Titre: Title:	Les techniques de normalisation appliquées à la vérification du locuteur
Auteur: Author:	Charles Dugas
Date:	1998
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Dugas, C. (1998). Les techniques de normalisation appliquées à la vérification du locuteur [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/6894/

UNIVERSITÉ DE MONTRÉAL

LES TECHNIQUES DE NORMALISATION APPLIQUÉES À LA VÉRIFICATION DU LOCUTEUR

CHARLES DUGAS
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ET DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)
AOÛT 1998

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

LES TECHNIQUES DE NORMALISATION APPLIQUÉES À LA VÉRIFICATION DU LOCUTEUR

présenté par: DUGAS Charles

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. MOORE Marc, Ph.D., président

M. BRAULT Jean-Jules, Ph.D., membre et directeur de recherche

M. DUMOUCHEL Pierre, Ph.D., membre et codirecteur de recherche

M. O'SHAUGHNESSY Douglas, Ph.D., membre

Remerciements

Merci à mes deux amis et directeurs Jean-Jules Brault et Pierre Dumouchel, pour leurs commentaires judicieux qui m'ont permis d'orienter efficacement ma recherche et me lancer sur des pistes profitables. Merci de m'avoir permis de vagabonder, d'être libre.

Merci au Fonds pour la formation de Chercheurs et l'Aide à la Recherche (FCAR), au Conseil de Recherche en Sciences Naturelles et Génie et au Centre de Recherche Informatique de Montréal (CRIM) pour leur soutien financier.

Merci à l'équipe GREP (Groupe de Reconnaissance de la Parole): France, Pierre, Gilles, Nathalie, Julie et Andrée-Hélène pour leur amitié.

Merci à tous ceux qui m'ont supporté et tenté de me comprendre lors de mon retour aux études. Cette décision, longtemps mûrie, fut parmi les plus bénéfiques.

Merci à mes parents, ma soeur et mes amis pour leur simple présence rafraîchissante et leur intérêt par rapport à mes travaux.

Merci à Christiane qui m'accompagne depuis le début de mes études universitaires jusqu'à tout récemment. Les valeurs de vie qui régissent actuellement mes idées et actions résultent en grande partie de notre merveilleuse relation qui m'a permis de devenir une meilleure personne.

Résumé

Ce mémoire de maîtrise vise à couvrir l'ensemble des techniques de normalisation actuellement utilisées dans le domaine de la vérification du locuteur et de proposer certaines nouvelles techniques qui semblent, à la lumière des résultats obtenus et présentés, permettre une amélioration de la qualité des systèmes de vérification du locuteur. Ce document vise aussi à décrire clairement les algorithmes utilisés pour le traitement de signal et la modélisation des paramètres extraits de ce signal.

Les locuteurs utilisés comme imposteurs n'avaient pas d'intention délictueuse *a priori* et sont appelés, en anglais, "casual impostors". Autrement dit, ces locuteurs ne tentaient pas d'accéder aux ressources d'une autre personne en imitant sa voix. De tels corpora de données standards et avec un nombre important de locuteurs n'existent pas encore. Naturellement, la tâche de vérification serait certainement plus difficile et il est donc souhaitable qu'un tel corpus soit disponible sous peu, puisque certaines applications commencent déjà à voir le jour et que la tâche de vérification qui correspond à un tel corpus de données se trouve beaucoup plus conforme à la réalité à laquelle les systèmes de vérification seront confrontés. La paramétrisation d'un signal constitue en soi une hypothèse, i.e., que ces paramètres permettent d'extraire les caractéristiques distinctives du signal observé pour la tâche envisagée. Les

locuteurs sont modélisés par des mixtures de gaussiennes sur les paramètres extraits. Là aussi, la structure choisie pour la modélisation constitue une hypothèse.

Les expériences ont été réalisées à l'aide du système HTK de reconnaissance de la parole qui a été adapté à l'École de Technologie Supérieure (ÉTS) et au Centre de recherche informatique de Montréal (CRIM), pour la reconnaissance du locuteur. Ces expériences ont été réalisées au CRIM sur des machines de type Sun Solaris sur le système d'exploitation UNIX. Le corpus de données utilisé fut SPIDRE.

Les techniques de normalisation proposées de la muselière et de l'impact ont été comparées à la technique de la cohorte. Les résultats semblent indiquer que lorsque le système opère en situation d'appariement et que la durée des fichiers tests est suffisante, la muselière et l'impact permettent de surpasser les résultats obtenus par la cohorte. De plus, dans tous les cas observés, la muselière s'est révélée meilleure que la cohorte lorsque le seuil d'acceptation est situé à un niveau élevé, i.e., pour des applications de haute sécurité.

Les conclusions à tirer des expériences réalisées sont les suivantes: la muselière devrait être préférée à la cohorte en situation d'appariement et lorsque la durée des fichiers de test est supérieure à environ 10 secondes, ou bien, lorsque des applications à des fins relativement sécuritaires sont envisagées.

Certaines compagnies ont déjà implanté des systèmes de reconnaissance de locuteur. Les applications envisagées sont les suivantes: accès aux installations, guichets automatiques, cartes d'appels (système téléphonique), accès par le téléphone à cer-

tains services (banque, comptes personnels, impôt, etc.), accès aux réseaux informatiques et surveillance électronique.

La reconnaissance du locuteur par le biais du téléphone dispose d'un avantage clair sur les autres biométriques qui pourraient éventuellement être utilisées: un réseau de transmission déjà installé pratiquement partout sur la planète. Cet avantage lui réserve donc une panoplie d'applications pour lesquelles l'utilisation d'autres biométriques nécessiterait le déploiement d'une infrastructure trop coûteuse.

Abstract

This master's thesis tries to cover all presently used background modelling techniques in the field of speaker verification and to propose certain novel techniques that seem, from the results obtained and presented, to allow an improvement of the verification system's quality. This document also wishes to clearly describe the algorithms used for signal processing and the modelling of it.

Speakers used as imposters were casual imposters, that is they were not deliberately trying to access the resources of another person by imitating his or her voice. Yet, no such corpus with a sufficiently large number of speakers has been accepted as standard in the research community. Naturally, the verification task would be more difficult and hopefully, such a corpus will be available in the near future since some applications are already being implemented on the field and the task corresponding to such a corpus is closer to the reality that systems are about to face. The parameterization of a signal is a hypothesis in itself, i.e., that the parameters allow the extraction of the distinctive characteristics of the observed signal for the task at hand. The speakers are further modelled with gaussian mixture models on the extracted parameters. There again lies a hypothesis.

Experiments were done with the use of the HTK speech recognition system, adapted at the École de Technologie Supérieure (ÉTS) and at the Centre de recherche informatique de Montréal (CRIM), for the speaker recognition task. These experiments were conducted at CRIM on Sun Solaris machines with the UNIX operating system. The standard SPIDRE data corpus was used.

The proposed normalization techniques, namely the muzzle and impact, were compared to the impostor cohort normalization (ICN) technique. Results indicate that in matched conditions and with sufficient test duration, the muzzle and impact beat the cohort. Also, in all cases, the muzzle outperformed the cohort when the acceptance threshold was set to a high level, that is, for high security applications.

The conclusions to which this document leads are as follows: the muzzle should be preferred to the cohort in case of matched conditions and test durations greater than 10 seconds or when relatively secure applications are envisioned.

Some enterprises have already implemented speaker recognition systems. The potential applications of such systems are plant access, communication with automated teller machines, telephone cards, access through telephone to certain services (banking, income taxes), access to computer networks and electronic surveillance.

Speaker recognition through the use of telephone has a clear advantage over other biometrics: an almost ubiquitous transmission network. Some applications are therefore confined to choose speaker recognition among biometrics as the use of an other one would require the building of a far too costly network.

Table des matières

Liste des Tableaux

Liste des Figures

Liste des sigles et abréviations

$\forall$	pour toutes les valeurs
$E[\cdot]$	espérance mathématique de
$\in$	est élément de
Hz	un Hertz i.e., un cycle par seconde
kHz	mille Hertz
$\mathbb{N}$	ensemble des nombres naturels
$\prod$	produit sur les valeurs
$\mathbb{R}$	ensemble des nombres réels
$\mathbb{R}^+$	ensemble des nombres réels supérieurs ou égaux à zéro
$\sum$	somme sur les valeurs
$\mathbb{Z}$	ensemble des nombres entiers
$\rightarrow$	tens vers ou converge vers
$\Rightarrow$	relation d'implication
$\approx$	approximativement égal à
$\gg$	beaucoup plus grand que
$\ll$	beaucoup plus petit que

Introduction

Les grandes entreprises du domaine des télécommunications ont clairement identifié le défi qu'elles devront relever afin de se tailler une part du marché ou simplement de préserver celle dont elles disposent: permettre l'accès rapide et fiable à une panoplie de services à valeur ajoutée de même qu'à plusieurs banques de données importantes et parfois confidentielles. Nortel annonçait, ce printemps 1998, son intention d'accélérer la recherche dans la transmission de données numériques par la voie de communications mobiles. Pressée par le temps et peut-être aussi par l'évolution de la compétition, Nortel achetait, quelques mois plus tard, Bay Networks, une compagnie spécialisée dans ce domaine.

Nortel s'est spécialisée, depuis plusieurs années, dans la transmission des données de la voix (ou parole) entre divers agents mobiles. La demande croissante pour la transmission de données numériques diverses aidant, le développement de la technologie nécessaire à cette fin suivra. La quantité accrue de données devant ainsi être transmises fait ressurgir le problème fondamental de la rapidité de transmission. Les solutions peuvent provenir du développement de nouveaux algorithmes ou de la réallocation de la bande passante.

Advenant la réussite technologique permettant la réalisation d'un tel projet, son épanouissement à grande échelle pourrait tout de même être ralenti si elle n'est pas accompagnée, en parallèle, par le développement de systèmes sécuritaires performants.

Plusieurs histoires cauchemardesques se sont succédées au cours des dernières années, dont la plus récente, liée au Pentagone, qui relate l'intrusion d'un groupe international de "hackers" dans le réseau informatique de la défense américaine. La firme comptable Ernst and Young estime que de 3 à 5 milliards de dollars (américains) sont volés, chaque année, par voie informatique [3].

La sécurité est un concept général englobant, entre autres, la cryptographie et l'authentification [2] d'une personne appellée *utilisateur* du système. Actuellement, la plupart des systèmes automatisés d'authentification requièrent un mot de passe ou un numéro d'identification personnelle (NIP). Le déploiement de plusieurs systèmes de ce type a incité les clients à trouver des moyens mémotechniques allant souvent à l'encontre de leur propre sécurité. Sous cet angle, les systèmes actuels peuvent donc être vus comme victimes de leur propre succès.

Parallèlement, on voit poindre l'aboutissement concret de décennies de recherche dans le domaine des biométriques, c.-à-d. l'authentification par la mesure de paramètres biologiques, entre autres la voix.

La reconnaissance d'une personne par le biais des traits caractéristiques de sa voix, aussi appellée reconnaissance du locuteur, a largement évolué, depuis une recherche confinée aux laboratoires vers des applications concrètes. Plusieurs entreprises, dont

AT&T et TI (avec Sprint) ont implanté des applications, "sur le terrain", de la technologie de la reconnaissance du locuteur.

Il reste toutefois plusieurs problèmes ouverts dont la résolution est nécessaire au développement à grande échelle de telles applications.

Chapitre 1

Description des systèmes de reconnaissance du locuteur

Les systèmes de reconnaissance du locuteur peuvent être implantés pour accomplir diverses tâches qui sont décrites dans ce chapitre. Celle qui sera étudiée dans ce chapitre est identifiée en conclusion.

1.1 Schéma d'un système de vérification du locuteur

La figure 1.1 présente un schéma de vérification du locuteur. Suite à une procédure d'identification personnelle, le locuteur prononce les mots qui lui permettront d'accéder à ses ressources. L'identité présumée du locuteur indique au système lequel des modèles de locuteurs, parmi ceux du registre, doit être confronté à la parole. Du signal de parole sont extraits les paramètres d'intérêt qui serviront, avec le modèle du locuteur visé appelé *désiré*, à établir un niveau de similitude entre la parole de l'utilisateur

actuellement en présence du système et le modèle du locuteur désiré qu'il prétend être. Si la similitude est suffisamment grande, i.e., supérieure à un certain seuil, alors l'utilisateur est accepté et l'accès aux ressources lui est fourni. Sinon, il se voit refuser cet accès.

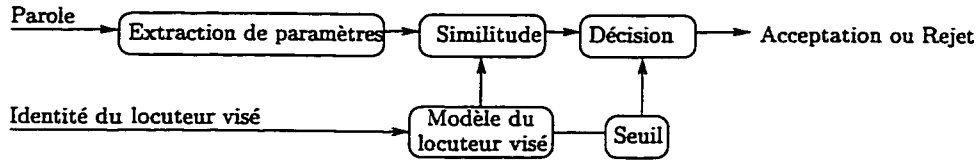

Figure 1.1: Schéma d'un système de vérification du locuteur

1.2 Reconnaissance, identification et vérification

Pour ce qui est de la vérification automatique du locuteur, l'utilisateur doit initialement s'identifier. Le système établit alors un lien mathématique entre les données du fichier test et les données du fichier d'entraînement. Cette tâche n'implique donc qu'un seul calcul de distance.

La tâche d'identification est généralement considérée comme étant plus complexe puisque le locuteur ne s'identifie pas au système. On doit donc choisir, parmi les membres du *registre*, lequel est le plus susceptible d'avoir prononcé les mots enregistrés. Cette tâche implique donc un calcul pour chaque membre du registre.

En d'autres termes, l'identification tente de répondre à la question, "Qui es-tu ?"; la vérification tente de répondre à la question "Es-tu la personne que tu prétends

être" [3].

Le système de reconnaissance, tant pour l'identification que pour la vérification, peut commettre deux types d'erreurs: des *faux rejets* et des *fausses acceptations*. Dans le domaine de la vérification du locuteur, un faux rejet se produit lorsque le système rejette un utilisateur qui tente d'accéder à ses propres ressources. Dans le domaine de l'identification, un faux rejet se produit lorsque le système refuse l'accès à un utilisateur faisant partie des membres du *registre*. En vérification, une fausse acceptation se produit lorsque le système accepte un imposteur qui tente d'accéder aux ressources d'une autre personne. En identification, une fausse acceptation correspond à l'acceptation par le système d'une personne n'étant pas membre du registre.

La décision d'accepter ou de rejeter dépend de ce qui est appelé le *seuil*. Le seuil correspond à la valeur numérique maximale (ou minimale) de la mesure de distance (ou de proximité) calculée entre le fichier test et le fichier d'entraînement.

1.3 Les méthodes à vocabulaire fermé, vocabulaire ouvert et vocabulaire dicté

Les méthodes à vocabulaire fermé exigent que l'utilisateur prononce une série de mots clés qui sont les mêmes que ceux utilisés lors de la phase d'entraînement. Ces méthodes sont généralement basées sur les *modèles cachés de Markov* ou bien sur des techniques d'anamorphose temporel ("dynamic time warping") ou de recherche dans une table d'exemples ("template matching") [13]. Puisque les méthodes à vo-

cabulaire fermé peuvent exploiter la répétition des mêmes phonèmes, elles obtiennent généralement des taux de reconnaissance supérieurs par rapport aux méthodes à vocabulaire ouvert.

Toutefois, certaines applications, telles la surveillance électronique, ne peuvent compter sur la répétition d'un texte prédéterminé. Ces applications doivent donc utiliser des méthodes à vocabulaire ouvert. Un autre avantage de ces méthodes est le fait que le locuteur n'ait pas à retenir un mot de passe ou une série de mots clés.

Ces deux types de méthodes souffrent malheureusement d'une sérieuse faiblesse: un imposteur pourrait, à l'aide d'un bon appareil électronique, faire jouer un enregistrement de la voix d'un locuteur du regitre et accéder aux ressources de ce locuteur.

Afin de contourner ce problème, des méthodes à vocabulaire dicté ont été proposées: l'utilisateur doit, chaque fois, prononcer une nouvelle phrase choisie au hasard par le système parmi un vocabulaire à toutes fins pratiques illimité. Un système de reconnaissance de la parole vérifie initialement que l'utilisateur a correctement prononcé la phrase qui lui a été demandée. Ensuite, le système de reconnaissance du locuteur tente de reconnaître l'utilisateur. Si la quantité de données d'entraînement fournies au système est suffisamment importante, chaque phonème des locuteurs du registre pourrait être modélisé.

1.4 Conclusion

Un schéma illustrant un système de vérification du locuteur a été présenté. Les termes reconnaissance, identification et vérification ont été définis tels qu'ils sont utilisés dans le cadre des systèmes biométriques. Enfin, le texte prononcé par l'utilisateur du système de vérification du locuteur est source d'une autre segmentation des tâches entre vocabulaire fermé, ouvert ou dicté. Les expériences décrites dans les prochains chapitres s'attardent à la vérification du locuteur à vocabulaire ouvert.

Chapitre 2

Les coefficients cepstraux

Dans le domaine de la reconnaissance de formes, le prétraitement du signal pour l'obtention de paramètres qui le caractérisent bien est essentiel. Bien que cet ouvrage ne soit pas voué à la recherche de ces paramètres, il est important de bien comprendre les étapes qui permettent le calcul des coefficients cepstraux, aussi appelés $MFCC$ (de l'anglais "Mel-Frequency cepstral coefficients"), utilisés ici pour la reconnaissance du locuteur. Cette compréhension servira à saisir les hypothèses qui sont sous-jacentes aux $MFCC$ dans l'espoir d'en faire une utilisation plus éclairée.

Lorsque l'on tente de bien caractériser un signal, une application particulière est généralement envisagée. Or, les paramètres développés en fonction de cette application peuvent s'avérer totalement inappropriés lorqu'utilisés pour une application différente, même si le signal traité reste le même.

Davis et Mermelstein [10] avait initialement proposé les paramètres $MFCC$ puisqu'ils permettaient de bien caractériser les phonèmes, indépendamment du locuteur. L'u-

tilisation de ces mêmes coefficients pour la reconnaissance du locuteur semble donc, a priori, paradoxale. Pourtant, les plus récents résultats de recherche montrent qu'ils restent, pour l'instant, parmi les meilleurs coefficients à utiliser.

Le développement des $MFCC$ (voir Figure 2.1) est obtenu à la suite d'une série d'étapes dont les plus importantes sont l'échantillonnage (section 2.1), la préemphase (section 2.2), le fenêtrage (section 2.3), la transformée rapide de Fourier (section 2.4) le calcul des amalgames (section 2.5) et la convolution cosinusoïdale (section 2.6). Ces six étapes ne diffèrent d'une implantation à l'autre que par l'ajustement de certains paramètres. Elles font donc chacune l'objet d'une section de ce chapitre. D'autres ajustements qui peuvent faciliter le travail subséquent de reconnaissance sont parfois utilisés et les plus importants sont brièvement décrits à la section 2.7. Parmi les $MFCC$, certains coefficients ont un pouvoir discriminant supérieur et il est intéressant de constater que ceux qui apportent le plus d'information discriminante en reconnaissance du locuteur ne sont pas les mêmes qu'en reconnaissance de la parole (section 2.8). Enfin, d'autres paramètres sont utilisés en reconnaissance du locuteur. Une justification de l'utilisation des $MFCC$ est donc présentée à la section 2.9, en guise de conclusion de ce chapitre.

2.1 Échantillonnage

L'échantillonnage est l'opération maîtresse pour la conversion de signaux continus en signaux discrets (conversion analogique-numérique). On peut modéliser l'échantillonnage comme la multiplication du signal de base $x(t)$ par un signal $p(t)$ constitué d'une série

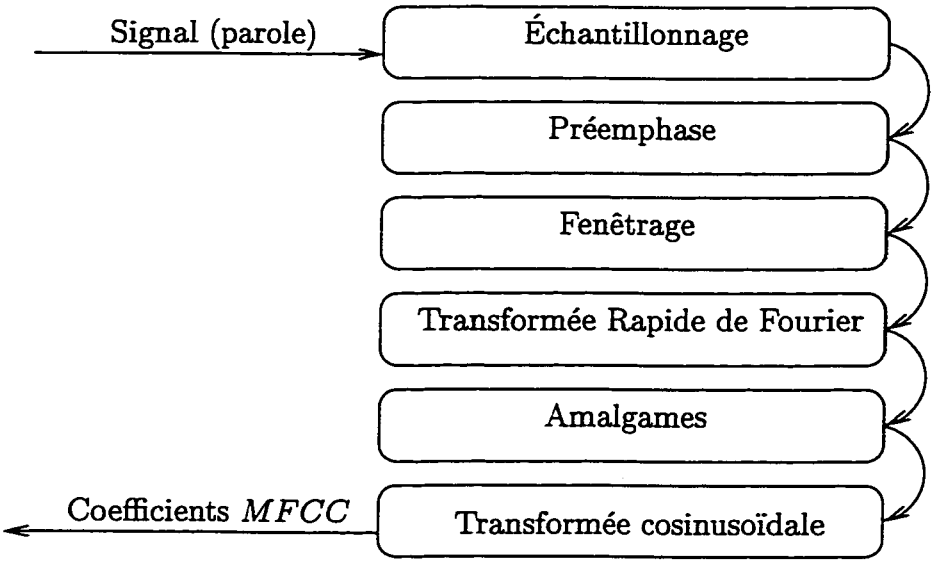

Figure 2.1: Développement des coefficients $MFCC$

d'impulsions unitaires [19] uniformément espacées sur la durée du signal. D'une impulsion à l'autre s'écoule un temps T appelé période.

Considérons tout d'abord une seule impulsion. Soit $\delta(t)$, la fonction d'impulsion unitaire, aussi appelée le delta de Dirac définie comme suit:

$$\delta(t) = \begin{cases} 1 & \text{si } t = 0 \\ 0 & \text{sinon} \end{cases} \tag{2.1}$$

En supposant que la durée du signal soit de $n \times T$, on définit $p(t)$, la série d'impulsions unitaires ainsi :

$$p(t) = \sum_{k=0}^{n} \delta(t - kT)$$

$$= \begin{cases} 1 & \text{si } t = 0, T, 2T, ..., nT \\ 0 & \text{sinon} \end{cases} \tag{2.2}$$

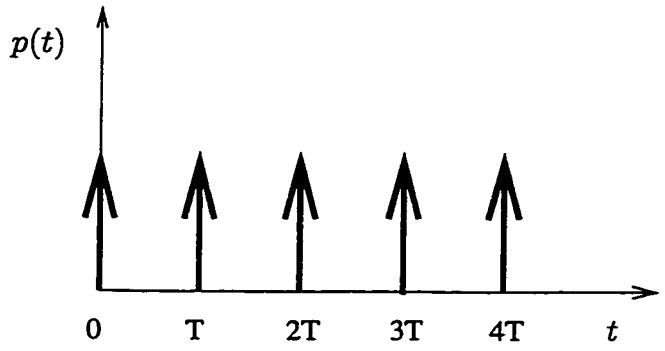

Figure 2.2: Série d'impulsions

Soit $y(t)$, le signal échantillonné, on a

$$y(t) = x(t) \cdot p(t)$$

Ainsi,

$$y(t) = \begin{cases} x(t) & \text{si } t = 0, T, 2T, ..., nT \\ 0 & \text{sinon} \end{cases} \tag{2.3}$$

Donc, $y(t)$, le signal échantillonné, correspond, quant à elle, dans le domaine temporel, au produit des signaux $x(t)$ et $p(t)$. La transformée de Fourier $\hat{y}(\omega)$ du signal $y(t)$ correspond à la convolution des transformées de Fourier $\hat{x}(\omega)$ et $\hat{p}(\omega)$ (des deux signaux $x(t)$ et $p(t)$) dans le domaine fréquentiel).

Démonstration:

$$
\begin{aligned}
\hat{y}(\omega) &= \int_{-\infty}^{\infty} y(t) \cdot e^{-j\omega t} dt \\
&= \int_{-\infty}^{\infty} x(t) \cdot p(t) \cdot e^{-j\omega t} dt \\
&= \int_{-\infty}^{\infty} x(t) \cdot \left[\frac{1}{2\pi} \cdot \int_{-\infty}^{\infty} \hat{p}(\xi) \cdot e^{j\xi t} d\xi \right] \cdot e^{-j\omega t} dt \\
&= \frac{1}{2\pi} \cdot \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(t) \cdot \hat{p}(\xi) \cdot e^{-j(\omega-\xi)t} d\xi dt \\
&= \frac{1}{2\pi} \cdot \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(t) \cdot \hat{p}(\xi) \cdot e^{-j(\omega-\xi)t} dt d\xi \\
&= \frac{1}{2\pi} \cdot \int_{-\infty}^{\infty} \hat{p}(\xi) \left[\int_{-\infty}^{\infty} x(t) \cdot e^{-j(\omega-\xi)t} dt \right] d\xi \\
&= \frac{1}{2\pi} \cdot \int_{-\infty}^{\infty} \hat{p}(\xi) \cdot \hat{x}(\omega - \xi) d\xi \\
&= \frac{1}{2\pi} \cdot \hat{p}(\omega) * \hat{x}(\omega)
\end{aligned}
$$

La fonction $p(t)$ étant périodique, $\hat{p}(\omega)$ doit être trouvée à l'aide de la transformée en séries de Fourier. La période de $p(t)$ étant égale à T, chaque coefficient c_k correspond au taux de corrélation entre le signal $x(t)$ et un phaseur de fréquence $\omega_k = \frac{2\pi k}{T}$, où $k \in \mathbb{Z}$.

De façon arbitraire, choisissons l'intervalle $[0, T)$.

$$
\begin{aligned}
c_k &= \frac{1}{T} \cdot \int_0^T p(t) \cdot e^{-jk\omega_0 t} dt \\
&= \frac{1}{T} \cdot \int_0^T \left(\sum_{m=0}^{n} \delta(t - mT) \right) \cdot e^{-jk\omega_0 t} dt \\
&= \frac{1}{T} \cdot \sum_{m=0}^{n} \int_0^T \delta(t - mT) \cdot e^{-jk\omega_0 t} dt \\
&= \frac{1}{T} \cdot \int_0^T \delta(t) \cdot e^{-jk\omega_0 t} dt
\end{aligned}
$$

$$= \frac{1}{T} \cdot e^{-jk\omega_0 0} dt$$
$$= \frac{1}{T}$$

Donc, une série uniforme d'impulsions dans le domaine temporel correspond à une série uniforme d'impulsions dans le domaine fréquentiel.

Le lecteur aura sans doute remarqué que la série de Fourier associée à une série d'impulsions est indépendante du nombre d'impulsions et même du statut fini ou infini de ce nombre.

Les coefficients c_k sont évalués sur une seule période du signal. La question à laquelle la tranformée en séries de Fourier ne répond pas est à quel moment sont présentes les fréquences $k\omega_0$. Dans notre cas, elles le sont aux temps $0, T, 2T, ..., nT$. Dans le cas d'une série infinie d'impulsions, elles le seraient aux temps $kT \; \forall k \in \mathbb{Z}$

Les conversations téléphoniques sont transmises à l'aide de canaux dont la bande passante va de $100Hz$ à $3500Hz$. Une fréquence, lorsque considérée comme phénomène physique, ne peut être que positive . Toutefois, les manipulations mathématiques de nombres complexes associées aux transformées de Fourier nécessitent la considération de fréquences négatives.

Par exemple, la simple fonction $x(t) = cos(\omega_0 t)$ engendre les coefficients de Fourier suivants:

$$c_k = \begin{cases} 1/2 & k = \pm 1 \\ 0 & \text{ailleurs} \end{cases}$$

On en déduit que $x(t)$ est composé d'une superposition de signaux émis à des fréquences de $-\omega_0$ et ω_0. Ainsi, la transformée de Fourier d'un signal téléphonique peut s'étendre sur une bande de $-3500Hz$ à $3500Hz$, à l'exception des fréquences

entre $-300Hz$ et $300Hz$.

$\hat{y}(\omega)$ consiste donc en une série de répliques de $\hat{x}(\omega)$, chacune de ces répliques étant déphasée de ω_0 par rapport à la précédente. Ainsi, en filtrant $\hat{y}(\omega)$ sur une bande passante de $(-\omega_b, \omega_b)$, on obtient $\hat{y'}(\omega) = \hat{x}(\omega)$. Donc, la transformée inverse de $\hat{y'}(\omega)$, $y'(t)$ nous redonne le signal initial $x(t)$. Il faut toutefois s'assurer que les répliques de $\hat{x}(\omega)$ sont disjointes afin d'éviter toute superposition qui impliquerait une perte irréversible d'information. Cette condition est respectée si et seulement si:

$$\begin{aligned} \omega_0 - \omega_b &> \omega_b \\ \omega_0 &> 2 \cdot \omega_b \end{aligned}$$

La fréquence d'échantillonnage doit donc être au moins deux fois supérieure à la fréquence maximale du signal d'entrée. C'est le théorème d'échantillonnage ou le théorème de Nyquist.

Dans le cas particulier du signal de parole, ce théorème implique une fréquence d'échantillonnage supérieure à $7000Hz$. Généralement, les études sont réalisées à l'aide de données échantillonnées à $8000Hz$. C'est le cas de celles qui seront présentées dans cet ouvrage.

2.2 Préemphase

Pour les sons voisés, l'intensité du signal de parole décroît en fonction de la fréquence. Cette décroissance est d'environ 6 décibels par octave. En termes absolus, certaines informations pertinentes de hautes fréquences sont donc noyées par les basses fréquences qui occupent une place prépondérante dans l'amplitude du signal. Afin

de donner à ces fréquences l'importance qui leur est due et puisque les sont voisés sont plus fréquents que les sons non-voisés, on utilise, dans le domaine temporel, une transformation ("autoregressive") $AR(1)$ du signal. Cette transformation permet de redresser le spectre du signal et ainsi de détecter certaines variations aux hautes fréquences. En termes relatifs, les hautes fréquences sont amplifiées par rapport aux basses fréquences. Toutefois, les relations relatives entre fréquences voisines restent à toutes fins pratiques intactes. Cette transformation permet aussi d'obtenir une quantification plus efficace [33].

$$\tilde{x}[n] \;=\; x[n] - a \cdot x[n-1] \tag{2.4}$$

Ceci correspond à la multiplication de la transformée en z du signal par un filtre "FIR" de premier ordre:

$$H(z) \;=\; 1 - a \cdot z^{-1} \tag{2.5}$$

où, généralement, $0,9 \leq a \leq 1,0$. Les expériences décrites au chapitre 6 utilisent une valeur de $a = 0,97$.

2.3 Fenêtrage

L'échantillonnage effectué à $8kHz$ a permis d'évaluer le signal de parole en un nombre fini de points sans affecter la quantité des fréquences transmises par le téléphone. Cette propriété ne s'applique toutefois que pour le cas théorique d'un signal de longueur infinie. De plus, l'information fréquentielle fournie représente une quantité

"moyenne" sur la durée totale du signal. Or, la parole étant hautement dynamique, on doit tenter d'extraire les paramètres fréquentiels sur une période beaucoup plus courte pendant laquelle il est raisonnable de supposer un signal stable dans le domaine fréquenciel. Ceci est rendu possible grâce à la relative lenteur de mouvement du conduit vocal. Cette période plus courte est appelée *trame*.

Le terme *fenêtrage* réfère à la multiplication d'un signal $x[n]$ par une séquence $w[n]$ de durée finie, c.-à-d. :

$$w[n] \begin{cases} = 0 & \text{pour} \quad n < 0 \\ \neq 0 & \text{pour} \quad 0 \leq n < N \\ = 0 & \text{pour} \quad n \geq N \end{cases}$$

Une fois qu'une portion du signal a été *fenêtrée*, on en extrait les fréquences en utilisant la transformée discrète de Fourier dont le calcul peut être simplifié à l'aide de l'algorithme de la transformée rapide de Fourier (FFT). Cet algorithme sera traité en détail dans la section 2.4.

Une fois que les paramètres ont été extraits sur la portion d'intérêt du signal, on fait "glisser" la fenêtre pour traiter une portion ultérieure et l'on répète le processus d'extraction de paramètres. Ce procédé est communément appelé "Transformée de Fourier à fenêtre glissante".

Dans le domaine de la reconnaissance de la parole, et en particulier dans le cas des expériences présentées au chapitre 6, la fenêtre s'étend sur une durée de 25ms et le glissement est de 10ms. Donc, à chaque "trame" de 10ms est associé un ensemble

de paramètres caractéristiques de la parole.

Tel que Fang [11] le mentionne, le choix de la durée de la fenêtre est dictée par deux objectifs qui se contre-balancent: La fenêtre doit être suffisamment courte afin que les paramètres sous-jacents puissent être considérés constants sur l'intervalle observé. Elle doit par contre être d'une longueur suffisante afin de fournir l'information suffisante à l'extraction fiable des paramètres évalués.

Plusieurs modèles de fenêtre ont été proposés. Le tableau 2.1 présente les valeurs de quelques paramètres qui servent à l'évaluation de quatre types de fenêtres et la figure 2.3 permet de visualiser ces paramètres pour la fenêtre de Hamming. L'avantage principal de la fenêtre de Hamming est la petitesse du lobe secondaire relativement au lobe principal. Cette fenêtre possède donc une bonne résolution dans le domaine fréquentiel. Cette fenêtre est utilisée dans le domaine de la reconnaissance de la parole et du locuteur, principalement pour cette raison.

Tableau 2.1: Paramètres pour l'évaluation de certaines fenêtres

Fenêtre	Lobe secondaire	Décroissance	Lobe principal
Rectangulaire	-13dB	-6dB/oct	$4\pi/N$
Bartlett	-25dB	-12dB/oct	$8\pi/N$
Hanning	-31dB	-18dB/oct	$8\pi/N$
Hamming	-41dB	-6dB/oct	$8\pi/N$

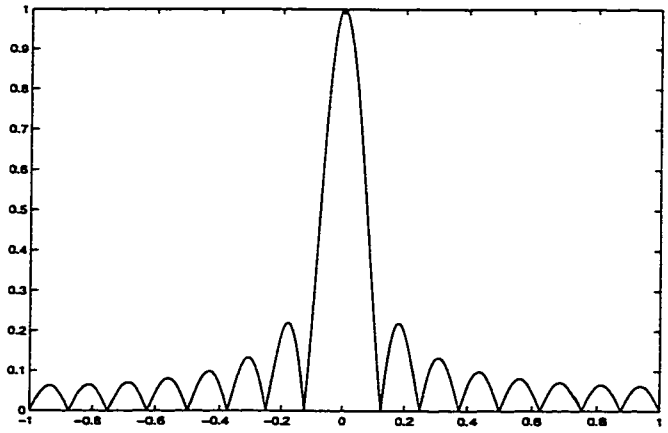

Figure 2.3: FFT de la fenêtre de Hamming avec $N = 32$ (tiré de Jackson [19])

L'équation de la fenêtre de Hamming est la suivante:

$$w_H[n] = 0.54 - 0.46 \cdot cos\left(\frac{2\pi n}{N-1}\right) \quad 0 \leq n \leq N-1 \tag{2.6}$$

2.4 Transformée rapide de Fourier

Après avoir fenêtré une portion du signal échantillonné, la transformée discrète de Fourier est utilisée afin de permettre l'étude de ce signal dans le domaine fréquentiel.

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi kn/N} \quad k = 0, \ldots, N-1 \tag{2.7}$$

L'équation 2.7 requiert un nombre de multiplications de l'ordre de $O(N^2)$. L'algorithme de la transformée rapide de Fourier permet d'obtenir les valeurs désirées avec un nombre de multiplications beaucoup inférieur (ordre $O(N \log N)$) en utilisant la programmation dynamique. La figure 2.4 illustre cet algorithme.

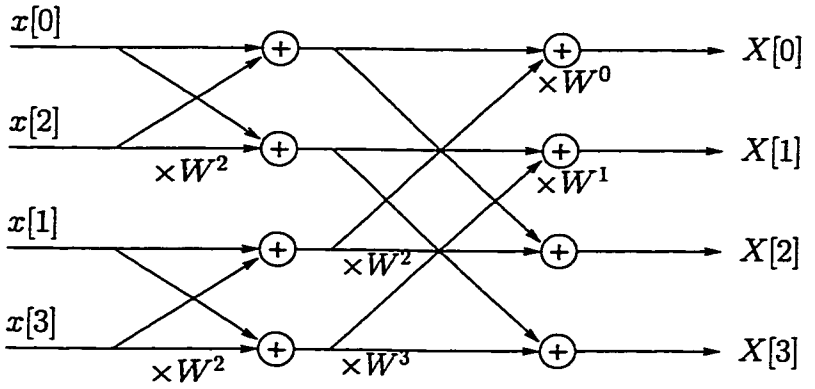

Figure 2.4: Algorithme de la transformée rapide de Fourier en 4 points illustré par le biais d'un treillis.

2.5 Les amalgames

Afin de réduire la quantité d'information fréquentielle générée par la transformée rapide de Fourier, Davis et Mermelstein [10] ont proposé de regrouper les fréquences en 20 *amalgames*. Les filtres (figure 2.5) sont triangulaires et répartis selon l'échelle de Mel, i.e., de façon linéaire entre 100 et $1000Hz$, puis logarithmique par la suite, jusqu'aux environs de $4500Hz$. Cette répartition des filtres correspond à la perception auditive humaine qui est moins précise pour les hautes fréquences.

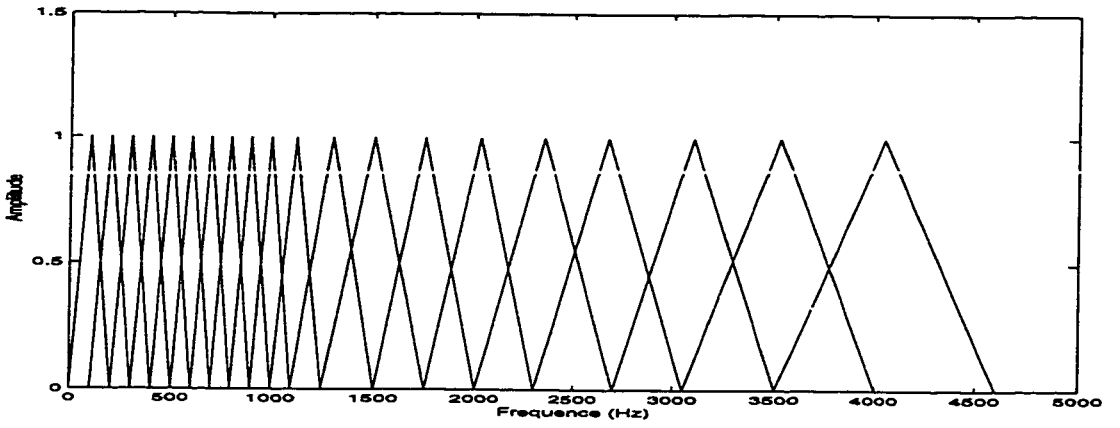

Figure 2.5: Filtres de Mermelstein selon l'échelle de Mel

L'équation 2.8 donne la correspondance entre l'échelle fréquentielle et l'échelle de Mel telle qu'établie dans le logiciel HTK [51]:

$$Mel(f) = 2595 \cdot \log_{10}(1 + f/700) \tag{2.8}$$

2.6 Transformée cosinusoïdale

Le *cepstrum* est la transformée de Fourier inverse du logarithme du *spectrum*. Le cepstrum peut être calculé en utilisant une transformée cosinusoïdale des amalgames [37].

$$c_i = \sum_{j=1}^{P} m_j \cos\left(\frac{\pi i}{P}(j - 0.5)\right) \quad i = 1, \ldots, N \tag{2.9}$$

Où m_j est le $j^{ième}$ amalgame et P est le *degré* d'analyse, i.e., le nombre d'amalgames qui ont été calculés. Généralement, N, le nombre de coefficients statiques calculés est compris entre 7 et 12. Les expériences décrites dans ce document ont été réalisées avec un ordre d'analyse de $P = 20$ et $N = 12$ coefficients statiques.

2.7 Autres ajustements apportés aux $MFCC$

En plus des $MFCC$, appelés coefficients statiques, obtenus à la suite des opérations décrites ci-haut, on calcule aussi leur variation d'une trame à l'autre, i.e., dans le temps. Ces coefficents dynamiques sont appellés *delta* $- MFCC$ ($\Delta MFCC$). De la même façon, certains chercheurs utilisent aussi les coefficients d'accélération

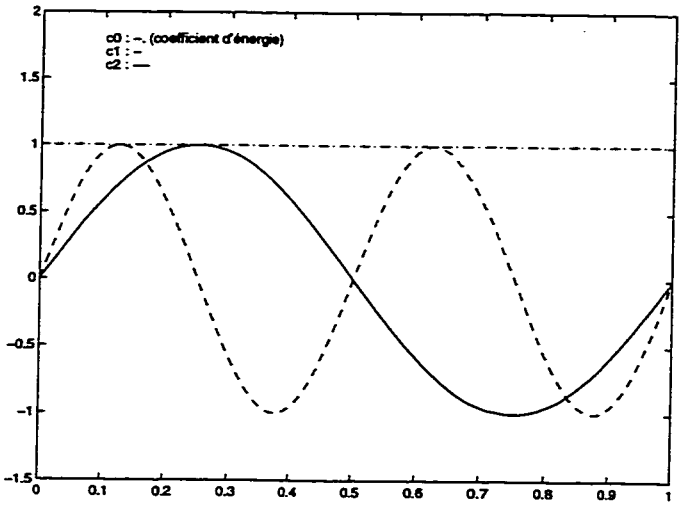

Figure 2.6: Trois fonctions utilisées successivement pour la transformée avec les amalgames: $\cos\left(\frac{\pi i}{P}(j-0.5)\right)$ pour $i = 0, 1, 2$.

$delta - delta - MFCC$ ($\Delta^2 MFCC$). Ces paramètres n'apportent toutefois pas tous la même quantité d'information discriminante quant à l'identité du locuteur.

2.8 Le pouvoir discriminant des $MFCC$

Il a été mentionné plus haut que l'utilisation des mêmes paramètres $MFCC$ à la fois pour la reconnaissance de la parole et la reconnaissance de locuteur semblait paradoxale. Les deux applications se distinguent toutefois par le pouvoir discriminant qu'elles retirent de ces paramètres. Alors que le coefficient d'énergie permet généralement l'identification d'une voyelle et possède un fort pouvoir discriminant en reconnaissance de la parole, certains chercheurs l'ignorent complètement pour la

reconnaissance du locuteur puisque la valeur du paramètre d'énergie est beaucoup plus influencée par le phonème prononcé et le canal de transmission de la voix que par le locuteur lui-même.

Charlet et Jouvet [6] ont exploré de façon relativement exhaustive le pouvoir discriminant des coefficients $MFCC$. Utilisant les coefficients statiques, dynamiques et d'accélération, de 9 transformées cosinusoïdales (incluant l'énergie), ils sont parvenus à montrer assez clairement la supériorité des coefficients de haut degré (i.e., c_6, c_7 et c_8) et des coefficients dynamiques (vs statiques et d'accélération). Enfin, la conclusion la plus sévère fut à l'endroit des trois coefficients d'énergie qui se classèrent parmi les cinq derniers quant au pouvoir discriminant, pour les raisons expliquées ci-haut.

2.9 Conclusion

Ce chapitre a présenté les étapes nécessaires à l'obtention des coefficients cepstraux, aussi appelés $MFCC$. Ces étapes sont l'échantillonnage, la préemphase, le fenêtrage, la transformée rapide de Fourier, le calcul des amalgames et la transformée cosinusoïdale. Certains autres ajustements peuvent être apportés et ont été décrits. Enfin, le pouvoir discriminant des différents paramètres a été brièvement analysé.

Chapitre 3

Les Modèles cachés de Markov

3.1 Introduction

La théorie des modèles cachés de Markov (ou "HMM" de l'anglais "hidden Markov models") fut publiée par Baum et ses collègues vers 1970 et par la suite implantée par Baker à CMU et Jelinek chez IBM. Rabiner et Juang [36] identifient clairement l'hypothèse fondamentale sur laquelle sont fondés les modèles cachés de Markov: le signal de parole se caractérise bien par un processus stochastique dont les paramètres peuvent être estimés de façon précise. Puisque le signal de parole est échantillonné avant d'être traité, nous aborderons simplement les modèles de Markov en temps discret.

Supposons qu'un processus se trouve dans un état quelconque parmi un certain nombre d'états possibles. Supposons, de plus, que la sortie (que nous appellerons observation) produite par ce processus dépende de l'état dans lequel il se trouve. Supposons, enfin, que le processus puisse évoluer de son état actuel vers un autre

état avec une certaine probabilité, ce qui aura donc une influence sur l'observation générée. Si les probabilités de transition vers un nouvel état ne dépendent que de l'état actuel, on peut qualifier ce processus de Markovien.

Ce chapitre décrit brièvement les chaînes (section 3.2) et les modèles cachés (section 3.3) de Markov. Par la suite, les solutions algorithmiques à deux problèmes importants des modèles cachés de Markov sont décrites: les procédures prospective et rétrospective pour la probabilité conditionnelle d'une séquence d'observations (section 3.4) et l'algorithme de Viterbi pour la séquence optimale d'états cachés (section 3.5).

3.2 Les chaînes de Markov

Une chaîne de Markov est formée d'un certain nombre N d'états. Ces états sont interconnectés par des probabilités de transitions pouvant être représentées sous forme matricielle:

$$\mathcal{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \\ \vdots & & \ddots & \\ a_{n1} & & & a_{nn} \end{pmatrix} \tag{3.1}$$

avec les propriétés suivantes:

$$a_{ij} \geq 0 \qquad \forall i, j \tag{3.2}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad \forall i \tag{3.3}$$

Autrement dit, un processus se trouvant dans l'état i au temps t a une probabilité a_{ij} d'évoluer vers l'état j au temps $t+1$. Ainsi, en supposant que l'état réel dans lequel le processus se trouve au temps t est q_t,

$$a_{ij} = P(q_{t+1} = j | q_t = i) \tag{3.4}$$

À chaque état correspond une observation. La figure 3.1 représente une chaîne de Markov à trois états pour laquelle l'observation $\mathbf{O}_1$ est associée à l'état 1, l'observation $\mathbf{O}_2$ à l'état 2 et l'observation $\mathbf{O}_3$ à l'état 3.

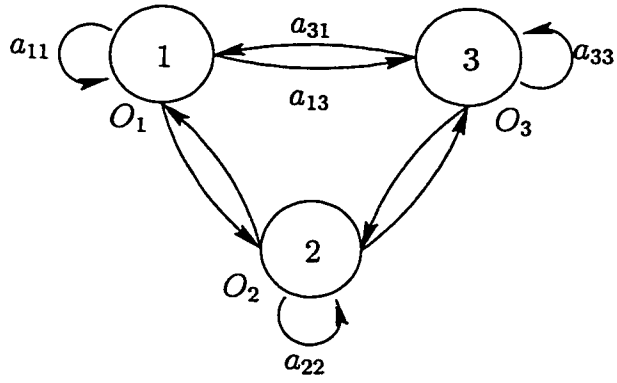

Figure 3.1: Chaîne de Markov avec $N = 3$

3.3 Les modèles cachés de Markov

Alors qu'une chaîne de Markov associe une seule observation possible à chaque état, un modèle de Markov associe une distribution de probabilité de l'observation générée par le processus à chaque état. C'est la distinction fondamentale entre une chaîne et un modèle de Markov. Généralement, les distributions associées aux différents états se recoupent. Donc, étant donné une observation, il n'est plus possible d'identifier avec certitude l'état dans lequel se trouve le processus. On peut simplement, à l'aide de la loi de Bayes pour les probabilités conditionnelles, identifier l'état le plus probable d'être à l'origine de l'observation en question.

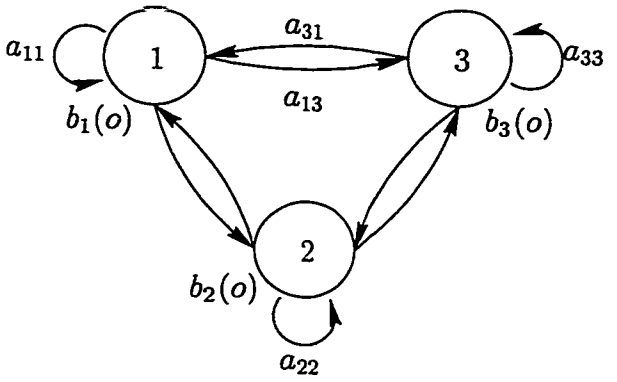

Figure 3.2: Modèle de Markov avec $N = 3$

La figure 3.2 présente un modèle de Markov à trois états. Nous utiliserons la notation compacte suivante pour symboliser la distribution de probabilité de l'observation o_t au temps t conditionnelle à la présence du processus dans l'état j à ce moment:

$$b_j(o_t) = P(o_t|q_t = j) \qquad 1 \leq j \leq N \tag{3.5}$$

$$b_j(o_t) \quad = \quad f(o_t|q_t = j) \qquad 1 \leq j \leq N \tag{3.6}$$

Les équations 3.5 et 3.6 s'appliquant dans le cas d'une distribution de probabilité discrète ou continue respectivement.

Rabiner et Juang identifient les composantes essentielles d'un modèle HMM:

1. N, le nombre d'états du modèle. Ce nombre est généralement inconnu puisque le modèle est caché. Il doit donc être estimé. Dans le cas de la reconnaissance de locuteur, les résultats expérimentaux suggèrent que la modélisation avec un seul état émettant est suffisante, en particulier dans le cas de reconnaissance du locuteur avec un texte ouvert. Pour la reconnaissance de la parole, on utilise généralement 5 états pour la modélisation d'un phonème: un état non-émettant d'entrée, trois états émettants et un cinquième état non-émettant de sortie. On note aussi q_t, l'état dans lequel se trouve réellement le processus au temps t.

2. $\mathcal{A}$, la matrice des probabilités de transition entre les états.

3. $\mathcal{B} = \{b_j(o)\} \qquad 1 \leq j \leq N$

4. $\Pi = \{\pi_i\}$ l'ensemble des probabiltés initiales

Dans un but de simplification, nous utiliserons la notation plus compacte suivante: $\lambda = (\mathcal{A}, \mathcal{B}, \Pi)$.

3.4 Les procédures prospective et rétrospective pour la probabilité conditionnelle d'une séquence d'observations

Nous pouvons poser un premier problème relié aux modèles cachés de Markov: Étant donné $\mathbf{O} = \{o_t\}$, $1 \leq t \leq T$, une série d'observations, quelle est la probabilité que cette série ait été générée par le modèle. $\lambda = (\mathcal{A}, \mathcal{B}, \Pi)$?

La résolution de ce problème nous permettrait d'établir, parmi une série de modèles HMM possibles, lequel serait le plus probable d'être à la source de cette série d'observations.

Par exemple, dans le cas de l'identification de locuteur (sans rejet), on voudra, à partir d'un fichier audio et une banque de modèles de locuteurs, trouver la personne la plus probable d'avoir généré ce fichier.

$$\begin{aligned}
\hat{r} &= \arg\max_{r \in \mathcal{R}}[P(r \equiv c)] \\
&\approx \arg\max_{r \in \mathcal{R}}[P(\mathbf{O}_r|\mathbf{O}_c)] \\
&\approx \arg\max_{r \in \mathcal{R}}[P(\lambda_r|\mathbf{O}_c)] \\
&= \arg\max_{r \in \mathcal{R}}[P(\mathbf{O}_c|\lambda_r) \cdot \frac{P(\lambda_r)}{P(\mathbf{O}_c)}] \\
&= \arg\max_{r \in \mathcal{R}}[P(\mathbf{O}_c|\lambda_r) \cdot P(\lambda_r)] \\
&= \arg\max_{r \in \mathcal{R}}[P(\mathbf{O}_c|\lambda_r)]
\end{aligned}$$

Où c désigne le locuteur qui tente d'accéder au système de reconnaissance, appellé client et $\mathbf{O}_c$ désigne la séquence d'observations utilisée pour reconnaître le client. De plus, $\mathbf{O}_r$ désigne la séquence d'observations utilisée pour développer le modèle λ_r du locuteur r du registre $\mathcal{R}$.

Les deux premiers passages sont rarement mis en évidence, malgré le fait qu'ils constituent une hypothèse et non une certitude. Le troisième passage s'effectue grâce à la loi de Bayes. Le quatrième profite du fait que la valeur de $P(\mathbf{O}_c)$ ne dépend pas du locuteur du registre et n'a donc pas d'influence sur la sélection du meilleur modèle. Enfin, l'hypothèse de locuteurs équiprobables permet le dernier passage.

Pour ce qui est de la vérification de locuteur ou de l'identification avec rejet, le calcul de $P(\mathbf{O}_c)$ reste un problème complexe à la base des modèles de normalization et sera discutée au chapitre 5.

L'hypothèse de locuteurs équiprobables ne pourrait être relaxée qu'én présence d'un système d'identification ou de vérification implanté depuis quelques temps. L'expérience réalisée pourrait alors servir à adapter les valeurs de $P(\lambda_r)$. Il est toutefois, a priori, peu plausible de trouver l'existence d'une corrélation entre les paramètres $MFCC$ d'un locuteur et sa fréquence d'utilisation d'un système automatisé. L'hypothèse utilisée semble donc valable.

Enfin, le calcul de $P(\mathbf{O}_c|\lambda_r)$ fait l'objet de cette section. L'algorithme vorace et deux algorithmes de programmation dynamique, très similaires, sont présentés.

3.4.1 L'algorithme vorace

Soient:

$\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T)$, une séquence de T observations

$\mathbf{q} = (q_1, q_2, \ldots, q_T)$, une séquence de T états cachés et

$\lambda = (\mathcal{A}, \mathcal{B}, \Pi)$, le modèle HMM à l'origine de la séquence $\mathbf{q}$

En supposant que le modèle comporte N états, il y a donc N^T séquences $\mathbf{q}$ possibles.

Le but est de calculer $P(\mathbf{O}|\lambda)$, la probabilité conditionnelle d'obtenir la séquence d'observations $\mathbf{O}$ à partir du modèle HMM λ.

À cette fin, la séquence d'états successifs $\mathbf{q}$ est, dans un premier temps, fixée, comme si elle était connue. Ceci permet le développement de certains résultats. Dans un second temps, cette hypothèse est relaxée pour obtenir le résultat désiré.

Il faut aussi souligner que les observations successives sont considérées indépendantes. Cette hypothèse est nécessaire à l'obtention du résultat désiré et ne pourra être relaxée. Elle est malheureusement très forte: par exemple, dans le cas d'une voyelle dont le spectre fréquentiel est relativement stable, jusqu'à 7 observations de 10ms (7 trames) lui sont attribuées. Ces observations sont certainement fortement corrélées et l'hypothèse d'indépendance y est clairement non valide. Cette approximation n'empêche toutefois pas les "HMM" d'obtenir d'excellents résultats dans une vaste série d'applications, en particulier en reconnaissance de la parole et du locuteur.

Donc, par indépendance des observations, on obtient que:

$$\begin{aligned} P(\mathbf{O}|\mathbf{q},\lambda) &= \prod_{t=1}^{T} P(\mathbf{o}_t|q_t,\lambda) \\ &= b_{q_1}(\mathbf{o}_1)\cdot b_{q_2}(\mathbf{o}_2)\cdot \ldots \cdot b_{q_T}(\mathbf{o}_T) \end{aligned} \tag{3.7}$$

Maintenant,

$$P(\mathbf{q}|\lambda) = \pi_{q_1}\cdot a_{q_1q_2}\cdot \ldots \cdot a_{q_{T-1}q_T} \tag{3.8}$$

Et

$$P(\mathbf{O},\mathbf{q}|\lambda) = \pi_{q_1}\cdot b_{q_1}(\mathbf{o}_1)\cdot a_{q_1q_2}\cdot b_{q_2}(\mathbf{o}_2)\cdot \ldots \cdot a_{q_{T-1}q_T}\cdot b_{q_T}(\mathbf{o}_T) \tag{3.9}$$

Enfin,

$$P(\mathbf{O}|\lambda) = \sum_{\forall \mathbf{q}} \pi_{q_1}\cdot b_{q_1}(\mathbf{o}_1)\cdot a_{q_1q_2}\cdot b_{q_2}(\mathbf{o}_2)\cdot \ldots \cdot a_{q_{T-1}q_T}\cdot b_{q_T}(\mathbf{o}_T) \tag{3.10}$$

L'algorithme vorace doit donc calculer $P(\mathbf{O},\mathbf{q}|\lambda)$ pour N^T séquences d'états possibles et sommer ces probabilités afin d'obtenir $P(\mathbf{O}|\lambda)$.

L'algorithme vorace roule donc dans l'ordre $O(T\cdot N^T)$. On peut toutefois, par programmation dynamique, faire beaucoup mieux.

3.4.2 La procédure prospective

De façon intuitive, on peut illustrer la propagation du processus dans le HMM sous forme de treillis en deux dimensions: le nombre d'états du HMM, N et le nombre d'observations T. Le treillis est donc composé de $N \cdot T$ noeuds et chaque séquence d'observations a la propriété de visiter exactement T noeuds, i.e., un noeud par colonne.

Définissons la variable prospective:

$$\alpha_t(i) = P(\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_t, q_t = i|\lambda) \tag{3.11}$$

C.-à-d., la probabilité d'observer $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t$ et que la séquence visite le noeud i au temps t étant donné le modèle sous-jacent λ.

1) Initialisation

$$\alpha_1(i) = \pi_i \cdot b_i(\mathbf{o}_1), \quad 1 \leq i \leq N$$

2) Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\mathbf{o}_{t+1}) \quad \begin{array}{l} 1 \leq t \leq T-1, \\ 1 \leq j \leq N \end{array}$$

3) Conclusion

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_t(i)$$

3.4.3 La procédure rétrospective

Définissons la variable rétrospective:

$$\beta_t(i) = P(\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, \mathbf{o}_T | q_t = i, \lambda) \tag{3.12}$$

1) Initialisation

$$\beta_T(i) = 1, \quad 1 \leq i \leq N$$

2) Induction

$$\beta_t(j) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, \ldots, 1 \quad 1 \leq i \leq N$$

3) Conclusion

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \pi_i \cdot b_i(\mathbf{o}_1) \cdot \beta_1(i)$$

3.5 L'algorithme de Viterbi pour la séquence optimale d'états cachés

Étant donné un modèle caché de Markov à N états et une séquence de T observations, l'algorithme de Viterbi permet de trouver, parmi les N^T suites d'états cachés, celle qui est la plus probable d'avoir généré la séquence en question. L'algorithme suppose la connaissance, a priori, des paramètres du modèle.

L'algorithme vorace, qui calculerait la probabilité de chacune des séquences roule dans l'ordre $\mathbf{O}(N^T)$. L'algorithme de Viterbi, par l'utilisation de la programmation dynamique permet, dans la plupart des cas, une réduction importante du nombre de calculs puisqu'il roule dans l'ordre de $\mathbf{O}(N^2 \cdot T)$.

1) Prétraitement

Le prétraitement des probabilités pour travailler dans le domaine logarithmique permet de transformer une série de multiplications en additions et ainsi d'accélérer le calcul de la séquence optimale.

$$\begin{aligned}
\tilde{\pi}_i &= \log(\pi_i) && 1 \leq i \leq N \\
\tilde{b}_i(\mathbf{o}_t) &= \log(b_i(\mathbf{o}_t)) && 1 \leq i \leq N,\ 1 \leq t \leq T \\
\tilde{a}_{ij} &= \log(a_{ij}) && 1 \leq i, j \leq N
\end{aligned}$$

2) Initialisation

$$\begin{aligned}
\tilde{\delta}_1(i) &= \tilde{\pi}_i + \tilde{b}_i(\mathbf{o}_1) && 1 \leq i \leq N \\
\psi_1(i) &= 0 && 1 \leq i \leq N
\end{aligned}$$

3) Induction

$$\begin{aligned}
\tilde{\delta}_t(j) &= \max_{1 \leq i \leq N}[\tilde{\delta}_{t-1}(i) + \tilde{a}_{ij}] + \tilde{b}_j(\mathbf{o}_t) && 1 \leq j \leq N,\ 2 \leq t \leq T \\
\psi_t(j) &= \arg\max_{1 \leq i \leq N}[\tilde{\delta}_{t-1}(i) + \tilde{a}_{ij}] && 1 \leq j \leq N,\ 2 \leq t \leq T
\end{aligned}$$

4) Conclusion

$$\begin{aligned}
\tilde{P}^* &= \max_{1 \leq i \leq N}[\tilde{\delta}_T(i)] \\
q_T^* &= \arg\max_{1 \leq i \leq N}[\tilde{\delta}_T(i)]
\end{aligned}$$

5) Retour sur le chemin optimal

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \qquad t = T-1, T-2, \ldots, 1$$

Les valeurs $\tilde{\delta}_t(i)$ conservent la probabilité associée au meilleur (le plus probable) chemin menant à l'état i, au temps t depuis le temps initial. Les valeurs $\psi_t(i)$, quant à elles, conservent l'indice de l'état précédemment visité sur ce meilleur chemin. Ainsi, au bout de T itérations, ces dernières valeurs serviront à retrouver, rétrospectivement, la série d'états successifs du chemin optimal.

3.6 Conclusion

Dans ce chapitre, les chaînes et les modèles cachés de Markov ont été présentés. Les algorithmes de programmation dynamique des procédures prospectives et rétrospectives de même que l'algorithme de Viterbi ont été décrits.

Les modèles cachés de Markov connaissent présentement un grand succès dans le domaine de la reconnaissance automatique de la parole. Toutefois, dans le domaine de la reconnaissance du locuteur, les modèles actuellement utilisés découlent d'une simplification des HMM et sont présentés au chapitre suivant.

Chapitre 4

Les Modèles GMM

Un troisième problème, lié aux modèles cachés de Markov, n'a pas été analysé. Il s'agit de l'estimation des paramètres des gaussiennes du modèle. Cette analyse a été reportée au chapitre présent puisqu'elle se trouve simplifiée par l'utilisation des GMM.

La section 4.1 offre une brève description des modèles GMM. La section 4.2 présente l'algorithme itératif EM pour l'estimation des paramètres d'une mixture de gaussiennes. La section 4.3 porte sur l'adaptation d'un modèle développé *a priori* par les données observées du locuteur.

4.1 Interprétation

Les modèles *GMM* (de l'anglais "Gaussian Mixture Models") ont été initialement proposés par Reynolds [38] pour la reconnaissance du locuteur. Un modèle GMM constitue en fait un modèle HMM à un seul état émettant. Cette simplification

est généralement compensée par une augmentation du nombre de mixtures dans le modèle. Il a été montré de façon empirique que l'utilisation d'un seul état émettant ne réduisait pas le taux de reconnaissance tout en permettant une simplification de l'estimation des paramètres du modèle. L'interprétation liée à l'utilisation d'un seul état est simple: le locuteur représente l'état et les mixtures représentent les différents phonèmes, ou classes de phonèmes prononcés par cette personne. Cette affirmation est surtout vraie dans les cas où peu de données d'entraînement sont disponibles, ce qui se produit généralement pour la vérification du locuteur.

4.2 La méthode de Baum-Welch (l'algorithme EM) pour l'estimation des paramètres

Ce problème ne possède pas de solution analytique. La technique de Baum-Welch, aussi appelée algorithme EM ("expectation-maximisation") a été proposée afin de résoudre le problème d'estimation des paramètres de façon itérative. La méthode développée dans cette section permet d'évaluer les paramètres maximum de vraisemblance d'un modèle GMM.

Les probabilités de transition entre les états (la matrice $\mathcal{A}$) et les probabilités de départ (le vecteur Π) ne s'appliquent plus aux modèles GMM. La notation compacte λ sera donc adaptée en conséquence:

Soient:

w_i, la probabilité que l'observation o_t provienne de la $i^{ième}$ mixture,

μ_i, la moyenne vectorielle des observations provenant de la $i^{ième}$ mixture,

Σ_i, la matrice de covariance des observations provenant de la $i^{ième}$ mixture,

M, le nombre de mixtures et

m_t, la mixture cachée qui a généré l'observation o_t.

On note $\lambda = \{w_i, \mu_i, \Sigma_i\}$, $i = 1, 2, \ldots, M$. Ainsi, l'algorithme EM peut maintenant être présenté:

1. Initialisation: Choisir un modèle initial $\lambda = \lambda^{(0)}$.

2. Itération k: Réestimer les paramètres pour trouver un nouveau modèle $\lambda^{(k)}$ à l'aide du modèle λ selon les équations suivantes:

$$
\begin{aligned}
w_i^{(k)} &= \frac{1}{T}\sum_{t=1}^{T} p(m_t = i|o_t, \lambda) \\
\mu_i^{(k)} &= \frac{\sum_{t=1}^{T} p(m_t = i|o_t, \lambda) \cdot o_t}{\sum_{t=1}^{T} p(m_t = i|o_t, \lambda)} \\
\Sigma_i^{(k)} &= \frac{\sum_{t=1}^{T} p(m_t = i|o_t, \lambda) \cdot o_t o_t'}{\sum_{t=1}^{T} p(m_t = i|o_t, \lambda)} - \mu_i^{(k)} \mu_i^{(k)'}
\end{aligned}
$$

où

$$
\begin{aligned}
p(m_t = i|o_t, \lambda) &= \frac{p(m_t = i, o_t, \lambda)}{p(o_t, \lambda)} \\
&= \frac{p(m_t = i, o_t, \lambda)}{\sum_{j=1}^{M} p(m_t = j, o_t, \lambda)}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{p(m_t = i, \lambda) \cdot p(o_t | m_t = i, \lambda)}{\sum_{j=1}^{M} p(m_t = j, \lambda) \cdot p(o_t | m_t = j, \lambda)} \\
&= \frac{p(\lambda) \cdot p(m_t = i | \lambda) \cdot p(o_t | m_t = i, \lambda)}{\sum_{i=j}^{M} p(\lambda) \cdot p(m_t = j | \lambda) \cdot p(o_t | m_t = j, \lambda)} \\
&= \frac{w_i \cdot p(o_t | m_t = i, \lambda)}{\sum_{j=1}^{M} w_j \cdot p(o_t | m_t = j, \lambda)} \\
&= \frac{w_i \cdot b_i(o_t)}{\sum_{j=1}^{M} w_j \cdot b_j(o_t)}
\end{aligned}
$$

3. Si $\lambda^{(k)}$ est meilleur que λ, alors mettre le modèle à jour ($\lambda \leftarrow \lambda^{(k)}$) et itérer une $k+1^{ème}$ fois. Sinon, arrêter.

4.3 Les modèles adaptés

Le problème classique de la présence de données partielles pour le développement d'un modèle se retrouve aussi en reconnaissance de locuteur. Les fichiers tests peuvent parfois être de très courte durée et les paramètres estimés à l'aide de ces données peu fiables. Certains chercheurs se sont attardés au développement de modèles adaptés: dans un premier temps, un modèle global est développé à l'aide des données d'entraînement de tous les locuteurs. Par la suite, le modèle global est adapté pour chaque locuteur du registre à l'aide de ses propres données.

Reynolds [39] a récemment proposé un algorithme de ce type:

1) Un modèle global est développé *a priori*, à l'aide des données de tous les locuteurs:

$$\lambda_p = \{w_{pi}, \mu_{pi}, \sigma_{pi}^2\} \qquad i = 1, 2, \ldots, M$$

2) Un modèle individuel est par la suite développé pour chaque locuteur du registre, strictement à l'aide de ses propres données:

$$n_i = \sum_{t=1}^{T} p(i|o_t) \tag{4.1}$$

$$\mu_{si} = \frac{1}{n_i} \sum_{t=1}^{T} p(i|o_t) \cdot o_t \tag{4.2}$$

$$^2\mu_{si} = \frac{1}{n_i} \sum_{t=1}^{T} p(i|o_t) \cdot diag(o_t o_t') \tag{4.3}$$

$$\sigma_{si}^2 = {}^2\mu_{si} - \mu_{si}^2 \tag{4.4}$$

$$w_{si} = \frac{n_i}{T} \tag{4.5}$$

$$\tag{4.6}$$

donc,

$$\lambda_s = \{w_{si}, \mu_{si}, \sigma_{si}^2\} \qquad i = 1, 2, \ldots, M$$

3) Un coefficient de crédibilité est calculé afin de mesurer le niveau de fiabilité qui peut être accordé aux données individuelles par rapport aux données globales.

$$\alpha_i = \frac{n_i}{n_i + r} \tag{4.7}$$

4) Le modèle individuel *a posteriori* est calculé pour chacun des locuteurs et correspond au modèle global adapté pour les données observées:

$$w_{ai} = \frac{\alpha_i w_{si} + (1-\alpha_i) w_{pi}}{\sum_{i=1}^{M} \alpha_i w_{si} + (1-\alpha_i) w_{pi}} \quad (4.8)$$

$$\mu_{ai} = \alpha_i \mu_{si} + (1-\alpha_i) \mu_{pi} \quad (4.9)$$

$${}^2\mu_{ai} = \alpha_i \cdot {}^2\mu_{si} + (1-\alpha_i) \cdot {}^2\mu_{pi} \quad (4.10)$$

$$\sigma_{ai}^2 = {}^2\mu_{ai} - \mu_{ai}^2 \quad (4.11)$$

donc,

$$\lambda_s = \{w_{ai}, \mu_{ai}, \sigma_{ai}^2\} \qquad i = 1, 2, \ldots, M$$

La valeur du paramètre r, $r \in \mathbb{R}^+$ est généralement établie empiriquement. Ce paramètre correspond à l'inertie du modèle *a priori*: lorsque $r \to \infty$, on obtient $\alpha_i = 0$, $\forall i$. Donc, $\lambda_a = \lambda_p$. À l'inverse, si $r = 0$, $\alpha_i = 0$, $\forall i$ et $\lambda_a = \lambda_s$.

Dans le domaine actuariel, le calcul du paramètre r a fait l'objet de nombreuses études et toute la théorie de la crédibilité y est reliée. Ces études pourraient certainement être utilisées à profit dans le domaine de la reconnaissance du locuteur si le calcul de modèles adaptés devenait un champ de recherche fructueux.

4.4 Conclusion

Les modèles GMM ont été présentés et décrits. L'algorithme EM pour l'estimation des paramètres a été énoncé dans sa version simplifiée, appliquée aux GMM. Enfin, un des algorithmes permettant le calcul de modèles adaptés a été présenté.

Chapitre 5

Les techniques de normalisation

5.1 Introduction

La question de l'évaluation de la probabilité d'une séquence d'observations ($P(\mathbf{O}_c)$) générée par un utilisateur a été soulevée au chapitre 3. Le problème est valide pour l'identification du locuteur avec la possibilité de rejet et la vérification du locuteur. Ce chapitre s'attardera uniquement au second cas.

La complexité du problème découle de la nécessité d'approximer un modèle acoustique universel à partir de données relativement fractionnaires.

Pour ce qui est de la vérification du locuteur, la règle de décision (acceptation ou rejet) peut s'exprimer ainsi:

$$D = \begin{cases} A & \text{si} \quad P(\lambda_d|\mathbf{O}_c) \geq S \\ R & \text{si} \quad P(\lambda_d|\mathbf{O}_c) < S \end{cases} \tag{5.1}$$

ou bien, selon Bayes,

$$D = \begin{cases} A & \text{si} \quad P(\mathbf{O}_c|\lambda_d) \cdot P(\lambda_d)/P(\mathbf{O}_c) \geq S \\ R & \text{si} \quad P(\mathbf{O}_c|\lambda_d) \cdot P(\lambda_d)/P(\mathbf{O}_c) < S \end{cases} \tag{5.2}$$

où λ_d est le modèle développé pour le locuteur propriétaire des ressources auxquelles l'utilisateur *désire* accéder. A et R désignent l'acceptation et le rejet, respectivement. Enfin, S représente le *seuil*, i.e., la valeur minimale de $P(\mathbf{O}_c|\lambda_d) \cdot P(\lambda_d)/P(\mathbf{O}_c)$ nécessaire afin que l'utilisateur puisse accéder aux ressources désirées.

De plus, en supposant des locuteurs équiprobables, on peut reformuler la règle de décision ainsi:

$$D = \begin{cases} A & \text{si} \quad P(\mathbf{O}_c|\lambda_d)/P(\mathbf{O}_c) \geq S \\ R & \text{si} \quad P(\mathbf{O}_c|\lambda_d)/P(\mathbf{O}_c) < S \end{cases} \tag{5.3}$$

Où S est ajusté de façon appropriée. Cette dernière considération est toutefois purement théorique puisque le seuil S est établi de façon empirique afin de minimiser une combinaison quelconque des erreurs de type I et II qui seront appelées FA, pour fausse acceptation et FR pour faux rejet.

Le calcul de $P(\mathbf{O}_c|\lambda_d)$ a été résolu au chapitre 3. Le calcul de $P(\mathbf{O}_c)$ fera l'objet de la première partie du chapitre. De nouvelles techniques de normalisation seront par la suite proposées.

5.2 Les cohortes

Soit Ω, l'ensemble universel de tous les locuteurs. On peut exprimer $P(\mathbf{O}_c)$ sous la forme suivante:

$$P(\mathbf{O}_c) = \sum_{j \in \Omega} P(\mathbf{O}_c|j) \cdot P(j) \tag{5.4}$$

Naturellement, personne ne dispose d'un corpus de données tel que cette somme puisse être évaluée. Toutefois, dans la mesure où le nombre ($N_{\mathcal{R}}$) de locuteurs faisant partie du registre $\mathcal{R}$ est suffisamment élevé, et que ceux-ci peuvent être considérés comme représentatifs des utilisateurs plausibles du système de reconnaissance (oublions les perroquets), l'équation 5.4 peut être approximée par la suivante:

$$P(\mathbf{O}_c) \approx \sum_{j \in \mathcal{R}} P(\mathbf{O}_c|j) \cdot P(j|j \in \mathcal{R}) \tag{5.5}$$

Il est important de souligner que le modèle λ_j est une approximation paramétrique de la distribution fondamentale inconnue des observations pouvant être générées par le locuteur. Cette dernière observation, bien que triviale, met en relief la faiblesse de la détermination, *a priori*, d'une architecture de modèle fixe et indépendante du locuteur. La relaxation de cette hypothèse, en particulier par le biais d'un nombre variable de gaussiennes, fera, sous peu, l'objet de travaux de recherche.

Cette modélisation nous conduit donc vers une seconde approximation:

$$P(\mathbf{O}_c) \approx \sum_{j \in \mathcal{R}} P(\mathbf{O}_c|\lambda_j) \cdot P(j|j \in \mathcal{R}) \tag{5.6}$$

La cardinalité ($N_{\mathcal{R}}$) du registre peut être telle que le calcul suggéré par l'équation précédente soit prohibitif et justifie la recherche d'une approximation plus forte. Certains auteurs ont suggéré l'établissement *a priori*, i.e., après le développement des modèles individuels des locuteurs du registre mais avant la phase test du système, d'une *cohorte*. La cohorte ($\mathcal{C}_d$) du locuteur (d) est formée d'un sous-groupe des locuteurs du registre pour lesquels la probabilité conditionnelle $P(\mathbf{O}_c|\lambda_j)$ sera calculée:

$$P(\mathbf{O}_c) \approx \sum_{j \in \mathcal{C}_d} P(\mathbf{O}_c|\lambda_j) \cdot P(j|j \in \mathcal{C}_d) \tag{5.7}$$

La sélection des membres de la cohorte d'un locuteur a fait elle-même l'objet de plusieurs publications. Initialement, il a été proposé [17] que la cohorte devait comprendre les locuteurs les plus susceptibles d'être de bons imposteurs pour le locuteur désiré. En second lieu, il fut remarqué que des observations éloignées pouvaient porter à confusion puisque la soustraction d'une série de queues de distributions déterminait la décision finale d'acceptation ou de rejet. Pour éviter ce problème, Reynolds [39] suggéra la présence, au sein de la cohorte, de locuteurs *éloignés* du locuteur désiré. Troisièmement, afin d'éviter la redondance des locuteurs et de maximiser l'information apportée par l'ensemble de la cohorte, il a été suggéré [17] de s'assurer que les locuteurs de la cohorte soient eux-mêmes éloignés les uns des autres.

La figure 5.1 illustre une situation où C_1, C_2 et C_3 sont des candidats potentiels pour la cohorte du locuteur D. Supposons que C_2 ait déjà été inséré au sein de la cohorte. Bien que D soit plus rapproché de C_1 que de C_3, on préférera tout de même ce dernier puisque son insertion dans la cohorte permet l'apport d'une quantité plus importante d'information nouvelle quant à la provenance des observations. L'infor-

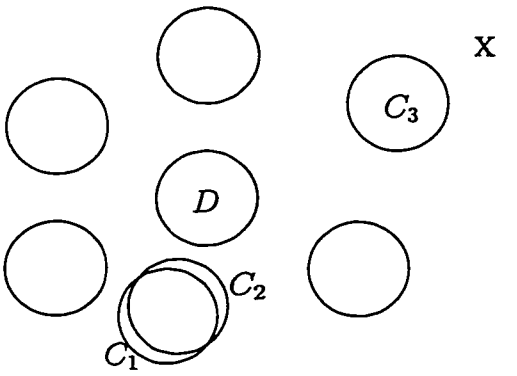

Figure 5.1: La sélection des membres d'une cohorte: locuteurs rapprochés

mation fournie par C_1 est trop redondante par rapport à celle procurée par l'insertion de C_2 au sein de la cohorte. L'observation X illustre l'importance de l'inclusion de C_3 au sein de la cohorte. Donc, dans l'éventualité où six locuteurs proches de D devaient former la cohorte, C_1 serait rejeté; C_2 et C_3 seraient insérés.

La figure 5.2 illustre une cohorte, pour le locuteur D, formée de six locuteurs rapprochés et bien dispersés les uns des autres. De plus, six autres locuteurs éloignés sont inclus. Ces derniers sont eux aussi bien dispersés les uns des autres. L'observation X illustre l'utilité de l'inclusion de locuteurs éloignés: sans la présence du locuteur C, il est probable que X ait été considérée comme provenant de D.

Étant donné l'impossibilité d'établir, *a priori*, la valeur de $P(j|j \in \mathcal{C})$, on l'approxime par une fonction quelconque, choisie pour ses propriétés particulières. Ceci nous conduit vers une quatrième approximation:

$$P(\mathbf{O}_c) \approx \sum_{j \in \mathcal{C}_d} P(\mathbf{O}_c|\lambda_j) \cdot f(j) \tag{5.8}$$

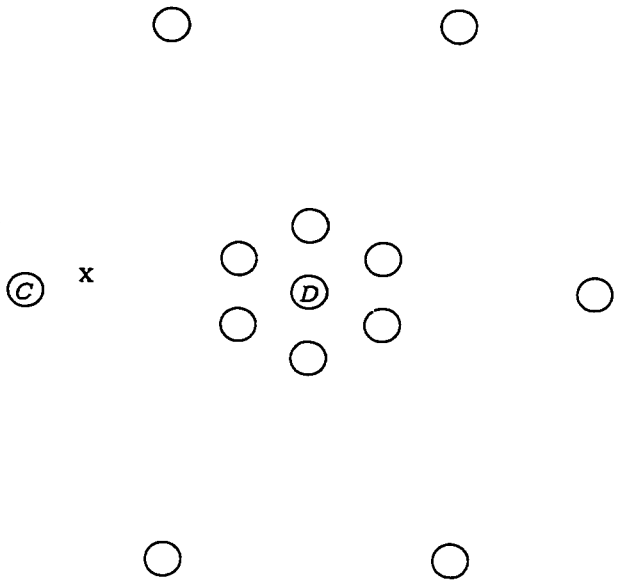

Figure 5.2: La sélection des membres d'une cohorte: locuteurs rapprochés et éloignés

où

$$\sum_{j \in C_d} f(j) \;=\; 1 \tag{5.9}$$

La fonction $f(j)$ est généralement établie de sorte que $P(\mathbf{O}_c)$ corresponde à la moyenne

$$P(\mathbf{O}_c) \;\approx\; \frac{1}{N_{C_d}} \sum_{j \in C_d} P(\mathbf{O}_c|\lambda_j) \tag{5.10}$$

ou au maximum,

$$P(\mathbf{O}_c) \;\approx\; \arg\max_{j \in C_d} P(\mathbf{O}_c|\lambda_j) \tag{5.11}$$

de l'ensemble des valeurs $P(\mathbf{O}_c|\lambda_j)$ où $j \in \mathcal{C}_d$.

5.3 La proximité de deux locuteurs

Comme mentionné au chapitre 4, les gaussiennes, aussi appelées "mixtures" représentent idéalement chacune un phonème ou une classe de phonèmes, parfois simplement une portion d'un phonème. On s'intéressera donc à connaître la proximité, pour deux locuteurs, de chacun de leurs phonèmes, i.e., la proximité inter-locuteurs et intra-phonétique.

Il faut donc utiliser une mesure de proximité dominée par la mesure entre les éléments rapprochés et associés plutôt qu'une mesure de distance, dominée par la mesure entre les éléments les plus éloignés, i.e., une mesure inter-phonétique. De cette constatation découle l'utilisation du terme "proximité" de préférence au terme "distance".

Les méthodes de mesure de proximité peuvent être classées en trois catégories: données vs données, données vs modèle et modèle vs modèle.

Supposons que les fichiers d'entraînement comportent tous un nombre similaire d'observations. Soit N, le nombre moyen d'observations par fichier d'entraînement. Soit M, le nombre de mixtures utilisées pour modéliser les locuteurs, avec $M \ll N$. Les trois catégories de mesure mentionnées ci-haut demandent un nombre d'opérations dans l'ordre de $O(N^2)$, $O(NM)$ et $O(M^2)$, respectivement.

Peu de gens se sont attardés au calcul fastidieux de la proximité données vs données. Ce calcul revêt toutefois un avantage certain: il est indépendant de la structure artificielle imposée par le modèle choisi et de sa possible faiblesse.

Reynolds [39] a proposé un calcul simple et symétrique de la proximité données vs modèle:

$$prox(A,B) = \frac{P(\mathbf{O}_A|\lambda_B) \cdot P(\mathbf{O}_B|\lambda_A)}{P(\mathbf{O}_A|\lambda_A) \cdot P(\mathbf{O}_B|\lambda_B)} \tag{5.12}$$

Finan et al. [12] ont analysé les effets de l'utilisation d'une méthode de type modèle vs modèle pour le calcul de la proximité. Les résultats obtenus montrent que le classement des locuteurs du registre selon leur proximité les uns par rapport aux autres est peu affecté par le type de la méthode utilisée (données vs modèle ou modèle vs modèle). Ainsi, la composition de la cohorte des locuteurs est peu affectée par le type de la méthode utilisée.

Enfin, étant donné que les valeurs nécessaires au calcul de proximité proposé par Reynolds étaient déjà disponibles et ne nécessitaient que peu de programmation, c'est le calcul données vs modèle qui a été choisi pour effectuer les expériences liées à la cohorte.

5.4 Le modèle universel de normalisation

Une des difficultés liée à l'utilisation de la cohorte est la sélection des membres de la cohorte pour chacun des locuteurs du registre. De plus, pour chaque membre de la cohorte d'un locuteur, un calcul de vraisemblance doit être effectué au moment de la

vérification.

Le modèle universel de normalisation, ou "UBM" (de l'anglais "universal background model") a l'avantage d'être indépendant du locuteur. On évite donc le calcul des proximités entre les différents modèles et l'établissement des cohortes. Un seul modèle global est entraîné une fois à l'aide des données de tous les locuteurs du registre et est utilisé lors de chaque vérification, quel que soit le locuteur désiré. Le UBM devrait être entraîné sur un large éventail de locuteurs, permettant ainsi de modéliser l'espace acoustique qui prévalait lors de l'entraînement. Le modèle ne devrait pas, idéalement, être trop influencé par les données individuelles d'un locuteur particulier [39].

5.5 Le modèle universel de normalisation adapté au combiné

Étudiant les résultats de reconnaissance de locuteur produits par différents systèmes, Reynolds [39] observa que les modèles de locuteurs produisaient différentes distributions de résultats pour les mêmes séquences d'observations tests, en particulier dans le cas de numéros de téléphone "mismatched".

Cette observation, en combinaison avec d'autres études préalables le mena à croire que le type de combiné utilisé, soit "carbon-button" ou "electret", pouvait être à l'origine de ces divergences. C'est-à-dire que les fichiers tests, prononcés à l'aide d'un combiné du même type que celui utilisé par le locuteur désiré lors de l'entraînement

de son modèle, obtenaient de meilleurs résultats que ceux qui avaient été prononcés à l'aide d'un combiné de l'autre type.

Afin de tenir compte de ces effets, Reynolds [39] a proposé la méthode "hnorm" selon laquelle, pour chaque locuteur du registre, certaines informations supplémentaires doivent être stockées en mémoire:

$$\lambda_d^{hnorm} = \{\lambda_d, \mu_{d,c}, \sigma_{d,c}, \mu_{d,e}, \sigma_{d,e}\} \tag{5.13}$$

où

λ_d	=	modèle GMM calculé pour le locuteur désiré.
$\mu_{d,c}$	=	moyenne des ratios de vraisemblance produits par le modèle λ_d pour les séquences d'observations provenant d'un combiné de type "carbon-button"
$\sigma_{d,c}$	=	variance des ratios
$\mu_{d,e}$ et $\sigma_{d,e}$	=	idem pour un combiné de type "electret"

Donc, en supposant que l'observation $\mathbf{O}_c$ ait été identifiée comme provenant d'un combiné de type "electret", le nouveau calcul de vraisemblance s'exprime ainsi:

$$\Lambda^{hnorm}(\mathbf{O}_c|\lambda_d) = \frac{\Lambda(\mathbf{O}_c|\lambda_d) - \mu_{d,e}}{\sigma_{d,e}} \tag{5.14}$$

Cette méthode est maintenant considérée comme l'une des plus performantes proposées à ce jour [3], suite aux résultats obtenus par Reynolds.

5.6 La vraisemblance des trames pour la normalisation

Nakagawa et Markov [26] ont étudié la normalisation au niveau de chacune des observations (trames), plutôt que pour la séquence complète d'observations.

La comparaison s'est réalisée au niveau de la normalisation par cohorte. Traditionnellement, on a:

$$\begin{aligned} \Lambda(\mathbf{O}_c|\lambda_d) &= \frac{P(\mathbf{O}_c|\lambda_d)}{\frac{1}{Nc_d}\sum_{j=1}^{Nc_d} P(\mathbf{O}_c|\lambda_j)} \\ &= \frac{\prod_{t=1}^{T} p(o_{c,t}|\lambda_d)}{\frac{1}{Nc_d}\sum_{j=1}^{Nc_d}\prod_{t=1}^{T} p(o_{c,t}|\lambda_j)} \end{aligned} \tag{5.15}$$

Le nouveau ratio proposé correspond donc à ceci:

$$\Lambda(\mathbf{O}_c|\lambda_d) = \prod_{t=1}^{T} \frac{p(o_{c,t}|\lambda_d)}{\frac{1}{Nc_d}\sum_{j=1}^{Nc_d} p(o_{c,t}|\lambda_j)} \tag{5.16}$$

$$= \frac{\prod_{t=1}^{T} p(o_{c,t}|\lambda_d)}{\prod_{t=1}^{T}\frac{1}{Nc_d}\sum_{j=1}^{Nc_d} p(o_{c,t}|\lambda_j)} \tag{5.17}$$

Cette méthode a elle aussi permis d'améliorer substantiellement les résultats obtenus par le système de vérification sur lequel elles ont été implantées.

5.7 La muselière

Observant les résultats de reconnaissance du locuteur, il semblait que certaines séquences d'observations obtenaient d'excellents résultats et ce, quel que soit le modèle auquel

elles étaient confrontées. Ces séquences étaient donc à la source de plusieurs fausses acceptations et constituaient donc, ce que l'on appelle des *loups*. Cette constatation fut faite *avant* même l'implantation d'une technique de normalisation (cohorte, UBM, etc.), alors que le simple calcul de vraisemblance de base était utilisé:

$$\Lambda(\mathbf{O}_c|\lambda_d) \;=\; P(\mathbf{O}_c|\lambda_d) \tag{5.18}$$

Le succès de ces séquences ne pouvait s'expliquer que par la présence d'un élément commun à tous les modèles de locuteurs du registre et aussi présent dans la séquence d'observations test du loup. Deux possibilités semblaient valides: l'environnement acoustique et la structure de modèle choisie.

À ce moment, la technique de normalisation envisagée était le UBM, généralement constitué d'un nombre substantiellement supérieur de mixtures et donc peu valable pour la vérification du succès général de la séquence d'observations sur la structure du modèle. L'objectif visé par l'implantation d'un modèle de type UBM est d'ailleurs plutôt relié à la première possibilité mentionnée: la représentation de l'environnement acoustique général, présent lors de l'entraînement des modèles des locuteurs du registre. Cet objectif est d'ailleurs clairement poursuivi avec l'introduction de la technique *hnorm*.

Afin d'étudier la seconde possibilité, i.e., vérifier la possibilité que la séquence d'observations puisse être bien représentée par le modèle choisi, il fallait calculer l'erreur d'entraînement (ou "goodness-of-fit") des données de la séquence par rapport au modèle. Pour ce faire, l'idée simple de *construire artificiellement un modèle selon la même structure utilisée pour les modèles du registre, mais avec les données de la*

séquence test observée fut envisagée. Le nouveau calcul de vraisemblance proposé s'exprime donc ainsi:

$$\Lambda(\mathbf{O}_c|\lambda_d) = \frac{P(\mathbf{O}_c|\lambda_d)}{P(\mathbf{O}_c|\lambda_c)} \tag{5.19}$$

La modélisation doit permettre d'optimiser un gain de simplification de calcul contrebalancé par une perte d'information. L'architecture du modèle choisie et fixée *a priori* permet un gain de simplification essentiellement semblable pour toutes les séquences d'observations. La perte d'information n'est, quant à elle, pas nécessairement constante d'une séquence à l'autre. Le calcul de $P(\mathbf{O}_c|\lambda_c)$, que nous appellerons *la muselière*, permet de mesurer la qualité de représentation de la séquence d'observations par l'architecture du modèle, i.e., la quantité d'information que revêt la séquence et qui a été retenue par le modèle.

La valeur numérique de $P(\mathbf{O}_c|\lambda_c)$ est d'autant plus élevée que la séquence $\mathbf{O}_c$ peut être bien représentée par la structure du modèle utilisé. Cette valeur devrait idéalement nous permettre d'identifier, *a priori*, les loups.

L'estimation des paramètres du modèle λ_c constitue le désavantage principal de la technique présentée, ce modèle n'ayant pas été développé *a priori*. Le modèle serait donc probablement encore en phase de développement à la fin de la prononciation du fichier test par le locuteur. Par la suite, la muselière devrait être calculée. En somme, la prise de décision du système ne pourrait s'effectuer en temps réel et être livrée après un temps constant suivant la fin de la prononciation. Elle nécessiterait un délai supplémentaire. L'analyse de ce délai supplémentaire n'a pas été effectuée et pourrait faire l'objet de recherches subséquentes, dépendamment de la réussite expérimentale

de la méthode et donc de l'intérêt qu'elle pourrait susciter. Toutefois, compte tenu de la longeur des séquences traitées (3, 10 ou 30 secondes), le problème du temps réel n'est pas aussi crucial que pour la reconnaissance de la parole. De plus, l'utilisation des techniques telles MLLR ou MAP permettrait de commencer l'estimation des paramètres avant la fin de la phrase.

Il convient maintenant de tenter de vérifier à quel point le calcul de la muselière est corrélé à l'*appétit* ($P(\mathbf{O}_c|\lambda_c)$) d'un fichier, i.e., d'un échantillon d'un locuteur. À cette fin, une mesure valable de cet appétit consiste à calculer, *a posteriori*, le succès moyen du fichier sous étude sur l'ensemble des modèles. Ce calcul (équation 5.20) a été effectué sur l'ensemble des 180 fichiers de la base de données du corpus de SPIDRE.

$$A_c = \frac{1}{179} \sum_{j=1, j\neq c}^{180} \log P(\mathbf{O}_c|\lambda_j) \tag{5.20}$$

$$W_c = \log P(\mathbf{O}_c|\lambda_c) \tag{5.21}$$

Le calcul de W_c constitue, quant à lui, la mesure, *a priori* de l'appétit du fichier en question. L'hypothèse qui fut initialement posée est qu'il existe une corrélation très fortement positive, entre les valeurs A_c et W_c.

Le nuage de points de la figure 5.3 illustre, de façon flagrante, l'existence de la corrélation recherchée avec un coefficient de $\rho = 0,97$. La droite de régression linéaire minimisant l'erreur quadratique moyenne suit l'équation suivante:

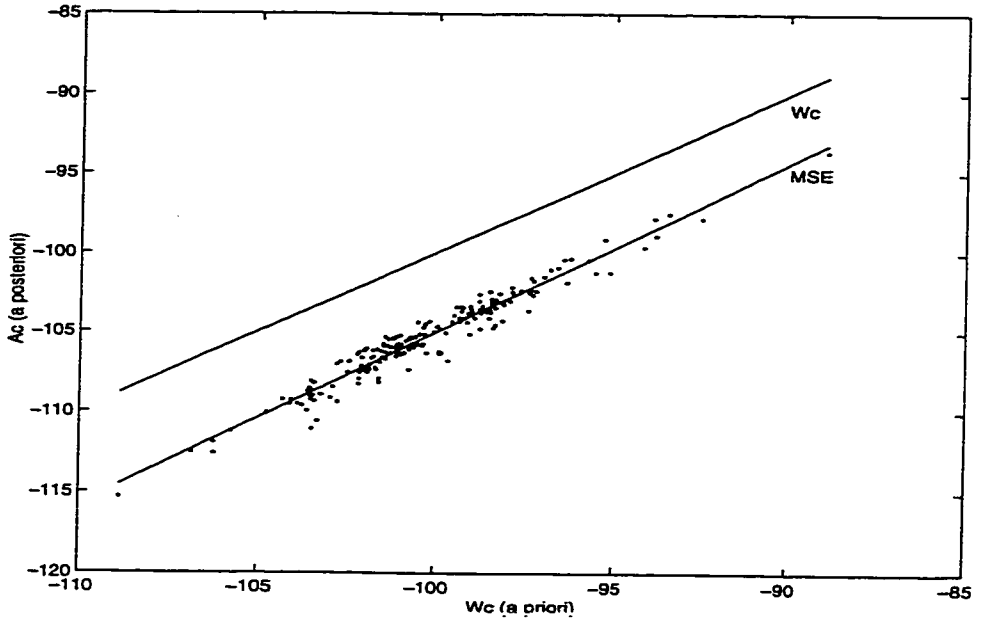

Figure 5.3: Résultats de A_c (*a posteriori*) en fonction de W_c (*a priori*)

$$\hat{A}_c = 1.0716 \cdot W_c + 2.1003 \tag{5.22}$$

Enfin, l'analyse ANOVA donne un coefficient de $R^2 = 94\%$, i.e., que 94% de la variation de A_c peut s'expliquer par la variation de W_c.

Il semble donc clair que le calcul de W_c permet d'établir de façon fiable, l'appétit d'un fichier. L'analyse précédente porte toutefois sur la moyenne des résultats obtenus par le fichier et une analyse plus complète sur les résultats individuels est de mise.

La figure 5.4 illustre les ($180 \times 180 =$) 32400 résultats de fichiers tests de 30 secondes sur les modèles développés à l'aide des fichiers entiers d'une durée moyenne

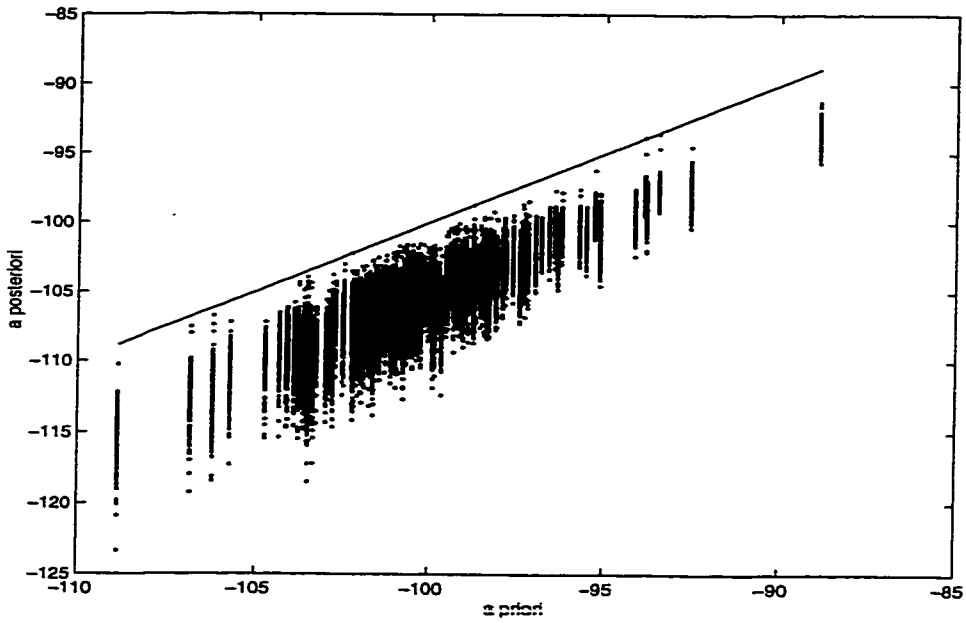

Figure 5.4: Résultats de $R_{c,d} = P(\mathbf{O}_c|\lambda_d)$ (*a posteriori*) en fonction de $W_c = P(\mathbf{O}_c|\lambda_c)$ (*a priori*)

de 66,4 secondes. En abscisse se trouve la valeur de W_c, en ordonnée, les valeurs individuelles de $R_{c,d} = \log P(\mathbf{O}_c|\lambda_d)$. Ainsi, le graphique est formé de 179 colonnes de points et d'une droite formée des valeurs de W_c mises en ordonnée, i.e., la droite d'équation $y = x$. Clairement, les valeurs de $R_{c,d}$ sont corrélées avec les valeurs de W_c. Pour les trois fichiers provenant du locuteur lui même, on obtient une corrélation de $\rho = 0,90$ (le fichier d'entraînement du locuteur c a été retiré afin d'éviter l'introduction d'un biais positif). Dans le cas des 176 fichiers provenant de locuteurs différents, $\rho 0,89$ $(3 + 176 + 1 = 180)$. Le coefficient R^2, quant à lui, tombe à 79%. Cette chute indique que la valeur de W_c n'est pas le seul facteur expliquant la variabilité des résultats. D'autres sources doivent être proposées.

La valeur de A_c, étant donné le nombre assez élevé de fichiers ayant servi à son calcul, devrait idéalement se trouver indépendante des observations individuelles des locuteurs du registre ("speaker independent") et ne dépendre que du fichier test et de l'architecture du modèle. Le coefficient de $R^2 = 94\%$ indique que la variabilité des résultats strictement due au fichier test peut s'expliquer essentiellement par la valeur de W_c. Donc, les sources de variabilité des résultats à trouver doivent donc provenir du fichier ayant servi au développement du modèle du locuteur désiré, à la structure du modèle, à leur interaction ou encore à l'interaction de l'un de ces trois éléments avec le fichier test.

5.8 La muselière, l'armature, la contre-attaque et l'impact

Dans le domaine de la reconnaissance de formes, le problème des *loups et moutons* (en anglais, "wolves and sheeps") est bien connu. Les loups (fichiers tests) réussissent à obtenir d'excellents résultats sur presque tous les modèles développés. Les moutons (fichiers d'entraînement), quant à eux, sont tels que plusieurs fichiers tests obtiennent d'excellents résultats lorsque confrontés aux modèles développés pour ces moutons. Les loups et les moutons sont donc à la source de plusieurs fausses acceptations (FA).

Les résultats obtenus avec le calcul de vraisemblance de base faisaient clairement ressortir la présence de certains loups féroces, d'où l'idée de la muselière. Cette vigueur des loups pouvait, en fait masquer la faiblesse de certains moutons. Cette

hypothèse fut toutefois rejetée puisque même après l'utilisation de la muselière, aucun mouton ne semblait se distinguer.

Pourtant, Campbell [3] avait fait face au problème inverse, i.e., la présence de moutons trop faibles. Il m'apparut donc que la présence/abscence de loups et de moutons dépende de la base de données et non d'une réalité universelle. La recherche d'un ratio permettant de protéger les moutons semblait donc justifiée par le désir de développer un calcul de vraisemblance plus robuste.

La première idée fut d'introduire, au dénominateur, *l'armature*, i.e., une protection pour les moutons similaire à la muselière utilisée pour les loups:

$$\Lambda(\mathbf{O}_c|\lambda_d) = \frac{P(\mathbf{O}_c|\lambda_d)}{P(\mathbf{O}_d|\lambda_d)} \tag{5.23}$$

La seconde idée fut d'utiliser conjointement la muselière et l'armature:

$$\Lambda(\mathbf{O}_c|\lambda_d) = \frac{P(\mathbf{O}_c|\lambda_d)}{P(\mathbf{O}_c|\lambda_c) \cdot P(\mathbf{O}_d|\lambda_d)} \tag{5.24}$$

Les deux calculs obtinrent peu de succès et furent rapidement rejetés.

Pourtant, l'observation des résultats laissait entrevoir une corrélation évidente entre le calcul de l'armature et la faiblesse de certains modèles. L'armature nécessitait toutefois une certaine normalisation, son comportement étant tantôt trop vif, tantôt trop faible. Le calcul de la *contre-attaque* ($P(\mathbf{O}_d|\lambda_c)$) apparut approprié et permettant le développement d'un calcul de vraisemblance symétrique, qui sera appelé *l'impact*:

$$\Lambda(\mathbf{O}_c|\lambda_d) = \frac{P(\mathbf{O}_c|\lambda_d) \cdot P(\mathbf{O}_d|\lambda_c)}{P(\mathbf{O}_c|\lambda_c) \cdot P(\mathbf{O}_d|\lambda_d)} \quad (5.25)$$

L'impact, comme le chapitre 6 le montrera obtint des résultats spectaculaires. Après m'être longuement concentré sur le problème des loups et moutons, il fut satisfaisant d'observer que ce calcul permettait de réduire substantiellement le problème des fausses acceptations.

L'impact et la muselière apportent la nouveauté du développement artificiel d'un modèle pour les données tests basé sur l'architecture utilisée pour les modèles des fichiers d'entraînement. Cette dernière idée correspond donc à l'apport essentiel de la recherche effectuée et présentée dans ce mémoire.

5.9 Combiner les techniques de normalisation

La muselière et l'impact qui ont été proposées ne tiennent pas compte du modèle de l'environnement acoustique et de sa possible disparité entre celui prévalant lors de l'enregistrement des données d'entraînement et celui prévalant lors de l'engistrement des données tests de l'utilisateur. Cette disparité pourrait être tenue en compte par le développement d'un modèle de type UBM, λ_b. Ainsi, la combinaison des différentes techniques serait, idéalement, complémentaire et non redondante:

$$\Lambda(\mathbf{O}_c|\lambda_d) = \frac{P(\mathbf{O}_c|\lambda_d) \cdot P(\mathbf{O}_d|\lambda_c) \cdot P(\mathbf{O}_d|\lambda_b)}{P(\mathbf{O}_c|\lambda_c) \cdot P(\mathbf{O}_d|\lambda_d) \cdot P(\mathbf{O}_c|\lambda_b)} \quad (5.26)$$

5.10 Conclusion

Ce chapitre a présenté les techniques de normalisation récemment proposées et couramment utilisées dans le domaine de la vérification du locuteur.

Les séquences d'observations tests sont généralement utilisées contre des modèles développés *a priori*. L'apport essentiel de cet ouvrage réside donc dans le simple fait de développer un modèle pour les données tests elles-mêmes et de tenter d'utiliser ce modèle de la meilleure façon possible.

Le désavantage de devoir développer ce modèle "on-line" et donc de ralentir le service fourni par le système est contrebalancé par une amélioration substantielle des résultats. Dans les applications requérant un niveau de sécurité élevé, un tel délai est plus susceptible d'être accepté du grand public. Il est donc important de noter que les améliorations apportées par les nouvelles techniques proposées, en particulier l'impact, se situent au niveau des fausses acceptations et s'adressent donc aux systèmes cherchant à augmenter leur niveau de sécurité, ceux qui accepteraient vraisemblablement une augmentation du délai de service.

Chapitre 6

Résultats expérimentaux

Ce chapitre décrit les résultats expérimentaux obtenus à l'aide du système de vérification du locuteur développé au CRIM et du logiciel HTK d'Entropie. Les critères d'évaluation des systèmes de vérification du locuteur sont discutés à la section 6.1. Le corpus de données utilisé fut SPIDRE et est décrit dans la section 6.2. Différentes segmentations des données disponibles entre corpus d'entraînement et corpus test furent utilisées. Ces segmentations seront appelées *schèmes* et seront décrites à la section 6.3. Certains paramètres utilisés ne furent pas modifiés et sont brièvement listés dans la section 6.4. L'essentiel des expérimentations a porté sur des modèles développés avec un nombre fixe de 16 gaussiennes. L'utilisation de ce nombre est justifiée a la section 6.5. Les résultats obtenus à l'aide de la muselière et de l'impact sont analysés à la section 6.6. Enfin, une comparaison de ces résultats avec ceux obtenus à l'aide de la cohorte est présentée à la section 6.7

6.1 Les critères d'évaluation des systèmes de vérification

6.1.1 La courbe "ROC" et le "EER"

La courbe ROC (de l'anglais "Receiver Operating Curve") est déployée sur le plan $FR \times FA$ (i.e., FR en abscisse, FA en ordonnée). La figure 6.1 illustre un exemple d'une courbe ROC. Lorsque le seuil S d'acceptation de l'utilisateur en tant que client cible est trop élevé, alors le système rejette tous les locuteurs: tous les imposteurs mais aussi toutes les personnes qui désirent accéder à leurs propres ressources. Ainsi, $FR = 100\%$ et $FA = 0\%$. À mesure que l'on abaisse le seuil d'acceptation, la quantité de faux rejets diminue. Toutefois, certains imposteurs sont acceptés. Donc FR diminue et FA augmente, jusqu'au point où le seuil est tel que tous les locuteurs sont acceptés. À ce moment, le système ne rejette plus de bons clients mais accepte tous les imposteurs. Donc, $FR = 0\%$ et $FA = 100\%$

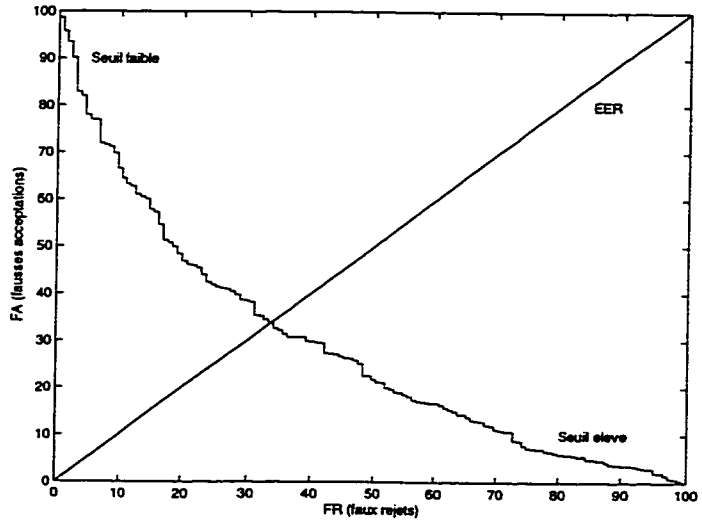

Figure 6.1: Illustration de la courbe ROC et de la droite EER

De façon générale, dans les domaines d'authentification, le courant actuel de la recherche encourage la publication des résultats d'expérimentation sur la base du EER (en anglais "equal error rate"). Afin d'établir la valeur du EER, on choisit le seuil d'acceptation faisant en sorte que les erreurs de types α et β (ou I et II ou encore $H0$ et $H1$) soient équivalentes, i.e., que la probabilité que le système accepte un faussaire (FA pour fausse acceptation) est égale à la probabilité de rejeter un client (FR pour faux rejet). Sur la figure 6.1, la droite $FA = FR$ est tracée depuis l'origine jusqu'à l'extrémité droite et haute du graphique. Le point de croisement de la droite $FA = FR$ avec la courbe ROC permet de bien voir le point correspondant au EER.

Plusieurs résultats sont souvent présentés avec la simple valeur du EER. Cette valeur ne représente pourtant qu'une fraction minime de l'information nécessaire à l'évaluation de la qualité d'un système de reconnaissance. Bien que l'utilisation de la valeur EER puisse être justifiée par des besoins de comparaison entre différents systèmes, la présentation de la courbe ROC permet au lecteur de visualiser, d'un simple coup d'oeil, toute l'information pertinente reliée à la performance du système. Trop de publications (e.g. [26]) ne se contentent que de fournir un tableau des valeurs du EER pour une multitude de combinaisons possibles des divers paramètres de leur système. La courbe ROC occupe, certes, beaucoup d'espace et les publications limitées à 4 pages ne peuvent se permettre l'insertion d'un trop grand nombre de graphiques. L'utilisation des deux mesures de performance s'avère donc souhaitable. Il semble toutefois que l'apport informatif du EER soit largement sur-estimé, au détriment de la courbe ROC.

6.1.2 La *droite-m* de distance minimale par rapport à l'origine

Fondamentalement, la qualité d'un système de reconnaissance peut être mesurée par le *coût* associé à son utilisation. On entend ici par coût, non pas les frais d'implantation ou de maintenance, mais les pertes encourues par le gestionnaire ou l'utilisateur du système en cas d'erreur de reconnaissance, que ce soit une fausse acceptation ou un faux rejet.

Étant donné la nature stochastique liée au processus, une mesure M valable du coût peut être représentée par l'espérance mathématique des frais encourus par utilisation du système. Le critère d'évaluation correspond donc à la minimisation, par l'ajustement du seuil S, de la valeur $M = E[C]$:

Soient:

- M, la mesure utilisée pour évaluer la faiblesse du système,
- C, le coût encouru,
- f, un faussaire (ou imposteur),
- c, un client honnête,
- A, la décision du système d'accepter l'utilisateur,
- R, la décision du système de rejeter l'utilisateur,
- FA, une fausse acceptation,
- FR, un faux rejet,

alors,

$$
\begin{aligned}
M &= E[C] \\
&= E[C|f] \cdot p_f + E[C|c] \cdot p_c \\
&= E[C|f, A] \cdot p_{f,A} + E[C|f, R] \cdot p_{f,R} + E[C|c, A] \cdot p_{c,A} + E[C|c, R] \cdot p_{c,R} \\
&= E[C|FA] \cdot p_{f,A} + E[C|FR] \cdot p_{c,R} \\
&= E[C|FA] \cdot p_{FA|f} \cdot p_f + E[C|FR] \cdot p_{FR|c} \cdot p_c \qquad (6.1)
\end{aligned}
$$

Les valeurs de $E[C|FA]$, $E[C|FR]$, p_f et p_c doivent être estimées *a priori*. Les valeurs de $p_{FA|f}$ et $p_{FR|c}$ sont estimées par les données empiriques obtenues, respectivement FA et FR. Elles dépendent donc du seuil S choisi.

En réalité, la probabilité (p_f) qu'un faussaire tente de déjouer le système dépend du gain ($E[C|FA]$) qu'il peut potentiellement ($p_{FA|f}$) en retirer. De façon plus subtile, il semble logique que dans le cas d'un petit nombre potentiel de faussaires (p_f faible), ceux qui se présenteront seront particulièrement féroces ($E[C|FA]$ élevée). Ces références circulaires laissent présumer l'existence de plus d'un point de convergence possible.

Lors de l'établissement initial du système on peut évaluer $E[C|FA]$ et p_f en supposant un état stable. Il faut toutefois prendre garde et considérer ces paramètres comme étant évolutifs, en particulier peu de temps après l'établissement initial du système, puisque la perception du public évoluera rapidement au fur et à mesure que les premiers succès ou échecs du système seront rapportés.

Une fois les quatre paramètres établis $E[C|FA]$, $E[C|FR]$, p_f et p_c, on peut

calculer le ratio suivant:

$$m = \frac{E[C|FA] \cdot p_f}{E[C|FR] \cdot p_c} \quad \text{où } m \in \mathbb{R}^+ \tag{6.2}$$

On peut interpréter m comme étant une mesure de sévérité qui sera imposée au système. Une valeur élevée de m signifie que l'on cherche fortement à se protéger des faussaires (e.g. applications bancaires) et le seuil S sera donc élevé. De la même façon, une valeur faible de m indique que les efforts sont concentrés sur la satisfaction du client (e.g. cartes d'appels), le seuil S sera ajusté à une valeur plus basse et le système sera plus tolérant, moins sévère.

De plus,

$$E[C] = (m \cdot p_{FA|f} + p_{FR|c}) \cdot E[C|FR] \cdot p_c \tag{6.3}$$

Ainsi, dans le plan $FR \times FA$, pour une certaine valeur M_o de M, l'ensemble des points de coût M_o forment une droite de pente $m' = -\frac{1}{m}$. Le but du système étant de minimiser M, on ajustera le seuil S afin de trouver le point de coût minimal parmi l'ensemble des points de la courbe ROC.

Au cours des deux dernières années, les concours organisés par NIST (National Institute for Standards in Technology) ont établi une règle de ce type en spécifiant les paramètres a priori [22]:

$$E[C|FA] = 10 \tag{6.4}$$

$$E[C|FR] = 1 \tag{6.5}$$

$$p_f = 0,01 \tag{6.6}$$

$$p_c = 0,99 \tag{6.7}$$

Ainsi, dans ce cas particulier, $m = \frac{10 \cdot 0,01}{1 \cdot 0,99} \approx 0,1$ et $m' = -9,9$. Les faux rejets (FR) sont donc particulièrement pénalisés.

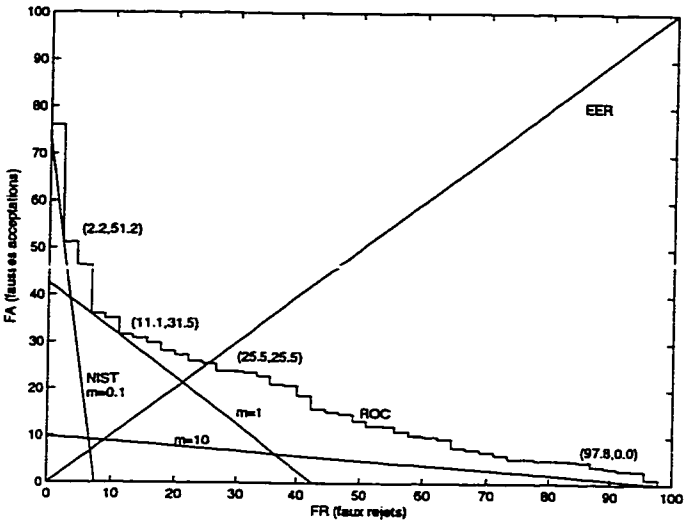

Figure 6.2: Quatre critères différents pour l'évaluation de la performance d'un système de vérification

La figure 6.2 illustre quatre mesures différentes de la performance d'un système de reconnaissance. Selon le EER, on observe le point de croisement de la courbe ROC et de la droite $FR = FA$. On obtient ici le point (25.5,25.5). Selon le critère établi pour les concours de NIST, les faux rejets sont pénalisés environ 10 fois plus que les fausses acceptations, i.e., $m \approx 0.1$. Le point obtenu est (2.2,51.2). La droite de coût minimal avec $m = 10$ correspond à l'optimisation pour un système à haute sécurité.

On obtient le point (97.8,0). Enfin, la droite de coût minimal avec $m = 1$ est illustrée afin de faire ressortir le fait que la mesure selon le EER ne lui correspond pas, une erreur d'interprétation parfois commise. Le point obtenu est (11.1,31.5). En fait, le calcul du EER ne correspond pas à un calcul d'optimisation: on restreint l'espace d'analyse, depuis le plan $FR \times FA$, à la simple droite $FR = FA$, sachant que la courbe ROC ne croisera cette droite qu'en un seul point. D'où la faiblesse d'une analyse effectuée strictement par le biais du EER.

6.1.3 Minimiser FR pour FA=0

Lors de l'implantation éventuelle d'un système à très haute sécurité, on voudra à tout prix éviter l'intrusion d'imposteurs. On peut facilement imaginer une série d'applications (accès à une centrale nucléaire, aux installations de la défense nationale ou utilisation d'un enregistrement audio comme preuve judiciaire, etc.) pour lesquelles une fausse acceptation entraînerait des conséquences catastrophiques (désastre écologique, atteinte à la sécurité des citoyens, exécution d'un innocent, etc.).

De façon mathématique:

$$E[C|FA] \gg E[C|FR] \tag{6.8}$$

$$m \rightarrow \infty \tag{6.9}$$

$$m' \rightarrow 0^- \tag{6.10}$$

Il est donc intéressant de vérifier quel pourcentage des clients honnêtes peuvent être acceptés avant même l'acceptation d'un seul imposteur. Ceci correspond à laisser glisser son regard le long de l'axe des abscisses, depuis l'extrême droite ($FR = 100\%$)

jusqu'au point où la courbe ROC quitte l'axe. Le plus près de l'origine est situé ce point, le mieux adapté sera le système en situation de haute sécurité. Les sections à venir montreront que la muselière et l'impact réussissent, selon ce critère, et dans certains cas, à battre les résultats publiés à ce jour.

6.1.4 La publication des résultats de cet ouvrage

Par conformisme, l'analyse des résultats publiés dans cet ouvrage sera essentiellement basée sur le EER. Toutefois, une analyse plus appropriée sera présentée afin d'améliorer la comparaison de certains systèmes qui performent similairement en regard de leur EER mais qui se distinguent sous d'autres aspects importants.

6.2 Le corpus de données SPIDRE

Le corpus de données SPIDRE ("SPeaker IDentification REsearch") est un sous ensemble de SWITCHBOARD et fait partie d'un nombre restreint de corpora de données standards utilisés pour l'identification et la vérification du locuteur.

Les locuteurs sont divisés en deux groupes distincts. Le premier est formé de 45 personnes (27 hommes et 18 femmes) pour lesquels quatre fichiers audio ont été enregistrés. De ces quatre fichiers, deux proviennent du même numéro de téléphone. Ces fichiers seront appelés *pairés*. Les deux autres fichiers proviennent de deux autres numéros de téléphone. Ces fichiers seront appellés *isolés*. Donc, trois numéros de téléphone ont été utilisés pour l'enregistrement des quatre conversations. Ces 45 locuteurs font partie du registre (en anglais "target speakers"). Nous les appellerons

locuteurs *internes*. Le second groupe est formé de 160 locuteurs ayant prononcé un total de 200 conversations. Ces locuteurs ne peuvent faire partie du registre et sont utilisés strictement comme imposteurs (en anglais "nontarget speakers"). Ils sont donc utiles pour l'identification avec rejet et la vérification du locuteur. Nous les appellerons locuteurs *externes*.

6.3 Les schèmes

Les expérimentations en vérification de locuteur se classent généralement en deux situations: la situation d'*appariement* (en anglais "matched conditions") et la situation de *disparité* (en anglais "mismatched conditions"). La situation d'appariement, telle que généralement définie, requiert que les fichiers de test et d'entraînement proviennent d'un seul et même numéro de téléphone pour un locuteur particulier. Il est à noter que cette situation n'implique pas nécessairement l'utilisation d'un type semblable de combiné puisque de nos jours, une maison compte souvent plus d'un appareil téléphonique. La situation de disparité, quant à elle, exige que pour chaque locuteur, le fichier test provienne d'un numéro de téléphone différent de celui d'où provient le fichier d'entraînement. Soulignons encore qu'il est tout de même probable que deux combinés d'un même type aient été utilisés par le biais de deux lignes téléphoniques différentes.

Quatre schèmes ont été développés pour les expérimentations présentées dans ce chapitre. Pour le premier schème, les fichiers pairés ont été utilisés pour l'entraînement de deux modèles différents pour chaque locuteur. Les deux fichiers isolés

ont été, quant à eux, utilisés comme fichiers test. De plus, les fichiers isolés des 44 autres locuteurs internes ont été utilisés comme fichiers test d'imposteurs. Donc, chacun des 90 modèles développés s'est vu confronté à deux fichiers test provenant du bon locuteur et 88 fichiers d'imposteurs. Ce schème génère donc en tout 8100 *confrontations*. Idéalement, 90 de ces confrontations devraient être acceptées par le système et 8010 devraient être rejetées. Parmi les 4 schèmes développés, celui-ci représente la tâche la plus difficile puisque malgré l'utilisation de 2 fichiers d'entraînement, le système ne peut généraliser l'information acquise sur le locuteur puisque ces deux fichiers sont pairés, i.e., proviennent du même numéro de téléphone. De plus les deux fichiers test isolés proviennent de deux autres numéros. Le code utilisé pour identifier ce schème sera "9090".

Le second schème fut choisi puisqu'il avait été préalablement utilisé au sein de notre groupe et permettait ainsi certaines comparaisons initiales de performance. Pour chaque locuteur, un seul fichier isolé fut utilisé comme test. Les trois autres (le second fichier isolé et les deux fichiers pairés) furent utilisés comme fichiers d'entraînement. Les 44 autres locuteurs fournirent un fichier isolé, utilisé comme fichier d'imposteur. Trois modèles furent donc développés pour chaque locuteur, 135 en tout. Chaque modèle fut confronté à un bon fichier et 44 fichiers d'imposteurs. Globalement, 6075 confrontations furent générées, 135 (3 modèles $\times$ 1 test $\times$ 45 locuteurs) provenant de locuteurs tentant d'accéder légitimement à leurs propres ressources et 5940 (3 modèles $\times$ 44 tests $\times$ 45 locuteurs) provenant d'imposteurs. Le code utilisé pour identifier ce schème sera "45135".

Le troisième schème correspond à la situation inverse du précédent. Le fichier

isolé, utilisé comme test dans le schème précédent fut utilisé comme fichier d'entraînement pour le schème présent. Ainsi, chacun des 45 modèles fut confronté à trois bons fichiers et 132 fichiers d'imposteurs. Encore une fois, 6075 confrontations furent générées, 135 (1 modèle $\times$ 3 tests $\times$ 45 locuteurs) provenaient de bons locuteurs et 5940 (1 modèle $\times$ 132 tests $\times$ 45 locuteurs) provenaient d'imposteurs. Le code utilisé pour identifier ce schème sera "13545".

Contrairement au trois précédents, le dernier schème reproduit une situation d'appariement. Pour chaque locuteur, un fichier pairé fut utilisé pour l'entraînement et le second pour le test. Chaque modèle fut confronté à un bon fichier et 44 imposteurs. Il y eu donc 2025 confrontations, 45 bonnes et 1980 provenant d'imposteurs. Le code utilisé pour identifier ce schème sera "4545".

6.4 Les paramètres fixes

Les chercheurs utilisent généralement de 7 à 12 coefficients cepstraux statiques. Le logiciel HTK a été implanté au CRIM en fonction d'une utilisation de 12 MFCC statiques. Au total, 12 MFCC statiques, 12 MFCC dynamiques, 1 coefficient d'énergie et 1 coefficient de variation d'énergie ont été utilisés. Chaque trame de 10ms était donc représentée à l'aide de 26 paramètres.

Au sein de la communauté scientifique, les résultats en reconnaissance du locuteur sont généralement présentés pour des durées d'enregistrement des fichiers tests de 30, 10 et 3 secondes. Les fichiers tests utilisés ont donc été tronqués afin de respecter

cette convention. Une exception, toutefois: afin de déterminer le nombre approprié de gaussiennes à utiliser pour les expériences subséquentes, les fichiers tests n'ont pas été tronqués, par but de simplicité.

6.5 Le nombre de gaussiennes pour la modélisation

Tadj [48] a analysé l'effet de certaines variations du nombre de gaussiennes utilisées pour la modélisation des locuteurs sur la tâche d'identification (sans rejet). De façon empirique, une "règle du pouce" pour le calcul du nombre optimal N de gaussiennes à utiliser en fonction de la durée T (en secondes) du fichier d'entraînement a été établie:

$$N = 0,36 \cdot T + 4 \tag{6.11}$$

La durée moyenne des fichiers de SPIDRE pour les locuteurs du registre est de 66,4 secondes. Donc, selon l'équation précédente, le nombre optimal moyen de gaussiennes était de 27 (minimum:10; maximum:59). L'expérience fut donc réalisée à l'aide d'un nombre variable de gaussiennes pour la modélisation de chacun des locuteurs, dépendamment de la durée du fichier d'entraînement afin de déterminer les paramètres (0,36 et 4) de l'équation linéaire. La performance d'identification fut de 78% pour une durée de 30 secondes des fichiers tests.

Par la suite, la même expérience fut reconduite avec un nombre fixe de gaussiennes. Le tableau 6.1 donne les résultats obtenus pour différents nombres de mixtures. Le but était de vérifier si l'utilisation d'un nombre variable de gaussiennes, adapté au

Tableau 6.1: Pourcentage d'identification pour différents nombres de gaussiennes fixés *a priori* (tiré de Tadj [48])

Nombre de gaussiennes	10	16	27	32	59
Identification	67%	67%	73%	71%	69%

fichier d'entraînement du locuteur permettait d'obtenir de meilleurs résultats.

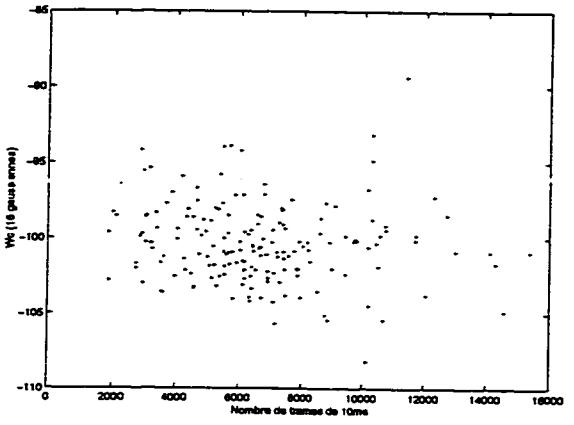

Figure 6.3: Graphique de W_c, calculé avec 16 gaussiennes, en fonction de T, le nombre de trames de 10ms (ou durée).

La conclusion fut donc qu'un nombre variable de gaussiennes, déterminé selon l'équation 6.11 était souhaitable. La question se pose, toutefois, à savoir si la durée du fichier d'entraînement est réellement représentative du nombre optimal de gaussiennes à utiliser pour la modélisation.

Les figures 6.3, 6.4 et 6.5 illustrent, en fonction de la durée du fichier d'en-

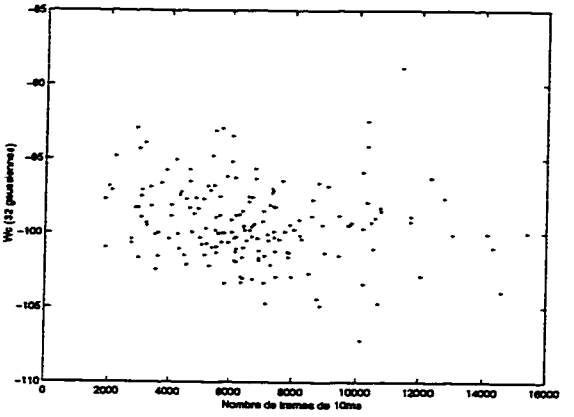

Figure 6.4: Graphique de W_c, calculé avec 32 gaussiennes, en fonction de T, le nombre de trames de 10ms (ou durée).

traînement, la valeur de W_c, calculée à l'aide de 16, 32 et 64 gaussiennes, respectivement. La valeur du coefficient de corrélation est de $\rho = -0,10$, $\rho = -0,16$ et $\rho = -0,25$, pour 16, 32 et 64 gaussiennes, tout aussi respectivement.

Les trois graphiques sont très similaires et les points semblent, dans les trois cas, se répartir relativement uniformément. Par contre, l'augmentation (en termes absolus) de la corrélation entre la durée d'entraînement et la valeur de W_c au fur et à mesure que le nombre de gaussiennes utilisées augmente nous force à jeter un second coup d'oeil: ce sont les fichiers de courte durée qui bénéficient le plus de l'augmentation de la complexité du modèle.

La figure 6.6 illustre la différence entre W_c calculé à l'aide de 64 gaussiennes et W_c calculé à l'aide de 16 gaussiennes en fonction de la durée T du fichier d'entraînement. Toutes les valeurs sont positives, ce qui signifie que l'augmentation de la complexité

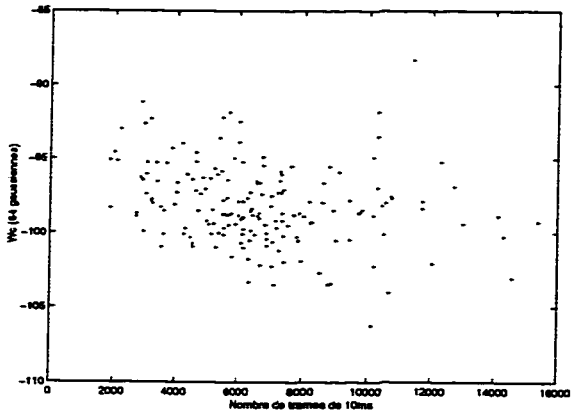

Figure 6.5: Graphique de W_c, calculé avec 64 gaussiennes, en fonction de T, le nombre de trames de 10ms (ou durée).

du modèle a permis une réduction de l'erreur d'entraînement. D'un simple coup d'oeil, on peut voir que les fichiers de courte durée sont les plus grands bénéficiaires de cette augmentation de la *capacité* du modèle. Plus précisément, la corrélation entre la différence et la durée est de $\rho = -0,65$. Encore plus forte est la corrélation entre la différence et la transformée logarithmique de la durée: $\rho = -0,74$. La figure 6.6 illustre la courbe de régression linéaire simple sur cette transformée.

Aucun test n'a été effectué pour observer l'évolution de l'erreur de généralisation. En fait, peu de chercheurs du domaine de la reconnaissance du locuteur s'y attardent, malgré l'intérêt que cette analyse peut comporter et qui fera probablement l'objet de recherches futures.

La figure 6.7 permet d'émettre certaines hypothèses pour l'interprétation des résultats de la figure 6.6. Selon la théorie des "learning machines" [27], l'augmentation

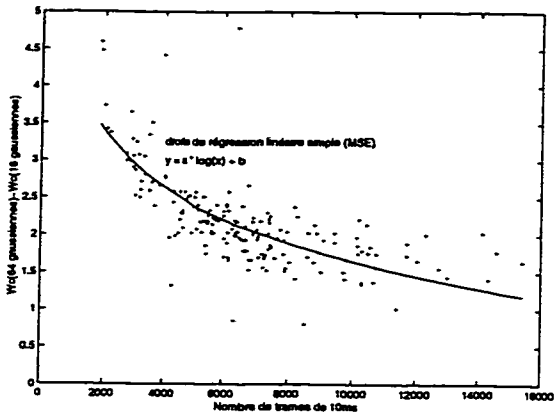

Figure 6.6: Réduction de l'erreur d'entraînement (lorsque 64 gaussiennes sont utilisées pour la modélisation au lieu de 16), en fonction de T, le nombre de trames de 10ms (ou durée).

de la capacité d'une machine, en particulier par le biais de l'augmentation du nombre de paramètres, permet d'obtenir, lorsque le nombre de données d'entraînement converge vers l'infini, non seulement une erreur d'entraînement inférieure, mais aussi et surtout, une erreur de généralisation inférieure. En fait, ces deux erreurs convergent vers une seul et même valeur. À l'autre extrême, alors que le nombre de données d'entraînement est minime, le modèle peut arriver à surentraîner les données. Sur la figure 6.7, pour une machine de capacité h, si le nombre de données d'entraînement est inférieur à N_h, l'erreur d'entraînement est nulle. Toutefois, l'erreur de généralisation est énorme. La question essentielle tourne autour de l'estimation de $N^*_{h,h'}$, le nombre de données d'entraînement nécessaires pour justifier le passage d'une capacité h à une capacité h'. Ce nombre correspond au point de croisement des courbes d'erreur de généralisation et ne peut être déduit des courbes d'erreur d'entraînement, d'où l'intérêt clair d'une analyse sur la base de l'erreur de généralisation.

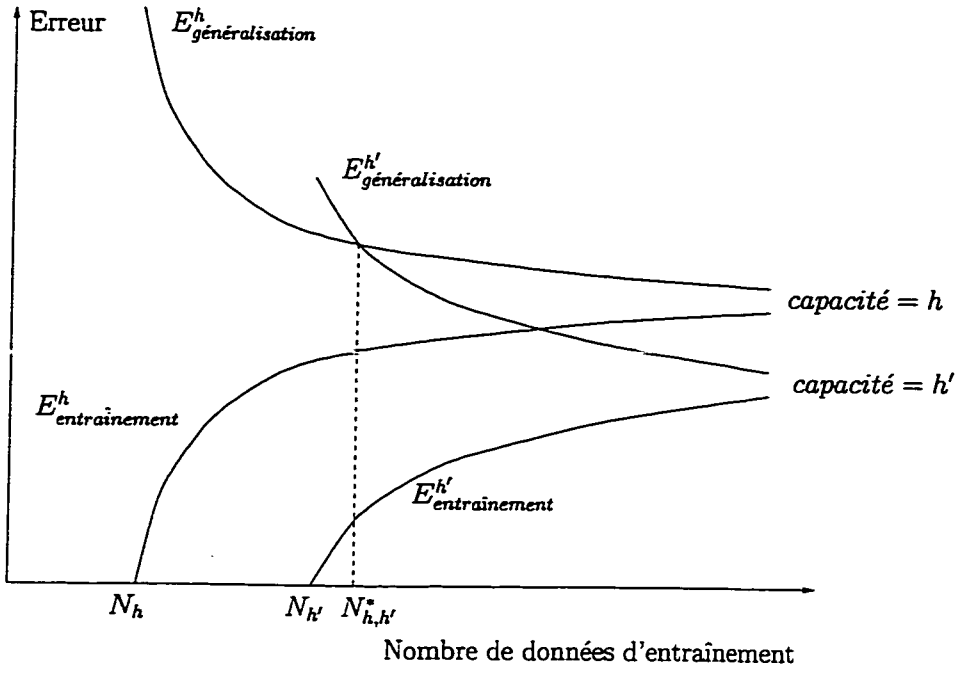

Figure 6.7: Interprétation de l'amélioration de l'erreur d'entraînement, à l'aide de la théorie des "learning machines", pour une augmentation de la complexité (ou capacité) du modèle de 16 à 64 gaussiennes.

Pour le cas particulier des modèles de mixtures de gaussiennes avec une matrice de covariance diagonale, le nombre N_h est légèrement supérieur au nombre de mixtures utilisées. Donc, le nombre (T) de données utilisées est amplement suffisant (i.e., $T \gg N_h$) pour éviter le surentraînement. Le fait que les fichiers de durée inférieure bénéficient le plus de l'augmentation de la capacité du modèle, correspond au rapprochement des deux courbes d'erreur d'entraînement: étant donné une variation fixe de capacité, la différence entre les erreurs d'entraînement est plus grande si le nombre de données est inférieur. La question reste toutefois en suspens à savoir si $N^*_{h,h'} < 2000 \approx T_{min}$, où h et h' sont les capacités des modèles à 16 et 64 gaussiennes respectivement. Cette situation justifierait le passage d'un modèle à 16 gaussiennes vers un modèle à 64 gaussiennes pour tous les locuteurs.

Donc, l'idée de choisir une architecture adaptée au fichier d'entraînement du locuteur pour le modéliser semble prometteuse. Par contre, le choix de la durée du fichier d'entraînement comme seule variable déterminante de ce nombre n'est sans doute pas la plus heureuse. L'avantage marqué de cette méthode est l'extrême simplicité de calcul du nombre (présumément) approprié du nombre de gaussiennes. L'utilisation du calcul de la muselière, W_c, plutôt que la durée fera l'objet (encore une fois) de recherches sous peu.

Les figures 6.8, 6.9 et 6.10 montrent que les courbes *ROC* sont très similaires, que le nombre de gaussiennes utilisées soit de 16, 32 ou 64. Cette observation est vraie pour les trois schèmes 9090, 45135 et 4545. La table 6.2 donne les valeurs du EER Les fichiers test ont été utilisés en entier pour obtenir ces valeurs. Encore une fois,

le nombre de gaussiennes utilisées semble n'avoir qu'un effet marginal sur les résultats.

Les résultats obtenus par Tadj, indique que le nombre optimal de gaussiennes est de 27. Nous nous sommes toutefois limité à l'expérimentation de modèles développés à l'aide de 16, 32 et 64 gaussiennes, i.e., les puissances de 2 rapprochées de 27. Sur la base de ces observations initiales, il semble donc que la modélisation à l'aide 16 gaussiennes offre le meilleur rapport qualité/complexité. Les résultats présentés dans les sections suivantes sont donc tous basés sur des modèles développés à l'aide de 16 gaussiennes.

Tableau 6.2: Calcul du EER. Modèles à 16, 32 et 64 gaussiennes. Schèmes: 9090, 45135 et 4545. Les fichiers d'entraînement n'ont pas été tronqués.

	Gaussiennes		
Schème	16	32	64
9090	35.5%	34.4%	34.0%
45135	34.1%	32.6%	32.6%
4545	23.2%	22.2%	20.9%

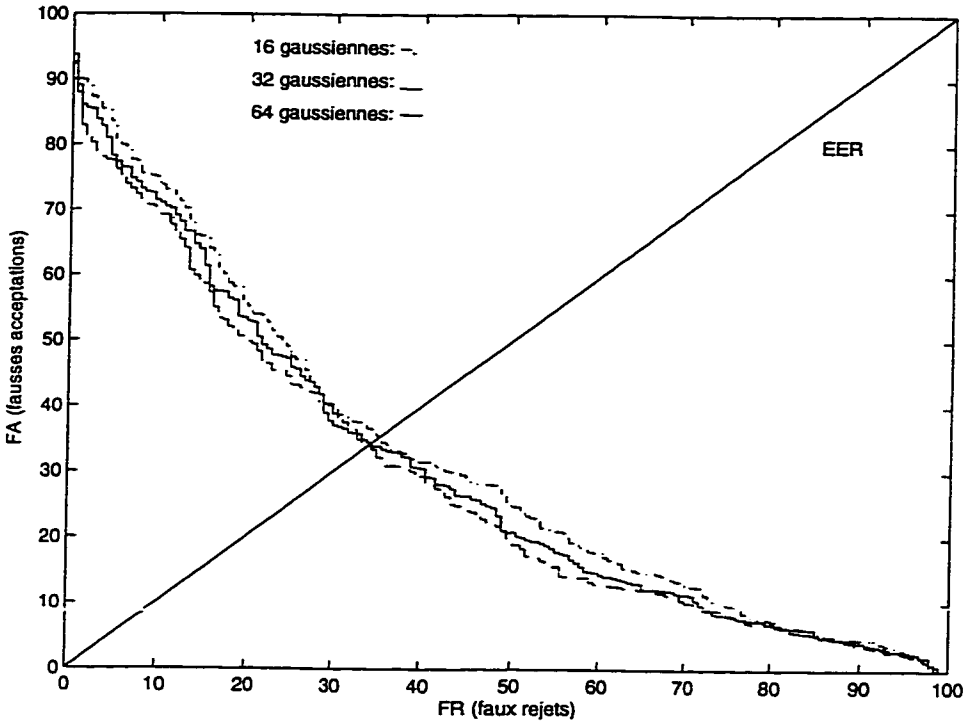

Figure 6.8: Courbe ROC pour modèles à 16, 32 et 64 gaussiennes. Schème 9090, utilisation du fichier d'entraînement en entier, aucune normalisation.

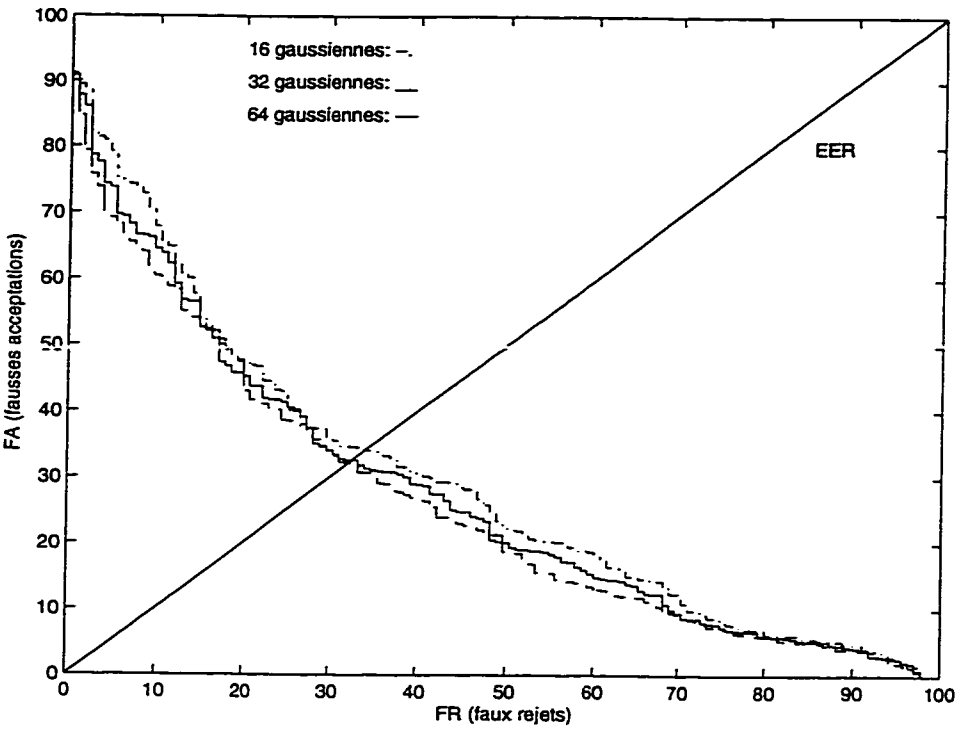

Figure 6.9: Courbe ROC pour modèles à 16, 32 et 64 gaussiennes. Schème 45135, utilisation du fichier d'entraînement en entier, aucune normalisation.

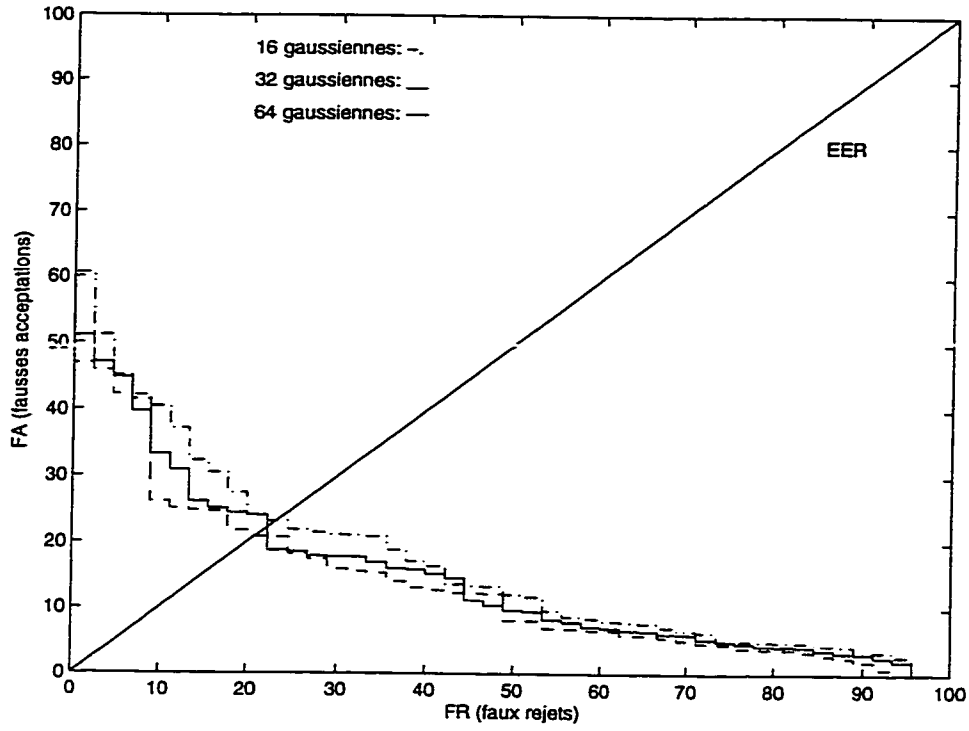

Figure 6.10: Courbe ROC pour modèles à 16, 32 et 64 gaussiennes. Schème 4545, utilisation du fichier d'entraînement en entier, aucune normalisation.

6.6 La muselière et l'impact

Le tableau 6.3 donne la valeur du EER calculé pour les 4 schèmes décrits à la section 6.3, les 3 durées standard pour les fichiers test (30, 10 et 3 secondes) et 3 techniques de normalisation (aucune muselière et impact). Donc, 36 valeurs qui correspondent aux 36 courbes ROC illustrées sur les figures 6.11 à 6.22 inclusivement.

Chacune de ces 12 figures démontre clairement l'amélioration apportée par la muselière par rapport au calcul de vraisemblance de base. L'apport de l'impact est similaire à celui de la muselière.

Les résultats obtenus par les deux techniques sont difficilement dissociables. Une seule observation: il semble qu'à 3 secondes de test, la muselière performe mieux que l'impact. Cette différence doit être attribuée soit à l'armature ($P(\mathbf{O}_d|\lambda_d)$), soit à la contre-attaque ($P(\mathbf{O}_d|\lambda_c)$). Or, le calcul de l'armature ne dépend pas de la durée du fichier test et si la différence en question lui était due, alors les résultats à 10 et 30 secondes de test seraient eux aussi affectés, ce qui n'est pas le cas. Donc, par élimination, la contre-attaque doit être pointée du doigt.

Les schèmes reproduisant des situations de disparité obtiennent des résultats très similaires. À l'exception du résultat obtenu avec 3 secondes de test et sans normalisation, le schème 9090 obtient les pires résultats de tous, tel que prévu (voir section 6.3). Les deux schèmes 45135 et 13545 sont similaires en terme de difficulté.

Les résultats obtenus en situation d'appariement (schème 4545) sont nettement meilleurs. De plus, l'effet de la muselière et de l'impact est encore plus prononcé,

en particulier pour une durée de 30 secondes de test. Ainsi, dans la mesure où le modèle de normalisation "UBM" est utilisé afin de modéliser l'environnement acoustique d'entraînement, on peut présumer que les techniques de la muselière (ou de l'impact) et du "UBM" seront complémentaires et non redondantes.

Tableau 6.3: Calcul du EER. Schèmes: 9090, 45135, 13545 et 4545. Durée de test: 30, 10 et 3 secondes. Normalisation: aucune, muselière et impact.

Schème	Temps (secondes)	Sans Normalisation	Muselière	Impact
9090	30	35.8%	20.0%	20.6%
	10	35.9%	20.9%	23.9%
	3	39.4%	27.0%	28.3%
45135	30	34.8%	17.8%	15.9%
	10	35.6%	20.0%	20.7%
	3	40.0%	23.0%	27.2%
13545	30	33.8%	18.9%	17.8%
	10	35.6%	19.3%	18.5%
	3	38.5%	25.8%	25.2%
4545	30	25.5%	4.7%	4.4%
	10	28.1%	6.7%	6.7%
	3	32.9%	11.6%	15.3%

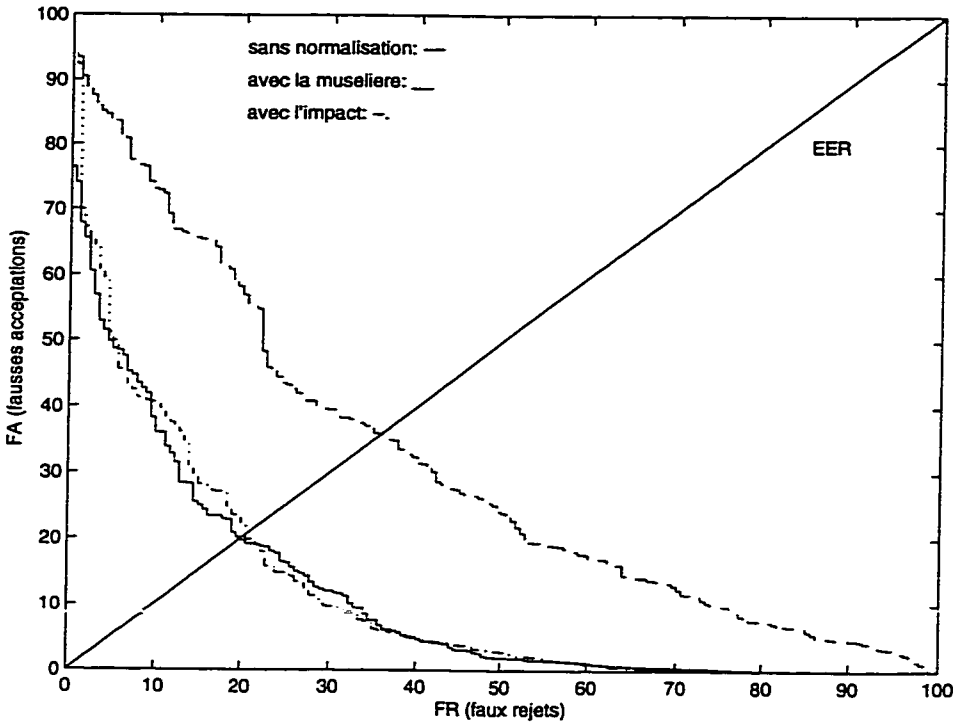

Figure 6.11: Courbes ROC. Schème 9090. Modèle à 16 gaussiennes. Fichier test tronqué à 30 secondes.

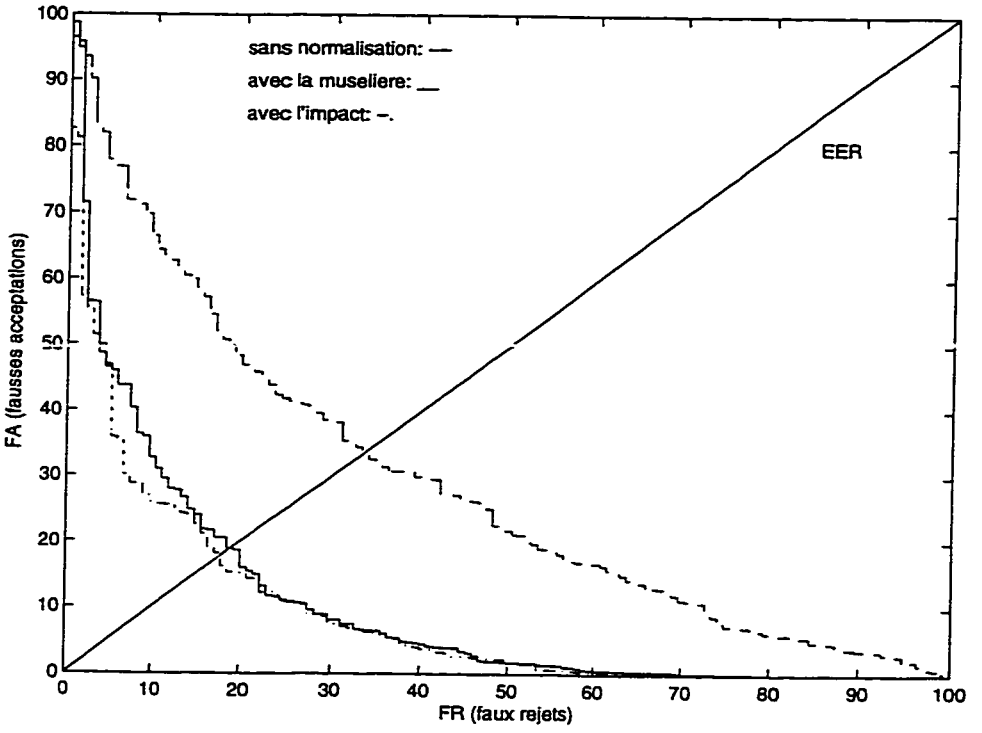

Figure 6.12: Courbes ROC. Schème 45135. Modèle à 16 gaussiennes. Fichier test tronqué à 30 secondes.

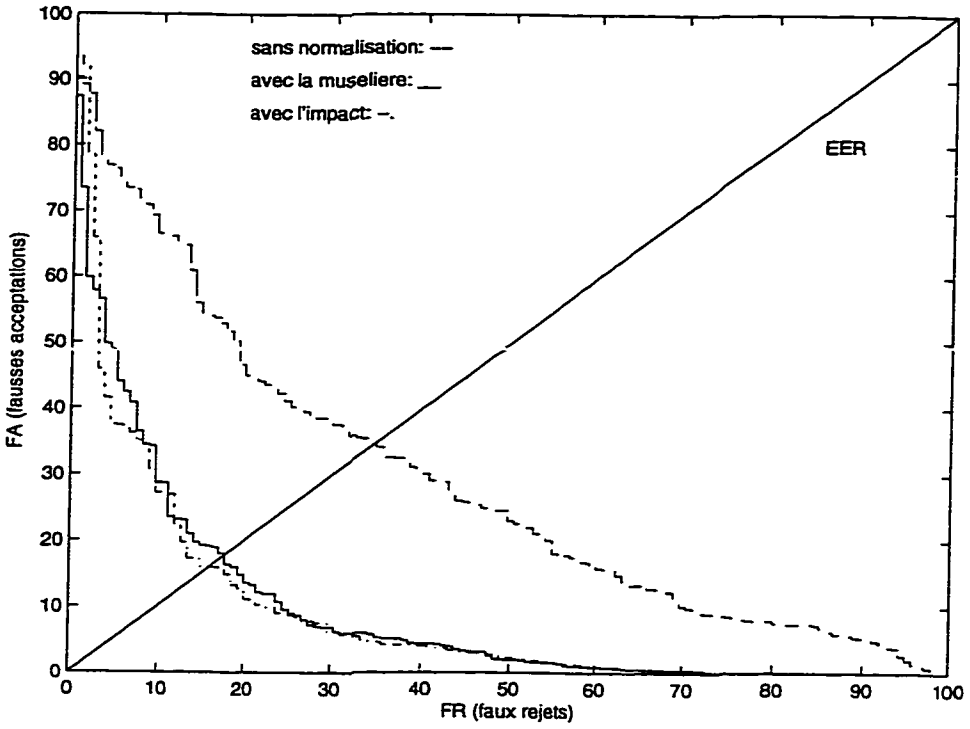

Figure 6.13: Courbes ROC. Schème 13545. Modèle à 16 gaussiennes. Fichier test tronqué à 30 secondes.

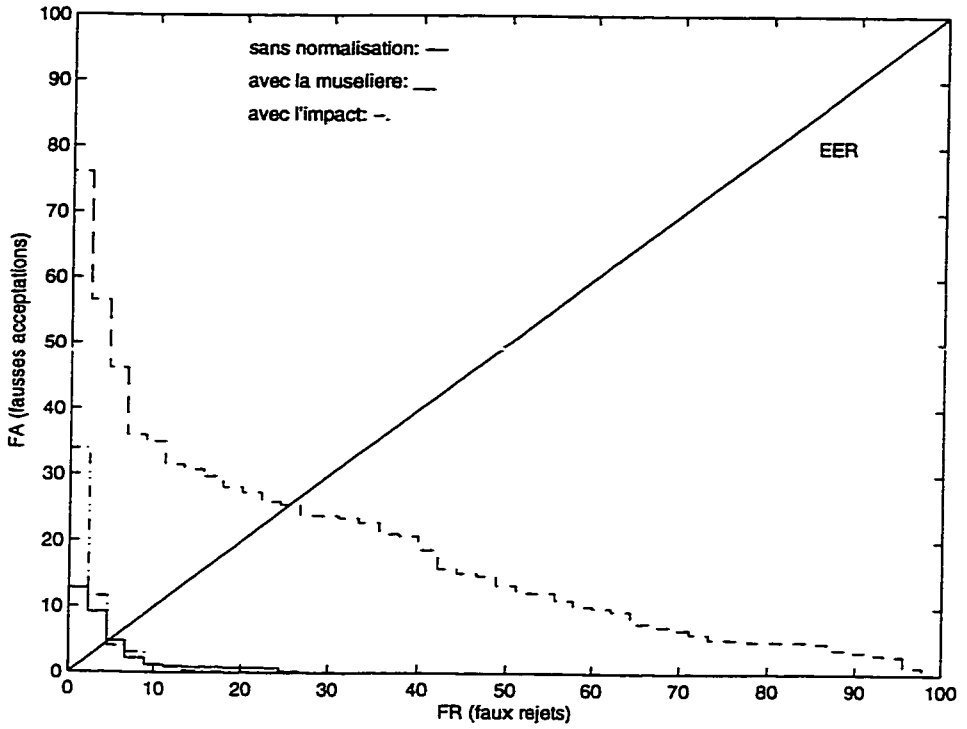

Figure 6.14: Courbes ROC. Schème 4545. Modèle à 16 gaussiennes. Fichier test tronqué à 30 secondes.

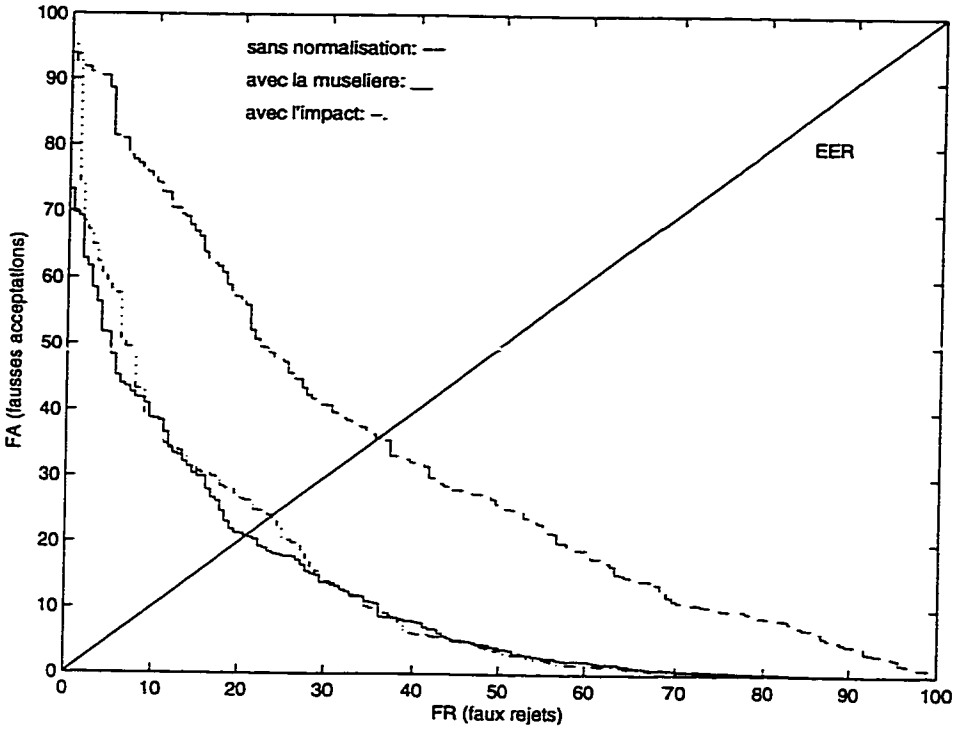

Figure 6.15: Courbes ROC. Schème 9090. Modèle à 16 gaussiennes. Fichier test tronqué à 10 secondes.

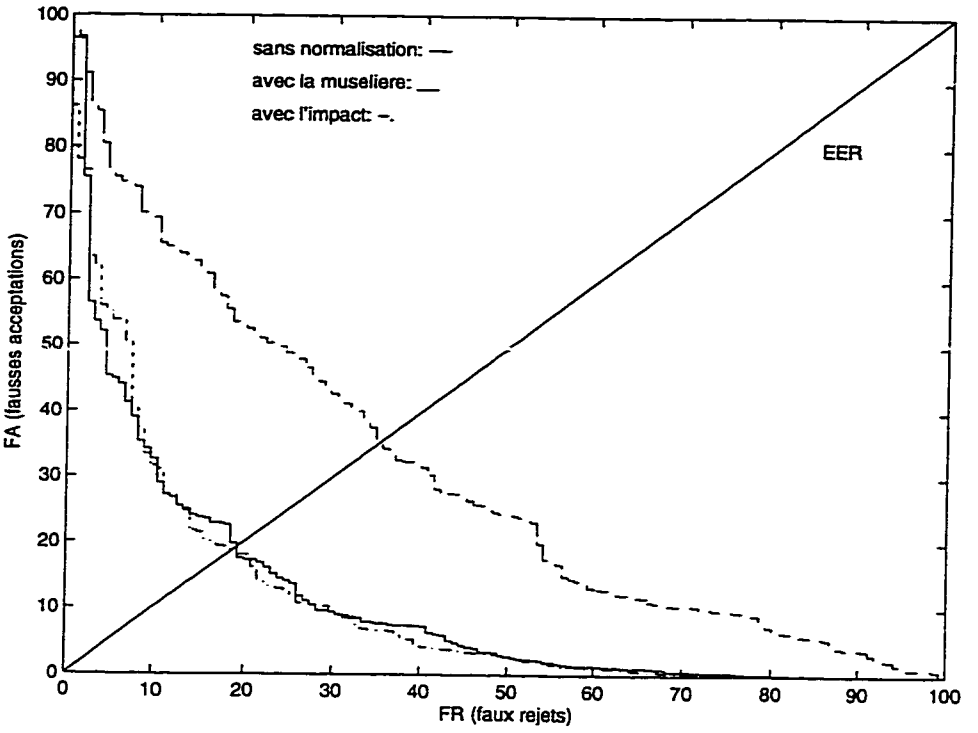

Figure 6.16: Courbes ROC. Schème 45135. Modèle à 16 gaussiennes. Fichier test tronqué à 10 secondes.

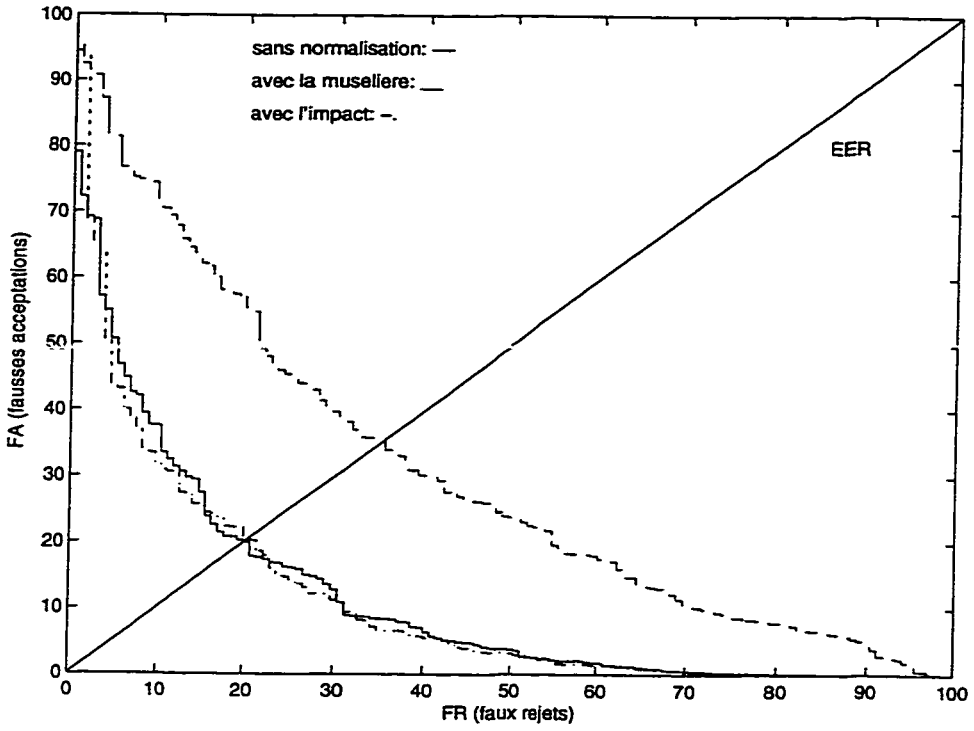

Figure 6.17: Courbes ROC. Schème 13545. Modèle à 16 gaussiennes. Fichier test tronqué à 10 secondes.

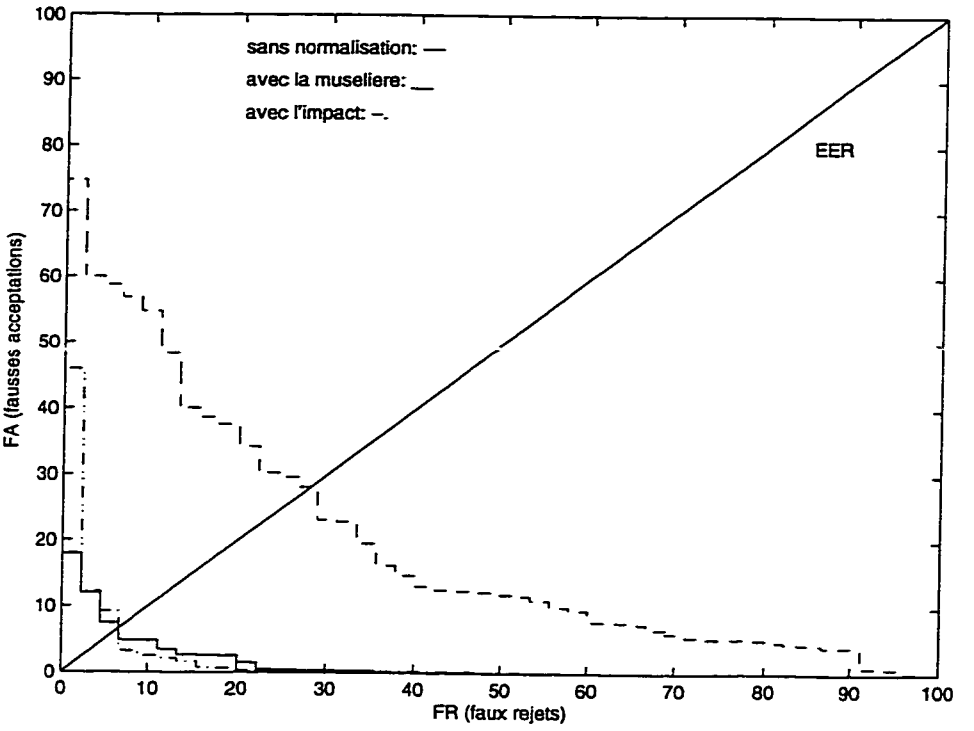

Figure 6.18: Courbes ROC. Schème 4545. Modèle à 16 gaussiennes. Fichier test tronqué à 10 secondes.

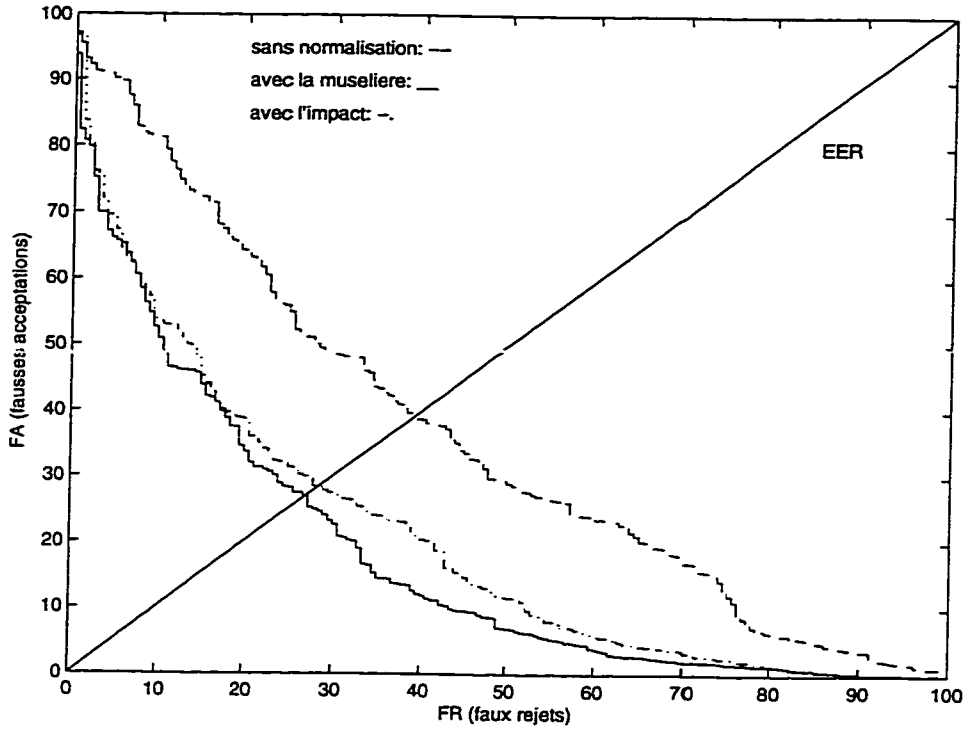

Figure 6.19: Courbes ROC. Schème 9090. Modèle à 16 gaussiennes. Fichier test tronqué à 03 secondes.

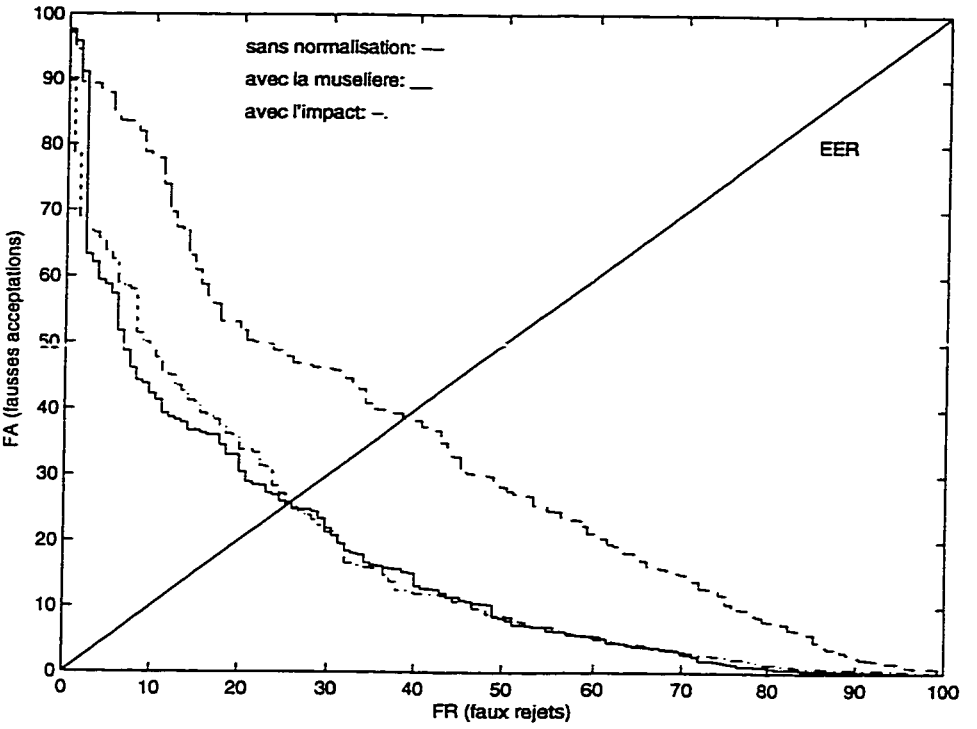

Figure 6.20: Courbes ROC. Schème 45135. Modèle à 16 gaussiennes. Fichier test tronqué à 03 secondes.

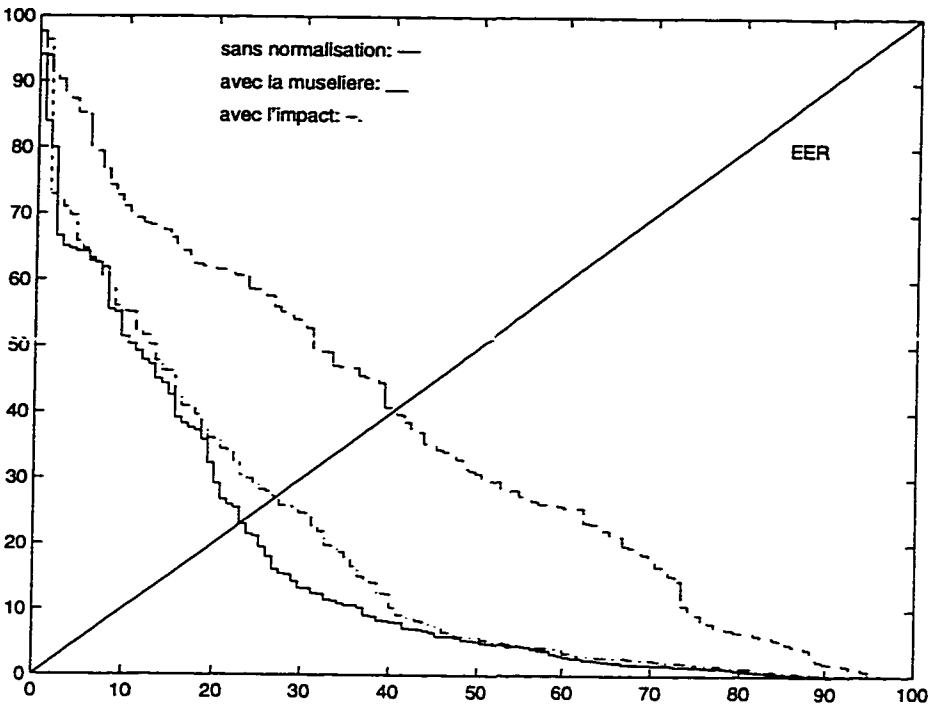

Figure 6.21: Courbes ROC. Schème 13545. Modèle à 16 gaussiennes. Fichier test tronqué à 03 secondes.

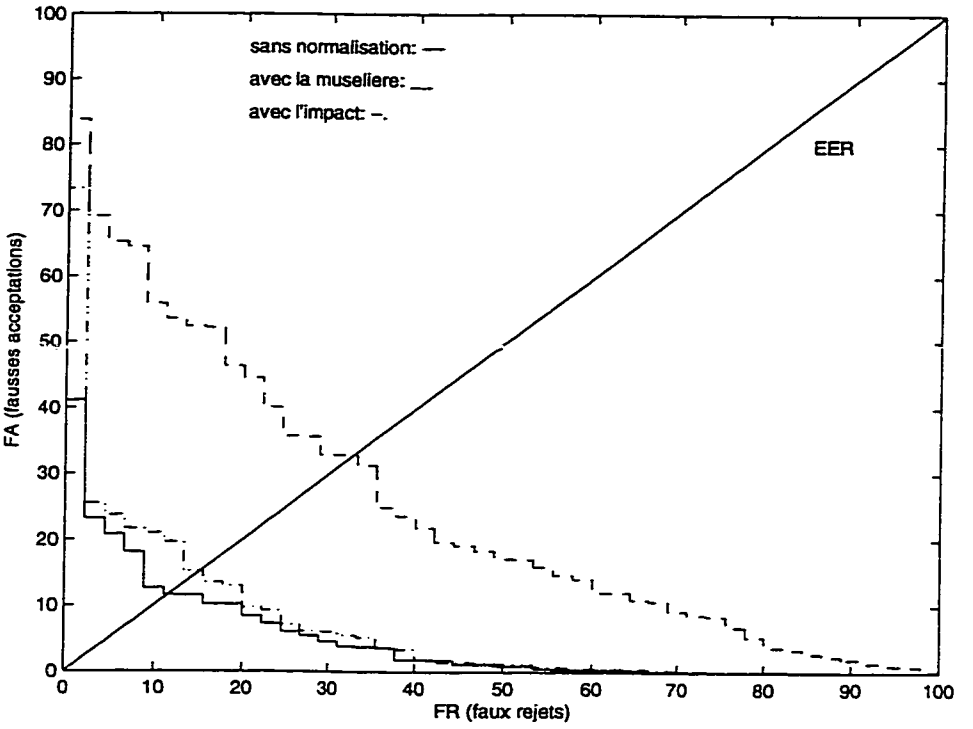

Figure 6.22: Courbes ROC. Schème 4545. Modèle à 16 gaussiennes. Fichier test tronqué à 03 secondes.

6.7 La cohorte et la muselière

La cohorte a été calculée en utilisant la moyenne des résultats des fichiers tests confrontés à 15 locuteurs choisis de façon aléatoire parmi les locuteurs internes. Chaque locuteur du registre disposait de sa propre cohorte. Bien qu'il ait été souligné que les fichiers tests d'imposteurs ne pouvaient provenir strictement de locuteurs internes dont certains enregistrements avaient servi au développement d'un modèle de normalisation tel le UBM ou la cohorte, cette règle n'a pas été respectée, et les résultats obtenus par la cohorte sont donc surestimés.

Toutefois, les techniques de la muselière et de l'impact sont développées de façon indépendante des autres locuteurs internes et l'utilisation d'imposteurs internes n'induit pas de biais sur les résultats. Ainsi, l'interprétation des résultats obtenus, qui tente de démontrer la supériorité de la muselière et de l'impact dans certains cas, est conservatrice.

Puisque les schèmes de disparité obtiennent des résultats similaires, seul le schème 9090, le plus difficile, a été retenu. Le schème 4545 a été aussi retenu afin de pouvoir comparer les techniques de normalisation en situation d'appariement.

Le tableau 6.4 présente le EER pour les schèmes 9090 et 4545 et pour des durées test de 30, 10 et 3 secondes. Les résultats montrent la supériorité de la muselière et de l'impact pour les durées de 30 et 10 secondes en situation d'appariement. Avec 3 secondes de test et donc, selon les expériences réalisées, avec 300 observations, la borne sur l'écart entre l'erreur de généralisation et l'erreur d'entraînement est

probablement trop élevée. Ainsi, l'estimateur utilisé, W_c, n'est pas suffisamment fiable. Ceci explique la contre-performance de la muselière et de l'impact lorsque la durée test est de 3 secondes. Le calcul de la muselière utilise la confrontation entre des données et un modèle dont les données proviennent nécessairement d'un même combiné (ce sont les mêmes données). Donc, la muselière ne permet pas la normalisation de façon indépendante du combiné, contrairement à la cohorte, puisque cette dernière utilise les résultats obtenus par le fichier test sur 15 locuteurs du registre.

Tableau 6.4: Calcul du EER. Schèmes: 9090 et 4545. Durée de test: 30, 10 et 3 secondes. Normalisation: cohorte, muselière et impact.

Schème	Temps (secondes)	Cohorte	Muselière	Impact
9090	30	17.9%	20.0%	20.6%
	10	18.5%	20.9%	23.9%
	3	20.0%	27.0%	28.3%
4545	30	6.0%	4.7%	4.4%
	10	7.3%	6.7%	6.7%
	3	9.8%	11.6%	15.3%

L'observation des figures 6.23 à 6.28 permet d'ajouter une remarque importante: pour l'ensemble des expériences, la muselière réussit à battre la cohorte lorsque le seuil d'acceptation est élevé, i.e., lorsque le critère d'optimisation du système correspond à celui d'une application à haute sécurité.

Les courbes présentées correspondent à la courbe ROC représentée sur une échelle logarithmique-logarithmique, afin de mieux percevoir les détails du comportement du système lorsque les taux d'erreur sont faibles. Ces courbes sont appelées courbes *DET* (de l'anglais "Detection Error Tradeoff").

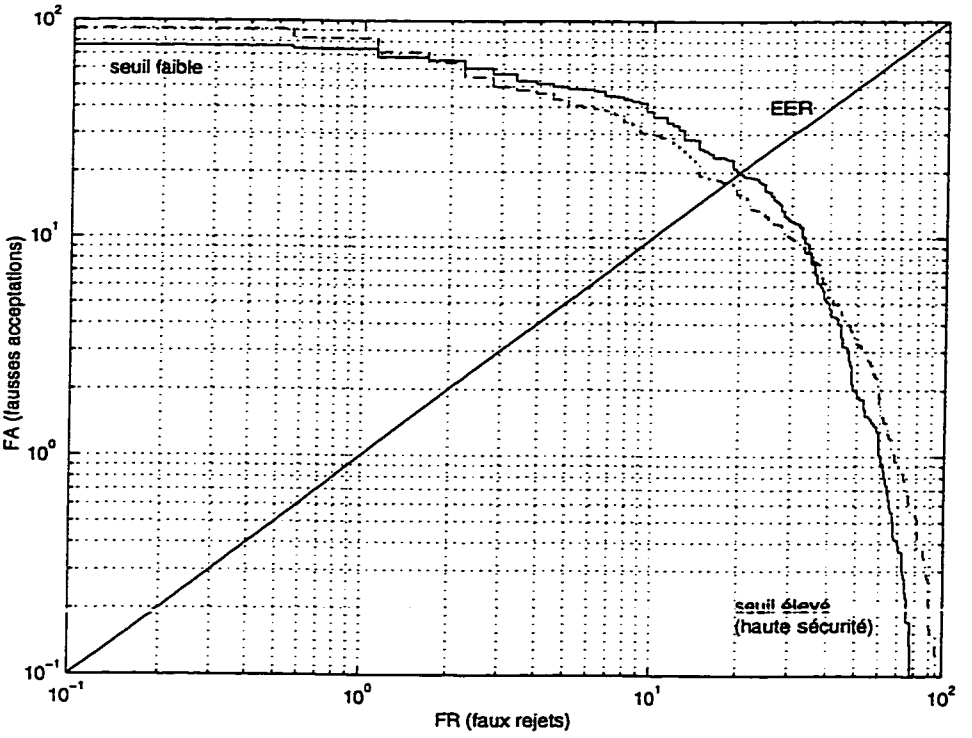

Figure 6.23: Courbes DET. Schème 9090. Modèle à 16 gaussiennes. Fichier test tronqué à 30 secondes. Normalisation: muselière (trait plein: -) et cohorte (trait pointillé: -.).

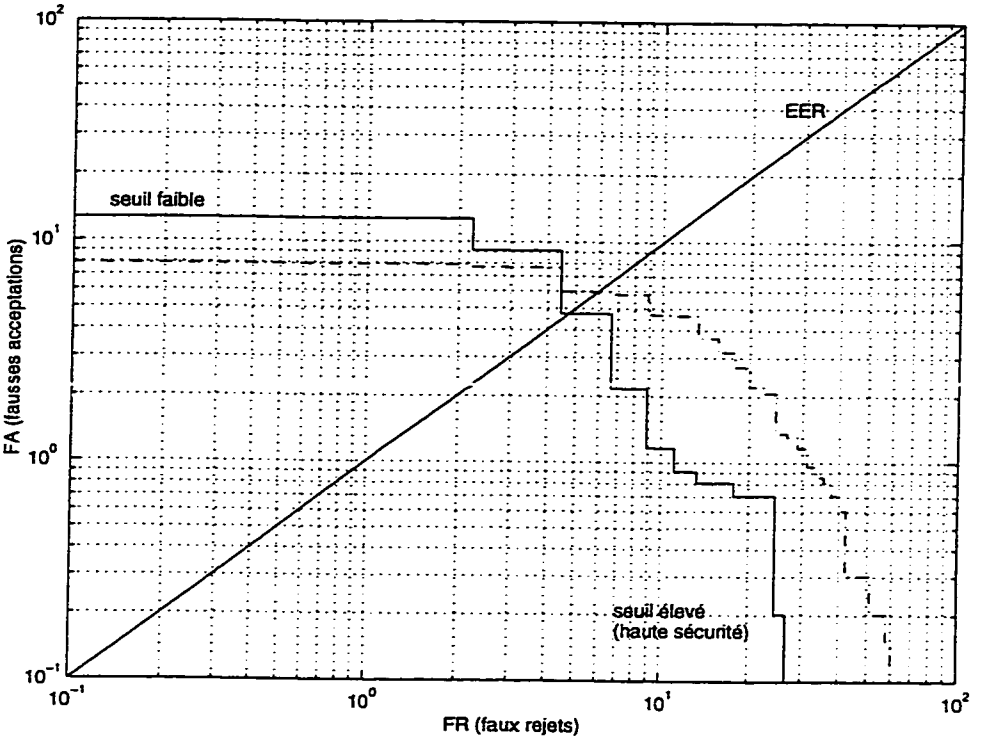

Figure 6.24: Courbes DET. Schème 4545. Modèle à 16 gaussiennes. Fichier test tronqué à 30 secondes. Normalisation: muselière (trait plein: -) et cohorte (trait pointillé: -.).

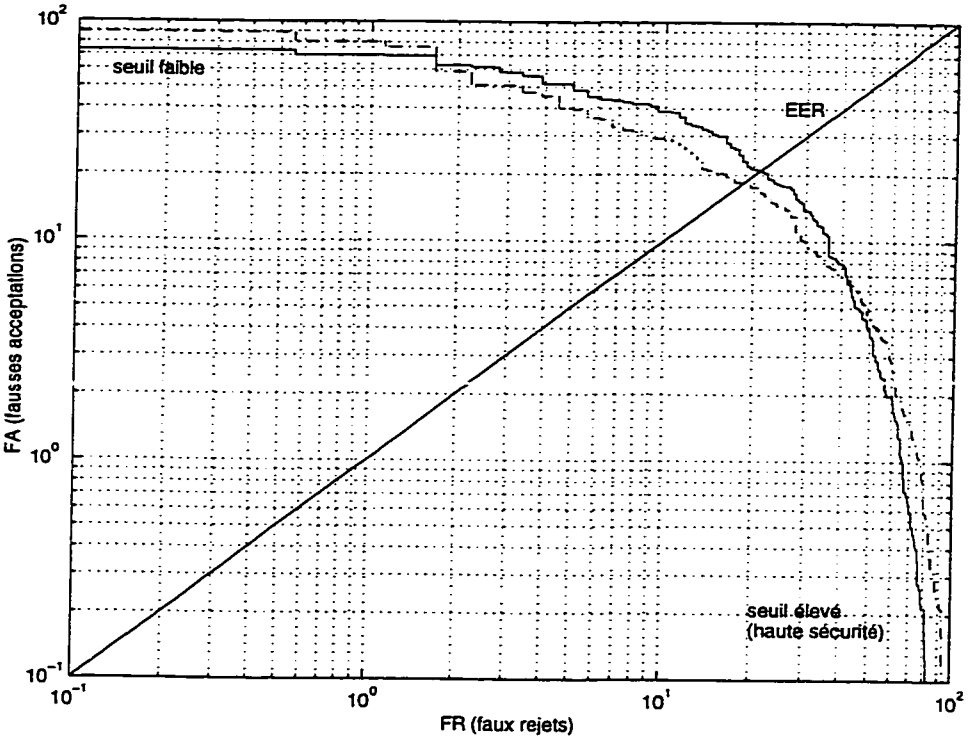

Figure 6.25: Courbes DET. Schème 9090. Modèle à 16 gaussiennes. Fichier test tronqué à 10 secondes. Normalisation: muselière (trait plein: -) et cohorte (trait pointillé: -.).

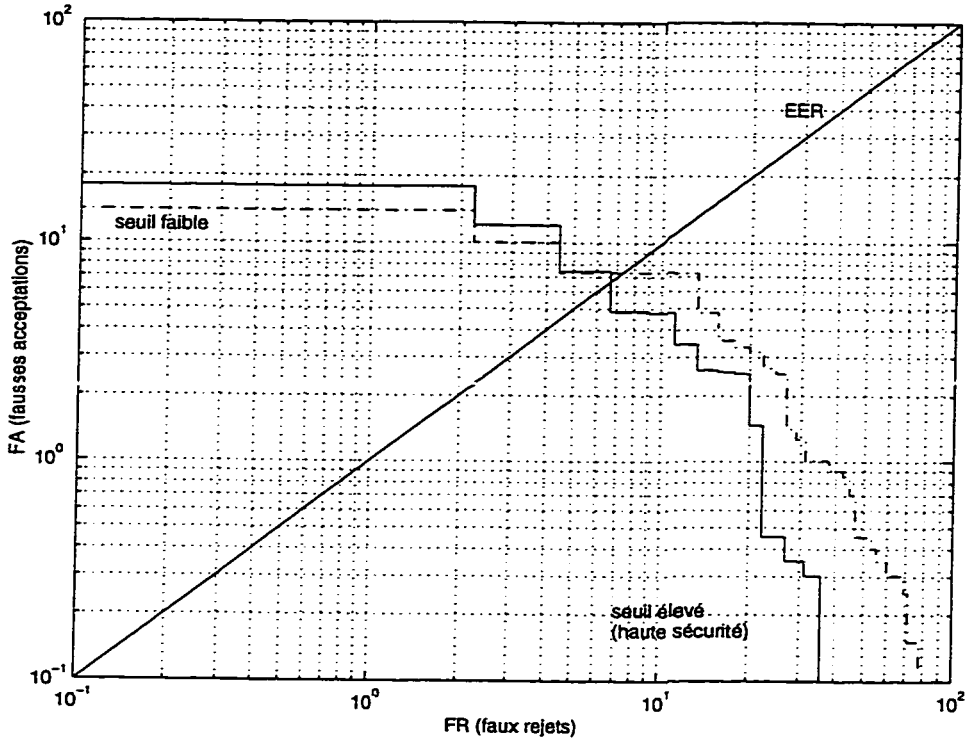

Figure 6.26: Courbes DET. Schème 4545. Modèle à 16 gaussiennes. Fichier test tronqué à 10 secondes. Normalisation: muselière (trait plein: -) et cohorte (trait pointillé: -.).

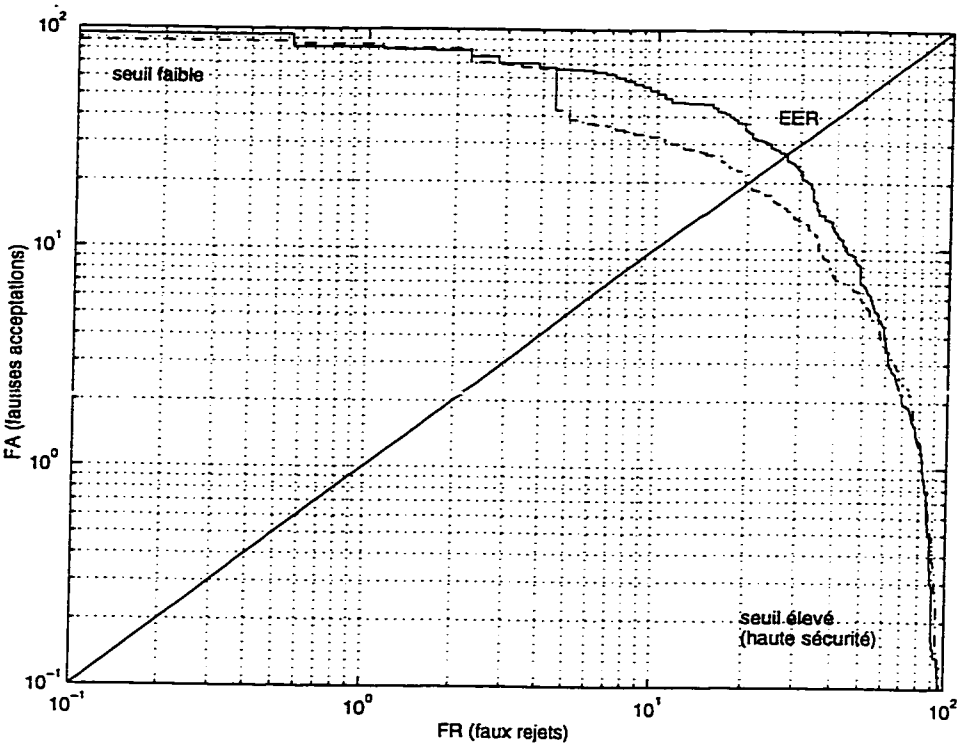

Figure 6.27: Courbes DET. Schème 9090. Modèle à 16 gaussiennes. Fichier test tronqué à 03 secondes. Normalisation: muselière (trait plein: -) et cohorte (trait pointillé: -.).

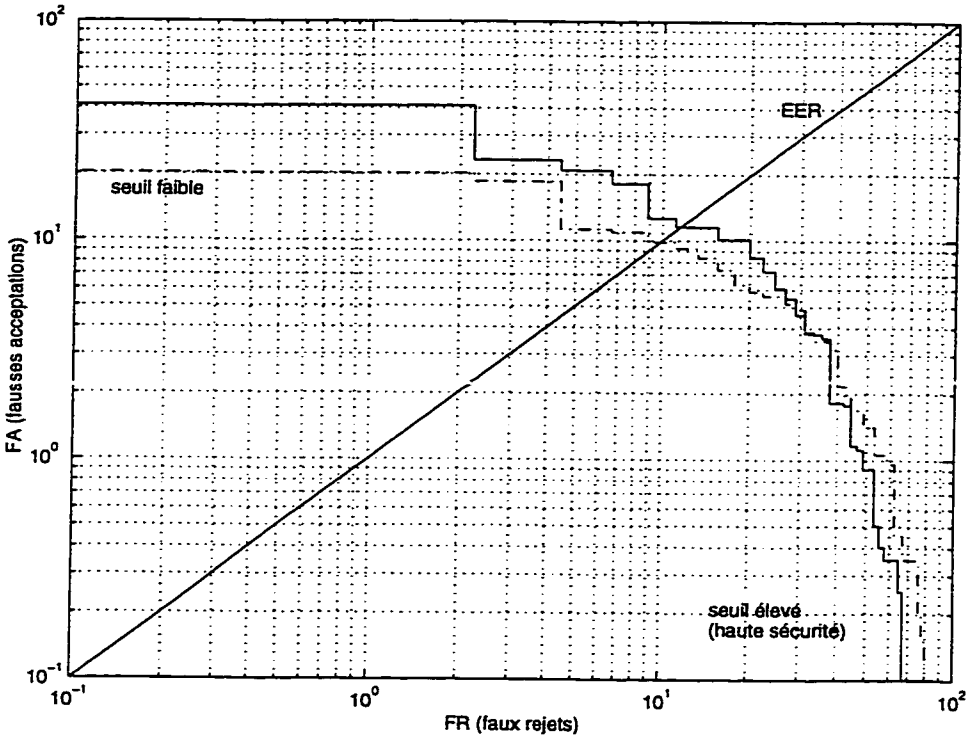

Figure 6.28: Courbes DET. Schème 4545. Modèle à 16 gaussiennes. Fichier test tronqué à 03 secondes. Normalisation: muselière (trait plein: -) et cohorte (trait pointillé: -.).

6.8 Conclusion

Au niveau du calcul du EER, les résultats obtenus par la muselière battent ceux de la cohorte pour des durées suffisamment longues selon le schème d'appariement utilisé. Toutefois, là où la muselière revêt le plus grand potentiel est lorsque le seuil d'acceptation est élevé et correspond à une situation de haute sécurité.

Conclusion

Pistes

Le standard établi par NIST pour l'évaluation des systèmes de vérification fait en sorte que les faux rejets sont particulièrement pénalisés. Selon ce critère, le calcul de la muselière ne pourra obtenir le même succès que celui de la cohorte. Dans l'éventualité où une équipe se formerait afin de participer à ce concours, il serait important de modifier l'objectif recherché au cours des expériences réalisées et présentées dans ce document: les loups ont été particulièrement ciblés alors que l'effort devrait plutôt être concentré sur l'amélioration des résultats obtenus par certains fichiers de locuteurs honnêtes, qui sont de piètre qualité et causent trop de faux rejets.

Le calcul de la muselière peut être interprété de diverses façons. Si l'on établit un lien avec l'erreur d'entraînement, on peut voir que le choix, *a priori*, d'un nombre fixe de gaussiennes pour la modélisation de chacun des locuteurs semble inapproprié.La faiblesse de la corrélation entre la durée du fichier et le calcul de la muselière (W_c) a été démontrée au chapitre 6. Certains algorithmes d'apprentissage pourraient être développés afin de choisir le nombre de mixtures approprié en minimisant l'erreur d'entraînement à laquelle serait ajoutée une pénalité à la complexité. Mieux encore,

le calcul de l'erreur de généralisation du modèle pourrait être utilisé afin de determiner la capacité optimale du modèle.

Le calcul de la muselière utilisé

$$\Lambda(\mathbf{O}_c|\lambda_d) = \frac{P(\mathbf{O}_c|\lambda_d)}{P(\mathbf{O}_c|\lambda_c)}$$

pourrait être adapté afin de tenir compte de l'équation 5.22 selon laquelle:

$$\hat{A}_c = 1.0716 \cdot W_c + 2.1003$$

La muselière adaptée pour ce résultat empirique serait la suivante:

$$\Lambda(\mathbf{O}_c|\lambda_d) = \frac{P(\mathbf{O}_c|\lambda_d)}{[P(\mathbf{O}_c|\lambda_c)]^{1.0716}}$$

La base de données utilisée n'a pas permis de mettre en valeur le calcul de l'impact qui semble toutefois intuitivement plus robuste. Il est possible que la présence de forts loups ne soit dépendante que de la base de données utilisée et que l'utilisation d'une base plus importante permette de faire ressortir sa supériorité. En particulier, la base de données utilisée en 1997 pour le concours de reconnaissance du locuteur et qui comprenait 417 locuteurs sera disponible sous peu auprès de DARPA. Il sera alors intéressant de pouvoir comparer les performances des diverses méthodes.

Synthèse

La contribution la plus importante de cet ouvrage est la proposition et l'expérimentation d'un nouveau calcul de vraisemblance pour la tâche de vérification du locuteur, appelé la muselière. L'impact, dérivé de la muselière a aussi été proposé. La muselière a réussi, dans certaines des expériences réalisées, à battre la cohorte, l'une des techniques présentement utilisées par les meilleurs systèmes.

Les premiers chapitres de ce document ont révisé les étapes liées au développement des coefficients MFCC. Les modèles cachés de Markov (HMM) ont été présentés, de même que les modèles à mixtures de gaussiennes (GMM). Les meilleures techniques de normalisation présentement utilisées de même que la muselière et l'impact ont été décrits. Enfin, une batterie de tests a été implantée sur la base de données SPIDRE afin de tester les idées proposées.

De ces expériences ressort la conclusion que la muselière pourrait s'avérer très utile dans le cas de l'implantation de systèmes de vérification à haute sécurité. Le développement d'un modèle artificiel à partir des données test reste, à la fois, l'idée originale liée à la muselière et le désavantage majeur de ce calcul: il doit être fait après la prononciation du fichier test et allonge donc le délai de réponse du système. Toutefois, dans la mesure où l'éviction des imposteurs est cruciale pour le système développé, il est vraisemblable que la durée de réponse du système soit justifiable, contrairement à une application où l'on s'attarde surtout à éviter les faux rejets afin de ne pas décevoir les utilisateurs.

Références

[1] Y. ARIKI et M. SAKURAGI. Unsupervised Speaker Normalization Using Canonical Analysis. In *Proceedings of the ICASSP*, pages 93–96, (1998).

[2] J. BIGÜN et G. CHOLLET. Special Issue on Audio- and Video-Based Person Authentication. *Pattern Recognition Letters*, 18:823–825, (1997).

[3] J.P. CAMPBELL. Speaker Recognition: A Tutorial. *Proceedings of the IEEE*, 85:1436–1462, (1997).

[4] M.J. CAREY, E.S. PARRIS, S.J. BENNETT, et H. LLOYD-THOMAS. A Comparison of Model Estimation Techniques for Speaker Verification. In *Proceedings of the ICASSP*, pages 1083–1086, (1997).

[5] M.J. CAREY, E.S. PARRIS, et J.S. BRIDLE. Speaker Verification Systems Using Alpha-Nets. In *Proceedings of the ICASSP*, pages 397–400, (1991).

[6] D. CHARLET et D. JOUVET. Optimizing Feature Set for Speaker Verification. *Pattern Recognition Letters*, 18:873–879, (1997).

[7] R. COMERFORD, J. MAKHOUL, et R. SCHWARTZ. The voice of the computer is heard in the land (and it listens too!). *IEEE Spectrum*, Décembre (1997).

[8] H.M. CUNG et Y. NORMANDIN. Noise Adaptation Algorithms for Robust Speech Recognition. *Speech Communication*, 12:267–276, (1993).

[9] S. DAS, R. BAKIS, A. NADAS, D. NAHAMOO, et M. PICHENY. Influence of Background Noise and Microphone on the Performance of the IBM Tangora Speech Recognition System. In *Proceedings of the ICASSP*, pages 71–74, (1993).

[10] S.B. DAVIS et P. MERMELSTEIN. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In *Proceedings of the ICASSP*, pages 357–366, (1980).

[11] Y. FANG. A GMM Speaker Recognition System Using HTK V1.5, (1997).

[12] R.A. FINAN, A.T. SAPELUK, et R.I. DAMPER. Impostor Cohort Selection for Score Normalisation in Speaker Verification. *Pattern Recognition Letters*, 18:881–888, (1997).

[13] S. FURUI. Recent Advances in Speaker Recognition. *Pattern Recognition Letters*, 18:859–872, (1997).

[14] M.F.J. GALES et S.J. YOUNG. Cepstral Parameter Compensation for HMM Recognition in Noise. *Speech Communication*, 12:231–239, (1993).

[15] L.P. HECK et M. WEINTRAUB. Handset-Dependent Background Models for Robust Text-Independent Speaker Recognition. In *Proceedings of the ICASSP*, pages 1071–1074, (1997).

[16] H. HERMANSKY, N. MORGAN, A. BAYYA, et P. KOHN. RASTA-PLP Speech Analysis Technique. In *Proceedings of the ICASSP*, pages 121–124, (1992).

[17] A. HIGGINS, L. BAHLER, et J. PORTER. Speaker Verification Using Randomized Phrase Prompting. *Digital Signal Processing*, 1:89–106, (1991).

[18] J. ISHII et T. FUKADA. Speaker Independent Acoustic Modeling Using Speaker Normalization. In *Proceedings of the ICASSP*, pages 97–100, (1998).

[19] L.B. JACKSON. *Signals, Systems and Transforms.* Addison-Wesley, (1990).

[20] Q. LI, S. PARTHASARATHY, et A.E. ROSENBERG. A Fast Algorithm for Stochastic Matching with Application to Robust Speaker Verification. In *Proceedings of the ICASSP*, pages 1543–1546, (1997).

[21] R.J. MAMMONE, X. ZHANG, et R.P. PAMACHANDRAN. Robust Speaker Recognition. *IEEE Signal Processing Magazine*, Septembre (1996).

[22] A. MARTIN. The (1997) Speaker Recognition Evaluation Plan. Technical report, National Institute for Standards in Technology, (1997).

[23] T. MATSUI et K. AIKAWA. Robust Model for Speaker Verification against Session-Dependent Utterance Variation. In *Proceedings of the ICASSP*, pages 117–120, (1998).

[24] T. MATSUI et S. FURUI. Similarity Normalization Method for Speaker Verification Based on A Posteriori Probability. In *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 59–62, (1994).

[25] J. NAIK. Speaker Verification over the Telephone Network: Databases, Algorithms and Performance Assessment. In *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 31–38, (1994).

[26] S. NAKAGAWA et K.P. MARKOV. Speaker Verification using Frame and Utterance Level Likelihood Normalization. In *Proceedings of the ICASSP*, pages 1087–1090, (1997).

[27] N.J. NILSSON. *The Mathematical Foundations of Learning Machines.* Morgan Kaufmann, (1990).

[28] J.Ø. OLSEN. Separation of Speakers in Audio Data. In *Proceeding of EUROSPEECH*, pages 355–359, (1995).

[29] J.Ø. OLSEN. A two-stage procedure for phone based speaker verification. *Pattern Recognition Letters*, 18:889–897, (1997).

[30] J.Ø. OLSEN. *Phoneme Based Speaker Verification.* PhD thesis, Aalborg University, (1997).

[31] J.Ø. OLSEN. Speaker Verification Based on Phonetic Decision Making. In *Proceeding of EUROSPEECH*, pages 1375–1378, (1997).

[32] J.Ø. OLSEN. Using Ensemble Techniques for Improved Speaker Modelling. to be presented at NORSIG, (1998).

[33] D. O'SHAUGHNESSY. *Speech Communication.* McGill University Press, (1997).

[34] S. PARTHASARATHY et A.E. ROSENBERG. General phrase speaker verification using sub-word background models and likelihood-ratio scoring. In *Proceedings of the ICSPL*, pages 2403–2406, (1996).

[35] J.-B. PIERROT, J. LINDBERG, J. KOOLWAAIJ, H.-P. HUTTER, D. GENOUD, M. BLOMBERG, et F. BIMBOT. A Comparison of A Priori Threshold Setting

Procedures for Speaker Verification in the CAVE Project. In *Proceedings of the ICASSP*, pages 125–128, (1998).

[36] L. RABINER et B.-H. JUANG. *Fundamentals of Speech Recognition.* Prentice-Hall, (1993).

[37] L. RABINER et R. SCHAFER. *Digital Processing of Speech Signals.* Prentice-Hall, (1978).

[38] D.A. REYNOLDS. *A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification.* PhD thesis, Georgia Institute of Technology, (1992).

[39] D.A. REYNOLDS. Comparison of Background Normalization Methods for Text-Independent Speaker Verification. In *Proceedings of Eurospeech*, pages 963–966, (1997).

[40] D.A. REYNOLDS. Speaker Identification and Verification using Gaussian Mixture Speaker Models. *Pattern Recognition Letters*, 18:91–108, (1997).

[41] D.A. REYNOLDS et R.C. ROSE. An Integrated Speech-Background Model for Robust Speaker Identification. In *Proceedings of the ICASSP*, pages 185–188, (1992).

[42] R.C. ROSE, J. FITZMAURICE, E.M. HOFSTETTER, et D. REYNOLDS. Robust speaker identification in noisy environments using noise adaptive speaker models. In *Proceedings of the ICASSP*, pages 401–404, (1991).

[43] A.E. ROSENBERG, J. DELONG, C.-H. LEE, B.-H. JUANG, et F.K. SOONG. The use of cohort normalization scores for speaker verification. In *Proceeedings of the ICSLP*, pages 599–602, (1992).

[44] A.E. ROSENBERG et S. PARTHASARATHY. Speaker background models for connected digit password speaker verification. In *Proceeedings of the ICASSP*, pages 81–84, (1996).

[45] A.E. ROSENBERG, O. SIOHAN, et S. PARTHASARATHY. Speaker Verification using Minimum Verification Error Training. In *Proceedings of the ICASSP*, pages 105–108, (1998).

[46] M. SCHMIDT, S. IRONSIDE, M. JACK, et F. STENTIFORD. User perspectives on the security of access data, operator handover procedures and "insult rate" for speaker verification in automated telephone services. *Pattern Recognition Letters*, 18:947–953, (1997).

[47] O. SIOHAN, A.E. ROSENBERG, et S. PARTHASARATHY. Speaker Identification using Minimum Classification Error Training. In *Proceedings of the ICASSP*, pages 109–112, (1998).

[48] C. TADJ, P. DUMOUCHEL, et P. OUELLET. GMM Based Speaker Identification using Training-Time-Dependent Number of Mixtures. In *Proceedings of the ICASSP*, pages 761–764, (1998).

[49] R. YANG et P. HAAVISTO. An Improved Noise Compensation Algorithm for Speech Recognition in Noise. In *Proceedings of the ICASSP*, pages 49–52, (1996).

[50] S. YOUNG. Speech Recognition, from the lab to the real world. *Telesis*, (101), (1996).

[51] S. YOUNG, J. ODELL, D. OLLASON, V. VALTCHEV, et P. WOODLAND. *The HTK Book, version 2.1.* Cambridge University Press, (1997).