utilisée dans cette expérience est égale à 400. En effet, passer en dessous de 400 pourrait produire des matrices de covariance singulières (utilisé dans l'équation 5.15).

Il est clair que pour les trois niveaux de SNR, l'EER ne varie pas beaucoup au-delà de 500. Par conséquent, nous allons définir le modèle de bruit sur un ensemble d'apprentissage de taille 500 i-vecteurs pour les prochaines expériences.

Sélection des i-vecteurs d'entraînement pour l'estimation de la distribution de bruit

Le but de cette expérience est de trouver un critère qui permet d'améliorer "globalement" la qualité du débruitage d'i-vecteurs sans mettre des contraintes strictes sur la durée des segments de test ou sur le contenu.

Une fois fixé à 500 le nombre d'i-vecteurs nécessaires pour estimer $P(N)$, nous nous concentrons sur leurs critères de sélection. Pour les six configurations différentes, nous avons créé un ensemble de 300 listes de 500 éléments choisis aléatoirement à partir de l'ensemble original de 3000 enregistrements propres qui seront utilisés pour estimer $P(N)$. Pour chaque liste, nous traçons l'EER résultante après compensation selon la durée moyenne des fichiers. La figure 5.2 montre la courbe obtenue en utilisant des données de test bruités avec le bruit de foule à 10dB.

Il est facile de voir que les segments ayant les durées les plus longues produisent de

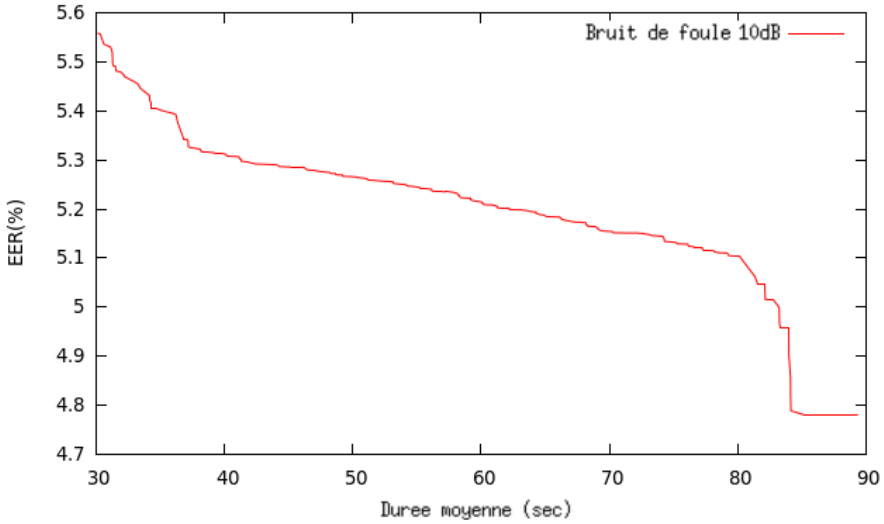

FIGURE 5.2 – *Variation du EER avec la durée moyenne des enregistrements utilisés pour estimer $P(N)$ pour le bruit de foule à 10dB.*

meilleurs résultats que les courts. Dans le reste de cette section, les 500 enregistrements d'entraînement les plus longs seront utilisés pour estimer $P(N)$. La forte baisse de la figure 5.2 est due aux données d'entraînement (peu d'enregistrements longs).

Seuil de compensation

L'un des avantages les plus importants de I-MAP est le fait que cet algorithme ne cause pas de distorsion sur les données propres et maintient de bonnes performances sur des données propres. Par conséquent, afin de gagner du temps et éviter les calculs inutiles, nous pouvons fixer un seuil de SNR au-delà duquel aucune transformation n'est appliquée (l'i-vecteur extrait est supposé être propre).

Afin de définir la valeur du seuil SNR_{seuil} au-dessus duquel un enregistrement de test est considéré comme propre, nous étudions la variation du EER avec le SNR maximum du segment de test débruité. La figure 5.3 montre que tenter de débruiter les sessions de test ayant des niveaux de SNR supérieurs à 25dB n'améliore pas les résultats. La variation du EER obtenu après compensation avec le seuil SNR est donnée pour deux bruits différents.

Dans ce qui suit, on utilisera $SNR_{seuil} = 25dB$ pour décider si l'opération de débruitage est requise.

5.4.3 Intégration de l'algorithme dans un système de RAL

La nouvelle méthode de débruitage d'i-vecteurs développée permet de construire un système de reconnaissance du locuteur qui prend en compte le niveau SNR d'une

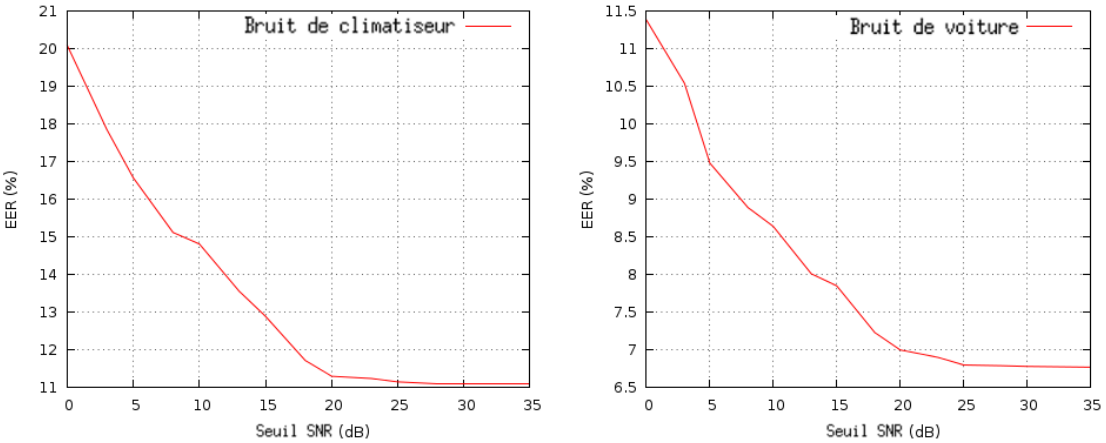

FIGURE 5.3 – *Variation du EER (après compensation I-MAP) avec le seuil SNR en dB pour deux bruits différents (bruit de voiture et de bruit de climatiseur) en utilisant des données d'apprentissage propres et des données de test bruités affectées avec le même bruit et sur des niveaux de SNR différents allant de 0dB à 35dB.*

session de test comme le montre la figure 5.4.

Avant de commencer, un seuil de SNR au-dessus duquel un segment est considéré comme propre doit être fixé (dans nos expériences, nous avons utilisé $SNR_{seuil} = 25dB$).

Puis, l'algorithme suit ces étapes :

— **Vérification de SNR :** Le niveau SNR est estimé pour le segment de test et comparé au seuil SNR_{seuil}.
— **Le cas propre :** Si le segment est propre, on extrait l'i-vecteur correspondant.
— **Le cas bruité :** Si l'i-vecteur est bruité :

1. L'i-vecteur bruité Y_0 correspondant est calculé.
2. Un VAD est utilisé pour extraire la partie du signal correspondant au bruit en sélectionnant les trames à basse énergie dans le signal (correspondant à du silence. La structure du système VAD utilisé dans nos expériences est détaillée dans la section 4.4.1 ainsi que le seuil de décision pour la détection de parole.
3. Le bruit est rajouté aux fichiers d'entraînement propres dans le domaine temporel avec le SNR du segment de test (estimé dans la première étape).
4. Les i-vecteurs correspondant aux sessions d'entraînement bruitées sont calculés (correspondant à Y).
5. La distribution de bruit $P(N)$ dans l'espace des i-vecteurs est estimée en utilisant l'équation $N = Y - X$.
6. Le nouveau i-vecteur propre est estimé en utilisant l'équation (5.15).

Il est important de préciser que dans des environnements bruités correspondant à des niveaux de SNR faibles, la procédure de détection d'activité vocale devient moins précise ce qui peut affecter la qualité de l'estimation du bruit (bruit extrait du signal). Dans nos expériences, deux seuils différents sont utilisés pour remédier à ce problème

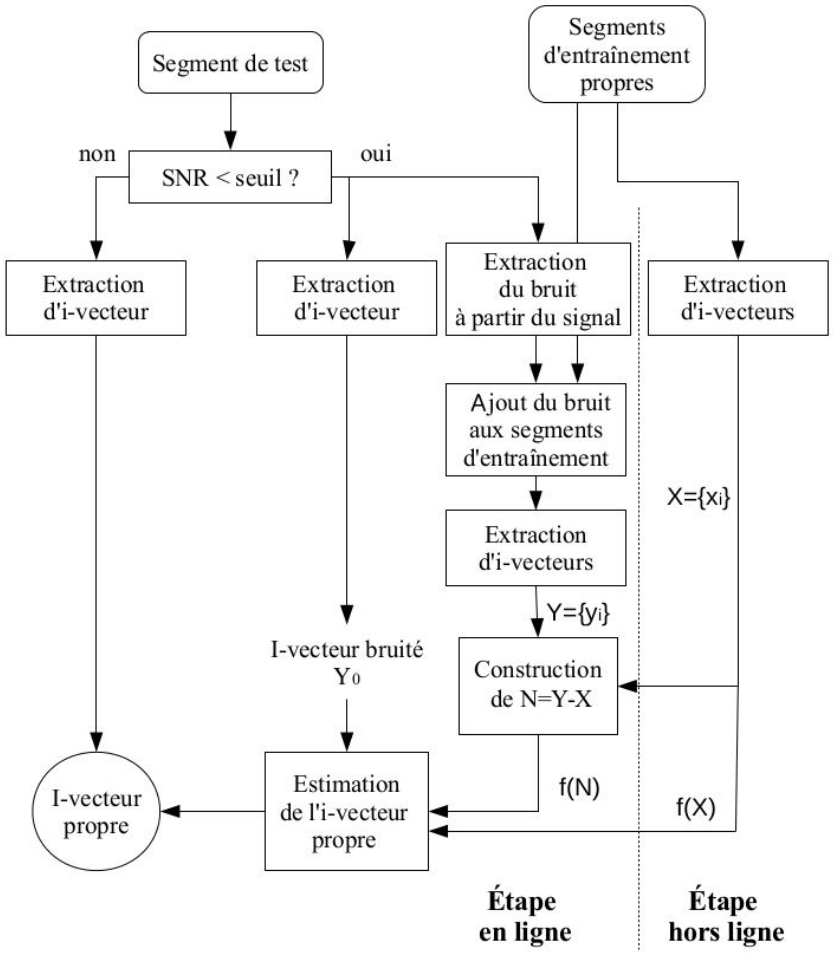

FIGURE 5.4 – *Algorithme de débruitage d'i-vecteurs. Tout d'abord, le niveau SNR du signal est estimé. Ensuite, si le segment est considéré bruité* ($SNR < seuil$)*, la distribution de bruit correspondante est estimée dans l'espace des i-vecteurs. Enfin, la procédure de débruitage I-MAP est appliquée.*

(un premier seuil est utilisé pour la détection d'activité vocale et un autre pour l'extraction du bruit). Pour chaque tâche, nous essayons de sélectionner les trames les plus utiles (en fonction de leur énergie), réduisant ainsi le risque d'erreur de reconnaissance. La procédure utilisée est expliquée dans la sous-section suivante.

5.4.4 Extraction de bruit

Le système VAD peut être utilisé pour faire une estimation du bruit de fond des segments bruités. En effet, les trames à faible énergie (qui correspondent essentiellement au silence) peuvent être extraites en prenant le complément des segments de parole.

Cette procédure est suffisante pour les segments avec des niveaux moyens et élevés SNR ($>$ 10dB), mais dans les niveaux de SNR faibles (par exemple. 0dB), il est difficile

de décider avec certitude si une trame correspond à de la parole ou du bruit. Pour cette raison, une configuration plus stricte est utilisée lorsque l'on veut extraire le bruit des signaux bruités. Dans toutes nos expériences, cette procédure est utilisée pour extraire le bruit d'un signal donné :

1. Calculer la valeur log-énergie de toutes les trames.
2. Modéliser la distribution log-énergie à l'aide d'un mélange de Gaussiennes à 3 composantes en utilisant l'algorithme EM.
3. Calculer τ_{seuil} en utilisant l'équation 4.1 avec $\alpha = 1$.
4. Prendre toutes les trames qui correspondent à $log\text{-}energie < \tau_{seuil}$ comme trames de bruit.

Régler $\alpha = 1$ lors de l'extraction du bruit permet de considérer plus de trames de haute énergie comme parole. En prenant le complément, cette approche permet de sélectionner moins de trames à faible énergie en tant que bruit. Ceci réduit le risque de sélectionner des trames de parole en tant que bruit dans des conditions de SNR faibles. Il est important de mentionner que cette procédure est utilisée comme une solution partielle qu'elle ne garanti pas que le système produise de bonnes estimations de $P(N)$ dans des conditions où le niveau SNR est faible.

5.4.5 Performances en utilisant I-MAP

Dans cette sous-section, les nouveaux i-vecteurs propres estimés (correspondant à deux segments de test ou d'apprentissage) seront appelés vecteurs "I-MAP".

Les données d'apprentissage et de test ont été bruités par deux ensembles différents de bruits :
— { bruit de nature, pluie et bruit d'un moteur } pour les sessions d'apprentissage.
— { bruit de climatiseur, bruit de voiture et bruit de foule } pour les sessions de test.
Les données ont été générés pour 4 niveaux de SNR différents : 0dB, 5dB, 10dB et 15dB.

Nous allons comparer dans cette section les performances des 4 systèmes :

— **Système de base :** i-vecteurs bruités utilisés avec une PLDA propre (une description détaillée du système est donnée dans la section 4.5).
— **Système *multi-style* :** i-vecteurs bruités utilisés avec un modèle PLDA entraîné avec des données bruitées. Les données d'entraînement sont bruités avec les bruits { applaudissements, sonnerie, bruit de fond d'une station de bus, bruit de vagues et bruit de pluie } à des niveaux de SNR variant entre 0dB à 25dB (une description détaillée du système est donnée dans la section 4.5).
— **Système PLDA bruité :** i-vecteurs bruités utilisés avec un modèle PLDA entraîné avec des données bruitées avec les bruits/SNRs des données de test (une description détaillée du système est donnée dans la section 4.5).
— **Système I-MAP :** Vecteurs I-MAP utilisés avec un modèle de scoring propre (l'algorithme décrit dans la section 5.4.3 est utilisé pour chaque i-vecteur).

Nous présentons d'abord les performances du système en utilisant des données d'apprentissage propres, puis nous présentons les résultats sur des données hétérogènes (apprentissage et test bruité avec différents bruits et niveaux de SNR).

Performances du système en utilisant des données de test bruitées

Pour trois bruits différents de test, le tableau 5.5 montre les performances des systèmes lorsqu'ils sont utilisés sur des données d'apprentissage propres et des données de test bruitées.

TABLE 5.5 – *Performances des différents systèmes sur des données d'apprentissage propres et des données de test bruitées.*

		EER			
Condition de test		**Système de base**	**Système Multi-style**	**Système PLDA Bruitée**	**I-MAP**
Bruit de climatiseur	**0dB**	26.85	23.53	22.01	**13.21**
	5dB	15.21	12.21	12.92	**7.25**
	10dB	9.51	8.62	7.32	**4.85**
	15dB	5.41	4.72	4.65	**2.85**
Bruit de voiture	**0dB**	25.54	22.85	22.21	**12.05**
	5dB	14.54	10.54	11.63	**6.65**
	10dB	8.32	7.24	6.40	**3.78**
	15dB	4.82	4.20	4.14	**2.36**
Bruit de foule	**0dB**	24.24	22.03	20.60	**11.55**
	5dB	13.94	10.01	10.73	**5.09**
	10dB	7.77	5.97	6.75	**3.05**
	15dB	4.01	3.82	3.12	**2.02**

Lorsque la compensation I-MAP est utilisée, une amélioration relative variant entre 48% et 64% est observée, alors que le système *multi-style* est limité à un maximum de 28% d'amélioration relative par rapport au système de base. Le système PLDA bruité surpasse le système *multi-style* (atteignant 33% d'amélioration relative), mais donne des résultats moins bons que ceux de I-MAP. Cependant, la construction d'un tel système (PLDA bruité) nécessite un grand nombre de sessions d'entraînement, ceci n'est pas pratique en particulier pour les applications réelles. D'un autre coté, I-MAP requiert moins de sessions d'entraînement tout en donnant de meilleurs résultats (cependant, I-MAP a un coût élevé en termes de temps de calcul). Cette expérience prouve clairement le potentiel de notre méthode dans des conditions de *mismatch* entre les conditions d'apprentissage et de test.

Performances du système sur des données hétérogènes

Une autre expérience a été effectuée pour prouver la validité de cette technique dans une situation où le niveau de bruit varie de façon aléatoire entre les segments d'apprentissage et de test. Dans cette expérience, toutes les sessions d'apprentissage et le test sont corrompues par un bruit choisi aléatoirement parmi les bruits suivants {bruit de climatiseur, bruit de voiture et bruit de foule} avec un niveau de SNR choisi au hasard entre 0dB à 20dB. Le tableau 5.6 montre les résultats obtenus avec les quatre systèmes.

TABLE 5.6 – *Comparaison de performance dans un contexte hétérogène. Toutes les sessions d'apprentissage et le test sont corrompues par un bruit choisi aléatoirement parmi les bruits suivants {bruit de climatiseur, bruit de voiture et bruit de foule} avec un niveau de SNR choisi au hasard entre 0dB à 20dB.*

	EER
Système de base	29.65
Système *multi-style*	23.12
Système PLDA bruité	20.72
I-MAP	**16.27**

En raison de la grande variabilité en termes de bruit et de niveau SNR, une amélioration significative est observée dans cette condition en utilisant I-MAP sur les données bruités avec un système de scoring appris sur des données propre comparé à un régime de scoring *multi-style*. En fait, cela montre les limites de la *multi-style* liés à sa propriété de généralisation. Cela rend notre méthode plus efficace dans des conditions de test / apprentissage inconnues car elle permet de s'adapter à tout bruit et niveau SNR présent dans un segment de test.

Le système PLDA bruité surpasse le système *multi-style*, mais ne peut pas être utilisé dans des applications réelles, car il suppose une connaissance préalable sur les conditions de test / apprentissage et nécessite l'ajout de bruit à un grand nombre de sessions d'entraînement. La différence entre les deux systèmes (*multi-style* et PLDA bruité) dans cette expérience est que le premier est construit en utilisant des segments propres et bruités affectés par des bruits qui ne figurent pas dans les conditions de test / apprentissage alors que le deuxième est construit en utilisant des bruits test / apprentissage à différents niveaux de SNR. Ceci explique la différence entre leurs performances.

5.4.6 Performances sur SITW

Dans (Ben Kheder et al., 2016a), on a testé la procédure de débruitage I-MAP sur l'ensemble de test de la base SITW. Dans cette expérience, l'*ensemble I* des données de test de SITW décrits dans la section 4.2 est utilisé. Cet ensemble correspond à des données d'apprentissage et de test bruitées (SNR inférieurs à 10dB) de longues durées (durée de parole 30s). Pour chaque session bruitée, l'algorithme décrit dans la figure 5.4

est appliqué (6000 i-vecteurs sont utilisés pour estimer $f(X)$ et 500 pour estimer $f(N)$). Enfin, le scoring est effectué avec le backend PLDA propre.

TABLE 5.7 – *Performances de I-MAP sur l'ensemble de test de SITW.*

	EER
Système de base	12.69
Système *Multi-style*	10.58
I-MAP	**6.34**

Il est clair que I-MAP améliore considérablement les performances du système de RAL pour atteindre 50% d'amélioration relative du EER sur les données de test bruitées par rapport aux performances du système de base. L'algorithme surpasse aussi le système *Multi-style* de 16% en gain relatif validant sa capacité à s'adapter aux conditions acoustiques des segments de test. Ceci confirme l'efficacité de l'algorithme proposé dans en présence de bruits réels.

Note :

La différence de gains obtenue entre les deux bases (NIST SRE 2008 et SITW) peut être due plusieurs facteurs. En effet, les performances de l'algorithme I-MAP dépendent de la qualité de la distribution de bruit estimée. Cette distribution est construite en se basant sur des données bruitées artificiellement et peut ne pas traduire fidèlement tous les effets induits par le bruit additif dans le cas des bruits réels (exp : effet Lombard). Un autre facteur à considérer est aussi le *dataset mismatch*[a]. Ce terme est généralement utilisé dans la littérature pour qualifier ce problème qui peut survenir lors de l'utilisation d'une base de test différente de celle utilisée pour entraîner le système. Ce *mismatch* peut se manifester sous forme d'une différence de performances entre les deux bases en raison de leurs propriétés acoustiques distinctes (types de microphones, différences de langues, etc).

a. Le terme *inter-dataset variability* (Aronowitz, 2014) est aussi utilisé au sein de la communauté de RAL.

5.5 Optimisation d'implémentation des méthodes de débruitage d'i-vecteurs pour des systèmes réels

5.5.1 Motivation

Dans les applications réelles, les exigences de temps de calcul et de mémoire sont deux facteurs importants à considérer, en particulier pour les machines à mémoire limitée tels que les smartphones. Pour une session de test contenant un bruit N_k, l'utilisation de la méthode proposée dans la section précédente (I-MAP) pour estimer les hyperparamètres de la distribution de bruit $d_{N_k} : (\mu_{N_k}, \Sigma_{N_k})$ est très coûteuse en termes de temps

de calcul en raison du nombre d'étapes nécessaires (ajout de bruit dans les fichiers d'entraînement, extraction des i-vecteurs bruités puis estimation de la distribution du bruit dans l'espace des i-vecteurs).

Pour faire face à ce problème, on a proposé dans (Ben Kheder et al., 2017) une solution qui évite l'étape d'estimation de la distribution de bruit en ligne en utilisant une base de distributions de bruit dans l'espace des i-vecteurs construite hors ligne avant la phase de reconnaissance. Au lieu d'estimer la distribution du bruit directement à partir du signal de test bruité (extraction des trames de bruit, puis les utiliser pour construire une distribution Gaussienne d'i-vecteurs bruités affectée par le même bruit), nous essayons de trouver la meilleure approximation du bruit de test en se basant sur les distributions présentes dans la base. En pratique, on ne dispose que de l'i-vecteur de test bruité Y_0, nous ne pouvons donc pas baser notre processus de sélection de distribution sur le bruit $(N_0 = Y_0 - X_0)$ correspondant. Afin de parer à ce problème, une solution possible serait de stocker, pour chaque configuration présente dans la base, à la fois la distribution $d_{\text{bruité}}$ des i-vecteurs bruités y_k (qui sera utilisée pour la sélection de distribution) et la distribution du bruit d_{bruit} de N_k (qui sera utilisée pour la compensation I-MAP). Pour un test de bruit i-vecteur donné Y_0, la distribution d'i-vecteurs bruités la plus probable $d_{y_k} : (\mu_{y_k}, \Sigma_{y_k})$ est d'abord sélectionnée à partir de la base. Ensuite, la distribution de bruit correspondante dans l'espace des i-vecteurs $d_{N_k} : (\mu_{N_k}, \Sigma_{N_k})$ est utilisée pour le débruitage comme le montre la figure 5.5.

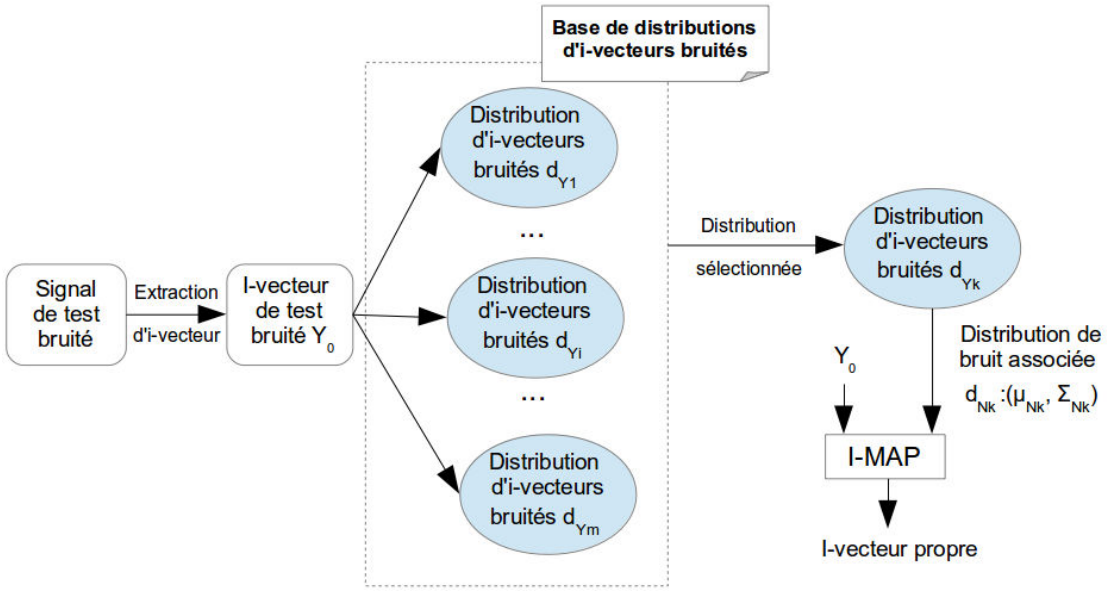

FIGURE 5.5 – *Utilisation d'une base de distributions de bruit dans l'espace des i-vecteurs pour le débruitage. D'abord, l'i-vecteur de test bruité est extrait. Puis, la distribution d'i-vecteurs bruités la plus probable d_{Y_k} est sélectionnée. Enfin, la distribution de bruit correspondante d_{N_k} est utilisée pour effectuer une compensation I-MAP.*

Dans la sous-section suivante, nous présentons la méthode utilisée pour construire la base de données, les critères de sélection de distribution de bruit et le nouveau régime de compensation résultant.

5.5.2 Construction de la base de distributions d'i-vecteurs bruités

La base de données de bruit est construite en utilisant 18 bruits différents provenant de différents environnements (bruit de vent, musique, bruit de voiture, le bruit d'un moteur, applaudissement, bruit de climatiseur, bruit de foule, ..) et 13 niveaux de SNR variant de 0dB à 30dB. Il est important de mentionner que les bruits utilisés pour construire la base de données sont différents de ceux utilisés pour bruiter les données d'apprentissage / de test. Cette condition permet de simuler un scénario réaliste. Pour chaque bruit différent et niveau SNR, nous suivons les étapes décrites dans l'algorithme 1. Nous nous retrouvons avec 234 distributions Gaussiennes différentes d'i-vecteurs bruités. L'étape suivante consiste à sélectionner la distribution qui correspond le mieux à un i-vecteur de test bruité donné Y_0.

Algorithm 1: Construction de la base de distributions d'i-vecteurs dans l'espace des i-vecteurs.

for *each* $(bruit_i, SNR_j)$ **do**
  1 - Ajout du bruit $bruit_i$ au niveau SNR_j aux fichiers d'entraînement.
  2 - Extraction des i-vecteurs bruités Y_{ij} correspondant aux segments bruités.
  3 - Calcul des données de bruit associées dans l'espace des i-vecteurs :
  $N_{ij} = Y_{ij} - X$
  4 - Calcul des hyper-paramètres de la distribution de bruit $d_{N_{ij}} : (\mu_{ij}, \Sigma_{ij})$.
end

Comme mentionné précédemment, les hyper-paramètres de la distribution des i-vecteurs bruités $d_{Y_{ij}} : (\mu_{ij}, \Sigma_{ij})$ sont également stockées car ils seront nécessaires lors de l'étape de sélection de distribution du bruit.

5.5.3 Sélection de la distribution de bruit

La sélection de la bonne distribution d'i-vecteurs bruités est une étape cruciale afin d'avoir une distribution de bruit qui reflète bien les conditions acoustiques de test. Nous considérons que chaque condition présente dans la base de données $(bruit_i, SNR_j)$ correspond à un bruit différent et essayons de sélectionner la distribution la plus proche des conditions de test étant donné un i-vecteur de test bruité Y_0 en se basant sur une distance ou une mesure de similarité. Un choix naturel pour la mesure de similarité dans ce contexte est la vraisemblance vu que nous avons supposé que la distribution des i-vecteurs bruités est gaussienne. Un autre choix possible est la distance euclidienne à n-dimensions entre un i-vecteur bruité et de la moyenne d'une distribution d'i-vecteurs bruités. Cette distance a l'avantage d'être très rapide par rapport à la mesure de la probabilité et pourrait être un meilleur choix pour des applications en temps réel, car elle nécessite beaucoup moins de calculs.

La distribution choisie d_p utilisée pour débruiter un i-vecteur bruité Y_0 correspond

à :

$$d_p = \underset{d_i}{\operatorname{argmin}} \{dist(Y_0, d_i) / i \in \{1, .., nb\_distribution\}\} \tag{5.16}$$

— **Distance Euclidienne :** La distance euclidienne peut être utilisée entre un i-vecteur bruité et les moyennes de toutes les distributions bruitées présentes dans la base. Pour un i-vecteur de bruit de test donné Y_0, et une distribution d'i-vecteurs bruités k : $d_k \sim \mathcal{N}(\mu_{Y_k}, \Sigma_{Y_k})$, la distance utilisée est :

$$dist_{Eucl}(Y_0, \mu_{Y_k}) = \left(\sum_{i=1}^{n} (Y_{0i} - \mu_{Y_{ki}})^2 \right)^{\frac{1}{2}} \tag{5.17}$$

— **Mesure de vraisemblance :** Utilisation de la vraisemblance d'un i-vecteur bruité Y_0 par rapport à toutes les distributions d'i-vecteurs bruités d_K est un choix naturel. Cette mesure tient compte des hyperparamètres des distributions d'i-vecteurs bruités qui la rend plus appropriée lorsqu'une connaissance à priori sur la distribution des données est disponible. Dans la pratique, il est possible d'utiliser la mesure de log-vraisemblance pour des raisons de simplicité. Pour un i-vecteur de test bruité donné Y_0, et une distribution d'i-vecteurs bruités k : $d_k \sim \mathcal{N}(\mu_{Y_k}, \Sigma_{Y_k})$, la distance peut être écrite sous la forme :

$$LLK(Y_0, d_k) = -\frac{1}{2} \ln(|\Sigma_{Y_k}|) - \frac{1}{2}(Y_0 - \mu_{Y_k})^{\mathrm{T}} \Sigma_{Y_k}^{-1} (Y_0 - \mu_{Y_k}) - \frac{p}{2} \ln(2\pi) \tag{5.18}$$

où p est la dimension de l'espace des i-vecteurs.

5.5.4 Performances en utilisant la base de distributions d'i-vecteurs bruités

Dans cette section, nous présentons d'abord les performances du système en utilisant deux mesures différentes (la distance Euclidienne et la mesure de vraisemblance) comme critère de sélection. Ensuite, nous étudions la validité de la méthode proposée dans les deux conditions d'apprentissage propres et bruités.

Dans le tableau 5.8, on compare les performances des différents systèmes :
— **I-MAP** : Système utilisant la compensation I-MAP basé sur l'algorithme décrit dans la section 5.4.3.
— **Base de distributions (BDD) + I-MAP + mesure LLK** : Système utilisant la base de distributions de bruits et la log-vraisemblance comme critère de sélection.
— **Base de distributions (BDD) + I-MAP + distance Euclidienne** : Système utilisant la base de distributions de bruits et la distance euclidienne comme critère de sélection.

Les performances sont données le tableau 5.8 en utilisant les données d'apprentissage propres et les données de test bruitées affectées par différents bruits à 4 niveaux de SNR (0dB, 5dB, 10dB et 15 dB). Le seuil de SNR utilisé dans ces expériences est : $SNR_{seuil} = 25dB$.

On peut clairement voir que l'utilisation de la mesure LLK comme critère de sélection produit le taux-d'égale-erreur le plus bas lorsque la base de distributions est

TABLE 5.8 – *Performances des systèmes pour des données d'apprentissage propres et des données de test bruitées.*

		EER			
		Système de base	**I-MAP**	**BDD + I-MAP + LLK**	**BDD + I-MAP + distance Euclidienne**
Appr. propre + test bruité	**0dB**	28.24	**14.01**	14.10	14.55
	5dB	15.94	**6.87**	6.93	7.09
	10dB	9.77	**3.84**	4.01	4.05
	15dB	4.31	**2.86**	2.88	2.92

utilisée. Mais comparé au système de base, le choix de la distance Eulidienne semble être un choix correct et semble être suffisant si un calcul rapide est nécessaire.

5.6 Combinaison itérative de techniques de débruitage d'i-vecteurs

Dans la section 5.4, on a présenté un algorithme bayésien de compensation de variabilités appelé I-MAP. En dépit de la simplicité du modèle utilisé, cette technique permet d'atteindre de bonnes performances et d'enlever une grande partie de la composante nuisible dans un i-vecteur corrompu donné. Cependant, cet algorithme n'enlève pas l'intégralité de l'information nuisible et peut, théoriquement, introduire des distorsions sur la version estimée dépendant de la qualité du modèle de bruit ($P(N)$) utilisé. Cette composante "résiduelle" pourrait être perçue comme un "bruit" qu'il est possible de modéliser et compenser. Vu que l'application de l'approche I-MAP ne garantit pas que l'hypothèse de Gaussianité soit vraie pour le bruit résiduel, on ne peut pas l'utiliser de façon itérative sur les données de test corrompues.

À la place, nous proposons d'ajouter une étape complémentaire à I-MAP en appliquant une autre approche basée sur MMSE qui utilise l'algorithme de Kabsch. Ce faisant, nous voulons atteindre deux objectifs : d'une part, nous voulons améliorer les performances de I-MAP en le combinant avec un autre algorithme qui utilise un critère d'optimisation différent (même si une technique bayésienne est généralement supérieure à un algorithme MMSE, la combinaison des deux peut surpasser chacune des approches). D'autre part, nous voulons pouvoir utiliser ces techniques (I-MAP + Kabsch) de manière itérative pour obtenir des performances encore meilleures et supprimer plus efficacement l'effet des nuisances acoustiques.

Performances en utilisant I-MAP + Kabsch

Toutes les données d'entraînement propres utilisées dans nos expériences ont une durée de parole moyenne de 90 secondes et un niveau de SNR supérieur à 25dB. En

outre, le SNR_{seuil} utilisé pour I-MAP est égal à 25dB.

Pour I-MAP, le nombre d'i-vecteurs d'entraînement N nécessaires pour estimer la distribution de bruit pour chaque bruit $\mathcal{N}(N; \mu_N, \Sigma_N)$ a été étudiée dans la sous-section 5.4.2. Nous allons utiliser $N = 500$ dans toutes nos expériences et le même ensemble de données d'entraînement seront utilisées dans l'algorithme Kabsch pour calculer la matrice de rotation R et vecteur de translation $\overline{P_Y}$ correspondant à chaque bruit.

Nous allons comparer les performances de cinq systèmes dans ces expériences (un modèle de scoring propre est utilisé pour tous les systèmes) :

— **système de base :** i-vecteurs bruités utilisés avec une PLDA propre (une description détaillée du système est donnée dans la section 4.5).

— **Algorithme Kabsch :** pour chaque bruit n / niveau SNR s :

1. Un ensemble de segments d'entraînement propres sont affectés par le bruit n au niveau SNR s dB dans le domaine temporel.
2. Les i-vecteurs bruités correspondants aux segments résultants $\{y_i\}_{i=1..N}$ et leurs homologues propres $\{x_i\}_{i=1..N}$ sont extraits.
3. Les étapes 1 et 2 de l'algorithme Kabsch sont appliquées à $\{x_i\}_{i=1..N}$ et $\{y_i\}_{i=1..N}$ et le vecteur de translation $\overline{P_Y}$ ainsi que la matrice de rotation R sont estimés.
4. L'étape 3 de l'algorithme Kabsch est appliquée sur les i-vecteurs de test bruités en utilisant R et $\overline{P_Y}$.

— **I-MAP + algorithme Kabsch (1 itération) :** pour chaque bruit n / niveau SNR s, la transformation I-MAP es appliquée aux i-vecteurs bruités d'apprentissage et de test, par la suite, l'algorithme Kabsch est appliqué :

1. Un ensemble de segments d'entraînement propres sont affectés par le bruit n au niveau SNR s dB dans le domaine temporel.
2. Les i-vecteurs bruités correspondants aux segments résultants $\{y_i\}_{i=1..N}$ et leurs homologues propres $\{x_i\}_{i=1..N}$ sont extraits.
3. L'équation 5.6 est appliquée, puis la distribution de bruit $P(N)$ est estimée.
4. I-MAP est appliqué (Équation 5.15) aux i-vecteurs de test bruités $\{t_i\}_{i=1..N}$ (générant $\{t_i^{'}\}_{i=1..N}$).
5. I-MAP est appliqué sur l'ensemble des i-vecteur d'entraînement bruités $\{y_i\}_{i=1..N}$ (générant $\{y_i^{'}\}_{i=1..N}$).
6. Les étapes 1 et 2 de l'algorithme Kabsch sont appliquées à $\{x_i\}_{i=1..N}$ et $\{y_i^{'}\}_{i=1..N}$ et le vecteur de translation $\overline{P_{Y'}}$ ainsi que la matrice de rotation R sont estimés.
7. L'étape 3 de l'algorithme Kabsch est appliquée sur les i-vecteurs bruités transformés avec I-MAP ($\{t_i^{'}\}_{i=1..N}$) en utilisant R et $\overline{P_{Y'}}$.

Les données d'apprentissage et de test ont été affectés en utilisant trois bruits (bruit de climatiseur, bruit de voiture et bruit de foule) à 4 niveaux de SNR différents : 0dB, 5dB, 10dB et 15dB.

Performances des systèmes en utilisant des données de test bruitées

Pour trois bruits de test différentes (bruit de climatiseur, le bruit de voiture et bruit de foule), les données de test propres sont corrompus dans le domaine temporel et les i-vecteurs correspondants sont évalués avant et après l'application de Kabsch, I-MAP et I-MAP + Kabsch. Le tableau 5.9 représente les performances des cinq systèmes pour différents bruits de test.

TABLE 5.9 – *Performances des 5 systèmes dans différentes conditions de test en utilisant des données d'apprentissage propres et des données de test bruités. Le nombre d'itérations indique combien de fois I-MAP et Kabsch sont été appliqués successivement.*

		EER				
Condition de test		**Système de base**	**Kabsch**	**I-MAP**	**I-MAP + Kabsch (1 itér.)**	**I-MAP + Kabsch (2 itér.)**
Bruit de climatiseur	**0dB**	26.85	17.18	13.21	8.86	**7.24**
	5dB	15.21	10.34	7.25	4.71	**3.89**
	10dB	9.51	5.70	4.85	2.94	**2.55**
	15dB	5.41	3.40	2.85	1.82	**1.63**
Bruit de voiture	**0dB**	25.54	15.83	12.05	7.91	**6.37**
	5dB	14.54	9.30	6.65	3.63	**3.04**
	10dB	8.32	5.15	3.78	1.99	**1.82**
	15dB	4.82	3.22	2.36	1.79	**1.65**
Bruit de foule	**0dB**	24.24	16.48	11.55	8.24	**6.78**
	5dB	13.94	9.20	5.09	4.18	**3.62**
	10dB	7.77	5.20	3.05	2.02	**1.81**
	15dB	4.01	2.52	2.02	1.84	**1.63**

Lorsque l'algorithme Kabsch est utilisé, une d'amélioration relative variante entre 33% et 40% est observée, alors que l'utilisation de I-MAP suivie par l'algorithme Kabsch donne une amélioration relative entre 65% à 85% par rapport au système de base.

Il est important de mettre en évidence la puissance de la combinaison de ces deux techniques. En effet, lorsque les deux algorithmes sont comparés séparément, I-MAP donne de meilleurs résultats que Kabsch en raison de sa nature bayésienne. Mais l'utilisation des deux algorithmes (soit pour une ou plusieurs itérations) peut être très efficace vu que les deux algorithmes utilisent différents critères d'optimisation (le critère MAP pour I-MAP et le critère RMSD pour Kabsch), améliorant ainsi de manière itérative la qualité du débruitage d'i-vecteurs. En outre, l'application de I-MAP produit un bruit résiduel qui ne respecte pas nécessairement l'hypothèse de Gaussianité (donc nous ne pouvons pas utiliser I-MAP itérativement sur les données de test bruités). Ce problème peut être corrigé par l'algorithme Kabsch et peut expliquer le bon ajustement des deux techniques lorsqu'ils sont utilisés plus d'une fois.

Performances des systèmes dans un contexte hétérogène

Nous avons effectué une autre expérience pour tester la validité de notre technique dans une situation où le niveau de bruit varie de façon aléatoire entre les segments d'apprentissage / test. Dans cette expérience, toutes les sessions d'apprentissage et le test sont corrompues par un bruit choisi aléatoirement parmi les bruits suivants {bruit de climatiseur, bruit de voiture et bruit de foule} avec un niveau de SNR choisi au hasard entre 0dB à 20dB. En conséquence, chaque session bruitée est affectée par un bruit unique, à un niveau SNR fixe. Le tableau 5.10 montre les résultats obtenus avec les cinq systèmes.

TABLE 5.10 – *Comparaison des performances dans une configuration hétérogène. Toutes les sessions d'apprentissage et le test sont corrompues par un bruit choisi aléatoirement parmi les bruits suivants {bruit de climatiseur, bruit de voiture et bruit de foule} avec un niveau de SNR choisi au hasard entre 0dB à 20dB.*

	EER
Système de base	29.65
Kabsch	18.78
I-MAP	16.27
I-MAP + Kabsch (1 iter.)	8.67
I-MAP + Kabsch (2 iter.)	**7.39**

Il est facile de voir que même si l'algorithme Kabsch et I-MAP améliorent respectivement les performances de 36% et 45%, la combinaison de ces deux techniques permet d'atteindre 75% dans une situation de données hétérogènes.

Conclusions

Dans ce chapitre, on a présenté deux techniques pour le traitement de variabilités dans le domaine des i-vecteurs. La première technique se base sur l'algorithme de Kabsch et modélise l'effet de la corruption acoustique dans l'espace des i-vecteurs sous forme d'une translation suivie d'une rotation. Cet algorithme est testé en présence de bruit additif ; une amélioration relative de 40% a été observée sur les données de l'évaluation NIST SRE 2008 bruitées artificiellement.

Le deuxième algorithme, appelé I-MAP, utilise le critère du maximum à posteriori. A la différence du premier algorithme, I-MAP considère la différence entre la version propre et la version corrompue d'un i-vecteur comme étant un bruit additif dans le domaine des i-vecteurs. La version propre correspondant à un i-vecteur de test corrompu est obtenue en utilisant le critère du maximum à posteriori (MAP) tout en supposant une distribution Gaussienne pour les i-vecteurs propres ainsi que pour le bruit dans l'espace des i-vecteurs. Cet algorithme a permis d'obtenir des gains pouvant atteindre 60% d'amélioration relative en termes d'EER sur les données de l'évaluation NIST SRE 2008 bruitées artificiellement et de 50% sur les données de SITW bruités naturellement.

Ces résultats confirment la possibilité de modéliser le bruit additif dans le domaine des i-vecteurs sans se baser directement sur son effet dans cet espace ou faire recours à des dérivations complexes.

Chapitre 6

Construction automatique de techniques de compensation des nuisances dans l'espace des i-vecteurs

Sommaire

6.1 Introduction et motivation

L'effet des nuisances acoustiques peut être très complexe dans l'espace des i-vecteurs en fonction de la nature des distorsions acoustiques affectant le signal d'origine. Dans les chapitres précédents, nous avons présenté un ensemble de procédures de compensation des nuisances basées sur des modèles de distorsion simples. Ceci a permis de monter qu'il est possible de créer des algorithmes efficaces tout en ignorant l'effet "réel" des nuisances acoustiques dans l'espace de variabilité totale. Ce cadre ouvre de nombreuses possibilités en ce qui concerne la modélisation et la compensation de ces nuisances dans cet espace et invite à explorer d'autres modèles de distorsion plus complexes.

Théoriquement, il est possible d'interpréter cette procédure de modélisation comme une recherche dans l'espace "algorithmes de compensation des nuisances acoustiques". À partir de cette idée, il est possible de créer un algorithme qui effectue une recherche dans cet espace et qui retourne la meilleure technique pour une nuisance donnée. Cette procédure nous permettrait d'explorer d'une manière efficace un grand ensemble d'algorithmes et donne un aperçu de la complexité de l'effet de chaque nuisance dans l'espace des i-vecteurs. Dans ce chapitre, nous introduisons un nouvel outil algorithmique basé sur la programmation génétique (PG) pour construire automatiquement des techniques de compensation de bruit dans l'espace des i-vecteurs et on détaille toutes ses composantes.

6.2 Construction d'une technique de compensation de variabilités nuisibles en utilisant la PG

6.2.1 I-MAP, un arbre dans le foret des solutions

I-MAP est une technique de compensation de bruit performante qui utilise des statistiques sur les distributions des i-vecteurs propres ainsi que la distribution de bruit $\{\mu_X, \Sigma_X, \mu_N, \Sigma_N\}$ en se basant sur un estimateur MAP. L'expression associée (équation 5.15) peut être exprimée en utilisant un arbre syntaxique comme suit :

Dans I-MAP, nous avons utilisé le bruit définit par $N = Y - X$ qui a permis d'avoir de bons résultats. Cela nous encourage à explorer d'autres relations entre les i-vecteurs propres et leurs versions bruitées pour un bruit donné. Pour ce faire, nous allons définir quatre variables aléatoires $\{N_i / i \in \{1, .., 4\}\}$ comme :

(1) $N_1 = Y - X$
(2) $N_2 = Y + X$
(3) $N_3 = exp(Y - X)$
(4) $N_4 = exp(Y) - exp(X)$

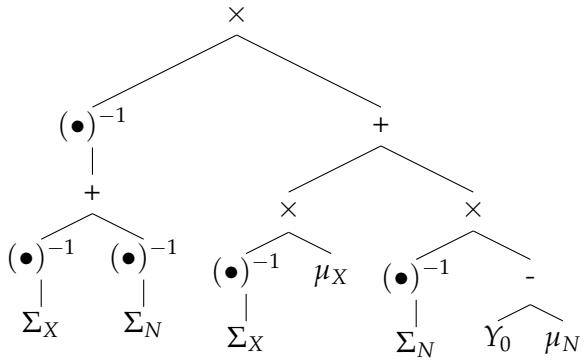

FIGURE 6.1 – *Représentation de I-MAP sous forme d'arbre.*

Chacune de ces variables aléatoires représente une relation entre les i-vecteurs propres et i-vecteurs bruités. Les statistiques associées (vecteur moyenne et matrice de covariance) pourraient être utilisées pour construire une technique de compensation ayant une structure arborescente similaire à celle de la figure 6.1. Les algorithmes évolutionnaires, et plus spécifiquement la programmation génétique (PG) (Banzhaf et al., 1998; Koza, 1992; Poli et al., 2008), donnent des outils appropriés pour la résolution de tels problèmes.

6.2.2 Recherche dans l'espace des solutions avec la PG

La PG est une méthodologie d'optimisation inspirée par le processus d'évolution biologique. C'est une spécialisation des algorithmes génétiques (GA) (Back, 1996) qui permet de générer des solutions en forme d'arbre pour des problèmes d'optimisation. Dans notre cas, il est possible d'alimenter l'algorithme de PG par l'ensemble des statistiques calculées sur $\{N_i/i \in \{1,..,4\}\}$ ainsi que les statistiques calculées sur les i-vecteurs propres $\{\mu_X, \Sigma_X\}$ en tant qu'éléments terminaux. Ces éléments seront utilisés pour construire un arbre solution. L'algorithme peut être adapté de manière à chercher la meilleure expression mathématique qui transforme un i-vecteur bruité y en sa version propre x.

Comme illustré dans la figure 6.2, l'algorithme de PG commence par générer la population initiale. Il génère un ensemble de solutions (arbres) créés au hasard. Ces arbres sont construits en utilisant un ensemble de terminaux (éléments utilisés comme feuilles d'arbres) et un ensemble d'opérations (éléments utilisés en tant que nœuds). Ces solutions sont évaluées progressivement à l'aide d'une fonction d'évaluation, puis évoluées [1] sur une série de générations d'une manière itérative (itérations de l'algorithme de PG).

L'opérateur de **croisement** "à point unique" est généralement utilisé avec des génomes en forme d'arbre. Il représente la création d'un ou deux arbres fils par recom-

1. Le processus d'évolution fait référence à la génération de nouvelles solutions à partir de solutions pré-existantes en utilisant les opérateurs de croisement et de mutation.

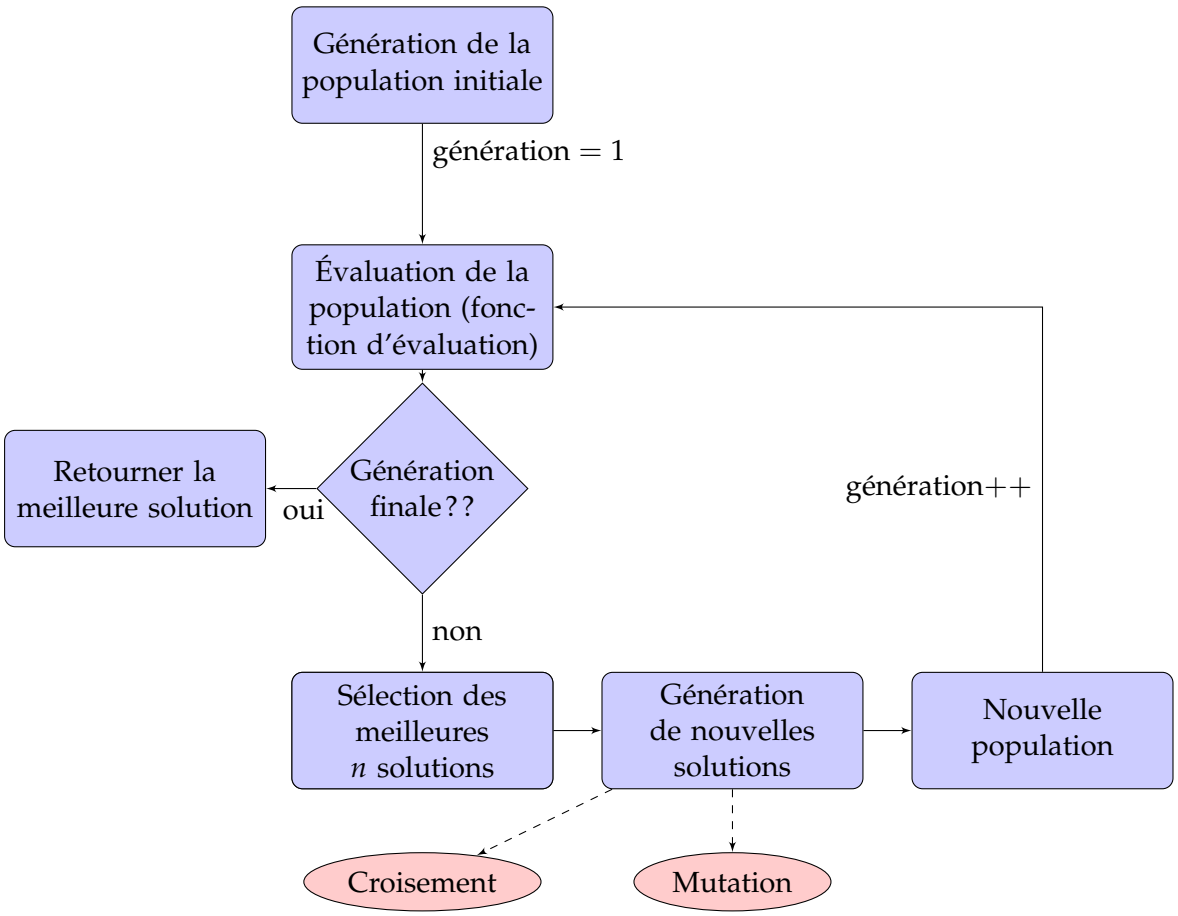

FIGURE 6.2 – *Diagramme de flux d'un algorithme génétique.*

binaison de parties choisies au hasard de deux individus sélectionnés de la population (les parents). Il est appliqué sur un individu en échangeant simplement un de ses nœuds avec un autre nœud d'un autre individu dans la population (remplacer un nœud signifie remplacer toute la branche correspondante). Cela ajoute une plus grande efficacité à l'opérateur de croisement vu que les expressions résultantes sont très différentes de leurs parents initiaux.

D'autre part, l'opérateur de **mutation** affecte une solution de la population et génère toute une nouvelle arborescence. Dans notre cas, nous utilisons l'opérateur "mutation de sous-arbre" qui remplace une branche d'un génome parent par un nouvel arbre généré aléatoirement.

6.2.3 Ensemble de terminaux

Les paramètres de distribution de $\{N_i\}$ sont estimés à l'aide d'un ensemble d'i-vecteurs d'entraînement propres X et les versions bruitées correspondantes Y affectées par un bruit donné. Le vecteur d'espérance ($\mu_i = \mu_{N_i}$) ainsi que la matrice de covariance ($\Sigma_i = \Sigma_{N_i}$) et sa décomposition de Cholesky ($\Sigma_i = \Sigma_i^{\frac{1}{2}} \Sigma_i^{\frac{1}{2}^T}$) sont calculés pour chaque $\{N_i\}$ et donnés à l'algorithme génétique en tant que partie des terminaux définis par :

$$\text{Terminaux} = \bigcup_{i=1}^{4} \{\mu_i, \Sigma_i, \Sigma_i^{\frac{1}{2}}\} \cup \{y\} \cup \{\mu_X, \Sigma_X, \Sigma_X^{\frac{1}{2}}\} \tag{6.1}$$

où y fait référence à un i-vecteur bruité.

Afin d'utiliser les informations à priori sur la distribution des i-vecteurs propres $d_X \sim \mathcal{N}(\mu_X, \Sigma_X)$ ainsi que l'information sur le bruit, nous avons ajouté une contrainte qui impose l'utilisation des terminaux qui incluent les terminaux (μ_X, Σ_X ou $\Sigma_X^{\frac{1}{2}}$) et les terminaux liés au bruit (μ_i, Σ_i or $\Sigma_i^{\frac{1}{2}} / i \in \{1, .., 4\}$) dans tous les arbres (pour éviter les expressions indépendantes du bruit).

L'un des éléments clés de notre implémentation de l'algorithme PG est que toutes les statistiques générées à partir de $\{N_1, .., N_4\}$ sont utilisées comme terminaux. De cette façon, les arbres générés par l'algorithme utilisent une combinaison des statistiques générées par $\{N_1, .., N_4\}$ au lieu de les utiliser séparément.

6.2.4 Ensemble d'opérations

Dans notre implémentation de l'algorithme PG, nous avons défini plusieurs versions de chaque opération afin de permettre son utilisation sur différents types d'entrées (scalaires, vectorielles ou matricielles). L'ensemble des opérations utilisées dans nos expériences peut être décomposé en opérations unaires, binaires et ternaires :

— Opération unaires = $\{minus(a), trace(a), invert(a), transpose(a)\}$

— Opération binaires $= \{add(a,b), sub(a,b), mul(a,b), quad(a,b)\}$
— Opération ternaires $= \{add3(a,b,c), mul3(a,b,c), centerAndScale(a,b,c)\}$

$$\text{Opérations} = \text{Opération unaires} \cup \text{Opération binaires} \cup \text{Opération ternaires} \quad (6.2)$$

Le comportement de chaque opération dépend des types de ses opérandes. Les opérations unaires et binaires sont définies respectivement dans les tableaux 6.1 et 6.2 et les opérations ternaires sont définies comme suit :

— $add3(a,b,c) = add(a, add(b,c))$
— $mul3(a,b,c) = mul(a, mul(b,c))$
— $centerAndScale(a,b,c) = mul(invert(c), sub(a,b))$
— $quad(a,b) = mul3(transpose(a), invert(b), a)$

TABLE 6.1 – *Comportement des opérations unaires utilisées dans l'algorithme PG en fonction des types d'opérandes. V_1, M_1 et S_1 représentent respectivement un vecteur de dimension D, une matrice $D \times D$ et un scalaire. J_D représente une matrice $D \times D$ de uns et $J_{1,D}$ représente un vecteur de uns de dimension D ($J_{D,1} = {J_{1,D}}^T$).*

	Scalaire (S_1)	**Vecteur** (V_1)	**Matrice** (M_1)
minus()	$= -S_1$	$= -V_1$	$= -M_1$
invert()	$= \begin{cases} 1/S_1, & \text{si } S_1 \neq 0 \\ S_1, & \text{sinon} \end{cases}$	$= V_1$	$= \begin{cases} M_1^{-1}, & \text{si } M_1 \text{ inversible} \\ M_1, & \text{sinon} \end{cases}$
trace()	$= S_1$	$= trace(diag(V_1))$	$= trace(M_1)$
transpose()	$= S_1$	$= {V_1}^T$	$= {M_1}^T$

TABLE 6.2 – *Comportement des opérations binaires utilisées dans l'algorithme PG selon le type des opérandes. V_i, M_i, S_i avec $i = 1..2$ représentent respectivement un vecteur de dimension D, une matrice $D \times D$ et un scalaire. J_D représente une matrice $D \times D$ de uns et $J_{1,D}$ représente un vecteur de uns de dimension D ($J_{D,1} = {J_{1,D}}^T$).*

	Scal./Scal. (S_1, S_2) **Vect./Vect.** (V_1, V_2) **Mat./Mat.** (M_1, M_2)	**Scal./Mat.** (S_1, M_2)	**Scal./Vect.** (S_1, V_1)	**Vect./Mat.** (V_1, M_2)
add()	$= S_1 + S_2$ $= V_1 + V_2$ $= M_1 + M_2$	$= S_1 \times J_D + M_2$	$= S_1 \times J_{1,D} + V_2$	$= J_{D,1} \times V_1 + M_2$
sub()	$= S_1 - S_2$ $= V_1 - V_2$ $= M_1 - M_2$	$= S_1 \times J_D - M_2$	$= S_1 \times J_{1,D} - V_2$	$= J_{D,1} \times V_1 - M_2$
mul()	$= S_1 \times S_2$ $= V_1 \cdot V_2$ $= M_1 \times M_2$	$= S_1 \times M_2$	$= S_1 \times V_2$	$= V_1 \times M_2$

Dans les tableaux 6.1 et 6.2, $V_1 \cdot V_2$ qualifie le produit scalaire entre V_1 et V_2. $J_{D,1} \times V_1 + M_2$ signifie que nous ajoutons le vecteur V_1 à chaque ligne de la matrice M_2 et $trace(diag(V_1))$ signifie que nous calculons la trace de la matrice ayant la diagonale V_1.

Vu que les vecteurs moyenne et les matrices de covariance (ou leur décomposition de Cholesky) sont utilisés comme terminaux en plus des i-vecteurs y bruités, aucune vérification de dimension n'est nécessaire. Pour les opérations nécessitant un alignement d'opérandes (comme la multiplication), le second opérande est transposé si les dimensions ne correspondent pas.

6.2.5 Fonction d'évaluation

La fonction d'évaluation (aussi appelée fonction de *fitness*) est l'une des composantes les plus importantes dans un algorithme de PG vu qu'elle guide la procédure d'optimisation. Elle est utilisée comme heuristique dans le processus de recherche et aide l'algorithme PG à décider si une certaine solution doit être conservée ou non dans les générations suivantes. Puisque l'objectif de l'algorithme PG dans notre travail est de débruiter les i-vecteurs, le critère d'optimisation utilisé est : Minimiser la distorsion moyenne entre les i-vecteurs transformés et leurs versions propres.

Soit Z une variable aléatoire représentant la sortie d'un arbre donné (i-vecteurs transformés). Pour évaluer un arbre solution, les i-vecteurs Y bruités sont transformés en utilisant l'expression correspondante ($Z = f(Y)$). Ensuite, nous calculons la distorsion moyenne entre les i-vecteurs propres et les versions transformées.

$$fonctionEvaluation(Z) = distortion(X, Z) = \frac{1}{n}\sum_{i=1}^{n} Eucl(X_i, Z_i) \tag{6.3}$$

n est le nombre d'i-vecteurs d'entraînement et $Eucl(a, b)$ calcule la distance euclidienne entre deux vecteurs a et b. L'algorithme PG vise à diminuer cette valeur à travers les générations.

6.2.6 Simplification d'expressions et vectorisation de la sortie

Dans notre implémentation de l'algorithme PG, nous avons ajouté une étape supplémentaire à l'algorithme avant l'étape d'évaluation afin d'éviter les calculs inutiles. Avant d'évaluer un arbre solution, l'expression correspondante est simplifiée et le type de sa sortie est vérifié.

Pour une matrice M et un vecteur V, ces règles ont été implémentées :

(1) $invert(invert(M)) = M$
(2) $transpose(transpose(M)) = M$
(3) $transpose(transpose(V)) = V$
(4) $invert(M) \times M = I_D$
(5) $invert(V) = V$

où I_D dénote une matrice d'identité définie dans l'espace des i-vecteurs.

Comme le type de la sortie de chaque arbre dépend de l'ensemble des terminaux utilisés comme feuilles et des opérations utilisées comme nœuds, la sortie de chaque programme doit être "vectorisée" pour pouvoir être interprétée comme i-vecteur. De cette façon, aucun arbre solution n'est rejeté et toute la population est évaluée à la fin de chaque génération. Pour cela, nous multiplions l'expression correspondant à l'arbre solution par l'i-vecteur original y et renvoyons le résultat comme i-vecteur transformé. En effet, multiplier la sortie de l'arbre (qu'il s'agisse d'un scalaire ou d'une matrice) par y garantit une sortie en forme de vecteur. L'algorithme 2 détaille la procédure d'évaluation d'un arbre.

Algorithm 2: Procédure d'évaluation d'un arbre

```
1: function EVALUER(arbre, entrées)
2:     expression ← simplifier(arbre)
3:     sortie ← evaluerExpression(expression, entrées)
         si(type(sortie) == scalaire) ou (type(sortie) == matrice)
4:     sortie ← y × sortie
         finSi
5:     retourner fonctionEvaluation(sortie)
6: end function
```

Les *entrées* de la fonction d'évaluation représentent le sous-ensemble de terminaux utilisé dans l'arbre et la fonction d'évaluation évalue la qualité de la sortie en calculant la distorsion entre les i-vecteurs transformés et leurs versions propres (voir sous-section 6.2.5).

6.3 Expériences et résultats

Afin d'évaluer cet algorithme, nous avons utilisé 2 types de bruit (bruit de foule et bruit de voiture) pour générer de nouveaux segments bruités pour chaque bruit / niveau de SNR. Toutes les données d'entraînement propres utilisées dans les expériences suivantes ont une durée de parole moyenne de 2 minutes et un niveau de SNR supérieur à 25dB. Un ensemble de 500 sessions d'entraînement propres sont utilisées pour estimer les statistiques $\{\mu_{N_i}, \Sigma_{N_i} / i \in \{1, .., 4\}\}$ pour chaque bruit ($n = 500$). Les hyper-paramètres de distribution i-vecteurs propres Σ_X et μ_X sont estimés une fois dans une étape *off-line* avant tout test en utilisant un ensemble de 6000 sessions d'entraînement propres.

L'algorithme de programmation génétique est implémenté en se basant sur la configuration spécifiée dans le tableau 6.3.

TABLE 6.3 – *Configuration de l'algorithme PG.*

Paramètres de l'algorithme PG	Valeur
Taux de croisement	0.9
Taux de mutation	0.02
Méthode de génération d'arbres	half-and-half ramped (Koza, 1992)
Profondeur maximale des arbres	6
Taille de la population	1000
Nombre de générations	30
Nombre d'exécutions pour l'algorithme PG	20

6.3.1 Premiers résultats en utilisant l'algorithme PG

Comme première expérience, nous lançons l'algorithme PG en utilisant les données d'entraînement affectées par un bruit et surveillons l'évolution de la meilleure valeur de fitness sur chaque génération (distorsion minimale) sur 20 exécutions différentes. Les figures 6.3 et 6.4 montrent respectivement l'évolution de la meilleure valeur donnée par la fonction d'évaluation avec le nombre de génération quand on l'utilise sur le bruit de voiture à 5dB et le bruit de foule à 0dB (la moyenne des 5 meilleures exécutions est montrée).

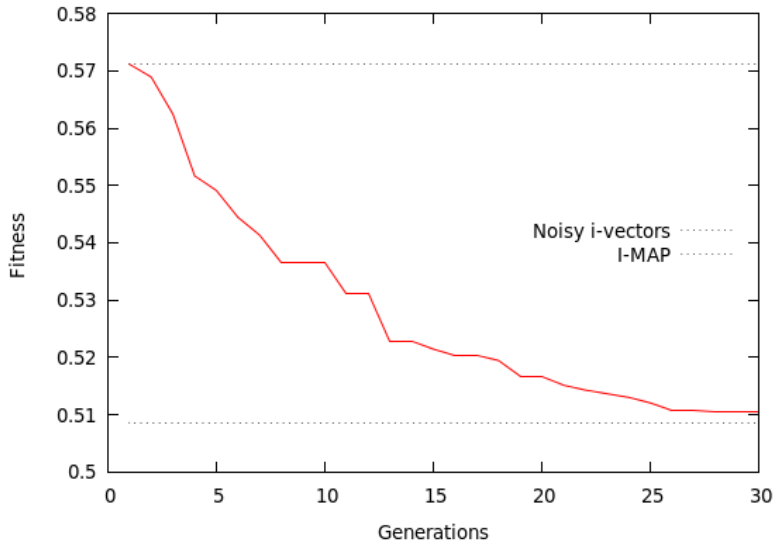

FIGURE 6.3 – *Évolution de la meilleure valeur de fitness avec le nombre de génération pour le bruit de voiture à 5dB (moyenne sur les 5 meilleures exécutions).*

Pour chaque bruit η, la procédure suivante est utilisée :

1. Le bruit est rajouté à l'ensemble d'enregistrements d'entraînement propres dans le domaine temporel à un niveau SNR donné ($n = 500$ fichiers d'entraînement).

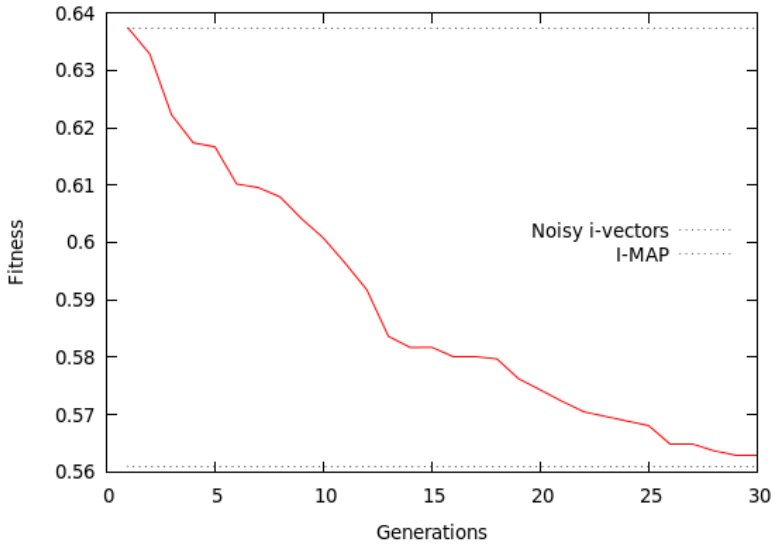

FIGURE 6.4 – *Évolution de la meilleure valeur de fitness avec le nombre de génération pour le bruit de foule à 0dB (moyenne sur les 5 meilleures exécutions).*

2. Les i-vecteurs bruités correspondants (Y_η) sont extraits.
3. Un ensemble de terminaux est généré pour ce bruit en utilisant les statistiques des variables aléatoires $\{N_1, .., N_4\}$ définis dans la sous-section 6.2.1 (en calculant l'espérance, les matrices de covariance et leurs décompositions de Cholesky), les paramètres de la distribution des i-vecteurs propres ainsi que l'i-vecteur bruité y qu'on veut débruiter.

$$Terminaux_\eta = \{\mu_1, \Sigma_1, {\Sigma_1}^{\frac{1}{2}}, .., \mu_4, \Sigma_4, {\Sigma_4}^{\frac{1}{2}}\}_\eta \cup \{\mu_X, \Sigma_X, {\Sigma_X}^{\frac{1}{2}}\} \cup \{y\} \tag{6.4}$$

4. L'algorithme PG est lancé (20 exécutions pour chaque bruit) : dans cette étape, toutes les statistiques générées à partir de $\{N_1, .., N_4\}$ sont utilisées dans l'ensemble de terminaux. De cette façon, les arbres générés par l'algorithme utilisent une fusion des statistiques générées par $\{N_1, .., N_4\}$.

Comme le montre la figure 6.3 et 6.4, aucun arbre n'a pu surpasser I-MAP en termes de distorsion minimale dans tous nos essais. Cependant, la meilleure solution trouvée pour chaque bruit après 20 exécutions de l'algorithme PG correspondait exactement à l'expression de I-MAP.

6.3.2 Au-delà de I-MAP

Dans une deuxième étape, nous appliquons I-MAP sur toutes nos données bruitées et utilisons l'algorithme PG sur la sortie (nous considérons la sortie d'I-MAP comme des i-vecteurs bruités devant être "re-débruitées"). Dans cette configuration, l'algorithme PG tentera de trouver une technique de compensation qui complémente I-MAP et minimise la distorsion des i-vecteurs transformée par rapport à leurs versions propres.

Pour le "bruit de voiture" à 5dB, la meilleure valeur de fitness correspond à l'expression :

$$f_1(y) = add(mul2(trace(\Sigma_2), \Sigma_X^{\frac{1}{2}}), sub(y, \mu_X)) \tag{6.5}$$

$$= trace(\Sigma_2) \times y \times \Sigma_X^{\frac{1}{2}} + (y - \mu_X) \tag{6.6}$$

Pour le "bruit de foule" à 0dB, la meilleure valeur de *fitness* correspond à l'expression :

$$f_2(y) = add3(mul(mul(y, \mu_3), minus(sub(y, \mu_X))), mul(y, \Sigma_X), sub(y, \mu_4)) \tag{6.7}$$

$$= -(y.\mu_3) \times (y - \mu_X) + (y \times \Sigma_X) + (y - \mu_4) \tag{6.8}$$

Le tableau 6.4 résume les EER donnés par chaque expression sur différents niveaux de SNR. Même si elle est efficace, cette approche implique qu'une expression différente doit être générée pour chaque bruit / niveau de SNR différent. Afin de résoudre ce problème et d'utiliser une seule expression sur de nombreux bruits / niveaux de SNR, nous présentons une autre expérience qui utilise des statistiques provenant de différents bruits.

TABLE 6.4 – *EER donné par la sortie de l'algorithme PG (f_1 et f_2) comparé à I-MAP et au système de base.*

	Bruit de voiture			**Bruit de foule**		
	Système de base	**I-MAP**	**Sortie de f_1**	**Système de base**	**I-MAP**	**Sortie de f_2**
0dB	25.54	12.05	**8.54**	24.24	11.55	**7.05**
5dB	14.54	6.65	**3.72**	13.94	5.09	**4.29**
10dB	8.32	3.78	**2.51**	7.77	3.05	**2.34**

6.3.3 Une expression pour plusieurs bruits

Afin de générer une expression utilisable sur différents bruits, nous utilisons dans l'algorithme PG des statistiques calculées sur des données provenant de 4 bruits / niveaux de SNR différents (bruit de voiture et bruit de foule à 0dB et 10dB). Dans cette configuration, 4 ensembles de terminaux différents sont générés (un pour chaque bruit / niveau de SNR) et la sortie de I-MAP est utilisée comme "i-vecteurs bruités". Après chaque évaluation d'arbre, 4 valeurs de distorsion sont générées (une pour chaque bruit / niveau de SNR) et optimisées simultanément.

Après 20 exécutions, la meilleure valeur de fitness correspond à l'expression :

$$f_3(y) = quad(\mu_X, add3(mul3(\Sigma_X^{\frac{1}{2}}, \Sigma_4^{\frac{1}{2}}, sub(y, \mu_2)), \tag{6.9}$$

$$mul(sub(y, \mu_3), sub(y, \mu_3)), \Sigma_X^{\frac{1}{2}})) \tag{6.10}$$

Le tableau 6.5 résume les EER donnés par f_3 sur chaque bruit / niveau de SNR. Il est facile de voir que f_3 améliore les EER par rapport à I-MAP et au système de base atteignant jusqu'à 80% d'amélioration relative.

TABLE 6.5 – *EER donné par la sortie de l'algorithme PG (f_3) comparé à I-MAP et au système de base.*

	Bruit de voiture			**Bruit de foule**		
	Système de base	**I-MAP**	**Sortie de f_3**	**Système de base**	**I-MAP**	**Sortie de f_3**
0dB	25.54	12.05	**8.64**	24.24	11.55	**5.23**
5dB	14.54	6.65	**4.01**	13.94	5.09	**4.35**
10dB	8.32	3.78	**2.01**	7.77	3.05	**2.41**

Conclusions

Dans ce chapitre, on a proposé une méthode de construction de techniques de débruitage qui opère dans le domaine des i-vecteurs. D'abord, plusieurs relations sont posées entre les i-vecteurs propres et leurs versions bruitées. Par la suite, une distribution de bruit est construite à partir de chacun de ces modèles. Les statistiques collectées sur ces distributions sont par la suite passés à un algorithme de programmation génétique (PG). Étant donné cet ensemble de statistiques et un i-vecteur bruité, l'algorithme PG construit un arbre syntaxique qui représente une technique de débruitage dans l'espace des i-vecteurs. Un processus itératif de sélection évolutive inspiré de l'évolution Darwinienne est par la suite utilisé ; génération d'un ensemble d'arbres solution, d'évaluation des solutions, sélection des meilleurs arbres, combinaison des solution par croisement ou génération de nouvelles solutions par mutation. Ce processus est répété itérativement pour la génération de nouvelles solution. Les meilleures solutions sont finalement gardées.

Nos expériences sur le bruit additif ont montré qu'il est possible de construire des techniques de compensation de bruit dans le domaine des i-vecteurs, et que ces techniques pourraient être apprises sur un ensemble de bruits et utilisés sur des bruits différents en phase de test. Cependant, ces algorithmes peuvent être sensibles au bruit et au niveau de SNR. Un résultat intéressant dans cette technique a été la re-découverte de l'algorithme I-MAP par l'algorithme génétique. De meilleures techniques de débruitage pourraient éventuellement être découvertes si un plus grand nombre de relations, bruit et niveaux de SNR sont utilisés en phase d'entraînement.

Chapitre 7

Une méthode générique pour la compensation de variabilités nuisibles

Sommaire

7.1 Introduction et motivations

Dans les chapitres précédents, nous avons présenté plusieurs techniques de compensation de nuisances acoustiques qui opèrent dans le domaine des i-vecteurs dont I-MAP. En dépit de son efficacité, l'algorithme I-MAP suppose une indépendance statistique entre les distributions des i-vecteurs propres et leurs versions corrompues. Dans ce chapitre, nous développons une nouvelle technique qui corrige ce défaut et n'utilise pas une relation spécifique entre les i-vecteurs propres et leurs versions corrompues.

Ce nouvel algorithme estime directement la distribution jointe entre les deux représentations. Cette distribution est par la suite intégrée dans un estimateur MMSE dans la phase de test pour calculer une version "améliorée" des i-vecteurs de test corrompus.

Nous présentons d'abord le formalisme derrière cette technique puis nous l'utilisons pour traiter deux types de nuisances : le bruit additif ainsi que les courtes durées respectivement dans les sections 7.3 et 7.4. Dans ce chapitre, les i-vecteurs affectés par une nuisance acoustique sont appelés *i-vecteurs corrompus* et les i-vecteurs correspondant à de longues sessions propres sont appelés *i-vecteurs de bonne qualité*.

7.2 Compensation de variabilités nuisibles dans l'espace des i-vecteurs en utilisant la modélisation jointe

Nous définissons deux variables aléatoires x et y représentant respectivement des i-vecteurs de bonne qualité et leurs versions corrompues (affectés par une certaine nuisance acoustique) avec M la dimension de l'espace des i-vecteurs. Nous définissons une troisième variable aléatoire appelée z et définie comme étant la concaténation de x et y :

$$z = \begin{pmatrix} x \\ y \end{pmatrix} \tag{7.1}$$

Cette variable est définie dans un espace de dimension $2M$ et peut être modélisée par un mélange de Gaussiennes :

$$P(z) = \sum_{k=1}^{K} c_k \mathcal{N}(z; \mu_{z,k}, \Sigma_{z,k}) \tag{7.2}$$

où :

— K est le nombre de composantes du GMM.
— c_k est le poids de la $k^{\text{ème}}$ Gaussienne.
— $\mu_{z,k}$ correspond au vecteur moyenne de la $k^{\text{ème}}$ composante.
— $\Sigma_{z,k}$ correspond à la matrice de covariance plaine de la $k^{\text{ème}}$ composante.

Ce GMM représente la distribution jointe entre les i-vecteurs propres et les i-vecteurs corrompus pour chaque Gaussienne k. Pour chaque composante, il est possible d'écrire la moyenne $\mu_{z,k}$ et la matrice de covariance $\Sigma_{z,k}$ comme suit :

$$\mu_{z,k} = \begin{pmatrix} \mu_{x,k} \\ \mu_{y,k} \end{pmatrix} \tag{7.3}$$

$$\Sigma_{z,k} = \begin{pmatrix} \Sigma_{xx,k} & \Sigma_{xy,k} \\ \Sigma_{yx,k} & \Sigma_{yy,k} \end{pmatrix} \tag{7.4}$$

où : $\Sigma_{yx,k} = \Sigma_{xy,k}^T$ et $\Sigma_{xy,k}$ modélise l'information jointe entre les deux représentations (i-vecteurs propres et i-vecteurs corrompus).

Le problème de compensation des nuisances acoustiques peut être formulé en utilisant un estimateur MMSE. Pour un i-vecteur de test corrompu donné y_0, la version propre correspondante peut être estimée par :

$$\begin{aligned}\hat{x} = E[x|y_0] &= \int_x P(x|y_0)x dx = \sum_k \int_x P(x,k|y_0)x dx \\ &= \sum_k P(k|y_0) \int_x P(x|k,y_0)x dx = \sum_k P(k|y_0)E[x|k,y_0]\end{aligned} \tag{7.5}$$

Pour chaque composante k, l'espérance $E[x|k,y_0]$ peut être écrite sous la forme (la démonstration de cette expression est détaillée dans l'annexe B) :

$$E[x|k,y_0] = \mu_{x,k} + \Sigma_{xy,k}\Sigma_{yy,k}^{-1}(y_0 - \mu_{y,k}) \tag{7.6}$$

La solution finale peut être alors écrite sous la forme :

$$\begin{aligned}\hat{x} &= \sum_k P(k|y_0)E[x|k,y_0] \\ &= \sum_k P(k|y_0)(\mu_{x,k} + \Sigma_{xy,k}\Sigma_{yy,k}^{-1}(y_0 - \mu_{y,k}))\end{aligned} \tag{7.7}$$

L'équation 7.7 peut être ré-écrite comme :

$$\hat{x} = \sum_{k=1}^{K} P(k|y_0)(F_k y_0 + g_k) \tag{7.8}$$

avec :

$$F_k = \Sigma_{xy,k}\Sigma_{yy,k}^{-1} \tag{7.9}$$

$$g_k = \mu_{x,k} - \Sigma_{xy,k}\Sigma_{yy,k}^{-1}\mu_{y,k} \tag{7.10}$$

En d'autres termes, cet algorithme est une somme pondérée de transformations linéaires. Ces transformations représentent la contribution de chaque composante k du GMM qui représente la distribution de P(z)=$P(x,y)$. Le poids correspond à la probabilité à posteriori $P(k|y)$ et chaque transformation linéaire est construite en utilisant les hyper-paramètres de chaque composante. La figure 7.1 donne un aperçu de la structure d'un système de compensation des nuisances basé sur la modélisation jointe.

Note :

Il est important de noter que l'expression finale donnée par cette modélisation (équation 7.7) n'utilise pas la matrice Σ_{xx} qui représente la covariance des i-vecteurs de bonne qualité. Cependant, cette matrice est utilisée pour estimer la matrice de covariance $\Sigma_{x|y}$ donnée par (annexe B) :

$$\Sigma_{x|y} = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} \tag{7.11}$$

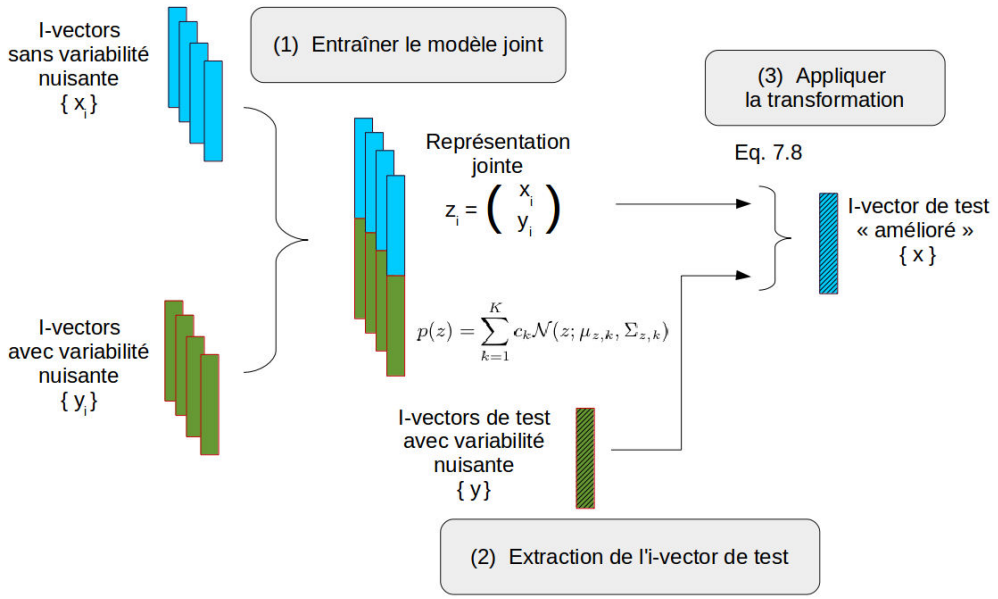

FIGURE 7.1 – *Aperçu du système basé sur la modélisation jointe pour la compensation des variabilités nuisibles dans l'espace des i-vecteurs. Après la génération de paires d'i-vecteurs correspondant à des sessions de bonne qualité et à leurs homologues corrompus, l'algorithme suit trois étapes :* ***(1) Entraînement du modèle joint :*** *où la distribution jointe est construite en utilisant la concaténation des i-vecteurs d'entraînement de bonne qualité et leurs homologues corrompus ;* ***(2) Extraction de l'i-vecteur de test :*** *où l'i-vecteur correspondant à des sessions de test affectées par une variabilité nuisible est extrait ;* ***(3) Appliquer la transformation :*** *L'équation 7.8 est utilisée pour transformer l'i-vecteur de test en utilisant la distribution jointe entraînée $P(z)$.*

Vu que la technique proposée dans cette section fournit une estimation ponctuelle $\hat{x} = E[x|y]$, il serait possible d'étendre cette présentation pour représenter la distribution conditionnelle des i-vecteurs propres étant donné une observation corrompue y; $P(x|y) = \mathcal{N}(\mu_{x|y} = E[x|y], \Sigma_{x|y})$. Suivant cette modélisation, le calcul de score entre les versions "débruités" de deux i-vecteurs y_1 et y_2 seraient équivalent à une mesure de similarité entre les deux distributions $P(x|y_1)$ et $P(x|y_2)$.

7.3 Utilisation d'un modèle joint d'i-vecteurs propres et bruités pour compenser le bruit additif

Dans (Ben Kheder et al., 2016d), on a utilisé l'algorithme présenté dans la section 7.2 dans le contexte de bruit additif où les *i-vecteurs corrompus* correspondent à des i-vecteurs bruités et les *i-vecteurs de bonne qualité* correspondent à des i-vecteurs propres. Dans ce contexte, nous tirons parti de la reproductibilité du bruit additif dans le domaine temporel afin de générer les ensembles appariés d'i-vecteurs propres et bruités.

Contrairement à I-MAP, l'algorithme proposé dans ce chapitre n'est pas basé sur un modèle de distorsion car il ne suppose pas de relation spécifique entre i-vecteurs bruités et leurs versions propres. De plus, il ne suppose pas l'indépendance statistique entre les distributions des i-vecteurs propres et i-vecteurs bruités (comme pour I-MAP), et permet de tirer parti de l'information jointe entre les deux représentations.

Dans les paragraphes suivants, nous testons la technique proposée dans les contextes de données d'apprentissage propres et test bruitées, puis nous analysons la quantité de données d'entraînement nécessaire pour obtenir les meilleurs résultats. Enfin, nous présentons une implémentation réelle de cet algorithme en l'utilisant pour compenser plusieurs bruits "non-observés" et on compare sa performance à celle du modèle de scoring générique *multi-style*.

7.3.1 Systèmes utilisés et résultats préliminaires

Nous commençons par tester la technique de compensation de bruit proposée sur des données d'apprentissage propres et des données de test bruitées. Dans cette expérience, les segments de test sont affectés par le bruit de voiture et le bruit de climatiseur à 0dB, 5dB, 10dB et 15dB.

Nous comparons les performances de 4 systèmes :
— **Système de base :** Un backend PLDA propre est utilisé (aucune compensation de bruit n'est effectuée).
— **Système *multi-style* :** Un backend *multi-style* est appris sur un ensemble de bruits et niveaux de SNR différents de ceux utilisés pour affecter les données de test (la description détaillée de ce système est fournie dans la section 4.5.3).

— **Système I-MAP :** Les i-vecteurs bruités de test sont débruités en utilisant I-MAP comme décrit dans la section 5.4 (500 i-vecteurs d'entraînement sont générés pour estimer la distribution de bruit pour chaque bruit de test), puis scoré à l'aide d'un backend PLDA propre.
— **Système ModeleJoint :** Pour chaque bruit $\mathcal{N}$ et niveau SNR $\mathcal{S}$:

1. Les données d'entraînement propres (15660 sessions) sont affectés par le bruit $\mathcal{N}$ à $\mathcal{S}$ dB.
2. Les i-vecteurs sont extraits pour les sessions propres $\{x_i\}$ et leurs versions bruitées $\{y_i\}$.
3. La distribution $P(z)$ est estimée pour un nombre différent de composantes ; $K \in \{1, 2, 3\}$.
4. Chaque i-vecteur d'apprentissage / test affecté par $\mathcal{N}$ à $\mathcal{S}$dB est débruité en utilisant l'équation 7.8.

Ensuite, les i-vecteurs résultants sont scorés en utilisant un backend PLDA propre.

Le tableau 7.1 montre les performances de ces 4 systèmes pour des données de test bruitées. Dans ces expériences, I-MAP et le ModeleJoint sont appris en utilisant des données affectées par le bruit de test alors qu'un ensemble différent de bruits est utilisé pour l'entraînement du système *multi-style*.

TABLE 7.1 – *Les performances des 4 systèmes (système de base, multi-style, I-MAP et ModeleJoint) pour des données d'apprentissage propre et des données de test bruitées. Le bruit de test est utilisé pour apprendre I-MAP et le ModeleJoint tandis qu'un ensemble de bruits différent est utilisé pour l'apprentissage du système multi-style.*

		EER					
		Système de base	Système Multi-style	I-MAP	ModeleJoint		
Condition de test					1 Gauss.	2 Gauss.	3 Gauss.
Bruit de voiture	**0dB**	11.15	8.64	4.99	**3.87**	8.03	9.11
	5dB	5.90	3.98	2.96	**2.28**	4.36	4.96
	10dB	3.64	2.67	2.28	**1.82**	2.49	2.80
	15dB	2.54	2.14	2.03	**1.80**	1.89	2.12
Bruit de climatiseur	**0dB**	11.84	8.69	4.78	**3.47**	7.59	8.81
	5dB	6.80	4.85	3.64	**2.74**	5.01	5.67
	10dB	4.11	3.21	2.93	**2.31**	3.02	3.38
	15dB	3.18	2.30	2.23	**1.82**	2.22	2.53

Il est facile de voir que la technique de compensation de variabilités basée sur la modélisation jointe proposée permet d'atteindre jusqu'à 80% d'amélioration relative par rapport aux performances données par le système de base. Ce système surpasse également le backend *multi-style* (50% d'amélioration relative de l'EER) et I-MAP (60% d'amélioration relative de l'EER). Nous observons que l'apprentissage de $P(z)$ en utilisant plus d'une Gaussienne détériore les résultats (comparée à la version à une seule Gaussienne). Ces résultats étaient attendus en raison de la Gaussianité de la distribution des i-vecteurs (Dehak et al., 2011).

Un grand ensemble d'i-vecteurs d'entraînement a été utilisé dans cette expérience pour estimer $P(z)$. Cette contrainte n'est pas pratique dans les applications réelles. Dans ce qui suit, nous essayons de trouver la quantité minimale de données nécessaire pour obtenir des performances optimales en utilisant le modèle proposé.

7.3.2 Effet de la quantité de données d'entraînement utilisée pour apprendre le modèle joint sur les performances

Nous utilisons dans les expériences suivantes la modélisation à une Gaussienne puisqu'elle donne les meilleurs résultats et on étudie la quantité de données nécessaires au modèle joint pour obtenir de bons résultats. La figure 7.2 présente les performances du système ModeleJoint par rapport au système de base lorsqu'il est utilisé sur des données de test bruitées (affectées par le bruit de voiture à 0dB, 5dB, 10dB et 15dB).

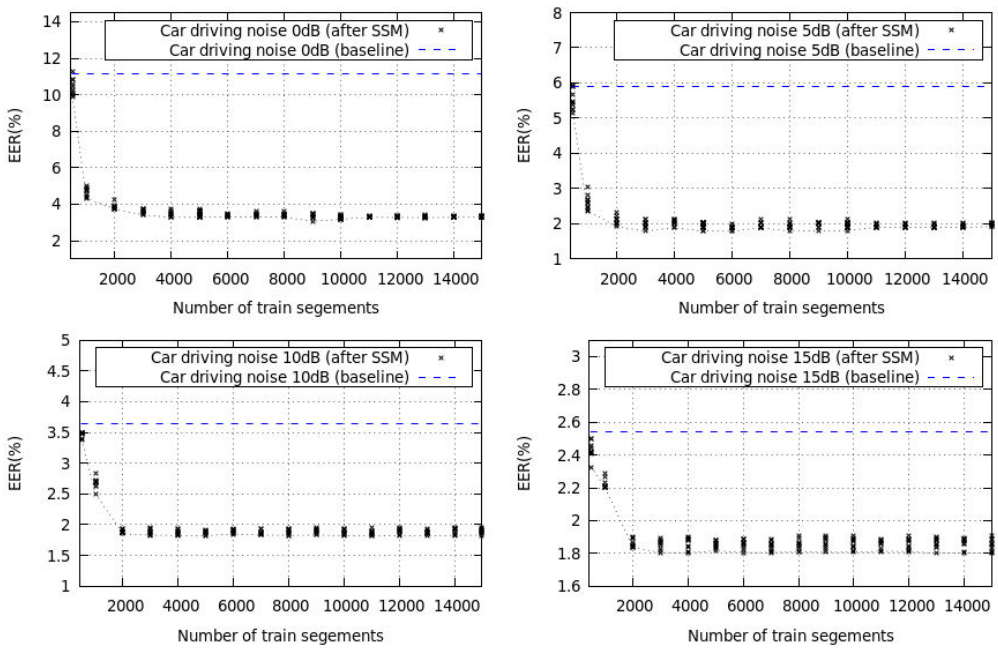

FIGURE 7.2 – *Variation du EER avec la quantité des i-vecteurs utilisés pour entraîner $P(z)$ pour le bruit de voiture à 0dB, 5dB, 10dB et 15dB (10 mesures pour chaque quantité de segments).*

Il est clair à partir de la figure que plus de 3000 paires d'i-vecteurs sont nécessaires pour estimer $P(z)$ afin d'obtenir des résultats optimaux en utilisant le modèle joint. Il est important de mentionner que l'algorithme I-MAP requiert seulement 500 sessions pour en phase d'entraînement. La différence entre les deux modèles (6 fois plus de données nécessaires pour construire le ModeleJoint) peut être expliquée par le nombre de paramètres estimés par chaque algorithme. En effet, étant donné que $M(M+3)/2$ paramètres sont nécessaires pour estimer chaque distribution Gaussienne (où M représente la dimension de l'espace des i-vecteurs), I-MAP nécessite l'estimation de $M(M+3)$ paramètres (correspondant aux deux distributions $P(X)$ et $P(N)$) alors que le ModeleJoint

nécessite l'estimation de $K \times (M(2M+3)+1)$ paramètres (où K représente le nombre de composantes de $P(z)$).

Dans les applications réelles, le temps est un facteur très important et générer 3000 i-vecteurs pour chaque bruit de test n'est pas réalisable. Cela nous motive à utiliser un seul modèle générique pour compenser plusieurs bruits. Cette expérience sera effectuée dans les sous-sections suivantes afin d'avoir une meilleure évaluation des performances du système ModeleJoint dans le contexte de bruits non-observés (bruits inconnus).

7.3.3 Utilisation d'un modèle générique avec des bruits non-observés

Nous appliquons l'approche de modélisation jointe dans le contexte de bruits de test inconnus et évaluons les performances de cette technique lorsqu'on l'utilise pour traiter différents bruits. Pour ce faire, cinq bruits (bruit d'applaudissements, sonnerie, bruit de fond d'une station de bus, bruit de vagues et bruit de tempêtes) sont utilisés pour affecter des segments d'entraînement distincts aléatoirement à différents niveaux de SNR entre 0dB et 15dB (un bruit et un niveau SNR sont utilisés pour chaque segment) et les i-vecteurs résultants sont utilisés pour construire une version générique du système ModeleJoint.

Le tableau 7.2 compare les performances du système de base et du ModeleJoint lorsqu'elle est apprise en utilisant le bruit de test et la version générique. Une amélioration relative des EER atteignant jusqu'à 70% est observée en utilisant les versions génériques du modèle développé, surpassant d'une manière consistante le backend PLDA *multi-style*, bien que les mêmes i-vecteurs soient utilisés pour apprendre les deux modèles. Ces résultats sont surprenants puisque les bruits utilisés pour estimer $P(z)$ sont différents de ceux présents dans les segments de test. Cela peut être expliqué par la nature des bruits utilisés dans nos expériences (les bruits utilisés en entraînement, en apprentissage et en test) qui partagent une caractéristique commune et peuvent être décrits comme des bruits de "basses fréquences". Cela permet au modèle générique de capturer des informations communes entre différents bruits et de les utiliser pour compenser efficacement l'effet des bruits de test inconnus.

Jusqu'à ce point, toutes les données d'apprentissage et de test bruitées ont été générées artificiellement sur la base de données NIST propre. Dans la prochaine expérience, nous évaluons la version générique du ModeleJoint en conditions réelles en utilisant la base de données SITW qui fournit des segments naturellement bruités.

7.3.4 Performances sur SITW

Afin de tester notre technique en conditions réelles, nous allons appliquer la technique de débruitage basée sur les modèles joints sur la base de données SITW. Dans cette expérience, on a comparé les performances du système de base à la version générique du système ModeleJoint ; Cinq bruits (applaudissements, sonnerie, bruit de fond

TABLE 7.2 – *Comparaison des performances de modèles joints construit en utilisant le bruit de test et un mélange de bruits (non observés dans les conditions de test). Le modèle multi-style et la version générique du système ModeleJoint sont appris en utilisant le même ensemble de données, correspondant à des bruits différents de ceux utilisés en test.*

		EER			
Conditions de test et d'apprentissage		**Système de base**	**Système Multi-style**	**ModeleJoint (bruit de test)**	**ModeleJoint (générique)**
Bruit de voiture	**0dB**	21.18	15.25	**4.78**	6.37
	5dB	15.67	9.92	**3.84**	3.91
	10dB	11.64	8.26	**2.61**	2.73
	15dB	8.46	6.31	**2.05**	2.28
Bruit de climatiseur	**0dB**	18.39	12.47	**5.01**	6.51
	5dB	16.17	11.39	**3.18**	3.82
	10dB	13.21	8.96	**2.72**	2.95
	15dB	10.47	7.83	**2.50**	2.61

d'une station de bus, bruit des vagues et bruit de tempêtes) sont utilisés pour bruiter les segments d'entraînement NIST à différents niveaux de SNR entre 0dB et 15dB (un bruit et un niveau SNR sont utilisés pour chaque segment) et les i-vecteurs résultants sont utilisés pour estimer $p(z)$. Dans cette expérience, l'*ensemble I* des données de test de SITW décrits dans la section 4.2 est utilisé. Cet ensemble correspond à des données d'apprentissage et de test bruitées (SNR inférieurs à 10dB) de longues durées (durée de parole 30s). Le tableau 7.3 montre les résultats de la technique proposée par rapport aux performances du système de base.

TABLE 7.3 – *Performances du système ModeleJoint et du système multi-style utilisés sur les sessions de test longues et bruitées de SITW. Le modèle multi-style et la version générique du système ModeleJoint sont appris en utilisant le même ensemble de données, correspondant à des bruits différents ce ceux présents dans le test.*

EER		
Système de base	**Système *multi-style***	**ModeleJoint (générique)**
12.69	10.58	**4.24**

Le système générique atteint 66% d'amélioration relative d'EER par rapport au système de base et surpasse le système *multi-style* qui à son tour donne 17% d'amélioration relative. Ces résultats confirment l'efficacité de la technique proposée sur les conditions réelles.

7.4 Utilisation d'un modèle joint d'i-vecteurs de longue de de courte durée pour traiter la variabilité des durées dans l'espace des i-vecteurs

Il est connu que les sessions longues donnent de meilleurs résultats que les sessions de courtes durées dans les tâches de RAL indépendante du texte (Sarkar et al., 2012; Rao et Mak, 2013; Kanagasundaram et al., 2011) vu qu'ils sont plus riches en information locuteur. Il est également important de mentionner que les systèmes de RAL dépendants du texte surpassent les systèmes de RAL indépendants du texte dans le contexte des courtes durées où il devient possible de cibler les similitudes dans les segments d'apprentissage et de test lorsque le contenu lexical de l'énoncé de test est connu (Hébert, 2008). La variabilité du contenu linguistique dans le cas de RAL indépendant du texte rend la tâche beaucoup plus difficile et plusieurs travaux ont essayé de modéliser la différence entre i-vecteurs longs et courts sous forme d'un "bruit" qui peut être modélisé et compensé dans l'espace des i-vecteurs (Hasan et al., 2013; Sarkar et al., 2012).

Ceci nous a motivé à utiliser dans (Ben Kheder et al., 2016e) l'algorithme présenté dans la section 7.2 afin d'améliorer les i-vecteurs de test correspondant aux sessions courtes. Ceci est fait en construisant un modèle joint entre des i-vecteurs de session de longue et de courte durées. Dans ce contexte, l'insuffisance de données est perçue comme une corruption qui peut être présente dans les données de test et l'écart entre un i-vecteur correspondant à une session courte et sa version longue comme un "bruit" qui peut être compensé en utilisant l'algorithme développé dans la section 7.2.

Note :

Il est important de préciser une différence fondamentale entre les approches citées précédemment et l'approche qu'on propose dans cette section.
Les modèles *multi-condition* PLDA ou la variante de la PLDA basée sur le décodage d'incertitude (Kenny et al., 2013) utilisent un modèle d'analyse factorielle qui modélise un seul espace qui couvre les i-vecteurs de longues durées et de courtes durées et estiment un sous-espace locuteur ainsi qu'une deuxième composante résiduelle qui capture l'ensemble les variabilités inutiles (variabilité canal, bruit d'estimation, etc.).
Dans notre cas, le but est de passer d'un premier espace i-vecteur hétérogène (qui représente des segments de longues et de courtes durées) vers un deuxième espace i-vecteur qui représente exclusivement les i-vecteurs de longue durée. Une fois les données d'apprentissage et de test sont "projetés" dans ce deuxième espace, la décomposition PLDA est réalisée.

7.4.1 Utilisation du modèle joint i-vecteurs pour la transformation des i-vecteurs courts

Dans ces expériences, les i-vecteurs corrompus correspondent à des i-vecteurs courts tandis que les i-vecteurs de bonne qualité correspondent aux i-vecteurs longs. Les sessions courtes sont générées artificiellement en utilisant la base de données NIST et le ModeleJoint est évalué pour différentes durées suivant cette procédure :

Pour chaque $\mathcal{D} \in \{5s, 10s, 15s, 20s, 30s, \text{durée complète}\}$:

1. Les i-vecteurs sont extraits pour les sessions de durée complète $\{x_i\}$ et leurs versions courtes $\{y_i\}$ de durée $\mathcal{D}$.

2. La distribution $p_{\mathcal{D}}(z)$ est estimée pour un nombre différent de composantes; $K \in \{1, 2, 3\}$.

3. Chaque i-vecteur court d'apprentissage ou de test est transformé en utilisant l'équation 7.8.

Le tableau 7.4 montre les performances du modèle joint par rapport à une performance du système de base (PLDA appris à l'aide de segments longs). La distribution $p_{\mathcal{D}}(z)$ est estimée pour chaque durée $\mathcal{D}$ indépendamment puis appliquée sur les i-vecteurs de test courts correspondants avant le scoring.

TABLE 7.4 – *Performances du modèle joint appris avec 15660 paires de segments de courte et de longue durée. La distribution $p_{\mathcal{D}}(z)$ est apprise pour chaque durée $\mathcal{D} \in \{30s, 20s, 15s, 10s, 5s\}$ indépendamment puis appliquée sur les i-vecteurs de test courts correspondants à la même durée avant le scoring.*

		EER			
		Entraînement de durée complète	**ModeleJoint 1 Gauss.**	**ModeleJoint 2 Gauss.**	**ModeleJoint 3 Gauss.**
Durée de parole de test	**30s**	3.59	**2.98**	3.12	3.25
	20s	5.26	**4.09**	4.69	4.87
	15s	7.28	**5.21**	5.88	6.31
	10s	11.84	**7.06**	8.32	9.35
	5s	21.83	**13.21**	15.32	17.12

Ce modèle améliore les performance jusqu'à 40% (modèle à 1 Gaussienne) et ne nécessite pas de GMM à 2 ou 3 composantes pour $p_{\mathcal{D}}(z)$ mais utilise une grande quantité de données pour être efficace. Dans la prochaine sous-section, nous allons essayer de trouver le minimum de données nécessaires pour apprendre la distribution jointe tout en restant efficace. Ces résultats sont surprenants car ils suggèrent qu'il est possible de modéliser et compenser efficacement la variabilité de durée dans l'espace des i-vecteurs

7.4.2 Effet de la quantité de données utilisée pour estimer $P(z)$ pour différentes durées

Dans cette sous-section, nous analysons la quantité de données nécessaire pour obtenir de bonnes performances en utilisant la méthode de modélisation jointe. La figure 7.3 montre la variation du EER avec le nombre de segments d'entraînement pour apprendre $P(z)$.

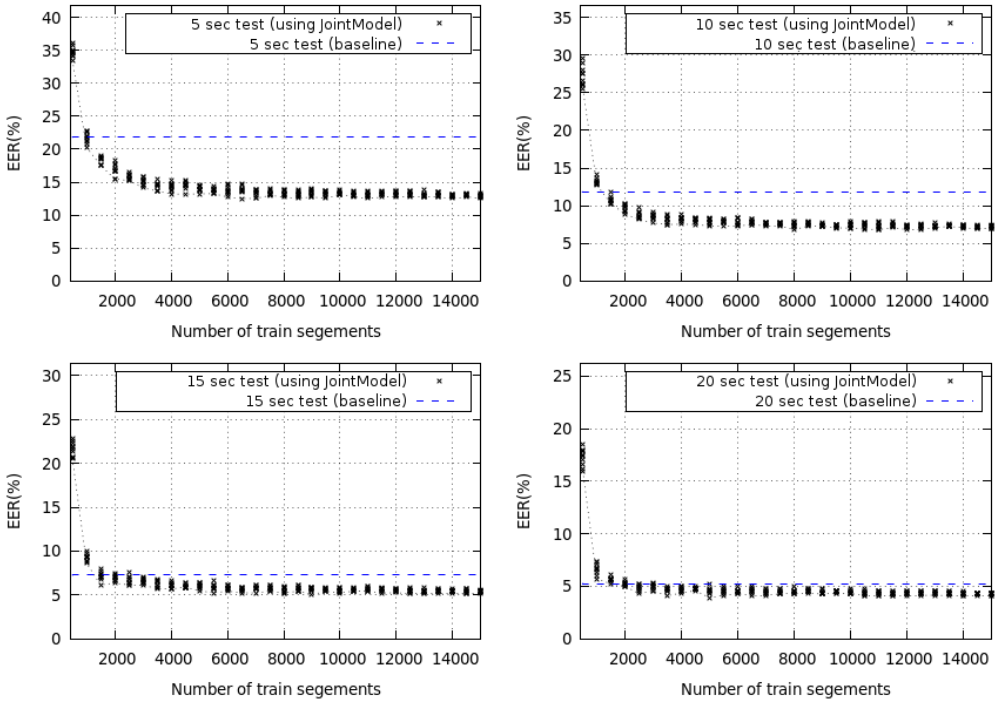

FIGURE 7.3 – *Variation du EER avec la quantité d'i-vecteurs utilisés pour apprendre le modèle de compensation pour les durées de test 5s, 10s, 15s et 20s (10 mesures pour chaque nombre de segments).*

Dans cette sous-section, nous utilisons un modèle à une Gaussienne pour le modèle joint vu que cette configuration a donné les meilleurs résultats dans la sous-section précédente et on étudie l'effet de la quantité de données utilisée pour apprendre chaque modèle. Il est clair à partir de la figure que plus de 3000 paires d'i-vecteurs sont nécessaires pour estimer $p(z)$ pour chaque durée afin d'obtenir de bons résultats en utilisant le modèle joint et que l'utilisation de plus de données d'entraînement ne donnerait pas de meilleurs résultats.

7.4.3 Utilisation du modèle joint avec des sessions de durée arbitraire

En pratique, la durée des segments d'apprentissage et de test varie et il serait intéressant de tester la technique proposée dans le contexte des segments de durée arbitraire.

Dans cette expérience, les segments d'apprentissage et de test ont été choisis aléatoirement afin de générer des enregistrements avec une durée de parole comprise entre 5s et 30s (le choix aléatoire des durées est effectué uniformément sur tout l'intervalle et le segment de parole choisi correspond à une portion continue de l'enregistrement d'origine).

Ensuite, deux systèmes de transformation sont construits (seuls des modèles joints à une Gaussienne sont utilisés dans cette expérience puisqu'ils ont donné les meilleurs résultats dans toutes les expériences précédentes) :

1. **ModeleJoint (générique) :** un seul modèle générique est utilisé pour traiter toutes les durées. Dans ce système, les données d'entraînement correspondant à toutes les durées $\mathcal{D} \in \{30s, 20s, 15s, 10s, 5s\}$ ont été utilisées pour estimer $p(z)$.
2. **ModeleJoint (adaptatif) :** Dans ce système, on a construit cinq modèles joints correspondant à $\mathcal{D} \in \{30s, 20s, 15s, 10s, 5s\}$ et pour chaque segment court d'apprentissage / test, le modèle correspondant à la durée la plus proche est utilisé. Pour un i-vecteur court donné iv de durée $duree(iv)$, le modèle joint $p_{\mathcal{D}}(z)$ choisi correspond à :

$$\underset{D}{\operatorname{argmin}} |duree(iv) - D|; D \in \{30s, 20s, 15s, 10s, 5s\} \tag{7.12}$$

   Cette expérience est rendue possible grâce au fait que tous les i-vecteurs transformés provenant de différents modèles joints partagent le même espace cible (l'espace des i-vecteurs long) et peuvent donc être utilisés ensemble dans la phase de scoring.

Nous entraînons également un modèle PLDA multi-conditions utilisant les mêmes données d'entraînement utilisées pour construire le modèle joint générique. Le tableau 7.5 compare les performances du système de base avec le modèle PLDA multi-condition, le ModeleJoint générique et le ModeleJoint adaptatif.

TABLE 7.5 – *Performances de la modélisation jointe sur des segments d'appr./test de durée arbitraire entre 5s et 30s. Le ModeleJoint générique et la PLDA multi-condition sont appris en utilisant des données correspondant à toutes les durées $\mathcal{D} \in \{30s, 20s, 15s, 10s, 5s\}$ et le modèle de compensation adaptatif utilise le modèle de compensation $P_{\mathcal{D}}(z)$ correspondant à la durée la plus proche.*

EER			
Système de base	**PLDA Multi-condition**	**ModeleJoint (générique)**	**ModeleJoint (adaptatif)**
10.58	8.78	7.08	**6.76**

Une amélioration relative des EER de 33% est observée à l'aide de la version générique du modèle développé et de 36% en utilisant le modèle joint adaptatif qui surpasse le backend PLDA *multi-condition* qui ne dépasse pas une amélioration relative de 17%. Ces résultats montrent que ces deux systèmes peuvent être utilisés pour gérer des durées arbitraires.

Jusqu'à ce point, toutes les données d'apprentissage et de test courtes ont été générées artificiellement sur la base des segments NIST longs. Dans la prochaine expérience, nous évaluons les performances du ModeleJoint sur la base de données SITW qui fournit une condition de test de segments courts.

7.4.4 Performances sur SITW

Dans (Ben Kheder et al., 2016e), on a testé la procédure de débruitage I-MAP sur l'ensemble de test de la base SITW. Dans cette expérience, l'*ensemble II* des données de test de SITW décrits dans la section 4.2 est utilisé. Cet ensemble correspond à des données d'apprentissage et de test propres (SNR supérieur à 20dB) et de courtes durées (< 15s de parole). Les modèles utilisés dans cette expérience sont appris à partir de données d'entraînement NIST propres.

Le tableau 7.6 montre les performances de la technique de modélisation jointe sur les sessions de tests courts de SITW. Différentes distributions $p_{\mathcal{D}}(z)$ sont estimées et appliquées sur tous les i-vecteurs de test courts (25s de durée de la parole). Les distributions apprises correspondent à $\mathcal{D} \in \{30s, 20s, 15s, 10s, 5s\}$, le modèle "générique" utilise des i-vecteurs correspondant à toutes les durées et le modèle de transformation "adaptatif" décrit dans la sous-section 7.4.3 en utilisant le modèle joint correspondant à la durée la plus proche de chaque segment court.

TABLE 7.6 – *Performances de la technique de modélisation jointe sur SITW. La distribution $p_{\mathcal{D}}(z)$ est estimée pour chaque durée $\mathcal{D} \in \{30s, 20s, 15s, 10s, 5s\}$ et la condition "générique" correspond à $p_{\mathcal{D}}(z)$ estimé en utilisant toutes les durées. La condition "adaptatif" utilise le modèle joint $p_{\mathcal{D}}(z)$ correspondant à la durée la plus proche pour chaque segment court.*

EER							
Système de base (entraîn. long)	**Après l'utilisation du modèle joint**						
	30s	**20s**	**15s**	**10s**	**5s**	**Générique**	**Adaptatif**
11.62	9.96	9.53	8.95	9.03	9.53	8.71	**8.42**

Le modèle générique atteint 25% d'amélioration relative des EER par rapport aux performances du système de base tandis que le modèle adaptatif atteint 27% d'EER relatif ce qui montre l'efficacité de la technique proposée.

Conclusions

Dans ce chapitre, on a présenté une approche de traitement de variabilités nuisibles qui opère dans le domaine des i-vecteurs. Contrairement aux algorithmes développés dans les chapitre précédents, cette approche n'utilise pas une relation spécifique entre les i-vecteurs propres et corrompus. Cette technique se base sur un modèle qui représente la distribution jointe entre les i-vecteurs propres et corrompus. Ce modèle permet d'intégrer à la fois des informations à priori sur la distribution des i-vecteurs propres,

bruitées ainsi que l'information jointe entre les deux distributions. Des gains importants en termes d'EER sont observées pouvant atteindre jusqu'à 80% d'amélioration relative. Ce modèle a aussi été testé dans le contexte de bruits inconnus en phase de test et s'est montré efficace pouvant apporter un gain de 70% en EER.

Troisième partie

Conclusions et perspectives

Conclusions et perspectives

Conclusions

Tout au long de cette thèse, nous avons présenté notre travail portant sur le développement de techniques de compensation des nuisances acoustiques dans le domaine des i-vecteurs et visant à rendre les systèmes de RAL plus robustes dans des conditions difficiles.

Dans la première partie de cette thèse, nous avons fait un état de l'art de la reconnaissance automatique du locuteur dans les milieux difficiles. Dans le chapitre 1, nous avons présenté le problème de reconnaissance du locuteur et nous avons détaillé la structure des systèmes de RAL à base d'i-vecteurs. Par la suite, nous avons introduit dans le chapitre 2 les variabilités et nuisances acoustiques qui peuvent dégrader les performances des systèmes de RAL. Cet aperçu a été complémenté par une illustration de l'effet de certaines nuisances acoustiques (bruit additif, distorsion canal, réverbération) sur le signal de parole. En fin de cette partie, nous avons fait dans le chapitre 3 un survol des principales techniques de traitement de nuisances acoustiques développées dans la littérature et qui opèrent à différents niveaux (signal, paramètres acoustiques, modèles et scores).

Dans le chapitre 4, nous avons présenté les jeux de données et les systèmes utilisés tout au long de cette thèse. Par la suite, nous avons accordé une attention particulière à deux types de nuisances acoustiques : le bruit additif ainsi que la variabilité des durées. L'effet de chacune de ces deux nuisances sur les performances des systèmes de RAL a été analysé. Dans un premier temps, nous avons observé une dégradation significative des performances en présence du bruit additif qui devient de plus en plus intense à mesure que le niveau SNR diminue. Par la suite, nous avons analysé la contribution de la distorsion des probabilités à posteriori dans cette dégradation ce qui nous a conduit a mettre en évidence la sensibilité des statistiques au bruit additif.

Dans un deuxième temps, l'évaluation des performances d'un ensemble de modèles PLDA correspondant à différentes durées nous a permis de montrer l'importante du *matching* de durées entre les données de test et les données d'entraînement. Bien que les i-vecteurs de longue durée soient plus riches en information locuteur, il s'est avéré qu'un modèle PLDA entraîné avec de longues durées n'est pas optimal pour les données de courte durée. En effet, l'utilisation d'un modèle PLDA *multi-durées* permet

d'améliorer les performances dans ce contexte et les meilleurs résultats sont atteints en utilisant des segments de même durée que le test pour construire le modèle de scoring.

Dans le chapitre 5, nous avons évoqué le besoin d'étude et de développement de techniques de compensation des nuisances qui opèrent directement dans le domaine des i-vecteurs. Après, nous avons proposé deux algorithmes de compensation des nuisances qui se basent sur une relation spécifique entre les i-vecteurs corrompus et leurs versions propres.

Le premier algorithme suppose que le décalage entre une version corrompue d'un i-vecteur et la version propre correspondante peut être modélisé sous forme d'une translation suivie par une rotation. Ce modèle utilise l'algorithme de Kabsch pour l'estimation de la meilleure matrice de rotation entre deux ensembles appariés d'i-vecteurs générés artificiellement (propres et corrompus). Une amélioration significative des performances du système en présence de bruit additif a été observée ; 40% de gains relatifs en EER sur les données de NIST SRE 2008 bruitées artificiellement. Cependant, nos expériences ont montré que cette approche est sensible au niveau de SNR et au bruit utilisé. En effet, l'utilisation d'un seul vecteur de translation et une matrice de rotation pour la compensation de plusieurs bruits ou niveaux de SNR réduit significativement les gains précédemment obtenus. Cette expérience a permis de remarquer une dépendance de la transformation apprise par cet algorithme des conditions acoustiques d'entraînement. En plus, les i-vecteurs estimés par cette technique peuvent être distordus vu que l'algorithme de Kabsch ne tient pas compte de la distribution des i-vecteurs ciblés.

Ces raisons ont motivé le développement d'un deuxième algorithme bayésien qui permet de faire la compensation de nuisances acoustiques tout en tenant compte de la distribution des i-vecteurs propres cible. Cette approche, nommée "I-MAP", modélise la distorsion dans le domaine des i-vecteurs sous forme de bruit additif. Le critère MAP est par la suite utilisé en supposant que les i-vecteurs propres et les i-vecteurs bruités sont indépendants et suivent une loi Gaussienne. Cette approche a permis d'améliorer significativement les performances du système surpassant les performances de l'algorithme de Kabsch. Une évaluation de I-MAP sur deux bases de test différentes en présence de bruit additif nous a permis de valider son efficacité ; 60% d'amélioration relative sur les données de NIST SRE 2008 bruitées artificiellement et de 50% sur les données de SITW bruitées naturellement. Cette différence de gains entre les deux bases peut être due à quelques facteurs. En effet, la performance de l'algorithme I-MAP dépend de la qualité de la distribution de bruit estimée. Cette distribution est construite en se basant sur des données bruitées artificiellement et peut ne pas traduire fidèlement tous les effets induits par le bruit additif dans le cas des bruits réels (exp : effet Lombard). Un autre facteur à considérer est aussi le *dataset mismatch*. Ce terme est généralement utilisé dans la littérature pour qualifier ce problème qui peut survenir lors de l'utilisation d'une base de test différente de celle utilisée pour entraîner le système. Ce *mismatch* peut se manifester sous forme d'une différence de performances entre les deux bases en raison de leurs propriétés acoustiques distinctes (types de microphones, différences de langues, etc). Pour pallier à ce problème, une solution aurait été d'utiliser les données propres de SITW bruitées artificiellement pour construire les distributions

de bruit.

En pratique, l'algorithme I-MAP est coûteux en temps de calcul vu qu'il requiert l'estimation d'une distribution de bruit pour chaque segment de test. Afin d'accélérer cette procédure, nous avons proposé une solution qui consiste à construire une base de distributions de bruits en utilisant différents bruits et niveaux de SNR. En phase de test, la meilleure distribution de bruit correspondant à un i-vecteur de test bruité est sélectionnée. Cette approche a montré des gains proches de ceux de I-MAP en utilisant une mesure de similarité adaptée. Théoriquement, il était attendu que cette procédure sélectionne une distribution de bruit ayant des propriétés acoustiques proches de celle de la session de test. Cependant, nous avons remarqué que l'algorithme sélectionne parfois une distribution correspondant à un niveau de SNR différent de celui de la session de test (exp : une distribution de 10dB peut être sélectionnée pour une session ayant un SNR de 0dB). Cette observation suggère l'existence de propriétés statistiques communes entre des distributions de bruit qui correspondent à des conditions acoustiques (bruit/SNR) différentes. Les bons résultats donnés par la base de distributions peuvent être expliqués d'un coté par cette hypothèse. D'un autre coté, la grande taille de la base (234 distributions) permet à l'algorithme de retrouver une distribution de bruit ayant des propriétés statistiques proches de celles de la distribution de bruit réelle.

Dans le chapitre 6, nous avons proposé une méthode de construction de techniques de débruitage qui opère dans le domaine des i-vecteurs. Cette technique a permis de construire des algorithmes de débruitage efficaces, mais peut être dépendante du bruit et du niveau de SNR pour des résultats optimaux. Il est intéressant de noter que cette procédure a pu retrouver l'équation de la technique I-MAP développée dans le chapitre 5. Ceci laisse à penser que l'utilisation de cette approche sur une grande variété de bruits et niveaux de SNR pourrait (théoriquement) résulter en des techniques de compensation de nuisances plus efficaces. Nos expériences avec la programmation génétique ont aussi fait apparaître une tendance intéressante. En effet, la plupart des algorithmes générées correspondent à des équations de plus en plus complexes à mesure que le niveau de SNR diminue. Bien que cette procédure soit très dépendante du bruit et du niveau SNR, il pourrait être stipulé que l'effet des bruits dans des niveaux de SNR très bas est complexe et requiert des méthodes plus sophistiquées pour compenser ses effets.

Dans le chapitre 7, nous avons présenté une technique de compensation des nuisances acoustiques qui corrige un défaut de l'algorithme I-MAP. En effet, ce dernier se base sur une hypothèse d'indépendance entre la distribution des i-vecteurs propres et celle des i-vecteurs bruités. Afin de résoudre ce problème, nous avons développé un algorithme qui modélise la distribution jointe entre les i-vecteurs propres et leurs versions bruitées. Après, cette distribution est intégrée dans un estimateur MMSE qui transforme un i-vecteur corrompu en sa version propre. Cette procédure nous a permis d'intégrer à la fois des informations à priori sur la distribution des i-vecteurs propres, bruitées ainsi que la distribution jointe entre les deux représentations. Nous avons testé cette approche dans le contexte de deux nuisances acoustiques ; le bruit additif et les sessions de courtes durées. Nous avons observé des gains significatifs en termes d'EER pouvant atteindre jusqu'à 80% d'amélioration relative sur les données de NIST SRE

2008 bruitées artificiellement et 66% sur les données de SITW bruitées naturellement. Nous avons testé ce modèle dans le contexte de bruits non-observés (différents en entraînement et en test) et des gains relatifs de 70% ont été observés sur les données de NIST SRE 2008 bruitées artificiellement. Nous avons émis l'hypothèse que ces performances pourraient être dues aux propriétés statistiques partagées par les bruits utilisées en entraînement et en test. En effet. Bien que les bruits soient de nature différentes, ils peuvent être vus comme des bruits de basses fréquences (bruit de foule, bruit de la pluie, bruit de fond d'une station de bus, ..). Cette propriété pourrait en effet aider le système à mieux compenser ces bruits.

Nous avons conduit une deuxième expérience en utilisant le même algorithme pour traiter le problème de la variabilité des durées. Dans ce contexte, le décalage entre les i-vecteurs de la version longue et la version courte d'une session donnée est modélisé sous forme de bruit dans le domaine des i-vecteurs. La distribution jointe entre les deux représentations (longue/courte) est utilisée dans ce contexte pour estimer l'i-vecteur long correspondant à un segment court donné. Nous avons observé des gains de 40% en termes d'EER en utilisant cette approche sur les données de NIST SRE 2008 et un gain de 35% sur les segments de courtes durée de la base SITW. Ces performances ont confirmé la possibilité de modéliser la variabilité de durées dans le domaine des i-vecteurs. Dans l'état de l'art, le manque de données est généralement perçu comme un problème difficile à aborder. En effet, la plupart des techniques proposées dans la littérature se restreignent à la modélisation de l'incertitude relative à l'extraction des i-vecteurs de courtes durées. Les résultats obtenus dans nos expériences prouvent qu'il est possible d'utiliser des informations à priori sur les segments de longues et de courtes durées et de les utiliser pour "récupérer" une partie de l'information acoustique manquante. Cette idée ouvre de nouvelles possibilités pour le développement d'algorithmes encore plus efficaces pour traiter les segments de courtes durées.

Perspectives

Cette section propose quelques pistes pour compléter les études que nous avons réalisées ou pour améliorer les méthodes proposées. Ces pistes concernent essentiellement cinq aspects complémentaires :

Le calcul robuste des probabilités à posteriori : Le calcul des statistiques (d'ordre zéro et de premier ordre) est l'étape la plus importante dans le processus d'extraction d'i-vecteurs. En présence de bruit additif, les probabilités à posteriori peuvent être sévèrement distordues causant une dégradation de performances. On a montré dans la section 5.2 que cette dégradation peut atteindre jusqu'à 20% en termes d'EER pour des niveaux de SNR faibles.

Tout on long de cette thèse, on a choisi de ne pas traiter d'une manière directe cette distorsion. A la place, les algorithmes développés visent à corriger à la fois la corruption

des données causée par les nuisances acoustiques ainsi que la distorsion des probabilités à posteriori.

En pratique, il serait possible d'implémenter des techniques de calcul robuste de statistiques et de les tester avec les algorithmes développés dans cette thèse. Une première possibilité serait de construire un UBM bruité qui permettrait de calculer d'une manière plus robuste les probabilités à posteriori. Une deuxième solution serait d'utiliser un réseau de neurones profond pour le calcul robuste de statistiques comme proposé dans (Lei et al., 2014b) et de l'entraîner sur des données propres et bruitées.

L'utilisation de distributions pour représenter les i-vecteurs corrompus : Les algorithmes proposés dans cette thèse se basent sur un i-vecteur corrompu et estiment la version propre correspondante. En réalité, la nature stochastique des nuisances acoustiques (tel que le bruit additif ou la réverbération) peut générer différentes versions corrompues correspondant à un segment propre donné. La même remarque peut se faire dans le contexte des segments de courtes durées où un i-vecteur long peut correspondre à plusieurs i-vecteurs de courte durée extraits de différents segments de l'enregistrement original. Pour cette raison, il serait intéressant de concevoir des techniques qui manipulent des distributions d'i-vecteurs corrompus.

Intégration des techniques développées dans des DNN : Au niveau le plus basique, les réseaux de neurones profonds se basent sur des multiplications matricielles et sur des transformations non-linéaires des données. La forme $h = f(Wx + b)$ est souvent utilisée pour représenter la sortie h d'une couche cachée où W est une matrice de poids et b est un vecteur de biais. $f(.)$ fait référence à la fonction d'activation non-linéaire du réseau de neurones.

Il est intéressant de noter que la forme $Wx + b$ est récurrente dans les algorithmes proposés dans cette thèse où la compensation des nuisances peut être réduite à une transformation linéaire. En pratique, il serait possible de combiner les techniques développées dans cette thèse en les utilisant comme couches cachées dans un réseau de neurones. Cette procédure a été proposée récemment dans (Seuret et al., 2017) dans le contexte du traitement d'images en utilisant la matrice issue d'une PCA sous forme de couche d'un réseau de neurones profond.

Intégration d'informations phonétiques dans les techniques de compensation de la variabilité des durées : La technique de compensation de variabilités basée sur les modèles joints (JointModel) se base sur les i-vecteurs correspondant à des segments de courte et de longue durée. Cependant, il serait intéressant d'étendre ce modèle pour prendre en compte l'information phonétique contenue dans les segments. En effet, les études de l'effet des catégories phonétiques sur la discrimination entre locuteurs dans le contexte de la RAL (Ajili et al., 2016, 2017b) montrent que tirer parti de ces connaissances pourrait améliorer la capacité de discrimination des systèmes.

Compensation de variabilités multiples : En présence de plusieurs nuisances acoustiques, il est préférable d'avoir des approches capables de compenser les effets complexes résultants. En pratique, ceci pourrait être réalisé par la combinaison séquentielle d'algorithmes qui ciblent chacune des nuisances. Les algorithmes développés dans cette thèse sont conçus sous forme de techniques de transformation qui peuvent être utilisées en tant que "boite noire" pour projeter les i-vecteurs d'un espace vers un autre (de l'espace des i-vecteurs corrompus vers l'espace des i-vecteurs propres). Ceci facilite l'utilisation séquentielle d'algorithmes et la combinaison de l'algorithme de Kabsch et de I-MAP dans le chapitre 5 en est un exemple. Cependant, cette approche peut être problématique en raison des distorsions induites par chaque algorithme sur les données d'origine. Afin de résoudre ce problème, il serait intéressant d'évaluer les performances des techniques développées dans cette thèse pour la compensation simultanée de plusieurs nuisances acoustiques (exp : bruit additif + distorsion canal) et de développer éventuellement des approches qui seraient plus adaptées à ce genre de problèmes.

Liste des illustrations

Liste des tableaux

Bibliographie

(Acero, 1990) A. Acero, 1990. *Acoustical and environmental robustness in automatic speech recognition*. Thèse de Doctorat, Carnegie Mellon University Pittsburgh.

(afcp2002, 2002) afcp2002, 2002. Communiqué de l'AFCP du 3 Décembre 2002 concernant "l'identification des individus par leur voix". http://www.afcp-parole.org/doc/NB-AFCP-dec02.htm.

(Afify et al., 2009) M. Afify, X. Cui, & Y. Gao, 2009. Stereo-based stochastic mapping for robust speech recognition. *IEEE transactions on audio, speech, and language processing 17*(7), 1325–1334.

(Ajili et al., 2016) M. Ajili, J.-F. Bonastre, W. B. Kheder, S. Rossato, & J. Kahn, 2016. Phonetic content impact on forensic voice comparison. Dans les actes de *Spoken Language Technology Workshop (SLT), 2016 IEEE*, 210–217. IEEE.

(Ajili et al., 2017a) M. Ajili, J.-F. Bonastre, W. B. Kheder, S. Rossato, & J. Kahn, 2017a. Homogeneity measure impact on target and non-target trials in forensic voice comparison. *Proc. Interspeech 2017*, 2844–2848.

(Ajili et al., 2017b) M. Ajili, J.-F. Bonastre, W. B. Kheder, S. Rossato, & J. Kahn, 2017b. Phonological content impact on wrongful convictions in forensic voice comparison context. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

(Alegre et al., 2013) F. Alegre, A. Amehraye, & N. Evans, 2013. A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns. Dans les actes de *IEEE Sixth International Conference on Biometrics : Theory, Applications and Systems (BTAS), 2013*, 1–8. IEEE.

(Ambikairajah et al., 2012) E. Ambikairajah, J. M. K. Kua, V. Sethu, & H. Li, 2012. Pncc-ivector-src based speaker verification. Dans les actes de *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, 1–7. IEEE.

(Aronowitz, 2014) H. Aronowitz, 2014. Inter dataset variability compensation for speaker recognition. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014.*, 4002–4006. IEEE.

(Atal, 1976) B. S. Atal, 1976. Automatic recognition of speakers from their voices. *Proceedings of the IEEE 64*(4), 460–475.

(Auckenthaler et al., 2000) R. Auckenthaler, M. Carey, & H. Lloyd-Thomas, 2000. Score normalization for text-independent speaker verification systems. *Digital Signal Processing 10*(1), 42–54.

(Back, 1996) T. Back, 1996. *Evolutionary algorithms in theory and practice : evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

(Banzhaf et al., 1998) W. Banzhaf, P. Nordin, R. E. Keller, & F. D. Francone, 1998. *Genetic programming : an introduction*, Volume 1. Morgan Kaufmann San Francisco.

(Barras, 2016) C. Barras, 2016. Reconnaissance du locuteur.

(Barras et Gauvain, 2003) C. Barras & J.-L. Gauvain, 2003. Feature and score normalization for speaker verification of cellular data. Dans les actes de *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. (ICASSP'03).*, Volume 2, II–49. IEEE.

(Ben Kheder et al., 2016a) W. Ben Kheder, M. Ajili, P.-M. Bousquet, D. Matrouf, & J.-F. Bonastre, 2016a. Lia system for the sitw speaker recognition challenge. *INTERSPEECH*.

(Ben Kheder et al., 2016b) W. Ben Kheder, M. Ajili, P.-M. Bousquet, D. Matrouf, & J.-F. Bonastre, 2016b. Lia system for the sitw speaker recognition challenge. *Interspeech 2016*, 848–852.

(Ben Kheder et al., 2016c) W. Ben Kheder, D. Matrouf, M. Ajili, & J.-F. Bonastre, 2016c. Iterative bayesian and mmse-based noise compensation techniques for speaker recognition in the i-vector space. *Speaker and Language Recognition Workshop, IEEE Odyssey*.

(Ben Kheder et al., 2016d) W. Ben Kheder, D. Matrouf, M. Ajili, & J.-F. Bonastre, 2016d. Probabilistic approach using joint clean and noisy i-vectors modeling for speaker recognition. *INTERSPEECH*.

(Ben Kheder et al., 2016e) W. Ben Kheder, D. Matrouf, M. Ajili, & J.-F. Bonastre, 2016e. Probabilistic approach using joint long and short session i-vectors modeling to deal with short utterances for speaker recognition. *INTERSPEECH*.

(Ben Kheder et al., 2015) W. Ben Kheder, D. Matrouf, J.-F. Bonastre, M. Ajili, & P.-M. Bousquet, 2015. Additive noise compensation in the i-vector space for speaker recognition. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4190–4194.

(Ben Kheder et al., 2017) W. Ben Kheder, D. Matrouf, P.-M. Bousquet, J.-F. Bonastre, & M. Ajili, 2017. Fast i-vector denoising using map estimation and a noise distributions database for robust speaker recognition. *Computer Speech & Language*.

(Besacier et al., 2000) L. Besacier, S. Grassi, A. Dufaux, M. Ansorge, & F. Pellandini, 2000. Gsm speech coding and speaker recognition. Dans les actes de *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000. ICASSP'00.*, Volume 2, II1085–II1088. IEEE.

(Bimbot et Chollet, 1997) F. Bimbot & G. Chollet, 1997. Assessment of speaker verification systems. *Handbook of standards and resources for spoken language systems*, 408–480.

(Bimbot et al., 1994) F. Bimbot, G. Chollet, & A. Paoloni, 1994. Assessment methodology for speaker identification and verification systems-an overview of sam-a esprit project 6819-task 2500. Dans les actes de *Automatic Speaker Recognition, Identification and Verification*.

(Bishop, 2006) C. M. Bishop, 2006. Pattern recognition. *Machine Learning 128*.

(Blank et al., 2002) S. C. Blank, S. K. Scott, K. Murphy, E. Warburton, & R. J. Wise, 2002. Speech production : Wernicke, broca and beyond. *Brain 125*(8), 1829–1838.

(Blasband et al., 1999) M. Blasband, N. Bevan, M. King, B. Maegaard, L. des Tombe, S. Krauwer, S. Manzi, & N. Underwood, 1999. Expert advisory group on language engineering standards/evaluation working group final report 2.

(Bolt et al., 1970) R. H. Bolt, F. S. Cooper, E. E. David Jr, P. B. Denes, J. M. Pickett, & K. N. Stevens, 1970. Speaker identification by speech spectrograms : a scientists' view of its reliability for legal purposes. *The Journal of the Acoustical Society of America 47*(2B), 597–612.

(Bonastre et Meloni, 1992) J. Bonastre & H. Meloni, 1992. A study of spectral variability for speaker characterisation. *19èmes Journées d'Etudes sur la Parole 555*.

(Bonastre et al., 2003) J.-F. Bonastre, F. Bimbot, L.-J. Boë, J. P. Campbell, D. A. Reynolds, & I. Magrin-Chagnolleau, 2003. Person authentication by voice : a need for caution. Dans les actes de *INTERSPEECH*.

(Bonastre et al., 2000) J.-F. Bonastre, P. Delacourt, C. Fredouille, T. Merlin, & C. Wellekens, 2000. A speaker tracking system based on speaker turn detection for NIST evaluation. Dans les actes de *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000. ICASSP'00.*, Volume 2, II1177–II1180. IEEE.

(Bouaziz et al., 2016) M. Bouaziz, M. Morchid, P.-M. Bousquet, R. Dufour, K. Janod, W. Ben Kheder, & G. Linarès, 2016. Un sous-espace thématique latent pour la compréhension du langage parlé.

(Bousquet, 2014) P.-M. Bousquet, 2014. Bénéfices et limites des représentations en facteur de variabilité totale pour la reconnaissance du locuteur. Avignon.

(Bousquet et al., 2011) P.-M. Bousquet, D. Matrouf, & J.-F. Bonastre, 2011. Intersession compensation and scoring methods in the i-vectors space for speaker recognition. Dans les actes de *Interspeech*, 485–488.

(Brown et Hagoort, 2000) C. M. Brown & P. Hagoort, 2000. The neurocognition of language.

(Brümmer, 2010) N. Brümmer, 2010. *Measuring, refining and calibrating speaker and language information extracted from speech*. Thèse de Doctorat, Citeseer.

(Brummer et al., 2007) N. Brummer, J. Cernocky, M. Karafiát, D. A. van Leeuwen, P. Matejka, P. Schwarz, A. Strasheim, et al., 2007. Fusion of heterogeneous speaker recognition systems in the stbu submission for the NIST speaker recognition evaluation 2006. *IEEE Transactions on Audio, Speech, and Language Processing 15*(7), 2072–2084.

(Brümmer et De Villiers, 2010) N. Brümmer & E. De Villiers, 2010. The speaker partitioning problem. Dans les actes de *Odyssey*, 34.

(Brümmer et Du Preez, 2006) N. Brümmer & J. Du Preez, 2006. Application-independent evaluation of speaker detection. *Computer Speech & Language 20*(2), 230–275.

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

(Campbell, 1997) J. P. Campbell, 1997. Speaker recognition : a tutorial. *Proceedings of the IEEE 85*(9), 1437–1462.

(Campbell et al., 2003) W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, & T. R. Leek, 2003. Phonetic speaker recognition with support vector machines. Dans les actes de *Advances in neural information processing systems*, None.

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

(Campbell et al., 2006b) W. M. Campbell, D. E. Sturim, & D. A. Reynolds, 2006b. Support vector machines using gmm supervectors for speaker verification. *IEEE signal processing letters 13*(5), 308–311.

(Campbell et al., 2006c) W. M. Campbell, D. E. Sturim, D. A. Reynolds, & A. Solomonoff, 2006c. SVM based speaker verification using a gmm supervector kernel and NAP variability compensation. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006.*, Volume 1, I–I. IEEE.

(Chen, 1987) Y. Chen, 1987. Cepstral domain stress compensation for robust speech recogniton. Dans les actes de *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87.*, Volume 12, 717–720. IEEE.

(Colby et al., 2012) E. Colby, B. Hamacher, & L. Emmanuel, 2012. George zimmerman charged with second-degree murder. [(online). Retrieved June 8, 2012.].

(Cumani, 2015) S. Cumani, 2015. Fast scoring of full posterior plda models. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 23*(11), 2036–2045.

(Cumani et al., 2014) S. Cumani, O. Plchot, & P. Laface, 2014. On the use of i-vector posterior distributions in probabilistic linear discriminant analysis. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 22*(4), 846–857.

(Davis et Mermelstein, 1980) S. Davis & P. Mermelstein, 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.

(Dehak et al., 2011) N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, & P. Ouellet, 2011. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing 19*(4), 788–798.

(Delacourt, 2000) P. Delacourt, 2000. La segmentation et le regroupement par locuteurs pour l'indexation de documents audio. *These de doctorat, ENST-Eurecom*.

(Delacourt et al., 2000) P. Delacourt, J. Bonastre, C. Fredouille, S. Meignier, T. Merlin, & C. Wellekens, 2000. Différentes stratégies pour le suivi du locuteur. *RFIA2000 : Reconnaissance des Formes et Intelligence Artificielle*.

(Deng et al., 2001) L. Deng, A. Acero, L. Jiang, J. Droppo, & X. Huang, 2001. High-performance robust speech recognition using stereo training data. Dans les actes de *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01).*, Volume 1, 301–304. IEEE.

(Doddington, 1985) G. R. Doddington, 1985. Speaker recognition—identifying people by their voices. *Proceedings of the IEEE 73*(11), 1651–1664.

(Doddington, 1998) G. R. Doddington, 1998. Speaker recognition evaluation methodology–an overview and perspective–. Dans les actes de *Workshop on Speaker Recognition and its Commercial and Forensic Applications (RLA2C)*, 20–23.

(Doddington et al., 2000) G. R. Doddington, M. A. Przybocki, A. F. Martin, & D. A. Reynolds, 2000. The NIST speaker recognition evaluation–overview, methodology, systems, results, perspective. *Speech Communication 31*(2), 225–254.

(Douglas, 1986) O. Douglas, 1986. Speaker recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 4–17.

(Du et al., 2015) S. Du, X. Xiao, & E. S. Chng, 2015. Dnn feature compensation for noise robust speaker verification. Dans les actes de *IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), 2015*, 871–875. IEEE.

(Eckert et Rickford, 2001) P. Eckert & J. R. Rickford, 2001. *Style and sociolinguistic variation*. Cambridge University Press.

(Egan, 1975) J. P. Egan, 1975. Signal detection theory and ROC analysis.

(Fan et Hansen, 2011) X. Fan & J. H. Hansen, 2011. Speaker identification within whispered speech audio streams. *IEEE transactions on audio, speech, and language processing 19*(5), 1408–1421.

(Ferrer et al., 2007) L. Ferrer, E. Shriberg, S. Kajarekar, & K. Sonmez, 2007. Parameterization of prosodic feature distributions for svm modeling in speaker recognition. Dans les actes de *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, Volume 4, IV–233. IEEE.

(Freesound.org, 2017) Freesound.org, 2017. Freesound.org. http://www.freesound.org.

(Fukunaga, 2013) K. Fukunaga, 2013. *Introduction to statistical pattern recognition*. Academic press.

(Furui, 1974) S. Furui, 1974. An analysis of long-term variation of feature parameters of speech and its application to talker recognition. *Electronics and Communications in Japan 57*(12), 34–42.

(Furui, 1996) S. Furui, 1996. An overview of speaker recognition technology. Dans les actes de *Automatic speech and speaker recognition*, 31–56. Springer.

(Furui, 1997) S. Furui, 1997. Recent advances in speaker recognition. Dans les actes de *International Conference on Audio-and Video-Based Biometric Person Authentication*, 235–252. Springer.

(G.711specs, 2017) G.711specs, 2017. G.711 specifications. http://www.itu.int/ITU-T/recommendations/related_ps.aspx?id_prod=911.

(Galliano et al., 2005) S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, J.-F. Bonastre, & G. Gravier, 2005. The ester phase ii evaluation campaign for the rich transcription of french broadcast news. Dans les actes de *Interspeech*, 1149–1152.

(Gao et al., 2016) T. Gao, J. Du, Y. Xu, C. Liu, L.-R. Dai, & C.-H. Lee, 2016. Joint training of dnns by incorporating an explicit dereverberation structure for distant speech recognition. *EURASIP Journal on Advances in Signal Processing 2016*(1), 86.

(Garcia-Romero et Espy-Wilson, 2011) D. Garcia-Romero & C. Y. Espy-Wilson, 2011. Analysis of i-vector length normalization in speaker recognition systems. Dans les actes de *Interspeech*, 249–252.

(Garcia-Romero et al., 2014) D. Garcia-Romero, X. Zhang, A. McCree, & D. Povey, 2014. Improving speaker recognition performance in the domain adaptation challenge using deep neural networks. Dans les actes de *Spoken Language Technology Workshop (SLT), 2014 IEEE*, 378–383. IEEE.

(Garcia-Romero et al., 2012) D. Garcia-Romero, X. Zhou, & C. Y. Espy-Wilson, 2012. Multicondition training of gaussian plda models in i-vector space for noise and reverberation robust speaker recognition. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*, 4257–4260. IEEE.

(Gaubitch et al., 2012) N. D. Gaubitch, H. W. Loellmann, M. Jeub, T. H. Falk, P. A. Naylor, P. Vary, & M. Brookes, 2012. Performance comparison of algorithms for blind reverberation time estimation from speech. Dans les actes de *International Workshop on Acoustic Signal Enhancement ; Proceedings of IWAENC 2012.*, 1–4. VDE.

(Gauvain et Lee, 1994) J.-L. Gauvain & C.-H. Lee, 1994. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE transactions on speech and audio processing* *2*(2), 291–298.

(Godfrey et al., 1992) J. J. Godfrey, E. C. Holliman, & J. McDaniel, 1992. Switchboard : Telephone speech corpus for research and development. Dans les actes de *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992*, Volume 1, 517–520. IEEE.

(Gong, 1995) Y. Gong, 1995. Speech recognition in noisy environments : A survey. *Speech communication* *16*(3), 261–291.

(Gravier, 2003) G. Gravier, 2003. Spro : speech signal processing toolkit. *Software available at http ://gforge. inria. fr/projects/spro*.

(Greenberg et al., 2010) C. S. Greenberg, A. F. Martin, L. Brandschain, J. P. Campbell, C. Cieri, G. R. Doddington, & J. J. Godfrey, 2010. Human assisted speaker recognition in NIST sre10. Dans les actes de *Odyssey*, 32.

(Hansen, 1996) J. H. Hansen, 1996. Analysis and compensation of speech under stress and noise for environmental robustness in speech recognition. *Speech communication* *20*(1), 151–173.

(Hansen et Cairns, 1995) J. H. Hansen & D. A. Cairns, 1995. Icarus : Source generator based real-time recognition of speech in noisy stressful and lombard effect environments. *Speech Communication* *16*(4), 391–422.

(Hansen et Hasan, 2015) J. H. Hansen & T. Hasan, 2015. Speaker recognition by machines and humans : A tutorial review. *Signal Processing Magazine, IEEE* *32*(6), 74–99.

(Hansen et al., 2017) J. H. Hansen, M. K. Nandwana, & N. Shokouhi, 2017. Analysis of human scream and its impact on text-independent speaker verification a. *The Journal of the Acoustical Society of America* *141*(4), 2957–2967.

(Hansen et Shokouhi, 2013) J. H. Hansen & N. Shokouhi, 2013. Speaker identification : Screaming, stress and non-neutral speech, is there speaker content ? *IEEE SLTC Newsletter*, 2013–11.

(Hansen et al., 2000) J. H. Hansen, C. Swail, A. J. South, R. K. Moore, H. Steeneken, E. J. Cupples, T. Anderson, C. R. Vloeberghs, I. Trancoso, & P. Verlinde, 2000. The impact of speech under 'stress' on military speech technology. *NATO Project Report*.

(Hansen et Varadarajan, 2009) J. H. Hansen & V. Varadarajan, 2009. Analysis and compensation of lombard speech across noise type and levels with application to in-set/out-of-set speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing* *17*(2), 366–378.

(Hanson et Applebaum, 1990) B. A. Hanson & T. H. Applebaum, 1990. Robust speaker-independent word recognition using static, dynamic and acceleration features : Experiments with lombard and noisy speech. Dans les actes de *International Conference on Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990*, 857–860. IEEE.

(Hasan et al., 2013) T. Hasan, R. Saeidi, J. H. Hansen, & D. A. van Leeuwen, 2013. Duration mismatch compensation for i-vector based speaker recognition systems. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*, 7663–7667. IEEE.

(Hatch et al., 2006) A. O. Hatch, S. S. Kajarekar, & A. Stolcke, 2006. Within-class covariance normalization for svm-based speaker recognition. Dans les actes de *Interspeech*.

(Hébert, 2008) M. Hébert, 2008. Text-dependent speaker recognition. Dans les actes de *Springer handbook of speech processing*, 743–762. Springer.

(Hermansky, 1990) H. Hermansky, 1990. Perceptual linear predictive (plp) analysis of speech. *the Journal of the Acoustical Society of America* *87*(4), 1738–1752.

(Hermansky et Morgan, 1994) H. Hermansky & N. Morgan, 1994. Rasta processing of speech. *IEEE transactions on speech and audio processing* *2*(4), 578–589.

(Hirsch, 1993) H. G. Hirsch, 1993. *Estimation of noise spectrum and its application to SNR-estimation and speech enhancement*. International Computer Science Institute.

(Hirsch, 2017) H. G. Hirsch, 2017. FaNT - Filtering and Noise Adding Tool. http://dnt.kr.hsnr.de/download.html.

(Hurmalainen et al., 2015) A. Hurmalainen, R. Saeidi, & T. Virtanen, 2015. Noise robust speaker recognition with convolutive sparse coding. Dans les actes de *INTERSPEECH*, 244–248.

(Jin et al., 2007) Q. Jin, T. Schultz, & A. Waibel, 2007. Far-field speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing* *15*(7), 2023–2032.

(Jin et al., 2008) Q. Jin, A. R. Toth, A. W. Black, & T. Schultz, 2008. Is voice transformation a threat to speaker identification ? Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008.*, 4845–4848. IEEE.

(Johnson, 1999) S. Johnson, 1999. Who spoke when ?-automatic segmentation and clustering for determining speaker turns. Dans les actes de *Eurospeech*.

(Junqua, 1993) J.-C. Junqua, 1993. The lombard reflex and its role on human listeners and automatic speech recognizers. *The Journal of the Acoustical Society of America* *93*(1), 510–524.

(Junqua et Anglade, 1990) J.-C. Junqua & Y. Anglade, 1990. Acoustic and perceptual studies of lombard speech : Application to isolated-words automatic speech recognition. Dans les actes de *International Conference on Acoustics, Speech, and Signal Processing, 1990. ICASSP-90.*, 841–844. IEEE.

(Kabsch, 1976) W. Kabsch, 1976. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A : Crystal Physics, Diffraction, Theoretical and General Crystallography* *32*(5), 922–923.

(Kanagasundaram et al., 2011) A. Kanagasundaram, R. Vogt, D. B. Dean, S. Sridharan, & M. W. Mason, 2011. I-vector based speaker recognition on short utterances. Dans les actes de *Proceedings of the 12th Annual Conference of the International Speech Communication Association*, 2341–2344. International Speech Communication Association (ISCA).

(Kenny, 2010) P. Kenny, 2010. Bayesian speaker verification with heavy-tailed priors. Dans les actes de *Odyssey*, 14.

(Kenny et al., 2005) P. Kenny, G. Boulianne, & P. Dumouchel, 2005. Eigenvoice modeling with sparse training data. *IEEE transactions on speech and audio processing 13*(3), 345–354.

(Kenny et al., 2007) P. Kenny, G. Boulianne, P. Ouellet, & P. Dumouchel, 2007. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing 15*(4), 1435–1447.

(Kenny et Dumouchel, 2004) P. Kenny & P. Dumouchel, 2004. Experiments in speaker verification using factor analysis likelihood ratios. Dans les actes de *ODYSSEY04-The Speaker and Language Recognition Workshop*.

(Kenny et al., 2003) P. Kenny, M. Mihoubi, & P. Dumouchel, 2003. New map estimators for speaker recognition. Dans les actes de *INTERSPEECH*.

(Kenny et al., 2013) P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, & P. Dumouchel, 2013. Plda for speaker verification with utterances of arbitrary duration. Dans les actes de *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 7649–7653. IEEE.

(Kheder et al., 2014) W. B. Kheder, D. Matrouf, P.-M. Bousquet, J.-F. Bonastre, & M. Ajili, 2014. Robust speaker recognition using map estimation of additive noise in i-vectors space. *Statistical Language and Speech Processing*, 97–107.

(Kitamura et Ono, 2016) D. Kitamura & N. Ono, 2016. Efficient initialization for nonnegative matrix factorization based on nonnegative independent component analysis. Dans les actes de *International Workshop on Acoustic Signal Enhancement (IWAENC), 2016.*, 1–5. IEEE.

(Klatt, 1977) D. H. Klatt, 1977. Review of the arpa speech understanding project. *The Journal of the Acoustical Society of America 62*(6), 1345–1366.

(Kolbœk et al., 2016) M. Kolbœk, Z.-H. Tan, & J. Jensen, 2016. Speech enhancement using long short-term memory based recurrent neural networks for noise robust speaker verification. Dans les actes de *Spoken Language Technology Workshop (SLT), 2016 IEEE*, 305–311. IEEE.

(Koza, 1992) J. R. Koza, 1992. *Genetic programming : on the programming of computers by means of natural selection*, Volume 1. MIT press.

(Kuitert et Boves, 1997) M. Kuitert & L. Boves, 1997. Speaker verification with gsm coded telephone speech. Dans les actes de *EUROSPEECH*. Citeseer.

(Laaridh et al., 2017) I. Laaridh, W. B. Kheder, C. Fredouille, & C. Meunier, 2017. Automatic prediction of speech evaluation metrics for dysarthric speech. *Proc. Interspeech 2017*, 1834–1838.

(Lanitis, 2009) A. Lanitis, 2009. A survey of the effects of aging on biometric identity verification. *International Journal of Biometrics* *2*(1), 34–52.

(Larcher et al., 2013) A. Larcher, J.-F. Bonastre, B. G. Fauve, K.-A. Lee, C. Lévy, H. Li, J. S. Mason, & J.-Y. Parfait, 2013. ALIZE 3.0-open source toolkit for state-of-the-art speaker recognition. Dans les actes de *Interspeech*, 2768–2772.

(Larcher et al., 2012) A. Larcher, K.-A. Lee, B. Ma, & H. Li, 2012. Rsr2015 : Database for text-dependent speaker verification using multiple pass-phrases. Dans les actes de *INTERSPEECH*, 1580–1583.

(Lau et al., 2004) Y. W. Lau, M. Wagner, & D. Tran, 2004. Vulnerability of speaker verification to voice mimicking. Dans les actes de *International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004. Proceedings of 2004*, 145–148. IEEE.

(Lee et al., 2017) K. A. Lee, V. Hautamaki, A. Larcher, C. Zhang, A. Nautsch, T. Stafylakis, G. Liu, M. Rouvier, W. Rao, F. Alegre, et al., 2017. The i4u mega fusion and collaboration for NIST speaker recognition evaluation 2016. Dans les actes de *Interspeech 2017Annual Conference of the International Speech Communication Association*.

(Lee et al., 2015) K.-A. Lee, A. Larcher, G. Wang, P. Kenny, N. Brümmer, D. A. van Leeuwen, H. Aronowitz, M. Kockmann, C. Vaquero, B. Ma, et al., 2015. The reddots data collection for speaker recognition. Dans les actes de *INTERSPEECH*, 2996–3000.

(Lee et al., 2016) K. A. Lee, H. Sun, S. Aleksandr, W. Guangsen, et al., 2016. The i4u submission to the 2012 NIST speaker recognition evaluation. Dans les actes de *NIST Speaker Recognition Conference*.

(Lei et al., 2012) Y. Lei, L. Burget, L. Ferrer, M. Graciarena, & N. Scheffer, 2012. Towards noise-robust speaker recognition using probabilistic linear discriminant analysis. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4253–4256.

(Lei et al., 2013) Y. Lei, L. Burget, & N. Scheffer, 2013. A noise robust i-vector extractor using vector taylor series for speaker recognition. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6788–6791.

(Lei et al., 2014a) Y. Lei, M. McLaren, L. Ferrer, & N. Scheffer, 2014a. Simplified VTS-based i-vector extraction in noise-robust speaker recognition. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4037–4041.

(Lei et al., 2014b) Y. Lei, N. Scheffer, L. Ferrer, & M. McLaren, 2014b. A novel scheme for speaker recognition using a phonetically-aware deep neural network. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014.*, 1695–1699. IEEE.

(Levelt, 1993) W. J. Levelt, 1993. *Speaking : From intention to articulation*, Volume 1. MIT press.

(Li et Wrench, 1983) K. Li & E. Wrench, 1983. An approach to text-independent speaker recognition with short utterances. Dans les actes de *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'83.*, Volume 8, 555–558. IEEE.

(Li et Porter, 1988) K.-P. Li & J. E. Porter, 1988. Normalizations and selection of speech segments for speaker recognition scoring. Dans les actes de *International Conference on Acoustics, Speech, and Signal Processing, 1988. ICASSP-88.*, 595–598. IEEE.

(Li et Mak, 2015) N. Li & M.-W. Mak, 2015. Snr-invariant plda modeling for robust speaker verification. Dans les actes de *INTERSPEECH*, 2317–2321.

(Mak et Yu, 2014) M.-W. Mak & H.-B. Yu, 2014. A study of voice activity detection techniques for NIST speaker recognition evaluations. *Computer Speech & Language 28*(1), 295–313.

(Makhoul, 1975) J. Makhoul, 1975. Linear prediction : A tutorial review. *Proceedings of the IEEE 63*(4), 561–580.

(Mandasari et al., 2013) M. I. Mandasari, R. Saeidi, M. McLaren, & D. A. van Leeuwen, 2013. Quality measure functions for calibration of speaker recognition systems in various duration conditions. *IEEE Transactions on Audio, Speech, and Language Processing 21*(11), 2425–2438.

(Mandasari et al., 2015) M. I. Mandasari, R. Saeidi, & D. A. van Leeuwen, 2015. Quality measures based calibration with duration and noise dependency for speaker recognition. *Speech Communication 72*, 126–137.

(Markel et al., 1977) J. Markel, B. Oshika, & A. Gray, 1977. Long-term feature averaging for speaker recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing 25*(4), 330–337.

(Markel et Gray, 1982) J. E. Markel & A. Gray, 1982. Linear prediction of speech.

(Martin et al., 1997) A. Martin, G. Doddington, T. Kamm, M. Ordowski, & M. Przybocki, 1997. The DET curve in assessment of detection task performance. Rapport technique, DTIC Document.

(Martin et Przybocki, 2000) A. Martin & M. Przybocki, 2000. The NIST 1999 speaker recognition evaluation—an overview. *Digital signal processing 10*(1), 1–18.

(Martin, 1997) K. D. Martin, 1997. Echo suppression in a computational model of the precedence effect. Dans les actes de *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics, 1997*, 4–pp. IEEE.

(Martınez et al., 2014) D. Martınez, L. Burget, T. Stafylakis, Y. Lei, P. Kenny, & E. Lleida, 2014. Unscented transform for ivector-based noisy speaker recognition. *ICASSP, Florence, Italy*.

(Matvejka et al., 2016) P. Matvejka, O. Glembek, O. Novotnỳ, O. Plchot, F. Grézl, L. Burget, & J. H. Cernockỳ, 2016. Analysis of dnn approaches to speaker identification. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*, 5100–5104. IEEE.

(Matrouf et al., 2015) D. Matrouf, W. Ben Kheder, P.-M. Bousquet, J.-F. Bonastre, & M. Ajili, 2015. Dealing with additive noise in speaker recognition systems based on i-vector approach. *Signal Processing Conference (EUSIPCO), 2015 23rd European*, 2092–2096.

(Matrouf et al., 2006) D. Matrouf, J.-F. Bonastre, & C. Fredouille, 2006. Effect of speech transformation on impostor acceptance. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006.*, Volume 1, I–I. IEEE.

(Matrouf et al., 2007) D. Matrouf, N. Scheffer, B. G. B. Fauve, & J.-F. Bonastre, 2007. A straightforward and efficient implementation of the factor analysis model for speaker verification. Dans les actes de *Interspeech*, 1242–1245.

(McLaren et al., 2016) M. McLaren, L. Ferrer, D. Castan, & A. Lawson, 2016. The speakers in the wild (sitw) speaker recognition database. *Interspeech 2016*.

(McLaren et al., 2012) M. McLaren, M. I. Mandasari, & D. A. van Leeuwen, 2012. Source normalization for language-independent speaker recognition using i-vectors.

(McLaren et al., 2013) M. McLaren, N. Scheffer, M. Graciarena, L. Ferrer, & Y. Lei, 2013. Improving speaker identification robustness to highly channel-degraded speech through multiple system fusion. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*, 6773–6777. IEEE.

(McLaren et Van Leeuwen, 2012) M. McLaren & D. Van Leeuwen, 2012. Source-normalized lda for robust speaker recognition using i-vectors from multiple speech sources. *IEEE Transactions on Audio, Speech, and Language Processing 20*(3), 755–766.

(Mehrabani et Hansen, 2013) M. Mehrabani & J. H. Hansen, 2013. Singing speaker clustering based on subspace learning in the gmm mean supervector space. *Speech Communication 55*(5), 653–666.

(Morchid et al., 2016) M. Morchid, M. Bouaziz, W. Ben Kheder, K. Janod, P.-M. Bousquet, R. Dufour, & G. Linares, 2016. Spoken language understanding in a latent topic-based subspace. *Corpus 2*, d3.

(Moreno, 1996) P. J. Moreno, 1996. *Speech recognition in noisy environments*. Thèse de Doctorat, Carnegie Mellon University Pittsburgh.

(Moreno et al., 1998) P. J. Moreno, B. Raj, & R. M. Stern, 1998. Data-driven environmental compensation for speech recognition : A unified approach. *Speech Communication* *24*(4), 267–285.

(Mourjopoulos, 1985) J. Mourjopoulos, 1985. On the variation and invertibility of room impulse response functions. *Journal of Sound and Vibration* *102*(2), 217–228.

(Naik, 1994) J. Naik, 1994. Speaker verification over the telephone network : databases, algorithms and performance assessment. Dans les actes de *Automatic Speaker Recognition, Identification and Verification*.

(Nautsch et al., 2016) A. Nautsch, R. Saeidi, C. Rathgeb, & C. Busch, 2016. Robustness of quality-based score calibration of speaker recognition systems with respect to low-snr and short-duration conditions. *Odyssey 2016*, 358–365.

(Nemer et al., 1999) E. Nemer, R. Goubran, & S. Mahmoud, 1999. Snr estimation of speech signals using subbands and fourth-order statistics. *IEEE Signal Processing Letters* *6*(7), 171–174.

(Netsch et Doddington, 1992) L. P. Netsch & G. R. Doddington, 1992. Temporal decorrelation method for robust speaker verification. US Patent 5,167,004.

(nist2008eval, 2008) nist2008eval, 2008. The NIST year 2008 speaker recognition evaluation plan. http://www.itl.nist.gov/iad/mig//tests/sre/2008/.

(nist2010eval, 2010) nist2010eval, 2010. The NIST year 2010 speaker recognition evaluation plan. http://www.itl.nist.gov/iad/mig/tests/spk/2010/NISTSRE10evalplan.r6.pdf.

(nist2012eval, 2012) nist2012eval, 2012. The NIST year 2012 speaker recognition evaluation plan. http://www.nist.gov/itl/iad/mig/upload/NISTSRE12evalplan-v17-r1.pdf.

(nist2016eval, 2016) nist2016eval, 2016. NIST 2016 speaker recognition evaluation plan. https://www.nist.gov/sites/default/files/documents/itl/iad/mig/SRE16_Eval_Plan_V1-0.pdf.

(Novotnỳ et al., 2016) O. Novotnỳ, P. Matejka, O. Plchot, O. Glembek, L. Burget, & J. Cernockỳ, 2016. Analysis of speaker recognition systems in realistic scenarios of the sitw 2016 challenge. Dans les actes de *Interspeech*, 828–832.

(Openshaw et Masan, 1994) J. P. Openshaw & J. Masan, 1994. On the limitations of cepstral features in noise. Dans les actes de *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1994. ICASSP-94.*, Volume 2, II–49. IEEE.

(Paliwal et Basu, 1987) K. Paliwal & A. Basu, 1987. A speech enhancement method based on kalman filtering. Dans les actes de *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87.*, Volume 12, 177–180. IEEE.

(Patterson et al., 1992) R. D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, & M. Allerhand, 1992. Complex sounds and auditory images. *Auditory physiology and perception 83*, 429–446.

(Paul, 1987) D. Paul, 1987. A speaker-stress resistant hmm isolated word recognizer. Dans les actes de *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87.*, Volume 12, 713–716. IEEE.

(Perone, 2009) C. S. Perone, 2009. Pyevolve : a python open-source framework for genetic algorithms. *ACM SIGEVOlution 4*(1), 12–20.

(Pigeon et al., 2000) S. Pigeon, P. Druyts, & P. Verlinde, 2000. Applying logistic regression to the fusion of the nist'99 1-speaker submissions. *Digital Signal Processing 10*(1-3), 237–248.

(Plapous et al., 2006) C. Plapous, C. Marro, & P. Scalart, 2006. Improved signal-to-noise ratio estimation for speech enhancement. *IEEE Transactions on Audio, Speech, and Language Processing 14*(6), 2098–2108.

(Poli et al., 2008) R. Poli, W. B. Langdon, N. F. McPhee, & J. R. Koza, 2008. *A field guide to genetic programming*. Lulu. com.

(Prince et Elder, 2007) S. J. Prince & J. H. Elder, 2007. Probabilistic linear discriminant analysis for inferences about identity. Dans les actes de *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007.*, 1–8.

(Przybocki et Martin, 1999) M. Przybocki & A. Martin, 1999. Two channel telephone data for speaker detection and speaker tracking. Dans les actes de *European Conference on Speech Communication and Technology Eurospeech*.

(Ramos-Castro et al., 2006) D. Ramos-Castro, J. Gonzalez-Rodriguez, & J. Ortega-Garcia, 2006. Likelihood ratio calibration in a transparent and testable forensic speaker recognition framework. Dans les actes de *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006 : The*, 1–8. IEEE.

(Rao et Mak, 2013) W. Rao & M.-W. Mak, 2013. Boosting the performance of i-vector based speaker verification via utterance partitioning. *IEEE Transactions on Audio, Speech, and Language Processing 21*(5), 1012–1022.

(Ratnam et al., 2003) R. Ratnam, D. L. Jones, B. C. Wheeler, W. D. O'Brien Jr, C. R. Lansing, & A. S. Feng, 2003. Blind estimation of reverberation time. *The Journal of the Acoustical Society of America 114*(5), 2877–2892.

(Reynolds et al., 2003) D. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, et al., 2003. The supersid project : Exploiting high-level information for high-accuracy speaker recognition. Dans les actes de *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, Volume 4, IV–784. IEEE.

(Reynolds, 1992) D. A. Reynolds, 1992. *A Gaussian mixture modeling approach to text-independent speaker identification*. Thèse de Doctorat, Georgia Institute of Technology.

(Reynolds, 1996) D. A. Reynolds, 1996. The effects of handset variability on speaker recognition performance : Experiments on the switchboard corpus. Dans les actes de *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1996. ICASSP-96.*, Volume 1, 113–116. IEEE.

(Reynolds, 2003) D. A. Reynolds, 2003. Channel robust speaker verification via feature mapping. Dans les actes de *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, Volume 2, II–53. IEEE.

(Reynolds et al., 2000) D. A. Reynolds, T. F. Quatieri, & R. B. Dunn, 2000. Speaker verification using adapted gaussian mixture models. *Digital signal processing 10*(1), 19–41.

(Reynolds et Rose, 1995) D. A. Reynolds & R. C. Rose, 1995. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE transactions on speech and audio processing 3*(1), 72–83.

(Reynolds et al., 1995) D. A. Reynolds, M. A. Zissman, T. F. Quatieri, G. C. O'Leary, & B. A. Carlson, 1995. The effects of telephone transmission degradations on speaker recognition performance. Dans les actes de *International Conference on Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995.*, Volume 1, 329–332. IEEE.

(Ribas et al., 2016) D. Ribas, E. Vincent, & J. Calvo, 2016. A study of speech distortion conditions in real scenarios for speech processing applications. Dans les actes de *2016 IEEE Workshop on Spoken Language Technology*.

(Ribas et al., 2015a) D. Ribas, E. Vincent, & J. R. Calvo, 2015a. Full multicondition training for robust i-vector based speaker recognition. Dans les actes de *Interspeech 2015*.

(Ribas et al., 2015b) D. Ribas, E. Vincent, & J. R. Calvo, 2015b. Uncertainty propagation for noise robust speaker recognition : the case of nist-sre. Dans les actes de *Interspeech 2015*, 5.

(Richardson et al., 2015) F. Richardson, D. Reynolds, & N. Dehak, 2015. A unified deep neural network for speaker and language recognition. *arXiv preprint arXiv :1504.00923*.

(Rose et al., 1994) R. C. Rose, E. M. Hofstetter, & D. A. Reynolds, 1994. Integrated models of signal and background with application to speaker identification in noise. *IEEE Transactions on Speech and Audio Processing 2*(2), 245–257.

(Rosenberg et al., 1998) A. Rosenberg, I. Magrin-Chagnolleau, & S. Parthasarathy, 1998. Speaker detection in broadcast speech databases. Dans les actes de *Proceedings of International Conference on Spoken Language Processsing*.

(Rosenberg, 1992) A. E. Rosenberg, 1992. Recent research in automatic speaker recognition. *Advances in speech signal processing*, 701–738.

(Rouvier et al., 2016) M. Rouvier, P.-M. Bousquet, M. Ajili, W. Ben Kheder, D. Matrouf, & J.-F. Bonastre, 2016. Lia system description for NIST sre 2016. *arXiv preprint arXiv :1612.05168*.

(Rouvier et al., 2015) M. Rouvier, P.-M. Bousquet, & B. Favre, 2015. Speaker diarization through speaker embeddings. Dans les actes de *Signal Processing Conference (EUSIPCO), 2015 23rd European*, 2082–2086. IEEE.

(Sadjadi et al., 2016) S. O. Sadjadi, S. Ganapathy, & J. W. Pelecanos, 2016. The ibm 2016 speaker recognition system. *arXiv preprint arXiv :1602.07291*.

(Sadjadi et Hansen, 2015) S. O. Sadjadi & J. H. Hansen, 2015. Mean hilbert envelope coefficients (mhec) for robust speaker and language identification. *speech communication 72*, 138–148.

(Sadjadi et al., 2012) S. O. Sadjadi, T. Hasan, & J. H. Hansen, 2012. Mean hilbert envelope coefficients (mhec) for robust speaker recognition. Dans les actes de *INTERSPEECH*, 1696–1699.

(Sadjadi et al., 2014) S. O. Sadjadi, J. W. Pelecanos, & W. Zhu, 2014. Nearest neighbor discriminant analysis for robust speaker recognition. Dans les actes de *INTERSPEECH*, 1860–1864.

(Saeidi et al., 2012) R. Saeidi, A. Hurmalainen, T. Virtanen, & D. A. van Leeuwen, 2012. Exemplar-based sparse representation and sparse discrimination for noise robust speaker identification. Dans les actes de *Odyssey*, 248–255. Citeseer.

(Sahidullah et Saha, 2012) M. Sahidullah & G. Saha, 2012. Comparison of speech activity detection techniques for speaker recognition. *arXiv preprint arXiv :1210.0297*.

(Salakhutdinov et Hinton, 2009) R. Salakhutdinov & G. Hinton, 2009. Deep boltzmann machines. Dans les actes de *Artificial Intelligence and Statistics*, 448–455.

(Sambur, 1975) M. Sambur, 1975. Selection of acoustic features for speaker identification. *IEEE Transactions on Acoustics, Speech, and Signal Processing 23*(2), 176–182.

(Sarkar et al., 2014) A. K. Sarkar, C.-T. Do, V.-B. Le, & C. Barras, 2014. Combination of cepstral and phonetically discriminative features for speaker verification. *IEEE Signal Processing Letters 21*(9), 1040–1044.

(Sarkar et al., 2012) A. K. Sarkar, D. Matrouf, P.-M. Bousquet, & J.-F. Bonastre, 2012. Study of the effect of i-vector modeling on short and mismatch utterance duration for speaker verification. Dans les actes de *Interspeech*, 2662–2665.

(Sarkar et Sreenivasa Rao, 2014) S. Sarkar & K. Sreenivasa Rao, 2014. Stochastic feature compensation methods for speaker verification in noisy environments. *Applied Soft Computing 19*, 198–214.

(Schwartz et al., 1993) R. Schwartz, T. Anastasakos, F. Kubala, J. Makhoul, L. Nguyen, & G. Zavaliagkos, 1993. Comparative experiments on large vocabulary speech recognition. Dans les actes de *Proceedings of the workshop on Human Language Technology*, 75–80. Association for Computational Linguistics.

(Serizel et al., 2016) R. Serizel, S. Essid, & G. Richard, 2016. Group nonnegative matrix factorisation with speaker and session variability compensation for speaker identification. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016*, 5470–5474. IEEE.

(Seuret et al., 2017) M. Seuret, M. Alberti, R. Ingold, & M. Liwicki, 2017. Pca-initialized deep neural networks applied to document image analysis. *arXiv preprint arXiv :1702.00177*.

(Shriberg et al., 2005) E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, & A. Stolcke, 2005. Modeling prosodic feature sequences for speaker recognition. *Speech Communication 46*(3), 455–472.

(Solomonoff et al., 2004) A. Solomonoff, C. Quillen, & W. M. Campbell, 2004. Channel compensation for svm speaker recognition. Dans les actes de *Odyssey*, Volume 4, 219–226. Citeseer.

(Sönmez et al., 1999) M. K. Sönmez, L. P. Heck, & M. Weintraub, 1999. Speaker tracking and detection with multiple speakers. Dans les actes de *Eurospeech*.

(Soong et Rosenberg, 1988) F. K. Soong & A. E. Rosenberg, 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., 1987) F. K. Soong, A. E. Rosenberg, B.-H. Juang, & L. R. Rabiner, 1987. Report : A vector quantization approach to speaker recognition. *AT&T technical journal 66*(2), 14–26.

(Swets, 1964) J. A. Swets, 1964. Signal detection and recognition in human observers : Contemporary readings.

(Tan et al., 2016) Z. Tan, Y. Zhu, M.-W. Mak, & B. K.-W. Mak, 2016. Senone i-vectors for robust speaker verification. Dans les actes de *10th International Symposium on Chinese Spoken Language Processing (ISCSLP), 2016*, 1–5. IEEE.

(Variani et al., 2014) E. Variani, X. Lei, E. McDermott, I. L. Moreno, & J. Gonzalez-Dominguez, 2014. Deep neural networks for small footprint text-dependent speaker verification. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014.*, 4052–4056. IEEE.

(Vincent et al., 2008) P. Vincent, H. Larochelle, Y. Bengio, & P.-A. Manzagol, 2008. Extracting and composing robust features with denoising autoencoders. Dans les actes de *Proceedings of the 25th international conference on Machine learning*, 1096–1103. ACM.

(Vogt et al., 2008) R. J. Vogt, S. Kajarekar, & S. Sridharan, 2008. Discriminant nap for svm speaker recognition.

(Vollert, 1992) E. Vollert, 1992. Method for speaker recognition in a telephone switching system. US Patent 5,166,971.

(Wolf, 1972) J. J. Wolf, 1972. Efficient acoustic parameters for speaker recognition. *The Journal of the Acoustical Society of America 51*(6B), 2044–2056.

(Wu et al., 2015) Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, & H. Li, 2015. Spoofing and countermeasures for speaker verification : a survey. *Speech Communication 66*, 130–153.

(Wu et al., 2012) Z. Wu, C. E. Siong, & H. Li, 2012. Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition. Dans les actes de *Interspeech*, 1700–1703.

(Yamada et Hattori, 1999) E. Yamada & H. Hattori, 1999. Tree structured cohort selection for speaker recognition system. US Patent 6,006,184.

(Yamada et al., 2013) T. Yamada, L. Wang, & A. Kai, 2013. Improvement of distant-talking speaker identification using bottleneck features of dnn. Dans les actes de *Interspeech*, 3661–3664.

(Yaman et al., 2012) S. Yaman, J. Pelecanos, & R. Sarikaya, 2012. Bottleneck features for speaker recognition. Dans les actes de *Odyssey 2012-The Speaker and Language Recognition Workshop*.

(Ye et al., 2013) H. Ye, G. Deng, S. J. Mauger, A. A. Hersbach, P. W. Dawson, & J. M. Heasman, 2013. A wavelet-based noise reduction algorithm and its clinical evaluation in cochlear implants. *PloS one 8*(9), e75662.

(Yoshioka et al., 2012) T. Yoshioka, A. Sehr, M. Delcroix, K. Kinoshita, R. Maas, T. Nakatani, & W. Kellermann, 2012. Making machines understand us in reverberant rooms : robustness against reverberation for automatic speech recognition. *IEEE Signal Processing Magazine 29*(6), 114–126.

(Yu et al., 2014) C. Yu, G. Liu, S. Hahm, & J. H. Hansen, 2014. Uncertainty propagation in front end factor analysis for noise robust speaker recognition. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014*, 4017–4021. IEEE.

(Zen et al., 2009) H. Zen, Y. Nankaku, & K. Tokuda, 2009. Stereo-based stochastic noise compensation based on trajectory GMMS. Dans les actes de *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4577–4580.

(Zhang et Hansen, 2007) C. Zhang & J. H. Hansen, 2007. Analysis and classification of speech mode : whispered through shouted. Dans les actes de *INTERSPEECH*, Volume 7, 2289–2292.

(Zhang et Hansen, 2011) C. Zhang & J. H. Hansen, 2011. Whisper-island detection based on unsupervised segmentation with entropy-based speech feature processing. *IEEE Transactions on Audio, Speech, and Language Processing 19*(4), 883–894.

Bibliographie personnelle

Revues internationales avec comité de sélection

BEN KHEDER W., MATROUF D., BOUSQUET PM., BONASTRE JF ET AJILI M. « Fast i-vector denoising using MAP estimation and a noise distributions database for robust speaker recognition » *dans Computer Speech & Language, Elsevier*, 2017

Conférences d'audience internationale avec comité de sélection

BEN KHEDER W., AJILI M., BOUSQUET P-M., MATROUF D. ET BONASTRE J-F. « LIA system for the SITW Speaker Recognition Challenge » *dans Interspeech*, 2016

BEN KHEDER W., MATROUF D., AJILI M. ET BONASTRE J-F. « Probabilistic approach using joint long and short session i-vectors modeling to deal with short utterances for speaker recognition » *dans Interspeech*, 2016

BEN KHEDER W., MATROUF D., AJILI M. ET BONASTRE J-F. « Probabilistic approach using joint clean and noisy i-vectors modeling for speaker recognition » *dans Interspeech*, 2016

BEN KHEDER W., MATROUF D., AJILI M. ET BONASTRE J-F. « Local binary patterns as features for speaker recognition » *dans Speaker and Language Recognition Workshop Odyssey*, 2016

BEN KHEDER W., MATROUF D., AJILI M. ET BONASTRE J-F. « Iterative Bayesian and MMSE-based noise compensation techniques for speaker recognition in the i-vector space » *dans Speaker and Language Recognition Workshop Odyssey*, 2016

MATROUF D., BEN KHEDER W., BOUSQUET P-M., BONASTRE J-F. ET AJILI M. « Dealing with additive noise in speaker recognition systems based on i-vector ap-

proach » *dans Signal Processing Conference (EUSIPCO)*, 2015

BEN KHEDER W., MATROUF D., BOUSQUET PM., BONASTRE JF ET AJILI M. « Additive noise compensation in the I-vector space for speaker recognition » *dans ICASSP*, 2015

BEN KHEDER W., MATROUF D., BOUSQUET PM., BONASTRE JF ET AJILI M. « Robust speaker recognition using MAP estimation of additive noise in i-vectors space » *dans Statistical Language and Speech Processing*, 2014

Autres travaux

Mis à part les travaux réalisés dans le cadre de cette thèse, des collaborations ont été menés avec des collègues du laboratoire LIA ainsi que des chercheurs faisant partie d'autres équipes de recherche dans le domaine du TALN (Lee et al., 2016, 2017; Ajili et al., 2017a; Laaridh et al., 2017; Rouvier et al., 2016; Ajili et al., 2017b, 2016; Ben Kheder et al., 2016b; Morchid et al., 2016; Bouaziz et al., 2016).

Appendices

Annexe A

Apprentissage de la matrice de variabilité totale et estimation d'i-vectors

Le paradigme de variabilité totale se base sur le modèle d'analyse factorielle suivant :

$$m_{(s,h)} = m + \mathbf{T}\mathbf{x}_{(h,s)} \tag{A.1}$$

où la matrice de variabilité totale $\mathbf{T}$ est apprise en utilisant un grand ensemble de données d'entraînement et $\mathbf{x}_{(h,s)}$ est une estimation maximum à posteriori (MAP) correspondant aux observations correspondant à une session donnée. Les indices h et s font respectivement référence à un locuteur s et une session h.

Soient $\mathbf{N}_{(h,s)}$ et $\mathbf{X}_{(h,s)}$ deux vecteurs contenant respectivement les statistiques de premier et second ordre relatifs à une session donnée.

Les statistiques sont calculées par rapport à un modèle du monde (UBM) :

$$\mathbf{N}_{(h,s)}[g] = \sum_{t \in (h,s)} \gamma_g(t) \tag{A.2}$$

$$\{\mathbf{X}_{(h,s)}\}_{[g]} = \sum_{t \in (h,s)} \gamma_g(t) \cdot t \tag{A.3}$$

où $\gamma_g(t)$ représente la probabilité *à posteriori* de la Gaussienne g pour une observation t. Dans cette équation, $\sum_{t \in (h,s)}$ représente la somme sur toutes les trames appartenant à une session d.

Soient $\overline{\mathbf{X}}_{(h,s)}$ les statistiques dépendant de la session et du locuteur définis par :

$$\{\overline{\mathbf{X}}_{(h,s)}\}_{[g]} = \{\mathbf{X}_{(h,s)}\}_{[g]} - \mathbf{m}_{[g]} \cdot \sum_{(h,s)} \mathbf{N}_{(h,s)}[g] \tag{A.4}$$

Soit $\mathbf{L}_{(h,s)}$ une matrice de taille $R \times R$, et $\mathbf{B}_{(h,s)}$ un vecteur de taille R définis comme :

$$\begin{aligned} \mathbf{L}_{(h,s)} &= \mathbf{I} + \sum_{g \in \text{UBM}} \mathbf{N}_{(h,s)}[g] \cdot \{\mathbf{T}\}_{[g]}^{t} \cdot \mathbf{\Sigma}_{[g]}^{-1} \cdot \{\mathbf{T}\}_{[g]} \\ \mathbf{B}_{(h,s)} &= \sum_{g \in \text{UBM}} \{\mathbf{T}\}_{[g]}^{t} \cdot \mathbf{\Sigma}_{g}^{-1} \cdot \{\overline{\mathbf{X}}_{(h,s)}\}_{[g]} \, . \end{aligned} \tag{A.5}$$

En utilisant $\mathbf{L}_{(h,s)}$ et $\mathbf{B}_{(h,s)}$, $\mathbf{x}_{(h,s)}$ peut être obtenu rn utilisant l'équation :

$$\mathbf{x}_{(h,s)} = \mathbf{L}_{(h,s)}^{-1} \cdot \mathbf{B}_{(h,s)} \tag{A.6}$$

La matrice $\mathbf{T}$ peut être estimée ligne par ligne, avec $\{\mathbf{T}\}_{[g]}^{i}$ la i^{eme} ligne de $\{\mathbf{T}\}_{[g]}$, alors :

$$\mathbf{T}_{[g]}^{i} = \mathbf{LU}_{g}^{-1} \cdot \mathbf{RU}_{g}^{i}, \tag{A.7}$$

où $\mathbf{RU}_{g}^{i}$ et $\mathbf{LU}_{g}$ sont donnés par :

$$\begin{aligned} \mathbf{LU}_{g} &= \sum_{(h,s)} \mathbf{L}_{(h,s)}^{-1} + \mathbf{x}_{(h,s)} \mathbf{x}_{(h,s)}^{t} \cdot \mathbf{N}_{(h,s)}[g] \\ \mathbf{RU}_{g}^{i} &= \sum_{(h,s)} \{\overline{\mathbf{X}}_{(h,s)}\}_{[g]}^{[i]} \cdot \mathbf{x}_{(h,s)} \end{aligned} \tag{A.8}$$

Les algorithmes 3 et 4 donnent respectivement le pseudo-code pour la procédure d'apprentissage de la matrice de variabilité totale T et le processus d'estimation de l'i-vecteur correspondant à une sessions donnée. Une fonction de vraisemblance standard peut être utilisée pour évaluer la convergence comme détaillé dans (Matrouf et al., 2007).

Algorithm 3: Estimation de la matrice $\mathbf{T}$ et la variable latente $\mathbf{x}_{(h,s)}$.

```
Pour chaque session h et locuteur s :
x_(h,s) ← 0, T ← initialisation aléatoire ;
Estimer les statistiques : N_(h,s), X_(h,s) (équation A.3);
for i = 1 jusqu'à nb_iterations do
    for tous les h et s do
        Centrer les statistiques : X̄_(h,s) (équation A.4);
        Estimer L_(h,s) et B_(h,s) (équation A.5);
        Estimer x_(h,s) (équation A.6);
    end
    Estimer la matrice T (équation A.7 et A.8);
end
```

Algorithm 4: Estimation de la variable latente $\mathbf{x}_{(h_0,s_0)}$ pour une session donnée.

Pour chaque session h et locuteur s :
$\mathbf{x}_{(h_0,s_0)} \leftarrow 0$;
Estimer les statistiques : $\mathbf{N}_{(h_0,s_0)}$, $\mathbf{X}_{(h_0,s_0)}$ (équation A.3);
for $i = 1$ *jusqu'à nb_iterations* **do**
    Centrer les statistiques : $\overline{\mathbf{X}}_{(h_0,s_0)}$ (équation A.4);
    Estimer $\mathbf{L}_{(h_0,s_0)}$ et $\mathbf{B}_{(h_0,s_0)}$ (équation A.5);
    Estimer $\mathbf{x}_{(h_0,s_0)}$ (équation A.6);
end

Annexe B

Démonstration de l'équation 7.6

Étant donnés trois variables aléatoires $x \sim \mathcal{N}(\mu_x, \Sigma_x)$, $y \sim \mathcal{N}(\mu_y, \Sigma_y)$ et z définit comme $z = \begin{pmatrix} x \\ y \end{pmatrix}$, les hyper-paramètres de la distribution $p(z)$ peut être réécrite comme $z \sim \mathcal{N}(\mu_z, \Sigma_z)$ avec :

$$\mu_z = \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix} \tag{B.1}$$

$$\Sigma_z = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix} \tag{B.2}$$

En utilisant le théorème d'inversion de matrice, la matrice de covariance Σ_z^{-1} peut être réécrite sous la forme :

$$\Sigma_z^{-1} = \Lambda = \begin{bmatrix} \Lambda_{xx} & \Lambda_{xy} \\ \Lambda_{yx} & \Lambda_{yy} \end{bmatrix} \tag{B.3}$$

avec :

$$\Lambda_{xx} = (\Sigma_{xx} - \Sigma_{xy}{\Sigma_{yy}}^{-1}\Sigma_{yx})^{-1} \tag{B.4}$$

$$\Lambda_{yy} = {\Sigma_{yy}}^{-1} + {\Sigma_{yy}}^{-1}\Sigma_{yx}(\Sigma_{xx} - \Sigma_{xy}{\Sigma_{yy}}^{-1}\Sigma_{yx})^{-1}\Sigma_{xy}{\Sigma_{yy}}^{-1} \tag{B.5}$$

$$\Lambda_{xy} = -(\Sigma_{xx} - \Sigma_{xy}{\Sigma_{yy}}^{-1}\Sigma_{yx})^{-1}\Sigma_{xy}{\Sigma_{yy}}^{-1} \tag{B.6}$$

$$\Lambda_{yx} = -{\Sigma_{yy}}^{-1}\Sigma_{yx}(\Sigma_{xx} - \Sigma_{xy}{\Sigma_{yy}}^{-1}\Sigma_{yx})^{-1} \tag{B.7}$$

D'après le théorème des densités conditionnelles, on peut écrire :

$$p(x|y) = \frac{p(x,y)}{p(y)} = \frac{\frac{1}{(2\pi)^n|\Sigma_z|^{1/2}}exp\{-(z-\mu_z)^T\Sigma_z^{-1}(z-\mu_z)\}}{\frac{1}{(2\pi)^{n/2}|\Sigma_{yy}|^{1/2}}exp\{-(y-\mu_y)^T\Sigma_{yy}^{-1}(y-\mu_y)}\} = K.exp(E) \tag{B.8}$$

avec :

$$\begin{cases} E = -\frac{1}{2}(z-\mu_z)^T\Sigma_z^{-1}(z-\mu_z) + \frac{1}{2}(y-\mu_y)^T\Sigma_{yy}^{-1}(y-\mu_y) \\ K = \sqrt{\frac{|\Sigma_{yy}|}{(2\pi)^{n/2}|\Sigma_z|}} \end{cases} \tag{B.9}$$

Simplification de K :

En utilisant la formule du déterminant pour les matrices définies par blocs, $|\Sigma_z|$ peut être écrite sous la forme :

$$|\Sigma_z| = \begin{vmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{vmatrix} = |\Sigma_{yy}|.|\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}| \tag{B.10}$$

En injectant ce résultat dans l'équation B.9, on retrouve :

$$K = \frac{1}{(2\pi)^{n/2}|\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}|} = \frac{1}{(2\pi)^{n/2}|\Lambda_{xx}^{-1}|} \tag{B.11}$$

Simplification de E :

$$\begin{aligned} E =& -\frac{1}{2}(z-\mu_z)^T\Sigma_z^{-1}(z-\mu_z) + \frac{1}{2}(y-\mu_y)^T\Sigma_{yy}^{-1}(y-\mu_y) \\ =& -\frac{1}{2}\begin{pmatrix} x-\mu_x \\ y-\mu_y \end{pmatrix}^T \begin{bmatrix} \Lambda_{xx} & \Lambda_{xy} \\ \Lambda_{yx} & \Lambda_{yy} \end{bmatrix} \begin{pmatrix} x-\mu_x \\ y-\mu_y \end{pmatrix} + \frac{1}{2}(y-\mu_y)^T\Sigma_{yy}^{-1}(y-\mu_y) \\ =& -\frac{1}{2}(x-\mu_x)^T\Lambda_{xx}(x-\mu_x) - \frac{1}{2}(x-\mu_x)^T\Lambda_{xy}(x-\mu_x) \\ & -\frac{1}{2}(y-\mu_y)^T(\Lambda_{yy} - \Sigma_{yy}^{-1})(y-\mu_y) - \frac{1}{2}(y-\mu_y)^T\Lambda_{yx}(y-\mu_y) \end{aligned} \tag{B.12}$$

En réorganisant les termes de l'équation B.12, on retrouve :

$$\begin{aligned} E =& -\frac{1}{2}x^T\Lambda_{xx}x + x^T(\Lambda_{xx}\mu_x - \Lambda_{xy}(y-\mu_y)) - \frac{1}{2}(y-\mu_y)^T(\Lambda_{yy} - \Sigma_{yy}^{-1})(y-\mu_y) \\ & -\frac{1}{2}\mu_x^T\Lambda_{xx}\mu_x + \mu_x^T\Lambda_{xy}(y-\mu_y) \end{aligned} \tag{B.13}$$

En utilisant la propriété :

$$\Sigma_{yy}^{-1} = \Lambda_{yy} - \Lambda_{yx}\Lambda_{xx}^{-1}\Lambda_{xy} \tag{B.14}$$

L'équation B.13 devient :

$$\begin{aligned} E =& -\frac{1}{2}x^T\Lambda_{xx}x + x^T(\Lambda_{xx}\mu_x - \Lambda_{xy}(y-\mu_y)) - \frac{1}{2}(y-\mu_y)^T(\Lambda_{yx}\Sigma_{xx}^{-1}\Lambda_{xy})(y-\mu_y) \\ & -\frac{1}{2}\mu_x^T\Lambda_{xx}\mu_x + \mu_x^T\Lambda_{xy}(y-\mu_y) \\ =& -\frac{1}{2}(x - (\Lambda_{xx}\mu_x - \Lambda_{xy}(y-\mu_y)))^T\Lambda_{xx}(x - (\Lambda_{xx}\mu_x - \Lambda_{xy}(y-\mu_y))) \end{aligned} \tag{B.15}$$

En combinant des équations B.15, B.11 et B.8 :

$$p(x|y) = \frac{1}{(2\pi)^{n/2}|\Lambda_{xx}^{-1}|} exp\{-\frac{1}{2}(x - (\Lambda_{xx}\mu_x - \Lambda_{xy}(y - \mu_y)))^T \Lambda_{xx}(x - (\Lambda_{xx}\mu_x - \Lambda_{xy}(y - \mu_y)))\} \tag{B.16}$$

ce qui donne :

$$E[x|y] = \Lambda_{xx}\mu_x - \Lambda_{xy}(y - \mu_y) \tag{B.17}$$

en utilisant les équations B.4 et B.6, $E[x|y]$ et $\Sigma_{x|y}$ peuvent être écrits sous la forme :

$$\begin{cases} E[x|y] & = \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y) \\ \Sigma_{x|y} & = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} \end{cases} \tag{B.18}$$

ceci conclu la démonstration.