N° d'ordre : 11/2010-M/EL

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE
FACULTE D'ELECTRONIQUE ET D'INFORMATIQUE

MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER

En : ELECTRONIQUE

Spécialité : Traitement du Signal et des Images

Par : M^elle YALA NAWEL

Sujet :

Reconnaissance Automatique du Locuteur

Soutenu publiquement le 20/12/2010, devant le jury composé de :Jury composé de :

Mme L.BOUMGHAR	Professeur à l'USTHB	Présidente
Mr A.HOUACINE	Professeur/A,àl'USTHB	Directeur de mémoire
Mr A.AMROUCHE	Maitre de Conférences/A, àl'USTHB	Examinateur
Mr H.TEFFAHI	Maitre de Conférences/A, àl'USTHB	Examinateur
Mr B.FERGANI	Maitre de Conférences/A, àl'USTHB	Examinateur

Remerciements

Le travail présenté dans ce mémoire est réalisé au Laboratoire de Communication Parlée et Traitement de Signal (LCPTS) de la Faculté d'Electronique et d'Informatique (FEI) de l'USTHB. Il s'inscrit dans le cadre de la préparation d'un Magister en Traitement du Signal et des Images.

Ce mémoire, ne pourrait exister sans l'aide et l'engagement d'un certain nombre de personnes qui ont décidé de m'accompagner résolument dans mon parcours.

Tout d'abord, je tiens à remercier mon directeur de mémoire, Professeur A. Houacine, pour m'avoir proposé ce sujet ainsi que pour ses orientations et conseils judicieux et avisés qu'il m'a prodigué tout au long de ce travail, pour tout le temps qu'il m'a consacré à relire et à corriger ce mémoire sans oublier ses suggestions éclairées.

Je remercie, Mme L. BOUMGHAR, Professeur USTHB, pour l'honneur qu'elle me fait en acceptant de présider le jury de ce mémoire, Mr A. AMROUCHE, Maitre de Conférences A, USTHB , Mr H. TEFFAHI, Maitre de Conférences A, USTHB et Mr B. FERGANI, Maitre de Conférences A, USTHB pour avoir accepté de faire partie de ce jury et d'examiner ce travail, j'en suis très honorée.

Je remercie ma famille, en particulier mes parents, pour leur présence, leur soutien et leurs conseils.

Que tous ceux, qui de près ou de loin, ont contribué, par leurs conseils, leurs encouragements ou leur amitié, à l'aboutissement de ce modeste travail, trouvent ici l'expression de ma profonde reconnaissance.

Dédicaces

A toute ma famille
et mes amis...

Résumé

Le domaine de la reconnaissance automatique du locuteur regroupe les applications pour lesquelles on désire identifier une personne à partir de sa voix. Le champ d'application couvre de nombreux secteurs tels que l'accès sécurisé, les transactions téléphoniques, la surveillance, l'indexation audio ou encore l'expertise judiciaire.

Dans ce travail, nous nous intéressons à une méthode de modélisation statique basée sur les mélanges gaussiens GMM (Gaussian Mixture Model), qui est devenue, dans les dix dernières années, l'approche la plus performante et la plus répandue pour les systèmes de reconnaissance du locuteur indépendante du texte. Chaque locuteur est représenté par un modèle GMM. Ce dernier est une somme pondérée de M distributions gaussiennes multidimensionnelles.

Dans notre travail, la base de données VoxForge est utilisée pour l'extraction des paramètres pertinents caractérisant les locuteurs. La technique utilisée est celle de l'analyse cepstrale, dont les résultats sont les coefficients MFCC couramment utilisés en reconnaissance du locuteur.

Durant la phase d'apprentissage, la phase de construction des modèles des locuteurs, les paramètres du mélange de gaussiennes sont estimés par l'algorithme EM/ML. Deux phases d'apprentissage sont mises en œuvre pour pallier le manque de données d'apprentissage disponibles pour chaque locuteur:

1. Apprentissage d'un modèle générique (aussi appelé « modèle du monde ») estimé par l'algorithme EM/ML sur une grande quantité de données (population de locuteurs);
2. apprentissage du modèle locuteur dérivé du modèle du monde par application d'une technique d'adaptation (par exemple par Maximum a Posteriori).

Nous avons considéré deux méthodes pour la phase de décision; la première est celle du rapport de la maximum de la log-vraisemblance (LLR : Logarithm of the Likelihood Ratio), la deuxième approche considérée est basée sur les SVM. Les performances des deux systèmes sont comparées.

Notre système de reconnaissance du locuteur réalise deux tâches qui sont l'identification du locuteur et la vérification du locuteur. Le taux d'égale erreur est un indice de performance offert par le système de vérification, tandis que le taux d'identification correcte est un indice offert par le système d'identification. Les résultats obtenus en termes de taux d'identification correcte sont de 91.1%, et le taux d'égale erreur EER est de 7.4%.

Mots-clés: Identification du locuteur, vérification du locuteur, l'approche GMM-UBM. SVM.

Tables des matières

Liste des figures

Liste des tableaux

Liste d'abréviations

ANN	Artifcial Neural Networks
DCF	Detection Cost Function
DET	plot Detection Error Tradeoff
EER	Equal Error Rate
FA	Fausse acceptation
FR	Faux rejet
GMM	Gaussian Mixture Models
HMM	Hidden Markov Models
IAL	Identification automatique du locuteur
MAP	Maximum A Posteriori
MFCC	Mel-Frequency Cepstral Coefficients
ML	Maximum Likelihood
NIST	National Institute of Standards and Technology
RAL	Reconnaissance automatique du locuteur
ROC	Receiver Operating Characteristics
SVM	Machines à Vecteurs Support
UBM	Universal Background Model
VQ	Vector Quantization

Chapitre 1

Introduction Générale

Sommaire

De nos jours, l'authentification des utilisateurs est nécessaire dans différents domaines pour, par exemple, sécuriser les guichets automatiques bancaires ou gérer les accès à des ressources protégées. En règle générale, l'authentification des utilisateurs est réalisée par un identifiant associé à un mot de passe secret. Les identifiants et mots de passe, clés ou badges d'accès, sont largement employés, mais ceux ci peuvent être facilement falsifiés ou volés.

Afin d'élever le niveau de confiance de cette application, de nouvelles techniques d'authentification, basées sur la biométrie, sont apparues. Effectivement, l'authentification biométrique répond à cette problématique car chaque individu a ses propres caractéristiques physiques, qui sont invariantes ou peu variables dans le temps et donc ne peuvent être perdues ou volées, comme un simple mot de passe.

1.1 La biométrie

La Biométrie est l'identification des individus à partir de leurs caractéristiques physiques de l'utilisateur. Ces caractéristiques sont liées à l'individu, à son organisme. Elles sont uniques pour chaque individu et ne varient pas ou très peu dans le temps et ne peuvent être modifiées par un individu. Une authentification biométrique est une authentification basée sur les caractéristiques physiques de l'utilisateur. Nous pouvons citer les méthodes d'authentification basées sur les empreintes digitales, sur les

empreintes dentaires ou encore la méthode d'authentification par l'iris de l'œil. Ceux-ci reposent sur deux étapes:

1. la collecte d'un échantillon de référence, appartenant à l'utilisateur,
2. la comparaison d'un échantillon de test à l'échantillon de référence.

Il suffit de mesurer la correspondance entre les deux échantillons pour avoir une décision d'authentification. Deux modes de fonctionnement existent en authentification biométrique :
– l'identification: il s'agit de déterminer l'identité de l'utilisateur à partir d'une base de données d'échantillons de référence. L'échantillon de test est comparé à tous les échantillons de la base. L'identité de l'utilisateur reconnu de la base est retournée,
– la vérification d'identité: il s'agit de valider l'identité proclamée de l'échantillon test, à partir d'un échantillon de référence. Dans ce cas, l'échantillon de test n'est comparé qu'à un seul échantillon de référence. L'identité d'utilisateur est celle qui est proclamée si la mesure de correspondance dépasse un seuil prédéterminé.

Pourquoi la voix pour authentifier un utilisateur ?

Comme nous l'avons vu au paragraphe précédent, l'individu possède une diversité de caractéristiques physiques, les unes plus fiables que d'autres. Cela nous permet de dire qu'il existe une méthode d'authentification plus fiable que d'autres. La reconnaissance d'empreintes digitales ou de l'iris autorisent un niveau maximal de sécurité et des taux d'erreurs minimaux. Mais ces méthodes sont intrusives et restent lourdes à mettre en œuvre. Le caractère intrusif d'une méthode biométrique est défini par le niveau d'acceptation de la méthode par l'utilisateur. Les empreintes digitales ont une référence criminalistique et c'est pour cette raison qu'elles sont peu acceptées par le public. De même, l'authentification par empreinte rétinienne, est aussi peu acceptée par les utilisateurs, bien que considérée comme la plus fiable des technologies, car elle nécessite un faisceau lumineux pour éclairer le fond de l'œil afin de déterminer les positions des veines de la rétine **[1] [2]**.

C'est pourquoi, l'authentification par ces types de biométries est limitée au cas de nécessité d'une sécurité importante, par exemple dans le milieu militaire où l'acceptation de l'utilisateur compte moins que d'assurer la robustesse du système.

Une comparaison entre les méthodes biométriques les plus répandues (comme l'empreinte digitale, l'iris ou la rétine) est proposée par le groupe international de la biométrie **[3]**. Elle démontre que la reconnaissance du locuteur offre l'avantage d'être bien acceptée par l'utilisateur et d'être simple à mettre en œuvre. Basée sur un échantillon de voix du locuteur, elle n'implique que la prise de son à travers un microphone. De plus, c'est souvent le seul média disponible. Les systèmes de Reconnaissance Automatique du Locuteur (RAL) s'appuient sur les caractéristiques de la parole pour reconnaître les individus.

La reconnaissance du locuteur en tant que technique d'authentification présente les avantages suivants :

– l'acquisition du signal audio est très simple à mettre en œuvre,
– l'enregistrement du signal audio n'est pas considéré comme intrusif.
– le signal audio est naturellement véhiculé dans la majorité des réseaux de communication,
– les techniques de stockage et de compression du signal audio sont très efficaces,
– dans de nombreuses applications (serveurs vocaux), l'utilisateur emploie déjà la parole pour communiquer avec la machine. Le coût supplémentaire de la RAL est faible.

Depuis quelque dizaines d'années, la recherche dans le domaine de la RAL est en progression. Cependant, les performances des techniques utilisées en reconnaissance n'atteignent pas le niveau des performances des techniques biométriques les plus robustes, comme la reconnaissance des empreintes digitales ou de la rétine. De ce fait, la voix est la plus souvent utilisée en complément d'une autre modalité (voix + mot de passe, voix + image).

1.2 Application visée

Dans ce travail, nous nous intéressons à la reconnaissance automatique du locuteur en mode indépendant du texte. Pour réaliser cette tâche, plusieurs approches ont été proposées dans la littérature: approche vectorielle, connexionniste, statistique et prédictive. En RAL, l'approche statistique est l'approche la plus utilisée dans les dernières années. Nous l'avons choisi comme méthode de modélisation dans notre système de reconnaissance.

1.3 Organisation du document

Ce document se divise en six chapitres. Le chapitre suivant est consacré aux principes généraux de la RAL. Le troisième chapitre décrit les modules d'une plateforme de RAL. Au quatrième chapitre, nous présentons une mise en œuvre d'un système de vérification du locuteur en mode indépendant du texte en utilisant l'approche GMM-UBM. Au cinquième chapitre nous présentons l'approche GMM-SVM pour la vérification du locuteur, afin de comparer ses performances avec celles du système de l'état de l'art (GMM-UBM). Enfin, le dernier chapitre est consacré à la présentation de l'expérimentation et l'évaluation des résultats.

Chapitre 2

Principes généraux de la reconnaissance du locuteur

Sommaire

Ce chapitre est consacré aux principes de la reconnaissance automatique du locuteur. Tout d'abord les mécanismes de production de la parole et les principales sources de variabilités sont présentés, afin de comprendre comment un individu peut être reconnu par sa voix. Les difficultés majeures associées à la RAL sont mises en évidence. Nous exposons ensuite les traitements numériques appliqués au signal audio dans un système de reconnaissance du locuteur. Nous présentons aussi les différentes tâches liées à la RAL, telles que l'Identification et la Vérification Automatique du Locuteur (IAL, VAL), et l'Indexation en Locuteurs de flux audio. Enfin nous présentons quelques approches utilisées pour les systèmes de RAL.

2.1 La parole

2.1.1 La production de la parole

La production de la parole fait intervenir différents organes. La source de la parole provient des poumons qui émettent un flux d'air. Ce flux d'air va traverser le larynx pour faire vibrer ou non les cordes vocales. Il va ensuite traverser le conduit vocal (cavité nasale et buccale) et les articulateurs tels que les lèvres et la langue (Figure **2-1**). Cet ensemble agit comme un filtre, considéré comme linéaire, dont la réponse impulsionnelle comporte des fréquences de résonance caractérisées par des pics, appelés formants, dans le spectre du signal de sortie.

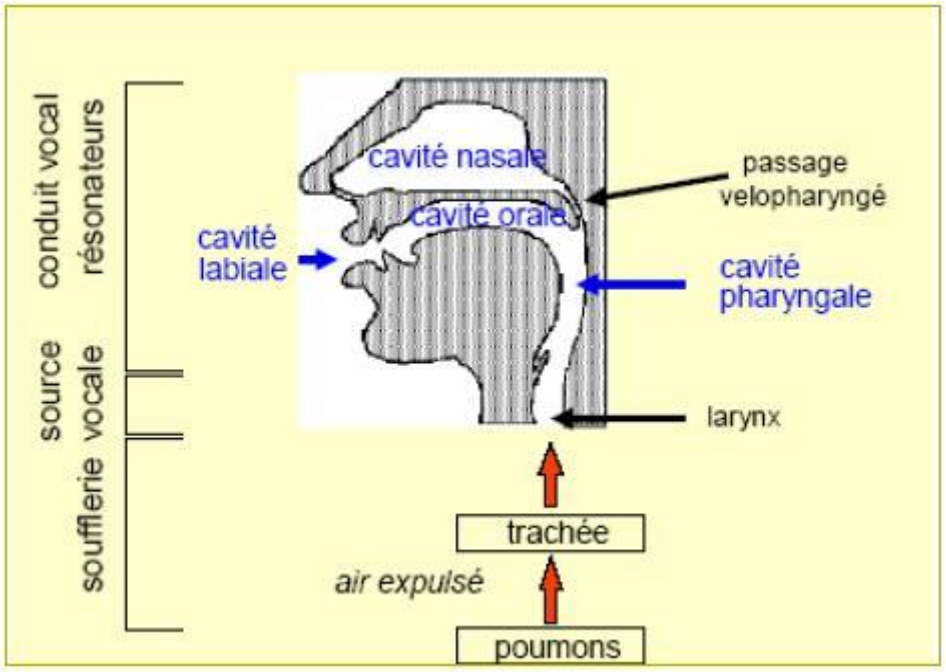


FIG 2-1. Modèle physiologique de la production de la parole (Extrait de **[4]**)

Le signal résultant est globalement non stationnaire mais peut être considéré comme stationnaire sur de très courtes périodes, de l'ordre de 20ms (signal pseudo-stationnaire). Sur un segment de parole de cette longueur la voix est habituellement et schématiquement séparée en deux classes distinctes :

1. voisée lorsqu'il y a vibration des cordes vocales, le signal est alors quasi-périodique,
2. non voisée dans le cas d'un simple soufflement, le signal est alors considéré comme aléatoire.

Dans le premier cas, la source d'excitation est modélisée par un train d'impulsions périodique, de fréquence dite de voisement F_0, qui correspond à la fréquence de vibration des cordes vocales, la fréquence fondamentale ou pitch ; dans le second cas, la source est modélisée par un bruit blanc. Cette représentation binaire de la production de la parole a été introduite par **[5]**. Elle est reprise sur la figure **2-2**.

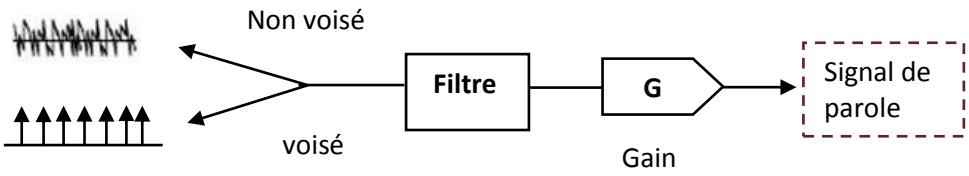


FIG 2-2. Modèle de production de la parole.

2.1.2 Les variabilités du signal de parole

2.1.2.1 Variabilité inter-locuteurs

Le signal de parole ne véhicule pas, seulement, un message, il porte aussi des informations sur l'individu qui l'émet. Il varie en fonction du locuteur. Cette variabilité utile pour différencier les locuteurs, est principalement due à des différences fonctionnelles et anatomiques (les fonctions de l'appareil phonatoire et de l'oreille) entre locuteurs (chacun a son propre appareil phonatoire). Autre origine de la variabilité inter-locuteurs revient aux différences de prononciation qui existent au sein d'une même langue et qui constituent les accents régionaux.

2.1.2.2 Variabilité intra-locuteur

La variabilité intra-locuteur identifie les différences dans le signal produit par une même personne. Cette variation peut résulter de l'état physique ou moral du locuteur (Rhume par exemple). L'humeur ou l'émotion du locuteur peut également influencer son rythme d'élocution, son intonation.

La variabilité intra-locuteur peut rendre l'identification du locuteur plus difficile, avec la variabilité de parole qui est due aux conditions d'enregistrement du signal de parole (bruit ambiant, microphone utilisé, lignes de transmission).

2.1.3 Paramètres caractéristiques

On distingue différents niveaux d'informations dans le signal de parole **[6]**. Les niveaux « bas » regroupent des informations liées à des traits physiques de locuteur (facteurs morphologiques et physiologiques). Les niveaux « hauts » regroupent les informations liées à des traits acquis de locuteur (facteurs socioculturels). Ben dans **[6]** a identifié six niveaux d'informations qu'il présente du plus bas niveau au plus haut :

1. le niveau acoustique : les paramètres sont relatifs au contenu spectral du signal de parole et sont liées aux caractéristiques physiques de l'appareil vocal. L'enveloppe du spectre du signal de parole caractérise principalement la morphologie du conduit vocal du locuteur et les harmoniques reflètent la fréquence fondamentale;
2. le niveau prosodique qui désigne la « mélodie » de l'énoncé de parole : hauteur de la voix (fréquence fondamentale), intensité de la voix (énergie) et durée des segments syllabiques ;
3. le niveau phonétique : la distinction des différents sons identifiables d'une langue;
4. le niveau idiolectal qui se rapporte aux particularités langagières propres à un individu ;
5. le niveau dialogique qui définit la façon de communiquer d'un individu, comme ses temps de parole dans une conversation ;
6. enfin, le niveau sémantique qui caractérise la signification du discours.

Historiquement, les paramètres de bas niveau ont été les premiers utilisés pour caractériser les personnes dans les systèmes de reconnaissance du locuteur, car ils sont faciles à extraire, robustes et caractérisent bien le conduit vocal des personnes. Les paramètres de haut niveau demandent une mise en œuvre plus lourde. Dans le cadre de notre travail, nous nous limitons à l'extraction des paramètres acoustiques qui caractérisent les locuteurs.

2.1.4 Les paramètres de la parole exploitables pour la RAL

Le signal de parole est quasi-stationnaire sur de courtes périodes, c'est pour cette raison qu'il est généralement analysé sur des trames pondérées par une fenêtre de pondération de 20 à 30ms avec un taux de recouvrement de 50% à 75%, puis représenté dans le domaine spectral.

Les paramètres du signal de parole décrits dans cette section permettent de discriminer un locuteur des autres. Idéalement, ces paramètres doivent avoir une forte variabilité entre les locuteurs et une faible variabilité pour un même locuteur. De plus, ils doivent être robustes aux perturbations d'enregistrement. Nous citons les paramètres exploitables pour la RAL.

2.1.4.1 L'énergie

L'énergie du signal s(t) est calculée à partir du signal temporel suivant l'équation **2.1**.

$$\mathrm{Es} = \int_{\infty}^{\infty} |\mathrm{s(t)}|^2 \, dt \qquad (2.1)$$

L'énergie est généralement exprimée en décibels :

$$\mathrm{Es(dB)} = 10 \log_{10} \mathrm{Es} \qquad (2.2)$$

L'évolution dans le temps du paramètre d'énergie peut déterminer le style d'intonation du locuteur. Utilisé seul, ce paramètre permet essentiellement de classer les trames par ordre énergétique, ce qui constitue une approche majoritaire pour la détection de la parole.

2.1.4.2 Les coefficients cepstraux

Nous avons vu au paragraphe **2.1.3** que le niveau acoustique est le plus utilisé en RAL pour caractériser les locuteurs. Il se base sur une représentation numérique de l'enveloppe du signal, définie comme une suite de paramètres acoustiques calculés à des intervalles de temps régulier sur des blocs de signal de longueur fixe .Cette suite de paramètres constitue un vecteur acoustique. Il ya plusieurs techniques de paramétrisation acoustique, on peut les regrouper en trois grandes familles :

- Paramétrisation par bancs de filtres.
- Paramétrisation par transformée de Fourier.
- Paramétrisation par prédiction linéaire.

Les cepstres

Le cepstre est défini comme étant la transformée de Fourier inverse du logarithme du la transformée de Fourier d'énergie d'une fonction **[7]**. Cette transformation porte le nom de cepstre d'énergie. L'équation donnant le cepstre, issue de cette définition, est :

$$s_{Ceps}(t) = \text{TFI}\,[\log(\,|\text{TF}(x(t)|\,)] \qquad (2.3)$$

A partir de l'équation **2.3**, le cepstre est une transformation qui transforme un produit de convolution en une addition.

Telle que mentionnée au paragraphe **2.1.1**, la parole est considérée comme étant le résultat d'une convolution entre la source d'excitation et le conduit vocal. La reconnaissance du locuteur bénéficie de la technique du cepstre pour séparer l'influence de la source d'excitation vocale et celle du conduit vocal **[7]**, cette dernière étant généralement la seule à être utilisée dans le domaine de la reconnaissance du locuteur.

En notant :

$$s(t) = w(t) * h(t) \qquad (2.4)$$

où s(t) est le signal de parole, w(t) est la source d'excitation et h(t) la réponse impulsionnelle du conduit vocal.

Par l'application du logarithme du module de la transformée de Fourier à l'équation **2.4**, cette dernière devient :

$$\log |S(f)| = \log |W(f)| + \log |H(f)| \quad (2.5)$$

Par l'application d'une transformée de Fourier inverse, on obtient :

$$s_{Ceps}(t) = w_{Ceps}(t) + h_{Ceps}(t) \quad (2.6)$$

D'où la séparation de l'influence de source d'excitation et celle du conduit vocal.
Les paramètres caractérisant la source du signal de parole sont moins largement utilisés en RAL. Ceci est en partie dû à la difficulté de leur estimation et à leur grande variabilité intra-locuteur. Parmi eux citons l'énergie, la fréquence fondamentale Fo et le taux de voisement **[8]**.

Les coefficients cepstraux sont utilisés dans le traitement du signal et sont des coefficients d'énergie calculés dans des bancs de filtres **[7]**. Selon la distribution des filtres dans la bande utile du signal l'un des deux cas suivants peut être considéré :

- les filtres sont uniformément distribués dans la bande utile du signal. Dans ce cas les coefficients calculés sont appelés LFCC (Linear Frequency Cepstral Coefficients) ;
- les filtres sont distribués selon une échelle non-linéaire, plus proche de la perception humaine appelée échelle Mel. Dans ce cas les coefficients calculés sont les MFCC (Mel Frequency Cepstral Coefficients).

Les coefficients LFCC sont similaires aux MFCC, sauf pour la distribution de filtres dans la bande utile du signal. Le calcul de ces coefficients est détaillé dans le chapitre suivant.

En reconnaissance du locuteur, nous prenons en considération que les coefficients cepstraux d'ordre faible(les premiers coefficients), celles qui contiennent l'information relative au conduit vocal. Cette information devient négligeable à partir d'un certain nombre de coefficients cepstraux. Les coefficients d'ordre élevé caractérisent la source vocale et reflètent les impulsions de la source **[9]**.

Nous trouvons aussi une méthode d'extraction des coefficients cepstraux différente de la méthode d'analyse cepstrale (elle résulte les coefficients MFCC et LFCC), c'est l'analyse paramétrique (par exemple, le codage prédictif linéaire (LPC)). Cette méthode est utilisée surtout en reconnaissance de la parole.

2.2 La Reconnaissance Automatique du Locuteur

L'objectif de la reconnaissance du locuteur est de reconnaître l'identité d'une personne à l'aide de sa voix. Les applications de la RAL sont principalement liées aux problèmes d'authentification ou de confidentialité.

2.2.1 Dépendance au texte

Selon le degré de dépendance au texte, nous pouvons classifier les systèmes de reconnaissance automatique du locuteur en deux catégories : systèmes indépendants du texte, où le texte prononcé par le locuteur n'est pas connu a priori ; et systèmes dépendants du texte, où le texte prononcé par le locuteur est connu à l'avance.

La connaissance a priori du contenu linguistique de texte prononcé par le locuteur (l'utilisateur du système) augmente les performances du système de reconnaissance. Mais aussi, dans ce cas, l'utilisateur doit se souvenir de son mot de passe, ou de lire un texte qui apparait instantanément.

2.2.2 Les différentes tâches

Les tâches de la reconnaissance du locuteur sont regroupées en trois catégories principales selon l'application proposée :

- l'identification de locuteur,
- la vérification du locuteur,
- l'indexation des locuteurs ou le suivi de locuteurs.

Dans la première tâche le système de RAL propose une identité à partir d'un ensemble de locuteurs, dans la seconde il valide une identité et dans la troisième, le système

détermine les durées de parole d'un locuteur, il compte aussi le nombre de locuteurs présents dans un signal.

2.2.1.1 Identification automatique du locuteur (IAL)

L'identification automatique du locuteur (IAL) consiste à déterminer l'identité d'un individu parmi un ensemble de personnes connues. Lors d'un accès à un système d'IAL, le signal de parole fourni à l'entrée du système est comparé à la référence caractéristique de chacun des locuteurs connus et l'identité retournée est celle dont la référence est la plus proche du signal de test. Le signal est la seule entrée du système d'IAL. Deux modes sont distingués : le fonctionnement en milieu fermé et le fonctionnement en milieu ouvert. En milieu fermé le locuteur est supposé être l'un des N locuteurs du système. En milieu ouvert, le système peut décider qu'aucune des N identités connues n'est celle du locuteur. Il doit pour cela disposer d'un modèle de rejet.

Les performances obtenues par les systèmes d'IAL sont directement liées au nombre N de locuteurs du système. La figure **2-3** représente un schéma illustrant le fonctionnement d'un système d'IAL.

Applications:

En ensemble fermé, les applications d'un système d'IAL sont peu nombreuses. L'identification automatique du locuteur peut cependant être utilisée de manière très efficace pour simplifier l'accès des membres d'une population d'individus à des données ou à des services personnalisés (mise en place automatique de paramètres d'utilisation, etc.). En ensemble ouvert, les applications de L'IAL sont essentiellement liées à des problèmes de sécurisation comme la protection de l'accès à des sites sensibles.

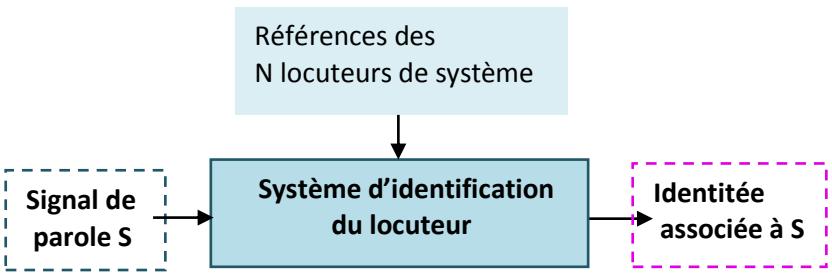


FIG 2-3. Schéma d'un système d'identification du locuteur

2.2.1.2 Vérification automatique du locuteur (VAL)

La vérification du locuteur consiste à déterminer si l'identité proclamée d'un message vocal correspond à la véritable identité du locuteur. La réponse est binaire, acceptation ou rejet. Les éléments mis en jeu sont donc une identité proclamée et la référence associée à un échantillon connu de l'identité proclamée. Une mesure de similarité entre le signal à vérifier et cette référence est calculée. Cette mesure est comparée à un seuil de vérification. Dans le cas où la mesure de similarité est supérieure au seuil, l'individu est accepté. Dans le cas contraire, l'individu est considéré comme un imposteur et rejeté.

2.2.1.3 Détection automatique des locuteurs

La détection automatique des locuteurs consiste à déterminer la présence ou non d'un locuteur donné sur un enregistrement audio. Si l'on fait l'hypothèse que le signal sonore est mono-locuteur, cette tâche est équivalente à la vérification automatique du locuteur. Comme dans le cas de la VAL, l'identité recherchée ainsi que le signal de parole constituent les deux entrées des systèmes de détection automatique du locuteur (Figure **2-4**).

Applications:

Comme pour la VAL, les applications actuelles de la détection des locuteurs sont principalement liées à la sécurisation de service (authentification de l'interlocuteur dans une communication téléphonique pour la validation de transactions, etc.). Cependant, d'autres applications du domaine de l'indexation de documents multimédia telles que la recherche d'information dans un document audio numérisé où la navigation dans les données sonores sont actuellement étudiées. Ainsi, les futurs moteurs de recherche permettront sans doute de retrouver des fichiers audio contenant la voix d'un individu donné.

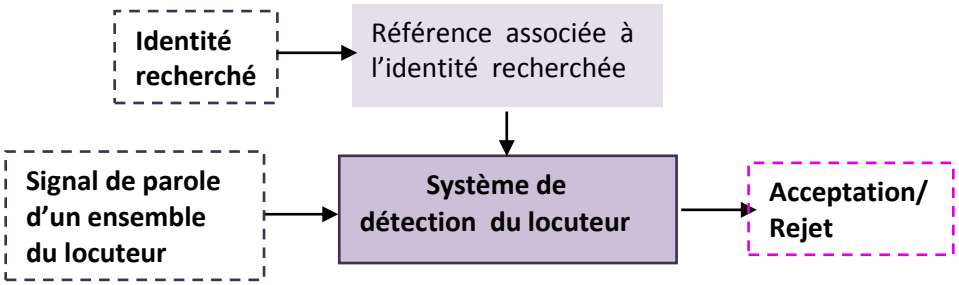


FIG 2-4. Schéma d'un système de détection du locuteur

2.2.1.4 Suivi de locuteurs - Indication par locuteurs

Le suivi de locuteurs consiste à segmenter un signal de parole pour indiquer les instants et durées de prise de parole d'un locuteur cibles (qui parle et quand ?). L'identité de ce locuteur ainsi qu'un signal de parole multi-locuteurs sont les entrées du système.

L'indexation par locuteur consiste à déterminer le nombre de locuteurs présents sur un document audio ainsi que leurs intervalles de prise de parole. Les systèmes d'indexation du locuteur fonctionnent sans aucun a priori sur l'identité des locuteurs présents sur l'enregistrement sonore.

Les schémas du fonctionnement d'un système d'indexation du locuteur et de suivi de locuteurs sont présentés respectivement sur les figures **2-5** et **2-6**.

Applications:

Le domaine d'application de ces deux tâches est principalement le traitement de bases de données audio. Citons par exemple **[10]**:

La transcription automatique de journaux télévisés, le regroupement automatique des messages ou la poursuite d'une personne dans des archives audio, la recherche d'information dans des séquences d'émissions télévisées ou radiophoniques, l'estimation du temps de parole de chaque intervenant lors d'un débat, etc.…

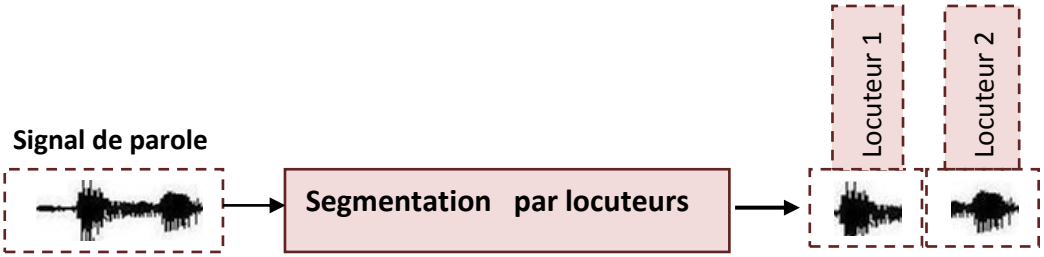


FIG 2-5. Schéma d'un système d'indication du locuteur

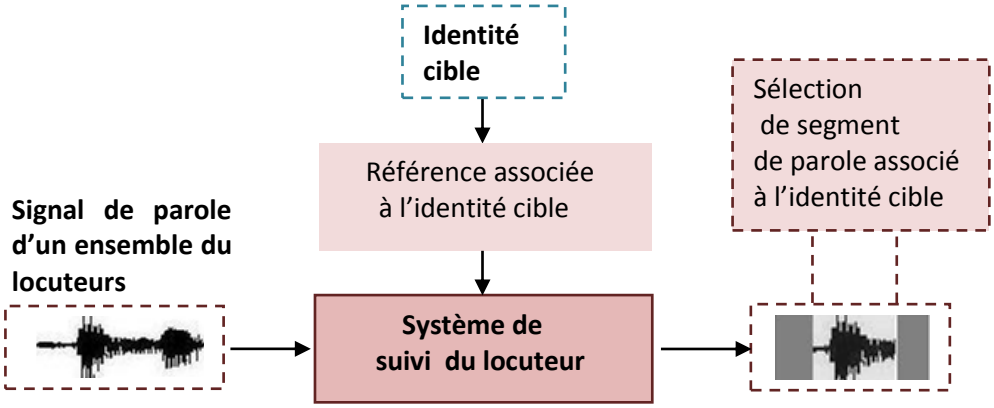


FIG 2-6. Schéma d'un système de suivi du locuteur

2.2.2 Scénarios

Savoir la tâche de reconnaissance du locuteur que nous voulons réaliser, ce n'est pas toute l'histoire pour construire un système de reconnaissance, il faut suivre toute une démarche en posant des questions, comme :

- **quelle est le mode de dépendance au texte ?**
  - Système à texte fixe dépendant du locuteur ?
  - Système à texte libre ?
  - ... etc.

- **Que savons-nous des références connues des locuteurs ?** : population inconnue, taille de la population connue, le sexe de la population, la ou les langue(s) parlée(s) par la population, la durée des références, la qualité d'enregistrement ;
- **Qu'en est-il du signal audio disponible pour la tâche ?** : sa taille, un locuteur présent ou plusieurs, sa qualité d'enregistrement ;
- **le matériel disponible** : la puissance de calcul et de stockage. Le traitement est-il différé ou en temps réel ?

Une telle étude est nécessaire pour limiter les méthodes ou techniques les plus efficaces et performantes pour une telle tâche de reconnaissance du locuteur. Par exemple, la technique utilisée en reconnaissance du locuteur en mode indépendant du texte est bien différente de celle utilisé en mode dépendant du texte.

Chapitre 3

Modules d'une plateforme d'un système de reconnaissance du locuteur

Sommaire

Nous avons jusque-là considéré les systèmes de RAL comme étant simplement des « boites noires » permettant d'associer une décision binaire d'acceptation ou de rejet à partir d'une identité prétendue X et d'un signal de parole Y, pour la vérification du locuteur, et d'associer une identité (parmi plusieurs identités) à un signal de parole, pour l'identification du locuteur. Dans ce chapitre, nous décrivons les différentes techniques associées à leur calcul.

Nous présentons ici, brièvement, pour chacun des modules d'une plateforme d'un système de reconnaissance du locuteur, certains des procédés de réalisation publiés dans la littérature et avec plus de détails le procédé utilisé dans notre réalisation du système de reconnaissance. Le choix d'un tel procédé doit être effectué en fonction de l'application de reconnaissance à réaliser (Paragraphe **2.2.2**).

La figure **3-1** représente une plateforme de reconnaissance du locuteur. On y trouve tous ses éléments essentiels. Pour chacun d'eux sont indiqués les flux de données entrants et sortants:

- le module d'extraction des paramètres caractéristiques, il s'agit du module d'analyse acoustique,
- le module de modélisation qui permet de calculer la référence caractéristique du locuteur,
- le module de calcul de la mesure de similarité,
- le module de décision.

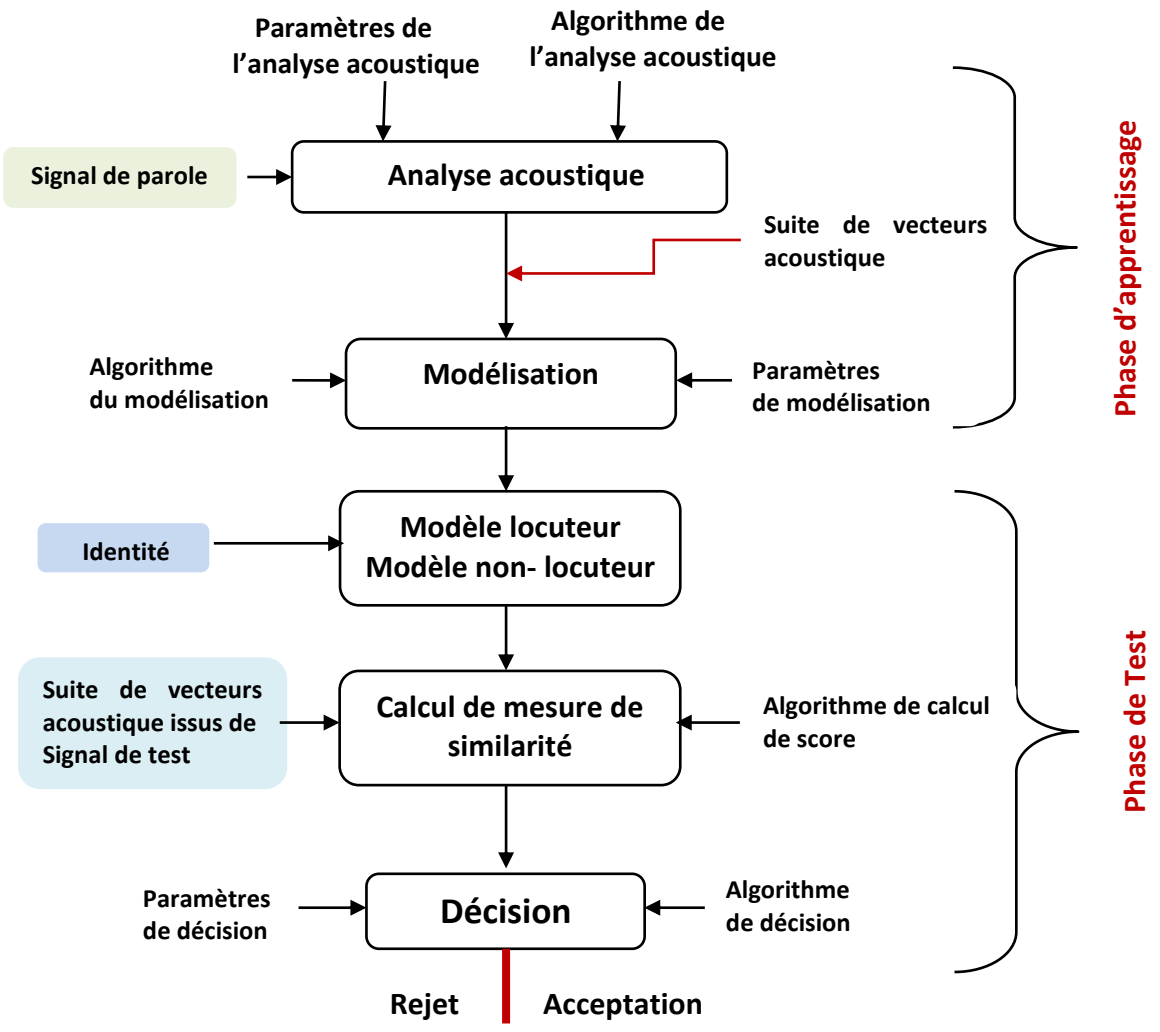


FIG 3-1. Description d'un système de vérification du locuteur.

3.1 Analyse acoustique

Le module d'analyse acoustique décrit sur le schéma de la figure **3-2** a pour but d'extraire la représentation du signal de parole **y** dans l'espace des paramètres acoustiques **Y**. Elle permet d'obtenir la suite $\mathbf{Y}=\{\boldsymbol{y_1} \ldots . \boldsymbol{y_N}\}$ de vecteurs acoustiques qui sont, selon le mode de fonctionnement du système (en phase d'apprentissage ou en phase test), utilisés pour estimer la référence caractéristique du locuteur ou calculer le score de décision. L'analyse acoustique permet de quantifier sous la forme d'une représentation multidimensionnelle toutes les grandeurs contenues dans **y** et capables de nous renseigner sur l'identité du locuteur.

Les performances des systèmes de RAL dépendent en grande partie de la qualité de la représentation acoustique choisie. Les propriétés souhaitées pour ces vecteurs acoustiques sont:

- qu'ils contiennent une information la plus caractéristique possible du locuteur,
- qu'ils soient robustes aux bruits et aux distorsions de canal qui perturbent le signal de parole lors de son acquisition,
- qu'ils soient rapidement calculables.

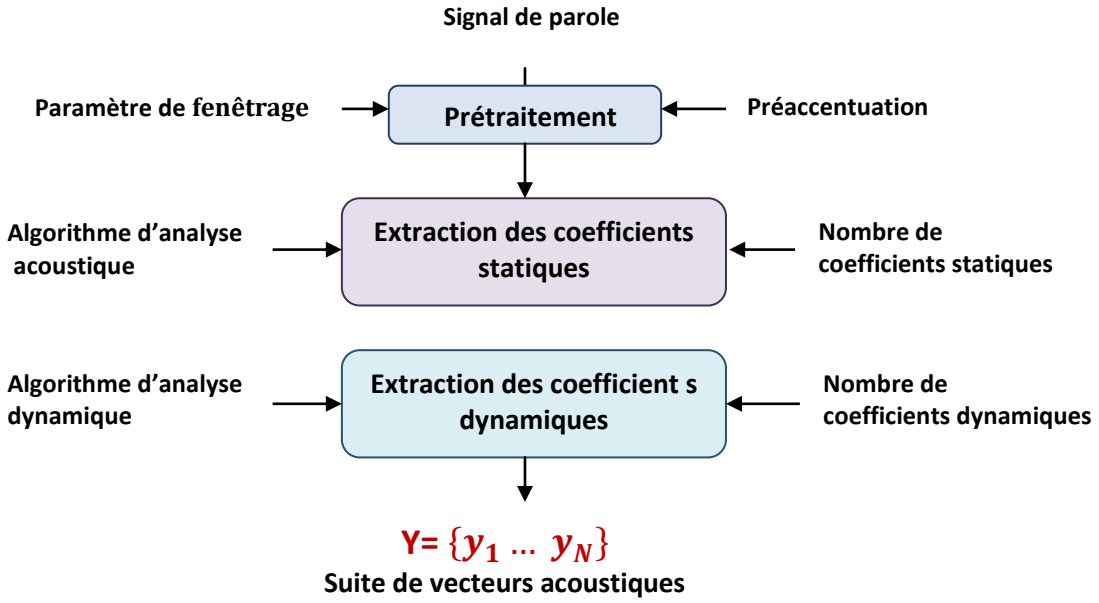


FIG 3-2. Principe d'analyse acoustique.

L'obtention du vecteur de représentation acoustique peut se décomposer en deux étapes. La première appelée étape de prétraitement a pour rôle principal d'augmenter la robustesse des paramètres qui sont calculés à l'étape suivante.

La seconde où l'on calcule les vecteurs acoustiques de la représentation. À l'issue de cette seconde étape, on dispose des vecteurs acoustiques homogènes à ceux qui sont utilisés par le module de modélisation ou de calcul du score.

3.1.1 Etape de prétraitement

La première étape de la paramétrisation acoustique consiste à découper, à cadence régulière, le signal de parole en fenêtres d'analyse appelées trame. C'est à partir de ces trames que sont extraits les vecteurs de paramètres acoustiques. La taille de la fenêtre est fixée a priori selon une approximation de la durée moyenne de stationnarité du signal de parole ≈20 ms. Á ce niveau, différents traitements peuvent être effectués tels que le filtrage de (pré-accentuation), la décomposition parole/non-parole afin de ne conserver que les zones de parole.

3.1.2 Calcul du vecteur de représentation

Comme nous l'avons vu auparavant (Paragraphe **2.1.4.2**), le signal de parole est modélisé par une composante source et une composante conduit vocal. Les paramètres caractérisant le conduit vocal sont les plus fiables à utiliser en RAL.

Il existe de nombreuses techniques permettant de caractériser le conduit vocal du locuteur, qui est représenté par l'enveloppe du spectre de fréquence, parmi lesquelles le codage prédictif linéaire LPC et l'analyse spectrale **[11]**. Cependant, l'analyse spectrale est la technique la plus utilisée pour représenter le signal de parole en RAL, parce qu'elle approxime l'enveloppe spectrale en un nombre réduit de coefficients, en plus elle présente une plus grande robustesse d'estimation sur des signaux bruités **[11]**.

3.1.3 L'extraction des coefficients cepstraux

En RAL, entre 15 et 20 coefficients cepstraux sont utilisés pour modéliser un locuteur. Ils sont généralement extraits toutes les 10 ms (hypothèse de pseudo-stationnarité), calculés sur une fenêtre d'analyse de type Hamming ou Hanning de 20 à 30 ms.

Une analyse en banc de filtre à échelle linéaire ou échelle de Mel est utilisée dans le calcul des coefficients cepstraux (LFCC ou MFCC). Les dérivées premières, ou coefficients Δ (vitesse), et parfois dérivables secondes, ou coefficients ΔΔ(accélération), des coefficients cepstraux sont ajoutés au vecteur de paramètres pour modéliser leurs trajectoires dans le temps.

L'énergie du signal joue aussi un rôle important en RAL tant au niveau de la sélection des données utiles que comme paramètre. En effet, ce paramètre est souvent utilisé pour la détection des segments de parole, et sa trajectoire (Δ-log-énergie) est souvent ajoutée au vecteur de paramètres.

3.2 Modélisation des locuteurs

La modélisation des locuteurs est une phase très importante en reconnaissance automatiques du locuteur. Elle permet, à partir d'une suite de vecteurs extraits du signal de parole d'un tel locuteur, d'obtenir la référence (le modèle) caractéristique qui le représente. Les principales propriétés souhaitées pour cette référence sont :

- Elle doit nécessiter le plus faible espace de stockage possible.
- Elle doit avoir une méthode d'estimation la moins complexe possible.
- Elle doit permettre une décision rapide lors de la phase de test.
- Elle doit être le plus robuste possible aux variations intra-locuteur.
- Elle doit permettre la meilleure séparation des locuteurs entres eux.
- Elle doit avoir la représentation la plus complète possible des paramètres acoustiques des locuteurs.

Il existe plusieurs approches de modélisation utilisées dans les systèmes de reconnaissance automatique du locuteur. Nous pouvons distinguer deux grandes familles pour la représentation des locuteurs :

- Approches basées sur le calcul d'une distance euclidienne entre les vecteurs extraits du signal d'accès et d'autres représentant le locuteur.
- Approches correspond à celles basées sur une représentation statistique du locuteur dans l'espace des paramètres. Notre travail est basé sur cette approche, et plus précisément, sur la méthode de mélange de gaussiennes **GMM**, qui sera étudié en détails dans le quatrième chapitre.

3.2.1 Approche vectorielle

Dans cette approche, la référence caractéristique (ou modèle) du locuteur est une suite de vecteurs. Lors de la phase de test, le score de décision est basé sur le calcul de la distance entre ces vecteurs et ceux extraits du signal d'accès. Parmi les technologies appartenant à cette famille de méthode, citons l'algorithme DTW et la quantification vectorielle (VQ).

3.2.1.1 L'alignement temporel dynamique (DTW- Dynamic Time Warping)

L'alignement temporel dynamique (**DTW [12])** est un modèle basé sur le calcul d'une distance entre deux vecteurs. Principalement, il fait la comparaison d'une séquence de $M\ (X_1,\ldots, X_M)$ vecteurs avec une autre séquence de $N\ (X_1,\ldots, X_N)$ vecteurs par le calcul de la distance euclidienne accumulée entre ces deux séquences. Si les deux séquences sont identiques alors le chemin entre eux est diagonal, et par conséquent, la distance qui les sépare est minimale. Cette méthode est souvent utilisée dans les systèmes de reconnaissance automatique du locuteur dépendante de texte.

3.2.1.2 Quantification vectorielle

Dans cette méthode l'espace acoustique est partitionné en régions qui seront représentées par leur vecteur moyen. La distance d'un vecteur acoustique à cet espace est obtenue en mesurant la distance avec chaque vecteur moyen des régions et en retenant celle qui est minimale. La génération du dictionnaire est donnée par la recherche d'un partitionnement qui minimise la distorsion moyenne des données d'apprentissage (les paramètres acoustiques). L'ensemble des clusters obtenus est appelé le dictionnaire des vecteurs, qui représente un seul locuteur **[13]**.

Dans les applications qui nécessitent une optimisation de l'espace de stockage, la quantification vectorielle est très utile, car la taille du dictionnaire des vecteurs est très petite. Lors de la phase de test, on fait la comparaison des trames extraites d'un signal vocal avec le dictionnaire des vecteurs. Cette comparaison est faite en calculant la distance qui sépare les trames en entrée avec les centres des partitions. La distance mesure le degré de similarité entre la voix de test et le modèle d'un locuteur. Cependant, cette méthode élimine beaucoup d'information sur les locuteurs et elle nécessite des paroles très longues pour avoir des informations statistiques stables.

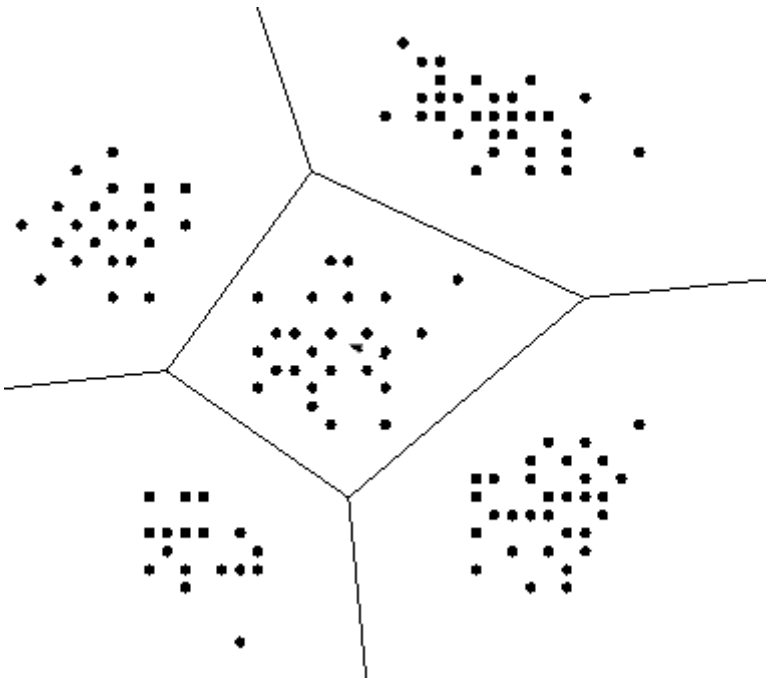


FIG 3-3. Définition des classes de données en quantification vectorielle.

3.2.2 Les approches statistiques

Cette approche est appelé « statistique » car tous les vecteurs de son espace d'entrée doivent avoir la même dimension.

L'approche statistique est la plus utilisée en reconnaissance automatique du locuteur, elle permet la réduction de la taille mémoire nécessaire à la modélisation du locuteur. Elle repose sur l'hypothèse qu'il existe une bijection entre l'ensemble des locuteurs et l'espace des fonctions de densité de probabilité. Cela signifie que les vecteurs de paramètres provenant d'un locuteur suivent une loi de probabilité propre à ce locuteur.

Trois densités sont principalement utilisées ; les Modèles Statistiques du Second Ordre (MSSO), les Modèles de Mélange de Gaussiennes (GMM) et les Modèles de Markov Cachés (HMM).

Nous citons aussi parmi les approches statistiques, les Machines à Vecteurs Supports (SVM), que nous allons présenter en détails au chapitre 5.

3.2.2.1 Machines à Vecteurs Support (SVM)

SVM (Support Vector Machines) est une technique d'apprentissage statique, proposée par V. Vapnik en 1995. Elle permet d'aborder des problèmes très divers comme la classification, la régression, la fusion,…etc.

Depuis son introduction dans le domaine de la Reconnaissance de Formes (RDF), plusieurs travaux ont pu montrer l'efficacité de cette technique principalement en traitement d'images **[14]**.

L'idée essentielle consiste à projeter les données de l'espace d'entrée (appartenant à des classes différentes) non linéairement séparables, dans un espace de plus grande dimension appelé espace de caractéristiques, de façon à ce que les données deviennent linéairement séparables **[15]**. Dans cet espace, la technique de construction de l'hyperplan optimal est utilisée pour calculer la fonction de classement séparant les classes. En tant que classifieurs binaires, les SVM sont très bien adaptés à la tâche de vérification du locuteur (sujet du chapitre 5).

3.2.2.2 Les Modèles Statistiques du Second Ordre

Dans cette approche **[16]**, les locuteurs sont représentés par une loi Gaussienne, un triplet (μ, Σ, X) où μ est le vecteur moyen de la Gaussienne et Σ la matrice de covariance estimée à partir de la suite de vecteurs acoustiques d'apprentissage. En considérant que chaque vecteur acoustique extrait d'un signal de parole est une réalisation d'une variable aléatoire multidimensionnelle, la densité d'une distribution normale pour ce vecteur noté X à D dimensions est exprimée par :

$$N(\mu, \Sigma, \mathrm{X}) = \frac{1}{(2\pi)^{D/2}\ |\Sigma_i|^{\frac{1}{2}}}\ exp\ .\left\{\frac{-1}{2}\ (\vec{x} - \vec{\mu})\Sigma_i^{-1}(\vec{x} - \vec{\mu})\right\} \qquad (3.1)$$

L'avantage des Méthodes Statistiques du Second Ordre (MSSO) est qu'elles permettent une modélisation simple, c'est à dire peu de paramètres à estimer. Elles fournissent de bons résultats sur des courtes durées.

3.2.2.3 Les Modèles de Mélange de Gaussiennes (GMM)

Dans cette approche, les locuteurs sont représentés par une combinaison de plusieurs composantes Gaussiens **[17]**. Ces mélanges de Gaussiennes multi-variées (Gaussian Mixture Model-GMM) constituent à l'heure actuelle l'essentiel des méthodes état de l'art. Lors de la phase d'apprentissage, un nombre de (K*(μ, Σ, X)) paramètres à estimer. La modélisation des locuteurs par GMM donne de très bonnes performances en reconnaissance du locuteur indépendante du texte **[18]**. Cependant, leur inconvénient est le fait que pour une bonne modélisation (i.e. beaucoup de Gaussiennes) nécessitent beaucoup de données. Cette méthode est présentée en détails au chapitre 4.

3.2.2.4 Les Modèles de Markov Cachés (HMM)

Cette technique intègre à la fois les propriétés des distributions de probabilité et celles d'une machine à état (dite chaine de Markov) et permet donc de modéliser des processus stochastiques variant dans le temps comme le signal de parole. Le modèle de Markov caché est un modèle statistique séquentiel qui suppose que les caractéristiques

observés forment une succession d'états distincts. Un tel modèle est entièrement caractérisé par la donnée de trois jeux de paramètres :

- Les probabilités initiales de se trouver dans chaque état.
- Les probabilités de transition qui décrivent les passages possibles entre les différents états.
- Les probabilités de sortie qui à proprement parler représentent les distributions conditionnelles des caractéristiques observées en fonction de l'état du modèle.

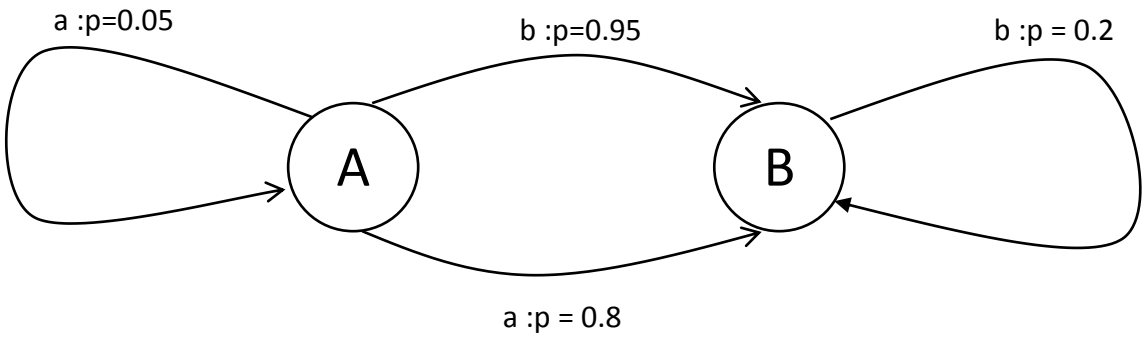


FIG 3-4. Exemple de chaîne de Markov.

Il existe plusieurs types de modèles de Markov qui correspondent aux différents choix possibles en ce qui concerne le second, et surtout, le troisième jeu de paramètres du modèle. Pour la reconnaissance du locuteur, le choix le plus fréquent consiste à utiliser un modèle de Markov où la distribution conditionnelle dans chaque état est un mélange de distributions gaussiennes.

L'approche HMM tient compte de l'enchainement temporel des trames du signal de parole, c'est pour cette raison elle est très utilisée en reconnaissance de parole et reconnaissance du locuteur en mode dépendant du texte et elle donne de très bons résultats.

3.2.3 L'approche connexionniste (réseaux neurones)

La technique basant sur les réseaux de neurones a été assez largement utilisés en reconnaissance du locuteur. Elle offre une bonne discrimination entre les locuteurs. Elle permet de séparer des classes, dans un espace de représentation donné, de façon non linéaire.

L'inconvénient important de l'application de cette technique en identification du locuteur est le coût important lié à l'ajout d'un nouveau locuteur dans la base de référence (ce n'est pas le cas en vérification du locuteur).

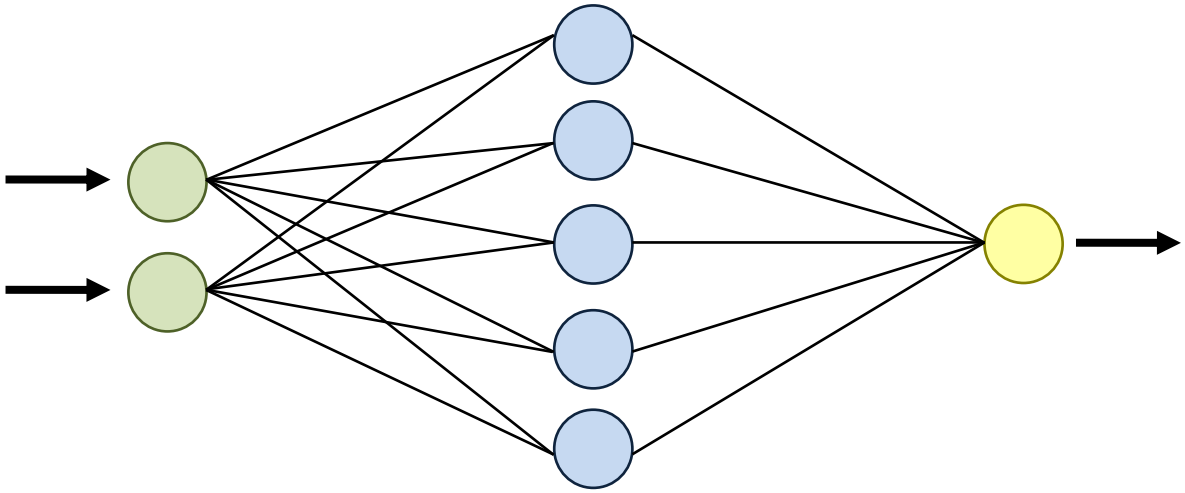


FIG 3-5. Exemple d'un réseau de neurone à deux entrées et une sortie.

3.2.3 L'approche relative

Cette nouvelle technique consiste à modéliser un locuteur non plus de façon absolue mais relativement à un ensemble de locuteurs bien appris.

3.3 La prise de décision

3.3 .1 Le calcul de mesure de la similarité

Le module de calcul de la similarité compare les paramètres extraits du signal à un modèle d'individu calculé lors de la phase de modélisation. Lors de cette comparaison, le module de calcul de la similarité calcule une mesure de similarité entre les données d'entrée et le modèle testé. C'est une valeur numérique, aussi appelée score. Ce dernier est dans la plupart du temps une distance, une probabilité ou une vraisemblance.
Cette mesure de similarité est ensuite transmise au module de décision qui doit, à partir de ce score, fournir une décision qui constituera la réponse finale du système de reconnaissance du locuteur.

La prise de décision en RAL est basée sur le formalise probabiliste. Elle est différente dans l'identification de celle qui suivi dans la vérification du locuteur.

3.3.2 Décision pour L'IAL

Considérons une population de locuteur i = 1,.....,N avec M_i la référence associée au locuteur i. L'identité retournée M, présente dans le signal X de test, est alors celle qui maximalise la probabilité :

$$M = \text{argmax}_i P(M_i|X) \tag{3.2}$$

Sans informations a priori sur l'apparition des locuteurs, P(Mi), et en appliquant la règle de Bayes la relation devient :

$$M = \text{argmax}_i P(M_i|X) = \text{argmax}_i \frac{p(X|M_i) \cdot P(M_i)}{P(X)}$$
$$\text{argmax}_i p(X|M_i) \tag{3.3}$$

où $p(X|M_i)$ est la fonction de vraisemblance du locuteur i qui approxime la densité de probabilité des observations du locuteur i. Les performances du système d'identification

du locuteur se dégradent en augmentant le nombre de locuteurs de l'ensemble de référence.

3.3.3 Décision pour la VAL

Dans une application de vérification, la décision est apprise par la méthode suivante :

Considérons une identité proclamée M. Selon l'approche probabiliste, le calcul de la probabilité que le signal $X = \vec{x}_1, \ldots . \vec{x}_T$ ait été prononcé par le locuteur M repose sur le test d'hypothèse suivant :

– H_0: X est une occurrence prononcée par le locuteur M ;
– H_1 : X n'a pas été prononcée par le locuteur M mais par un autre locuteur que M.

Une des deux hypothèses doit être validée par le système de VAL. L'hypothèse H_0 est représentée par la fonction de vraisemblance $p(X|H_0)$ et l'hypothèse H_1 est représentée par la fonction de vraisemblance $p(X|H_1)$. Le problème de vérification est résolu en comparant le rapport de ces deux hypothèses à un seuil de décision. Dans le cadre de la théorie de la décision bayésienne (son avantage principal est d'exprimer le degré de similitude entre deux échantillons différents), le rapport de vraisemblance des deux hypothèses (likelihood ratio) est défini par :

$$LR(X,H_0,H_1) = \frac{P(H_0|X)}{P(H_1|X)} \tag{3.4}$$

En appliquant la règles de Bayes :

$$P(H_i|X) = \frac{P(X|H_i).P(H_i)}{P(X)} \tag{35}$$

$$LR(X, H_0, H_1) = \frac{P(X|H_0).P(H_0)}{P(X|H_1).P(H_1)} \tag{3.6}$$

$LR(X, H_0, H_1) < \theta$ l'hypothèse H_0 est rejetée

$LR(\mathrm{X}, H_0, H_1)$ > θ L'hypothèse H_0 est validée

où θ est le seuil de décision. En pratique les probabilités a priori $\mathrm{P}(H_0)$ et $\mathrm{P}(H_1)$ sont reportées dans le calcul du seuil de décision θ. Le rapport de vraisemblance devient alors :

$$LR(\mathrm{X}, H_0, H_1) = \frac{\mathrm{P}(\mathrm{X}|H_0)}{\mathrm{P}(\mathrm{X}|H_1)} <> \theta.\frac{\mathrm{P}(H_0)}{\mathrm{P}(H_1)} \quad (3.7)$$

La modélisation de l'hypothèse de l'imposture H_1 est réalisée à l'aide d'un modèle du « non-locuteur ». Il représente l'ensemble des locuteurs autres que M. Son estimation est une tâche difficile. Différentes approches sont proposées. La première approche consiste à utiliser une cohorte de locuteurs **[19]**. Les locuteurs peuvent être sélectionnés selon un critère de proximité avec le locuteur M. La vraisemblance de l'hypothèse H_1 est alors une fonction (somme, max, ...) des vraisemblances du signal sur les modèles des locuteurs de la cohorte ($\overline{M_1}$,...,$\overline{M_n}$) :

$$LR(\mathrm{X}|H_1) = f\,(LR(\mathrm{X}|\overline{M_1}),...,LR(\mathrm{X}|\overline{M_n})) \quad (3.8)$$

Une seconde approche consiste à utiliser un modèle unique pour le modèle du « non-locuteur » **[20]**. Ce modèle, dénommé modèle du monde ou UBM (Universal Background Model), est estimé sur une grande quantité d'enregistrements de locuteurs. Il représente toute la variabilité de la parole. La modélisation de l'hypothèse H_0 utilise quant à elle les données disponibles du locuteur.

3.4 Evaluation des systèmes de RAL

Dans cette section, nous allons décrire les différentes mesures les plus utilisées pour évaluer un système de RAL. Mais avant, voici quelques définitions très utiles que nous utilisons très fréquemment dans le domaine de la RAL :

- Client : est un locuteur de la base de données dont le système dispose de son modèle et qui utilise ce système sous sa vraie identité.
- Imposteur : cette notion est spécifique au système de VAL. Un imposteur est tout locuteur utilisant le système sous une identité qui n'est la pas sienne.

3.4.1 Typologie d'erreurs et mesures de performances

En RAL, chaque tâche possède ses propres erreurs. Dans cette section nous rappelons la typologie d'erreurs des deux tâches les plus utilisées, IAL et VAL.

En identification du locuteur, on peut parler de deux types d'erreurs :

- mauvaise identification : c'est le cas où le système propose une identité qui ne correspond pas à celle du locuteur présenté.
- non-détection : cette erreur est caractéristique des systèmes d'identification de locuteur dans un ensemble ouvert. Elle correspond au cas où le système n'a pas pu identifier le locuteur présenté alors que ce dernier a son modèle dans la base de référence.

La mesure des performances des systèmes d'identification du locuteur se base sur le taux d'identification correcte **TIC** obtenu en phase de test :

$$\text{TIC} = \frac{Nombre\ \ de\ tests\ \ correctement\ \ \ identifi\ és}{Nombre\ \ total\ \ de\ tentatives} \tag{3.9}$$

En vérification du locuteur, il existe deux types d'erreurs :

- Fausse Acceptation (FA) : Elle correspond au cas où le système accepte un locuteur qui a proclamée une identité qui n'est pas la sienne. Une fausse acceptation est une erreur où le système accepte un imposteur.
- Faux Rejet (FR) : C'est le cas où le système rejette un locuteur qui a proclamé sa vraie identité. Autrement dit, c'est quand le système rejette un client.

Les mesures de performances d'un système de VAL se basent principalement sur le taux des fausses acceptations (TFA) et le Taux des Faux Rejets (TFR) obtenus en phase de test :

$$\text{TFA} = \frac{\text{Nombre\ \ de\ tests\ \ ayant\ \ amen é à une\ \ fausse\ \ acceptation}}{\text{tests\ \ imposteurs}} \tag{3.10}$$

$$TFR = \frac{\text{Nombre de tests ayant amené un faux rejet}}{\text{tests client}} \qquad (3.11)$$

Les performances d'un système de VAL peuvent être présentées sous forme d'une seule courbe appelée courbe DET (Detection Error Tradeoff) **[21]** (figure **3-6**) sur laquelle les TFA sont données en fonction des TFR.

Pour construire cette courbe, on calcule un couple (TFA, TFR) pour chaque valeur de seuil de décision variant de la plus petite valeur des scores obtenus en phase de test à la plus grande valeur. Les performances des systèmes de VAL sont souvent comparés selon un point particulier de ces courbes qui est le Taux d'Égale Erreur (TEE, ERR en anglais) et qui correspond au point de la courbe où TFA=TFR. Une autre mesure permet d'évaluer les performances d'un système de VAL est HTER (Half Total Error Rate). Cette mesure est utilisée quand le seuil de décision est fixé à priori. Le *HTER* représente la moyenne de TFA et TFR.

$$HTER = (TFA + TFR)/2$$

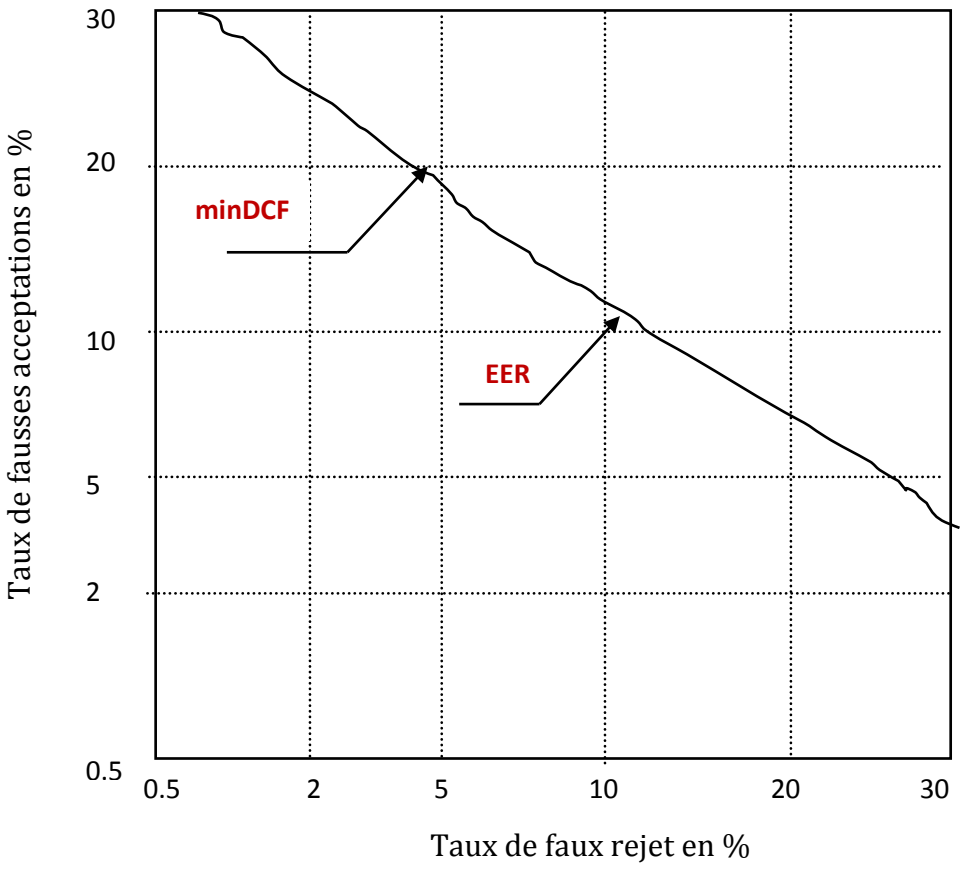


FIG 3-6. Exemple de représentation des performances d'un système de vérification du locuteur par une courbe DET.

Nous utilisons aussi les courbes ROC (Receiver Operating Characteristics) **[23]** pour représenter les valeurs TFA, TFR et EER en fonction du seuil, comme illustré dans la figure **3-7**.

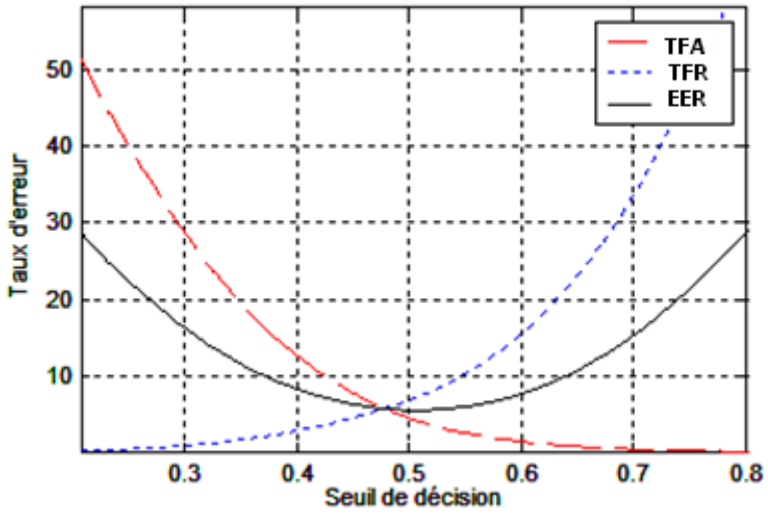


FIG 3-7. Courbe ROC

3.4.2 Points de fonctionnement

Le module de décision reçoit, en entrée et pour chaque test, un score, résultant de la comparaison entre les caractéristiques de l'utilisateur testé et la référence apprise lors de la phase de la modélisation. Un score élevé signifiera que la probabilité pour que l'utilisateur testé corresponde à l'identité qu'il annonce est élevée et un score faible signifiera que cette probabilité est faible. La décision binaire qui constitue la sortie du module résulte de la comparaison de ce score avec un seuil défini à l'avance. Si le score est supérieur au seuil, l'utilisateur est accepté et s'il est inférieur au seuil, l'utilisateur est rejeté.

Le choix d'un seuil a une influence directe sur les performances du système. Pour un système idéal, les scores obtenus par les clients seront tous plus élevés que les scores obtenus par les imposteurs. Dans ce cas, le seuil à fixer se situe entre le score imposteur le plus élevé et le score client le plus faible, assurant ainsi une authentification parfaite (figure **3-8**).

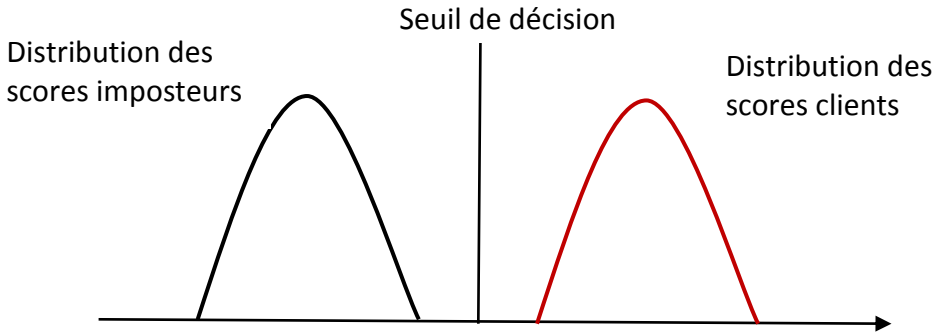


FIG 3-8. Répartition des scores clients et imposteurs et seuil de décision d'un système parfait.

En pratique, les distributions des scores clients et imposteurs se superposent partiellement. Ce cas ne permet pas une reconnaissance parfaite et des erreurs de type faux rejets et fausses acceptations apparaissent. Le choix du seuil influe sur le taux de faux rejets et de fausses acceptations. Le taux de faux rejet est proportionnel au seuil appliqué, plus le seuil est élevé plus le taux de faux rejet sera important. Concernant le taux de fausse acceptation, il est inversement proportionnel au seuil, plus le seuil est élevé, plus le taux de fausse acceptation est faible. En conséquence, il est impossible de réduire les deux taux en même temps en jouant uniquement sur la valeur du seuil. Les figures **3-9**, **3-10** et **3-11** illustre l'influence de choix de seuil de décision.

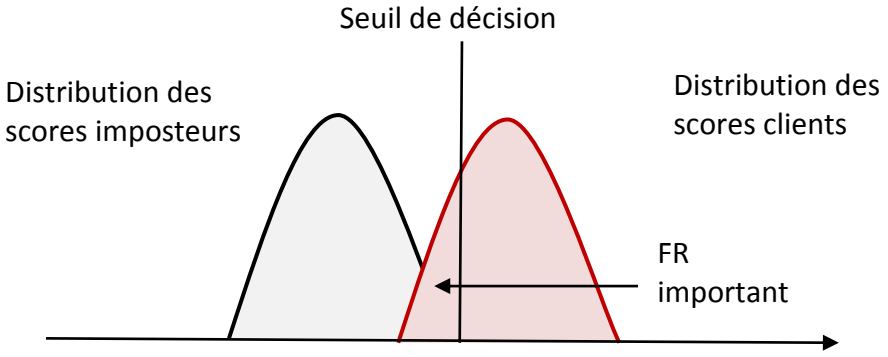


FIG 3-9. Répartition des scores clients et imposteurs avec FR >FA

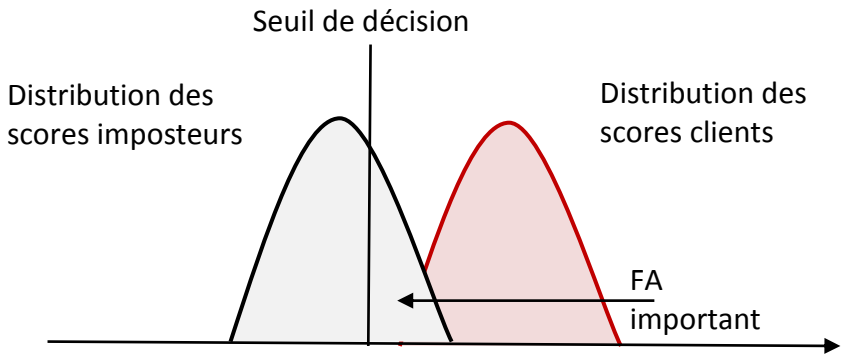


FIG 3-10. Répartition des scores clients et imposteurs avec FR < FA

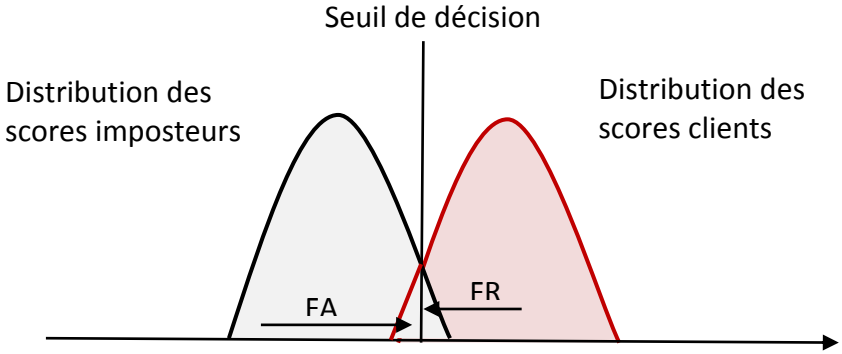


FIG 3-11. Répartition des scores clients et imposteurs avec FR = FA

Dans le cadre d'une application réelle, le seuil sera fixé pour une valeur minimale d'une fonction coût (DCF, Decision Cost Function). Du fait des conséquences liées aux fausses acceptations, on préfère, en général, rejeter un client qu'accepter un imposteur. Le coût associé au taux de fausse acceptation TFA sera donc plus élevé que le coût associé au taux de faux rejet TFR. La fonction, détaillée dans l'équation **3.12**, est souvent utilisée pour résumer les performances des systèmes de vérification automatique du locuteur **[23].**

$$DCF = Coût(FR)\ P\ (client)\ TFR + Coût(FA)\ P\ (imposteur)\ TFA \qquad (3.12)$$

Où Coût(FR) et Coût(FA) sont respectivement les coûts d'un faux rejet et d'une fausse acceptation pour l'application choisie. P (client) et P(imposteur) sont les probabilités

qu'un client ou un imposteur utilise le système. Une telle fonction correspond à des points de fonctionnement optimaux en haut à gauche de la courbe DET (faibles taux de fausses alarmes).

Une autre mesure de performance largement utilisée dans la vérification automatique du locuteur est le taux d'égale erreur (Equal Error Rate ou EER), où les deux taux d'erreurs sont égaux TFA= TFR.

Chapitre4

Approche GMM-UBM

Sommaire

Ce chapitre présente l'approche statistique GMM/UBM, majoritairement utilisée en VAL indépendante du texte. La modélisation des locuteurs repose sur les modèles à base de mélanges de Gaussiennes (GMM). L'hypothèse inverse (paragraphe **3.3.3**) dans la théorie bayésienne est réalisée à l'aide du modèle du monde (**Universal Background Model**). Le schéma de fonctionnement d'un système de VAL est représenté sur la figure **4-1**.

4.1 Schéma général

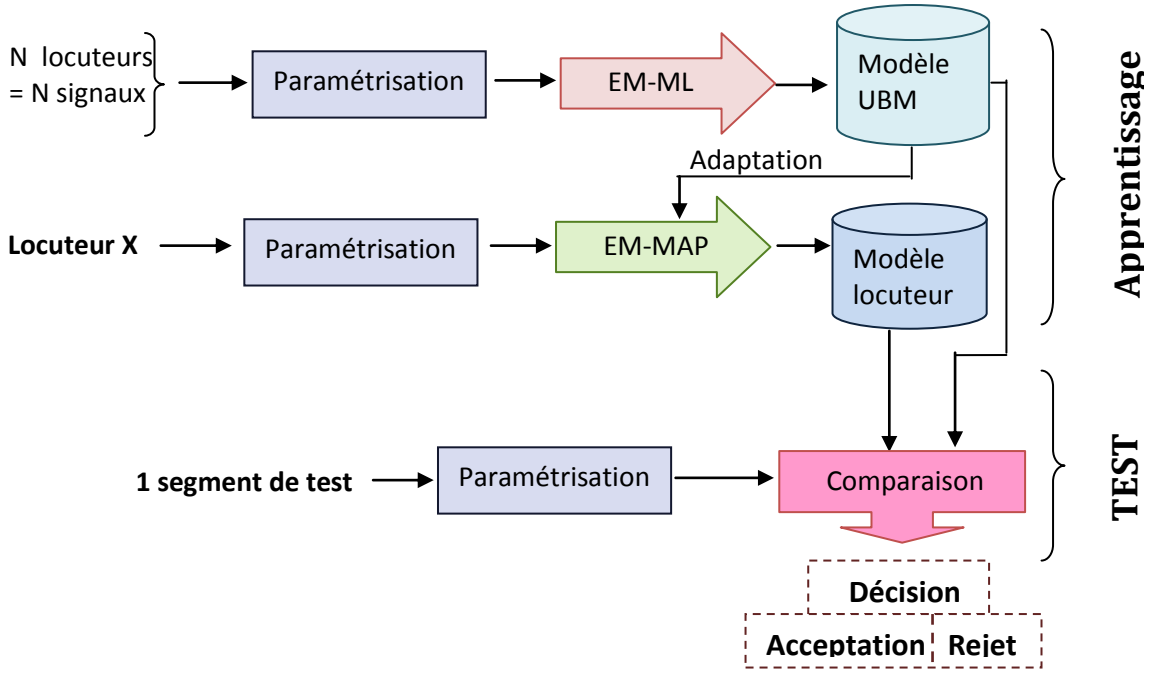


FIG 4-1. Schéma de la méthode GMM-UBM pour la VAL indépendante du texte.

Les différents modules représentés sont :

- le module « Paramétrisation ». Il permet d'extraire les paramètres du signal de parole pertinents pour la VAL.
- les modules «Modèle UBM et modèle locuteur» estiment, à partir des données d'apprentissage, le modèle statistiques non locuteur et le modèle de locuteur cible respectivement.
- le module « Comparaison » calcule la mesure de similarité entre l'échantillon de test et le modèle de locuteur cible. Il fournit la décision de vérification. La suite de ce chapitre décrit chacun de ces modules.

4.2 La paramétrisation du signal de parole

L'objectif de cette phase de reconnaissance est d'extraire des coefficients représentatifs du signal de parole. Ces coefficients sont calculés à intervalles temporels réguliers. En simplifiant les choses, le signal de parole est transformé en une série de vecteurs de coefficients, ces coefficients doivent représenter au mieux ce qu'ils sont

censés modéliser et doivent extraire le maximum d'informations utiles pour la reconnaissance.

Nous avons déjà vu, au chapitre précédant, que les coefficients cepstraux sont les coefficients les plus utilisés et qui représentent au mieux le signal de la parole en reconnaissance du locuteur. Dans notre système nous avons utilisé l'analyse cepstrale (Mel-ScaleFrequency Cepstral Coefficients (MFCC)) pour les extraire.

4.2.1 Les coefficients MFCC

L'extraction de coefficients MFCC est basée sur l'analyse par banc de filtres qui consiste à filtrer le signal par un ensemble de filtres passe-bande. L'énergie en sortie de chaque filtre est attribuée à sa fréquence centrale. Pour simuler le fonctionnement du système auditif humain, les fréquences centrales sont réparties uniformément sur une échelle perceptive. Plus la fréquence centrale d'un filtre est élevée, plus sa bande passante est large. Cela permet d'augmenter la résolution dans les basses fréquences, zone qui contient le plus d'information utile dans le signal de parole

Le processus de l'extraction des coefficients MFCC suit les étapes selon le schéma de la figure **4-2 [24]**.

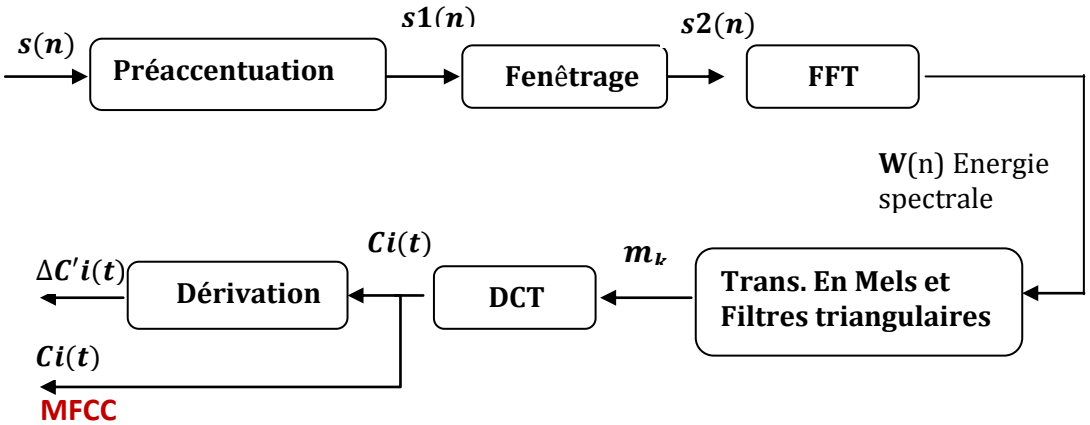


FIG 4-2. Calcul des coefficients MFCC.

4.2.1.1 Préaccentuation

Le signal de parole est représenté à partir de maintenant par une famille S(n) avec n [1, N] où N est le nombre d'échantillons dans le signal. Chaque élément de la famille est un réel.

Les sites traitant de la reconnaissance vocale par MFCC précisent que le signal doit tout d'abord suivre une préaccentuation pour palier le fait que les hautes fréquences seront moins puissantes que les basses fréquences. La formule de préaccentuation d'un signal est de la forme :

$$S_1(n) = S(n) - \alpha.S(n-1) \qquad n = 1, ..., N \text{ et} \qquad S_1(0) = S_1(0) \tag{4.1}$$

Dans notre système, on prend $\boldsymbol{\alpha} = 0.97$.

La parole est un son complexe dans son ensemble, mais si on considère des intervalles de temps très réduits (environ 10 - 30 ms), le signal vocal est alors presque stationnaire. C'est alors pour cela que l'on considère le signal vocal comme un signal quasi-stationnaire. En plus, un signal audio ne peut pas être traité dans son ensemble car cela demanderait énormément de calculs pour la machine (impossible à réaliser). Une fois la formule de préaccentuation du signal est effectuée, nous découpons le signal en fenêtres. Le signal est découpé en tranches de $2^{\text{échantillons}}$ appelées trames ou encore fenêtres qui ont la particularité de se recouvrir de moitié dans l'objectif d'avoir un meilleur traitement pour FFT (Fast Fourier Transform). Le nombre d'échantillon N dans une trame est déterminé par la formule suivante: N=Fe/D ; D est la duré de trame (10 à 30 ms), Fe est la fréquence d'échantillonnage, ce nombre doit être une puissance de 2, cela vient du fait que l'algorithme FFT que nous utilisons est bien plus rapide pour ces nombres. Dans notre programme principal de reconnaissance du locuteur nous utilisons des fenêtres de 1024 échantillons.

4.2.1.2 Fenêtrage Hamming

Dans cette étape on applique sur chaque trame une pondération de Hamming, cette pondération ayant pour le but d'une part de concentrer la répartition de l'énergie

sur les basses fréquences et d'autre part d'amoindrir les fortes variations du signal sur les bords de la fenêtre, afin de permettre un meilleur traitement pour l'algorithme FFT.
La fenêtre de pondération Hamming appliquée au signal :

$S_2(n) = S_1(n).\ (0{,}54\text{-}0{,}46.\cos\ (2л\ .n/\ (N\text{-}1)\ -\ л))$ avec : n = 0, ..., N-1 (4.2)

4.2.1.3 FFT

Cette étape consiste à prendre chaque trame et à appliquer la transformée de Fourier, on convertit ainsi chaque trame du domaine temporel en domaine fréquentiel.

La DFT sur la $i^{ème}$ trame est définie par la formule suivante :

$$X_n = \sum_{k=0}^{N-1} x_k\ .\ e^{-2\pi jkn\ /N} \qquad n = 0, ..., N\text{-}1 \qquad (4.3)$$

Où $j = \sqrt{-1}$

4.2.1.4 Calcul d'énergie

Le spectre du signal $s\ (f_n)$ est la sortie du module FFT et l'entrée de ce module. En réalité, c'est une chaîne des nombres dont la longueur dépende du nombre de FFT. Dans cette étape, l'énergie spectrale sera calculée :

$$W_n = |s\ (f_n)\ |^2 \qquad (4.4)$$

4.2.1.5 Banc de filtres Mels

L'étendue de fréquences présentées dans le spectre est très large, donc beaucoup de données à traiter. Pour réduire ces données, nous utilisons un banc de filtres dans l'échelle de Mels. L'usage de l'échelle Mels favorise les basses fréquences au détriment des hautes fréquences qui sont plus faible que les basses fréquences dans les sons de la

parole (Figure **4-3)**. Les coefficients MFCC ont prouvé leur efficacité en traitement automatique de la parole parce qu'ils utilisent cette échelle.

Cette échelle est définie par :

$$Mel(f) = x*\log_{10}(1+f_H/y) \quad (4.5)$$

Où f_H est la fréquence en Hz.

Plusieurs valeurs sont utilisées pour *x* et y par exemple :

$$x = 1000/\log(2) \text{ et } y = 1000.$$

Les valeurs les plus couramment utilisées sont :

$$x = 2595 \text{ et } y = 700.$$

Nous allons donc appliquer ce dernier couple des deux valeurs pour le module d'extraction des MFCC dans notre programme.

L'échelle en fréquence entre 0 et F_e /2 est ainsi partitionnée en M bandes sur l'échelle de Mel, si F_e est la fréquence d'échantillonnage du signal audio (typiquement M = 24). Ensuite, le filtrage est effectué en multipliant l'énergie du spectre du signal W_n par le gabarit des filtres ($H_{[m]}$). Chaque filtre va donner un coefficient cepstrale.

Filtres utilisés

Sur la figure **4-3**, nous voyons les filtres triangulaires dont les bandes passantes sont équivalents en domaine Mel-fréquence. Les points frontières *B[m]* des filtres en Mel-fréquence sont calculés ainsi :

$$B[m] = B(f_l) + m\frac{B(f_h)-B(f_l)}{M+1} \qquad 0 \le m \le M+1 \tag{4.6}$$

Où M est le nombre de filtres, f_h est la fréquence la plus haute et f_l est la fréquence la plus basse pour le traitement du signal.

Dans le domaine fréquentiel, les points *f[m]* discrets correspondants sont calculés par l'équation :

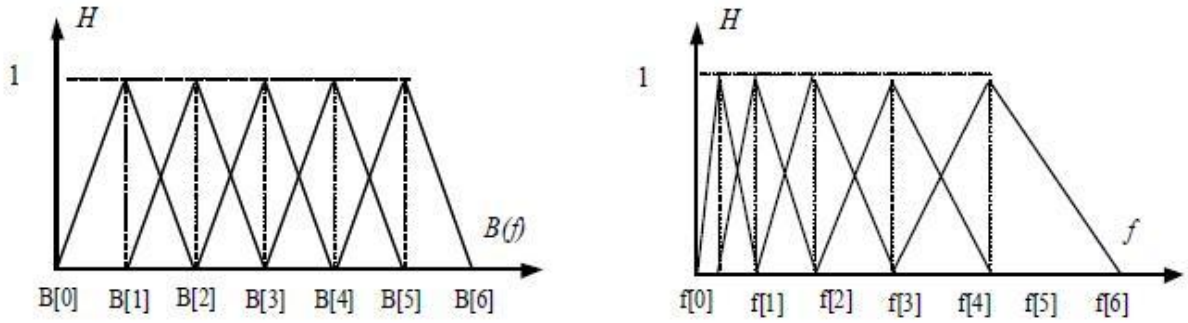
$$f[m] = \frac{N}{F_s} B^{-1}\left(B(f_l) + m\frac{B(f_h)-B(f_l)}{M+1}\right) \tag{4.7}$$

Où B^{-1} est la transformée de Mel-fréquence en fréquence.

$B^{-1}(b) = 700*(10^{\frac{b}{2595}} - 1)$

Le coefficient *Hm[k]* de chaque filtre est déterminé par le système suivant :

$$H_m[k] = \begin{cases} 0 \; si \; k \le f[m-1] \; ou \; k \ge f[m+1] \\ \frac{k-f[m-1]}{f[m]-f[m-1]} \; si \; f[m-1] \le k \le f[m] \\ \frac{f[m+1]-k}{f[m+1]-f[m]} \; si \; f[m] \le k \le f[m+1] \end{cases} \tag{4.8}$$



Mels **Hertz**

FIG 4-3. Figure du banc de filtres espacés sur l'échelle Mel en Hertz et en Mel.

La chaîne des coefficients m_k (k = 1,2, …, K) de K filtres est obtenue par la somme accumulée après avoir transformé W_n en échelle Mels et après avoir passé à travers de chaque filtre.

4.2.1.5 Transformation en Cosinus Discret Inverse

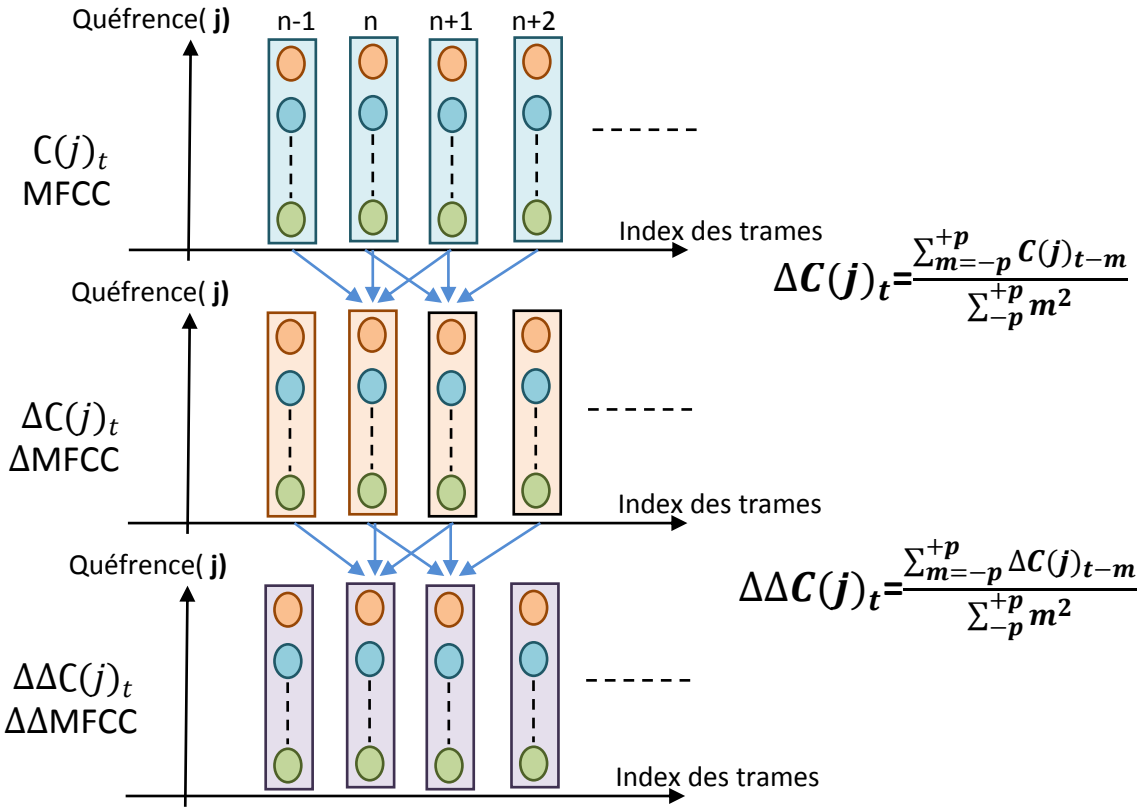
C'est l'étape finale, on transforme les données dans l'échelle des Mels (fréquentielle donc) vers l'échelle des temps. Le résultat de cette étape sera les MFCC proprement dit. Il suffit d'effectuer l'inverse de la transformée de Fourier. Dans la pratique, on effectue une transformée en Cosinus Discrète inverse (iDCT), ce qui revient au même puisque la transformée en Cosinus inverse donne la partie réelle de la transformée de Fourier ; or ici on a que des réels. Il faut noter que la transformée en sinus donnera la partie imaginaire de la transformée de Fourier.
Alors, les valeurs logarithmiques des m_k seront transformées en domaine temporel en utilisant la transformation en Cosinus Discrète inverse :

$$C_i=\sqrt{2/K} \cdot \sum_{j=1}^{K} \ln(m_j) \cdot \cos\left(\frac{\pi . i}{K}(j-0.5)\right) \quad i = 1, 2, ..., Nc \tag{4.8}$$

Où K est le nombre de filtres et Nc est le nombre de coefficients MFCC que l'on souhaite obtenir.

4.2.1.6 Les Delta-MFCC

Ces paramètres sont les dérivées temporelles des MFCC. Ils nous permettent d'avoir une information sur la variabilité temporelle du signal de parole, qui est une de ses caractéristiques importante. Une manière d'avoir la première et la deuxième dérivée des coefficients MFCCS est définie par la figure **4.4 [25]**:



$$\Delta C(j)_t = \frac{\sum_{m=-p}^{+p} C(j)_{t-m}}{\sum_{-p}^{+p} m^2}$$

$$\Delta\Delta C(j)_t = \frac{\sum_{m=-p}^{+p} \Delta C(j)_{t-m}}{\sum_{-p}^{+p} m^2}$$

FIG 4-4. Calcul des dérivées premières et secondes des coefficients MFCC.

4.2.1.7 L'Energie

L'énergie de chaque trame sera ainsi ajoutée au vecteur de coefficients MFCC, elle est mesurée par la formule :

$$E = \log \sum_{n=1}^{N} s_n^2 \,. \qquad (4.9)$$

4.3 Modélisation des locuteurs
Les mélanges de Gaussiennes (GMM)

Comme indiqué dans la section précédente, le signal de parole en reconnaissance du locuteur est représenté par une suite de vecteurs de paramètres caractérisant le locuteur. Chaque vecteur de cette suite est considéré comme une variable aléatoire multidimensionnelle. Les approches statistiques en reconnaissance du locuteur supposent que les vecteurs de paramètres provenant d'un locuteur suivent une loi de probabilité propre à ce locuteur. Les chercheurs proposent plusieurs méthodes pour

trouver une approximation à cette loi de probabilité. Certains d'entre eux ont présentés dans la partie **3.2.2**.

L'avantage d'utiliser la méthode des mélanges de Gaussiennes (GMM) est qu'elle permet d'obtenir une approximation plus précise de la fonction de densité de probabilité caractéristique des locuteurs, tout en restant relativement simple à estimer**[14][21]**. L'autre avantage qui permet d'expliquer le succès des GMM est l'existence d'un outil très puissant pour l'estimation des paramètres qui leur sont associés : l'algorithme *Expectation-Maximisation (EM)*. La mise en œuvre de cet algorithme, et plus particulièrement l'estimation des différents paramètres qui composent un modèle GMM, constitue l'objet principal de cette section.

La densité de probabilité d'un mélange de M distributions Gaussiennes (Figure **4-5**), pour des vecteurs acoustiques $\overrightarrow{x}$ de dimension D ($\overrightarrow{x}$= x_1,..., x_D), est définie par :

$$p(\overrightarrow{x}|\lambda)= \sum_{i=1}^{M} w_i \, P_i(\vec{x}, \mu_i, \Sigma_i) \qquad (4.10)$$

$P_i(\vec{x}, \mu_i, \Sigma_i)$ est la Gaussienne i dans le mélange, définie par un vecteur de paramètres $\vec{x}$ de moyennes μ_i de dimension D, une matrice de covariance Σ_i de dimension D * D et w_i est le poids associé à la Gaussienne dans le mélange (équation **4.11**).

$$Pi(\vec{x}, \mu_i, \Sigma_i)=\frac{1}{(2\pi)^{D/2} \, |\Sigma_i|^{\frac{1}{2}}} \; exp\,.\left\{\frac{-1}{2}\,(\vec{x}-\overrightarrow{\mu_i})\Sigma_i^{-1}(\vec{x}-\overrightarrow{\mu_i})\right\} \qquad (4.11)$$

Les paramètres qui caractérisent le GMM sont : λ = (w_i, Σ_i, μ_i) i [1..M]. La somme des poids w_i du mélange est égale à 1.

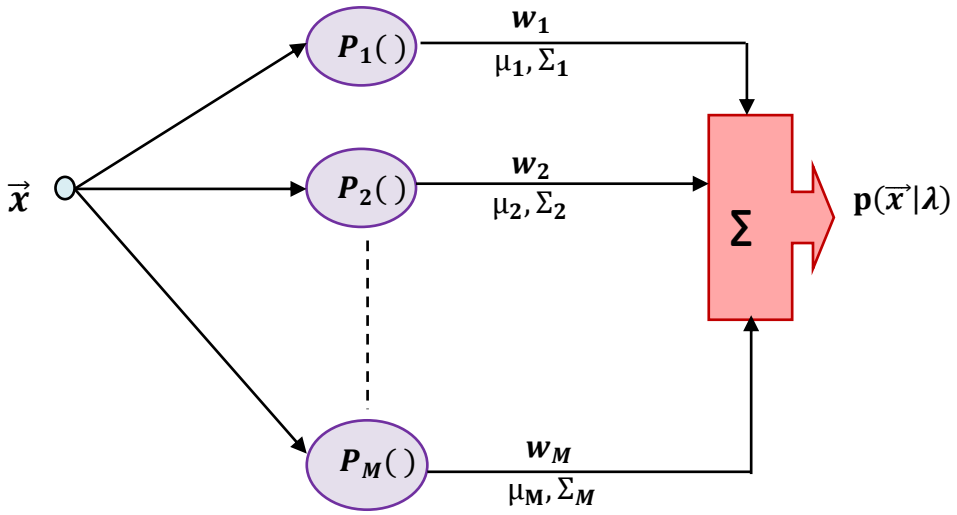


FIG 4-5. Représentation d'un mélange de M gaussiennes **[18]**.

Dans notre système de reconnaissance du locuteur, nous utilisons des matrices de covariances diagonales. Ils sont plus efficaces dans les calculs et plus performants que les matrices de covariance pleines **[21]**.

4.3.1 Modélisation de l'hypothèse de non -locuteur

La probabilité de l'hypothèse de non-locuteur, $p(X|H_1)$, dans le test Bayésien (paragraphe **3.3.3**) est souvent approximée grâce à une cohorte d'imposteurs ou un modèle du monde (Universal Background Model - UBM). La méthode basée sur une cohorte d'imposteurs consiste à réaliser un modèle non-locuteur pour chaque client du système. La difficulté de cette méthode a conduit la plupart des systèmes état de l'art à utiliser un modèle unique (modèle du monde).
Reynolds dans **[21]** propose d'approximer l'hypothèse $p(X|H_1)$ par un modèle universel **W** représentant l'ensemble des locuteurs, excepté le locuteur considéré (et ceci quelque soit le locuteur).

Ce modèle est appris en utilisant des dizaines d'heures de signal audio provenant de multiples locuteurs.

4.3.2 L'apprentissage des modèles GMM

Créer un modèle statistique du locuteur, cela veut dire qu'il faut déterminer les paramètres de ce modèle (w_i,Σ_i, μ_i). Cette étape est réalisée à partir des données extraites du signal de parole dite données d'apprentissage, réordonnées sous forme d'une suite de vecteurs. Un algorithme est utilisé pour estimer les paramètres du modèle en maximisant un critère choisi, par rapport aux données d'apprentissage. Le critère le plus utilisé pour l'apprentissage des modèles GMM, est le critère de maximum de vraisemblance **ML** (maximum likelihood). L'estimation des paramètres du GMM consiste à trouver ceux qui maximisent la fonction de vraisemblance des données d'apprentissage.

$$\widetilde{\lambda_X} = \text{argma}x_\lambda \ P(X|\lambda) \qquad (4.12)$$

Où X est l'ensemble des trames d'apprentissage : $X = \vec{x}_1, \ldots . \vec{x}_T$ et $P(X|\lambda)$ la vraisemblance de X sachant le modèle GMM λ :

$$P(X|\lambda) = \prod_t P(\vec{x}_t|\lambda) \qquad (4.13)$$

Il est très complexe de résoudre l'équation **4.12** à cause du manque d'information des données d'apprentissage. En effet, il est difficile de savoir quelle gaussienne dans le mélange a généré une trame d'apprentissage donnée. Pour résoudre ce problème, appelé problème des données manquantes, l'algorithme Expectation Maximisation (**EM**) **[26]** est souvent utilisé. L'idée principale de l'algorithme **EM** est, en commençant par les paramètres initiaux λ du modèle GMM, on estime les nouveaux paramètres $\tilde{\lambda}$, telle que la vraisemblance du nouveau modèle soit supérieur ou égale à la vraisemblance du modèle initial. En d'autre terme, $P\left(X\middle|\tilde{\lambda}\right) \geq P(X|\lambda)$.

Les paramètres du nouveau modèle seront les paramètres initiaux de l'itération suivante de l'algorithme **EM**, ce processus est répété plusieurs fois jusqu'à atteindre un seuil de convergence.

L'algorithme **EM** est divisé en deux étapes : étape d'expectation ; et étape maximisation. À l'étape **Expectation,** l'algorithme **EM** détermine les probabilités a posteriori que les gaussiennes aient généré les trames d'apprentissage. Les probabilités a posteriori sont calculées, à chaque itération t, à travers la formule suivante :

$$\gamma_{n,m}^{(t)} = \frac{w_m^{(t)} P\left(x_n \middle| \mu_m^{(t)}, \Sigma_m^{(t)}\right)}{\sum_{k=1}^{M} w_k^{(t)} P\left(x_n \middle| \mu_k^{(t)}, \Sigma_k^{(t)}\right)} \quad (4.14)$$

Avec n [1..N], N le nombre de trames, et m [1..M], M le nombre de composantes gaussiennes dans le mélange. $P\left(x_n \middle| \mu_m^{(t)}, \Sigma_m^{(t)}\right)$ est calculée par l'équation **4.11**.

À l'étape **Maximisation**, l'algorithme cherche à calculer les nouveaux paramètres de modèle maximisant la vraisemblance.
Dans chaque itération t de l'algorithme, les poids sont estimés par la formule ci-dessus :

$$w_m^{(t+1)} = \frac{1}{N} \sum_{n=1}^{N} \gamma_{n,m}^{(t+1)} \quad (4.15)$$

Les formules de ré-estimation vecteurs des moyennes sont données par:

$$\mu_m^{(t+1)} = \frac{\sum_{n=1}^{N} \gamma_{n,m}^{(t+1)} x_n}{\sum_{n=1}^{N} \gamma_{n,m}^{(t+1)}} \quad (4.16)$$

Et pour les variances:

$$\sigma_m^{2(t+1)} = \frac{\sum_{n=1}^{N} \gamma_{n,m}^{(t+1)} (x_n - \mu_m^{(t+1)})(x_n - \mu_m^{(t+1)})^T}{\sum_{n=1}^{N} \gamma_{n,m}^{(t+1)}} \quad (4.17)$$

La phase d'initialisation de l'algorithme **EM** est très importante lors de l'apprentissage d'un modèle GMM. Les techniques les plus courantes choisissent aléatoirement des données dans l'ensemble d'apprentissage pour initialiser les

moyennes, la matrice de variance est la matrice unité, et les poids suivent la loi d'équiprobabilité. Les moyennes initialisées du GMM peuvent être réactualisées par l'utilisation de l'algorithme de classification de type **k-means**.

Variance Flooring :

Durant l'apprentissage du modèle GMM, un seuillage des paramètres de variance pour l'estimation des variances des gaussiennes est utilisé, afin d'éviter le sur-apprentissage du modèle. En d'autres termes, si un grand nombre de composantes du mélange est utilisé, la quantité de données utilisée pour estimer chaque composante peut être faible, et l'estimation correspondante de la variance peut être trop faible, ce qui donne lieu à des problèmes numériques (division par zéro) lors des calculs de probabilité. Pour régler ce problème, on impose une valeur minimale pour l'estimation de la variance. En règle générale, ce paramètre est réglé à une fraction de la variance globale de toutes les données du modèle **[27] [28]**.

4.3.3 Adaptation de modèle, critère du Maximum a Posteriori

L'algorithme **EM –ML**, utilisant un critère de maximum de vraisemblance **ML**, donne des bons résultats lorsque la quantité de données disponible pour un locuteur est suffisante. Cependant, dans les meilleurs des cas, on dispose quelques minutes de parole pour chaque locuteur. Cette quantité n'est pas suffisante pour estimer les paramètres du modèle si le nombre de composantes dans le mélange GMM est élevé. Pour remédier à ce problème, une méthode courante consiste à apprendre le modèle GMM d'un locuteur en adaptant le modèle du monde avec les données de ce locuteur. Différents critères d'adaptation existent dans la littérature. La méthode la plus utilisée en reconnaissance du locuteur est celle du Maximum a Posteriori (**MAP**) **[21]**. Il est intéressant de remarquer que cette approche tend généralement vers l'estimation **ML** lorsque la quantité de données disponible pour un locuteur est infinie.

L'adaptation selon le critère **MAP** utilise l'algorithme **EM**, mais considère un modèle a priori. Si λ est le vecteur de paramètres qui doit être estimé d'après les données X, alors le vecteur de paramètres, λ_{MAP}, estimé est :

$$\lambda_{MAP} = \text{argmax}_{\lambda} \, P(X|\lambda)P(\lambda) \tag{4.18}$$

Dans l'approche GMM-UBM, le modèle a priori est le modèle du monde (**UBM**). L'adaptation **MAP** peut être interprétée comme une modification des paramètres du modèle GMM du monde en vue de le « rapprocher » d'un modèle appris sur l'ensemble des données d'apprentissage.
En pratique, seules les moyennes des distributions Gaussiennes sont adaptées pour la reconnaissance du locuteur, les vecteurs variances et poids sont ceux du modèle UBM.

Les nouveaux vecteurs moyennes sont estimés par :

$$\tilde{\mu}_i = \alpha_i \, \mu_i^{\boldsymbol{c}} + (1+ \alpha_i) \, \mu_i^{\boldsymbol{w}} \tag{4.19}$$

$$\alpha_i = \frac{n_i}{\tau+n_i} \tag{4.20}$$

$$n_i = \sum_{n=1}^{N} \gamma_{n,i} \tag{4.21}$$

$\boldsymbol{\gamma_{n,i}}$ est obtenu par l'équation **4.14**.
Où n_i représente l'occupation de la Gaussienne i, N est le nombre de trames, $\mu_i^{\boldsymbol{c}}$ est la moyenne du modèle du locuteur estimée selon le critère **ML**, $\mu_i^{\boldsymbol{w}}$ est la moyenne de modèle UBM, et $\tilde{\mu}_i$ la moyenne adaptée du modèle du locuteur.
$\boldsymbol{\tau}$ est un facteur de régulation. Ce facteur indique en pratique la confiance accordée aux statistiques provenant des données d'apprentissage par rapport à un a priori que sont les statistiques issues du modèle du monde. Le facteur $\boldsymbol{\tau}$ correspond au nombre de trames d'apprentissage nécessaires à l'adaptation d'un paramètre pour accorder le même poids au paramètre appris d'après les données d'apprentissage qu'au paramètre a priori (si l'occupation de la Gaussienne considérée est égale à ce paramètre, alors le

paramètre résultant du **MAP** est une moyenne du paramètre a priori et du paramètre appris par maximum de vraisemblance).
Le modèle ou les paramètres initiales de l'algorithme **EM-MAP** sont celles du modèle du monde.

4.4 Le test de vérification

Le test de vérification permet d'obtenir la mesure de similarité entre un modèle de locuteur et un signal de test. Cette mesure est appelée score de vérification.

4.4.1 Calcul du score vérification

En prenant l'hypothèse d'indépendance des réalisations du vecteur $\vec{x}$, la vraisemblance pour que le test X = $(\vec{x}_1, \dots, \vec{x}_t)$ ait été généré par le GMM λ, est définie comme :

$$\mathrm{P}(X|\lambda) = \prod_{t=1}^{T} w_i \, P_i(\vec{x}_t|\lambda) \tag{4.22}$$

Où P_i est une distribution Gaussienne de moyenne μ_i avec une matrice de covariance Σ_i et où w_i est le poids associé à la Gaussienne dans le mélange, calculée au moyen de l'équation **4.14**.

Le test de vérification repose sur le rapport d'hypothèse défini en section **3.3.2**. En utilisant le modèle du monde comme modèle de l'hypothèse inverse, le rapport d'hypothèse s'écrit sous la forme d'un rapport de vraisemblance (likelihood ratio) :

$$LR(X|S) = \frac{P(X|S)}{P(X|UBM)} \tag{4.23}$$

En pratique, on utilise le logarithmique des vraisemblances, ce qui donne le logarithme du rapport de vraisemblance Log Likelihood Ratio (LLR), pour éviter les problèmes de précision numérique dus aux multiplications de faibles valeurs. Le score de vérification utilisé en VAL est alors défini comme :

$$\text{Score}(X|S) = \text{LLR}\ (X|S) = \frac{1}{T}\ (\log P(X|S) - \log P(X|UBM)) \quad (4.24)$$

Où T est le nombre de trames ou vecteurs acoustiques.

4.5 Le module de Décision

La stratégie de décision en vérification du locuteur nous permet de choisir entre les deux alternatives suivantes: l'identité de l'utilisateur correspond à l'identité proclamée ou recherchée ou elle ne correspond pas. Elle est basée sur un **seuil** prédéfini. Le paragraphe **3.4.2** explique bien la stratégie de détermination du seuil de décision.

Chapitre 5

Vérification du locuteur en mode indépendant du texte par les SVM

Sommaire

Les Supports Vecteurs Machines (SVM) sont de nouvelles techniques discriminantes dans la théorie de l'apprentissage statistique. Elles ont été proposées en 1995 par V. Vapnik dans son livre « The nature of statistical learning theory » **[29]**. Elles permettent d'aborder plusieurs problèmes comme la régression, la classification, la fusion etc. Les SVM se caractérisent par des performances robustes obtenues même avec des bases de très haute dimension telles que les textes, les images et la parole. Le succès de cette méthode est justifié par les bases théoriques solides qui la soutiennent.

Dans le domaine de la reconnaissance des formes, les SVM ont été utilisées pour la reconnaissance de l'écriture des chiffres isolés **[30] [31]**, l'identification d'objet **[32]**, la détection de visage en images **[33]** et la catégorisation de texte **[34] [35]**. Leur utilisation en traitement automatique de la parole paraît également très prometteuse. La première application des SVM en reconnaissance du locuteur, notamment pour l'identification du locuteur, était en 1996 **[31]**. Elles ont été utilisées aussi pour la

vérification du locuteur **[36]**, la détection des mots clés **[37]**, la reconnaissance de langue **[38]**, etc.

Les SVM sont des classifieurs binaires et statistiques. C'est à dire qu'ils permettent de créer une surface de décision entre deux classes définies dans un même espace. Pour cela, ils construisent une frontière de décision par projection des caractéristiques provenant d'un espace d'origine dans un espace de caractéristiques de dimension supérieure dans le but de rendre les classes linéairement séparables.

L'hyperplan choisi est celui qui maximise la marge de séparabilité entre les deux ensembles de données. La sélection de l'hyperplan dans un espace de caractéristiques nécessite d'évaluer un produit scalaire dans cet espace. Ce qui peut être très coûteux en temps et en complexité si l'espace est de très grande dimension. Heureusement, ce calcul n'est pas obligatoire grâce à une opération mathématique appelé noyau (kernel). Le noyau calcule le produit scalaire de deux points dans l'espace de dimension supérieur sans avoir à les projeter. Le problème de recherche de l'hyperplan séparateur possède une formulation duale. Ceci est particulièrement intéressant car, sous cette formulation duale, le problème peut être résolu au moyen de méthode d'optimisation quadratique.

Dans ce chapitre, Nous allons faire l'étude théorique de la méthode SVM en détails. Nous allons voir comment arriver à formuler mathématiquement le problème de l'hyperplan optimal dans les deux cas linéairement séparables et non linéairement séparables. Nous verrons par la suite le passage à la forme duale pour l'obtention d'un problème de programmation quadratique, qu'on va résoudre par l'algorithme SMO. Nous verrons vers la fin l'application des SVM à la vérification du locuteur.

5.1 SVM : Théorie de l'apprentissage
5.1.1 Classification binaire par hyperplan

Considérons l'ensemble d'apprentissage { (x_1,y_1), (x_2,y_2),....., (x_i,y_i) } $x_i \in R^n$ avec i= 1...*l* et $y_i \in \{\pm 1\}$. Classons cet ensemble en utilisant une famille de fonctions linéaires définis par ‹w, x›+b=0 avec $w \in R^n$ et $b \in R$ de telle sorte que la fonction de décision concernant l'appartenance d'un vecteur à l'une des deux classes soit donnée par :

$$f(x)=sign\ (\langle w,x\rangle+b=0)$$

5.1.1.1 Cas de données linéairement séparables

Nous allons construire l'hyperplan *H* d'équation : ‹w, x›+b=0 qui sépare au mieux les deux classes et se trouvant à mi-distance des deux hyperplans H_1 et H_2 parallèles à H, d'équations respectives :

$$\mathbf{H_1} : \langle w, x_i\rangle+b=+1$$

$$\mathbf{H_2} : \langle w, x_i \rangle+b=-1$$

Telle que les deux conditions suivantes soient respectées :

Condition1

- il n'y a aucun point qui se situe entre H_1 et H_2.

Cette contrainte se traduit par les inégalités :

$$\langle w, x_i \rangle+b\geq+1 \text{ pour } \mathbf{y_i = +1}$$

Et

$$\langle w, x_i \rangle+b\geq-1 \text{ pour } \mathbf{y_i = -1}$$

Ces deux inégalités peuvent être combinées en une seule :

$$\mathbf{y_i}\ (\langle w, x_i\rangle+b)\geq+1$$

Condition 2

- La distance ou la marge entre H_1 et H_2 est maximale.

Dans ce cas, la distance entre H_1 et H_2 est donnée par : $M=\frac{2}{\|w\|}$

Maximiser M revient à minimiser $\|w\|$ ou à minimiser $\|w\|^2$ avec :

$\|w\|^2 = w^T w$ (carré de la norme euclidienne du vecteur w).

Le problème de séparation par hyperplan optimal peut être formulé comme suit :

$$\begin{cases} \boldsymbol{min}_{w \in R^n} \frac{1}{2}\|w\|^2 \\ (\boldsymbol{yi} < w, \boldsymbol{x_i} > +b) \geq +1 \ \forall \mathbf{i} \in \mathbf{1} \ldots \boldsymbol{l} \end{cases} \quad (5.1)$$

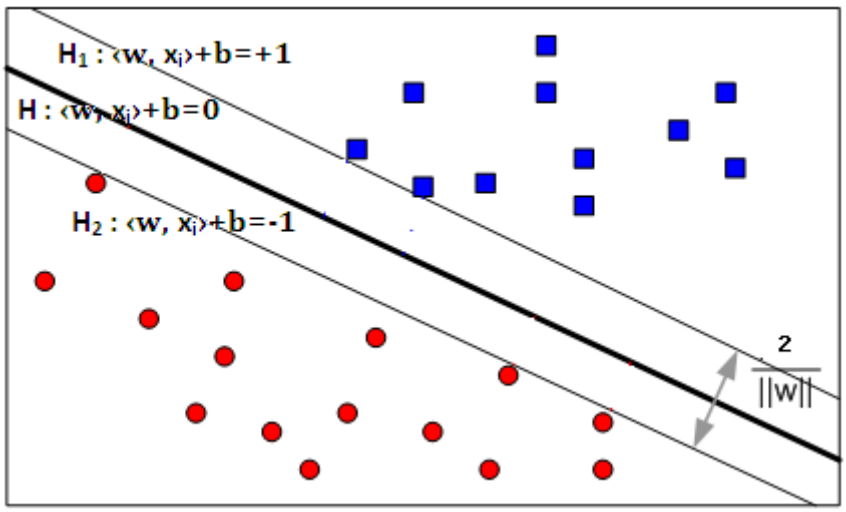


FIG 5-1. Données linéairement séparables.

Ce problème d'optimisation quadratique peut être résolu en introduisant des multiplicateurs de Lagrange $\alpha_i \geq 0$.

Le Lagrangien associé au problème précédent d'optimisation est :

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 \sum_{i=1}^{l} \alpha i(\mathbf{yi}\ (\langle w, \mathbf{xi}\rangle + b) - 1) \quad (5.2)$$

Le Lagrangien doit être minimisé par rapport à w et b et maximisé par rapport à α.

$\frac{\partial L}{\partial w} = 0$ (*)

$\frac{\partial L}{\partial b} = 0$ (**)

A partir des relations (*) et (**), nous pouvons déduire :

$$W = \sum_{i=1}^{l} \alpha_i y_i x_i \quad (5.3)$$

$$ET \sum_{i=1}^{l} \alpha_i y_i = 0 \quad (5.4)$$

En les remplaçant dans L (w, b, α) on obtient le problème dual :

$$\begin{cases} \boldsymbol{L_D} = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j \alpha_i y_i y_j x_i x_j & \text{àmaximiser sous les contraintes} \\ \sum_{i=1}^{l} \alpha_i y_i = 0 \text{ et } \alpha_i \geq 0 \text{ i} = 0, \dots, \text{l} \end{cases} \quad (5.5)$$

La fonction de décision est alors :

$$f(x) = sign\ (\sum_{i=1}^{l} \alpha_i y_i \langle x_i, x \rangle + b) \quad (5.6)$$

Cette fonction de décision est donc seulement influencée par les points correspondants à des α_i non nuls. Ces points sont appelés les *Vecteurs de Support*. Ils correspondent, dans un cas linéairement séparable, aux points les plus proches de la limite de décision, c'est à dire aux points se trouvant exactement à une distance égale à la marge. Il s'agit là d'une propriété très intéressante des SVM : seuls les Vecteurs de Support sont nécessaires pour décrire cette limite de décision, et le nombre de Vecteurs de Support pour le modèle optimal est généralement petit devant le nombre de données d'entraînement.

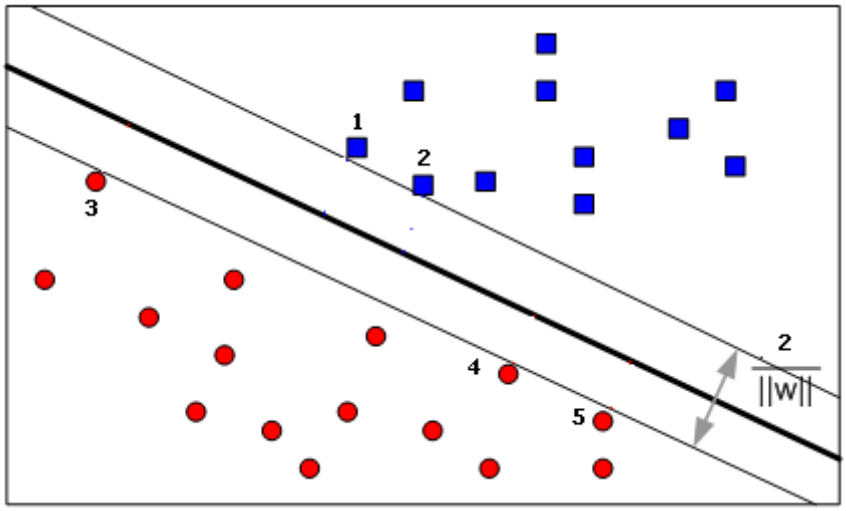


FIG 5-2. Représentation des vecteurs de support (Représentation par des numéros).

5.1.1.2. Cas des données non-linéairement séparables

En pratique, il est rare d'avoir des données linéairement séparables. Afin de traiter des données bruitées ou non linéairement séparable, les SVM ont été généralisées grâce à deux outils : la marge souple (soft margin) et les fonctions noyau (kernel functions).
Le principe de la marge souple est d'autoriser des erreurs de classification. Le nouveau problème de séparation optimale est reformulé comme suit :
L'hyperplan optimal séparant les deux classes est celui qui sépare les données avec le minimum d'erreurs, et satisfait donc les deux conditions suivantes :

Condition 1

- la distance entre les vecteurs bien classés et l'hyperplan doit être maximale.

Condition 2

- la distance entre les vecteurs mal classés et l'hyperplan doit être minimale.

Pour formaliser cela, on introduit des variables de pénalité non-négatives, ε_i pour i= 1,.., *l* appelées variables d'écart. Le principe de la marge souple se traduit par la transformation des contraintes (5.1) qui deviennent :

$$y_i \, (\langle w, x_i \rangle + b) \geq +1 - \varepsilon_i \quad \text{pour } i = 1,..,l \tag{5.7}$$

Avec l'introduction d'un terme de pénalité, la fonction objective devient :

$$\boldsymbol{min}_{w,b,\varepsilon} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \varepsilon_i \quad , C \geq 0 \tag{5.8}$$

Le paramètre C est défini par l'utilisateur. Il peut être interprété comme une tolérance au bruit de classificateur. C'est aussi la pénalité associé à toute violation des contraintes (5.1) du cas linéairement séparable. Pour de grandes valeurs de C, seule de très faibles valeurs de ε sont autorisées et, par conséquent, le nombre de points mal classées sera très faible (données faiblement bruitées).
Cependant, si C est petit, ε peut devenir assez grand et autorise alors bien plus d'erreurs de classification (données fortement bruitées).

La nouvelle formulation d'optimisation est alors :

$$\begin{cases} \boldsymbol{min}_{\mathrm{w,b},\varepsilon}\ \frac{\mathbf{1}}{2}w^T w\ +\ \mathrm{C}\ \sum_{i=1}^{l}\varepsilon_i \quad , \mathrm{C} \geq 0 \ \text{ sous contraintes} \\ \qquad \mathrm{yi}\ (\langle \mathrm{w}, \mathrm{x}\mathbf{i} \rangle + \mathrm{b}) \geq +1 - \ \varepsilon_i \\ \qquad\quad \varepsilon_i \geq 0 \ \text{pour i} = \ 1, .., l \end{cases} \tag{5.9}$$

En introduisant les multiplicateurs de Lagrange, le Lagrangien associé au nouveau problème d'optimisation devient :

$$\mathrm{L}\ (\mathrm{w}, \mathrm{b}, \varepsilon \mathrm{i}, \alpha, \mu) = \frac{\mathbf{1}}{2}w^T w + \mathrm{C}\sum_{i=1}^{l}\varepsilon_i - \sum_{i=1}^{l}\alpha_{\mathrm{i}}[\mathbf{y_i}\ (\mathrm{w}^{\mathrm{T}}\mathrm{x_i} - \mathrm{b}) + \varepsilon_i - 1] - \sum_{i=1}^{l}\varepsilon_i\ \mu_i$$

$$= \frac{\mathbf{1}}{2}w^T w + \sum_{i=1}^{l}(C - \alpha_{\mathrm{i}} - \mu_i)\ \varepsilon_i - (\sum_{i=1}^{l}\alpha_{\mathrm{i}}\mathbf{y_i}\ \mathrm{x_i})\mathrm{w} - (\sum_{i=1}^{l}\mathbf{y_i}\ \mathrm{x_i})\mathrm{b} - \sum_{i=1}^{l}\alpha_{\mathrm{i}} \tag{5.10}$$

Le Lagrangien doit être minimisé par rapport à w, b, εi et ε_i et maximisé par rapport à α etμ.

$$\frac{\partial L}{\partial w} = 0 \qquad (*)$$

$$\frac{\partial L}{\partial b} = 0 \qquad (**)$$

$$\frac{\partial L}{\partial \varepsilon_i} = 0 \qquad (***)$$

De ces dernières relations, on peut tirer les trois égalités suivantes :

$$\mathrm{W} = \sum_{i=1}^{l}\alpha_i\ y_i x_i \quad ; \quad \sum_{i=1}^{l}\alpha_i\ y_i = 0 \quad \text{et} \quad \alpha_{\mathrm{i}} = \mathrm{C} - \mu_{\mathrm{i}} \tag{5.11}$$

$$\begin{cases} \boldsymbol{L_D} = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\ \alpha_j \alpha_i y_i y_j x_i x_j \ \ \text{àmaximiser sous les contraintes} \\ \qquad 0 \leq \alpha_{\mathrm{i}} \leq \mathrm{C} \quad \mathrm{i} = 0, \ldots, \mathrm{l} \\ \qquad \text{et} \ \sum_{i=1}^{l}\alpha_i\ y_i = 0 \end{cases} \tag{5.12}$$

La seule différence avec le cas linéairement séparable est donc l'introduction d'une borne supérieure pour les paramètres α_i. Il est également intéressant de noter que les points se trouvant du « mauvais » côté de la limite de décision sont tous des vecteurs de

support, quelle que soit leur distance à cette limite, ce qui signifie qu'ils exercent une influence sur le calcul de cette limite.

Maintenant, que faire si les données ne sont pas linéairement séparables ?
L'idée est de projeter l'espace d'entrée (espace des données) dans un espace de plus grand dimension appelé espace de caractéristiques (feature space) afin d'obtenir une configuration linéairement séparable (à l'approximation de la marge souple près) de nos données, et d'appliquer alors l'algorithme des SVM.

Cette projection est équivalente à l'application d'une transformation sur les données initiales par l'intermédiaire d'une fonction Φ (figure **5-3**).

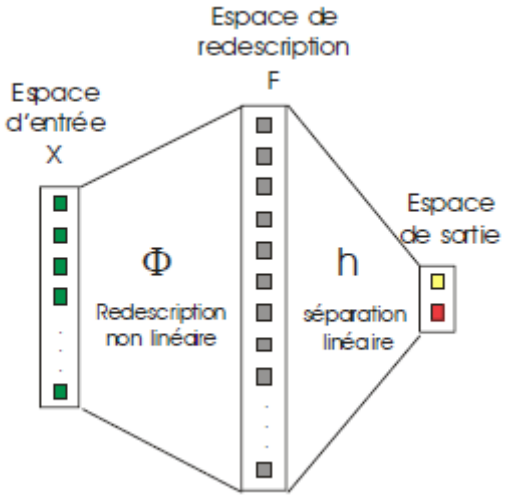


FIG 5-3. la transformation non linéaire (Φ).

Le nouvel algorithme peut être écrit ainsi :

Soit la fonction : $\Phi : R^n \longrightarrow R^m$, m>n

$x \longrightarrow \Phi(x)$

$$\begin{cases} \boldsymbol{L_D} = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i\, \alpha_j \alpha_i y_i y_j (\Phi(x_i)\, \Phi(x_j)\,) \\ \text{àmaximiser sous les contraintes} \\ 0 \le \alpha_i \le C \quad i = 0, \dots, l \\ \text{et } \sum_{i=1}^{l} \alpha_i\, y_i = 0 \end{cases} \tag{5.13}$$

Pour illustrer l'idée de la transformation des données initiale par une fonction Φ, prenons l'exemple du OU exclusif. Dans le plan, les données ne sont pas séparables par une droite. Par contre, si on plonge les données dans un espace qui associe aux coordonnées (x,y) les coordonnées (x,y,xy) on peut trouver un hyperplan séparateur **[39]**.

$$\Phi : R^2 \longrightarrow R^3$$

$$x \longrightarrow \Phi(x)$$

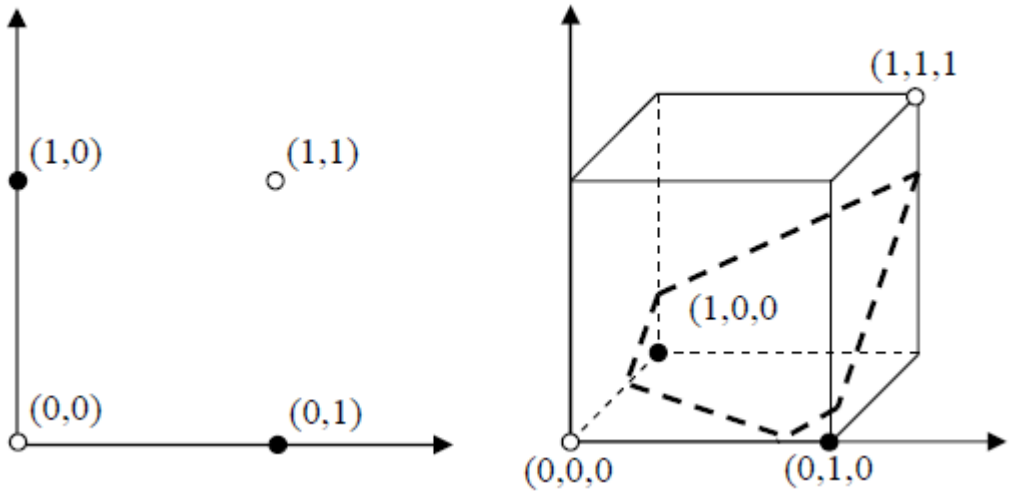


FIG 5-4. Plonger les données du OU-Exclusif dans un espace vectoriel de dimension supérieure permet de les rendre linéairement séparables.

Une fois que l'on a trouvé un hyperplan séparateur, il nous reste à trouver celui qui classe au mieux les nouvelles données.

5.1.2 Les fonctions noyau

5.1.2.1 Introduction du noyau

Afin d'effectuer des décisions non linéaires en utilisant le SVM, il n'est pas nécessaire de définir une transformation explicite car ce genre de transformation peut devenir très coûteux du point de vue calcul pour de grandes valeurs de m. En analysant les formules (5.5) et (5.12), on remarque que les vecteurs d'entrée se présentent dans les fonctions objectives sous formes de produits scalaires entre les paires de vecteurs.

L'astuce est de calculer le produit scalaire dans l'espace des caractéristiques en fonction des vecteurs de l'espace d'entrée directement.
Le produit scalaire dans l'espace d'entrée en utilisant la transformée utilisée dans l'espace du Ou exclusif est :

$$u'.v' = \begin{pmatrix} u_1 \\ u_2 \\ u_1 u_2 \end{pmatrix} . \begin{pmatrix} v_1 \\ v_2 \\ v_1 v_2 \end{pmatrix} = u.v + u_1 u_2 + v_1 v_2$$

Donc on peut définir le noyau :

$$K(u,v) = u.v + u_1 u_2 + v_1 v_2$$

Les produits scalaires dans les formules (5.5) et (5.12), peuvent être remplacés par une fonction noyau. On peut utiliser n'importe qu'elle fonction noyau valide (satisfaisant la condition de Mercer **[40]**) sans avoir besoin de connaître des informations sur la transformation linéaire qui lui a donné lieu. C'est également plus efficace que d'effectuer des transformations non-linéaires sur les données puis calculer leurs produits scalaires séparément.

5.1.2.2 Condition de Mercer [40]

La matrice contenant les similarités entre tous les exemples de l'entraînement est appelée matrice de Gram.

$$G = \begin{pmatrix} k_{11} k_{12} & \cdots & k_{1n} \\ \vdots & \ddots & \vdots \\ k_{n1} k_{n2} & \cdots & k_{nn} \end{pmatrix}$$

Théorème : (condition de Mercer) la fonction K(x,y) =XxX ⟶ R est un noyau si et seulement si :

$$G = (K(x_i, x_j)) \text{ , } i,j = 1,\ldots,n$$

Est définie positive.
Notons qu'une fonction K : XxX ⟶ R générant une matrice définie positive possède les trois propriétés fondamentales du produit scalaire : $\forall x_i, x_j \in X$

1. Positivité : $K(x_i, x_j) \geq 0$.
2. Symétrie : $K(x_i, x_j) = K(x_j, x_i)$.
3. Inégalité de Cauchy-Schwartz : $|K(x_i, x_j)| \leq \|x_i\| . \|x_j\|$

La condition de Mercer nous indique si une fonction est un noyau mais ne fournit aucune information sur la fonction Φ (et donc sur l'espace des caractéristiques) introduit par ce noyau.

5.1.2.3 Exemples de noyaux

- **Le noyau Linéaire**

$$K(x,y) = \langle x\ y \rangle \text{ avec } \Phi(x) = x \qquad \text{(La fonction identité).}$$

- **Le noyau Polynomial**

Sa forme générique est de la forme :

$$K(x,y) = (a.\langle x\ y \rangle + b)^d$$

Prenons une instance simple de cette fonction : $K(x,y) = (\langle x\ y \rangle)^2$ et essayons de trouver un candidat Φ tel que :

$$\langle x\ y \rangle^2 = \langle \Phi(x)\ \Phi(y) \rangle$$

En supposant que l'espace d'entrée est de dimension 2, on peut utiliser les projections suivantes :

$$\Phi_1 : R^2 \longrightarrow R^3$$
$$x \longrightarrow (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

$$\Phi_2 : R^2 \longrightarrow R^4$$
$$x \longrightarrow (x_1^2, x_1x_2, x_1x_2, , x_2^2)$$

Cet exemple montre que la fonction de projection Φ et l'espace des caractéristiques ne sont pas uniques. En générale, quand on utilise un noyau polynomiale, on prend des paramètres a et b égaux à 1.

- **Le noyau RBF (Radial Basis Function)**

La forme générique de ce noyau est :

$$K(x,y)= \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$

Où σ est un paramètre de régulation.

Notons que le noyau linéaire, le noyau polynomial et le noyau RBF sont les plus utilisés dans la classification automatique basée sur la technique des SVM.

En résumé, pour tout problème de la classification automatique, nous devons résoudre le programme quadratique suivant :

$$\begin{cases} \boldsymbol{L_D} = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i\, \alpha_j \alpha_i y_i y_j K(x_i, x_j) \\ \text{àmaximiser sous les contraintes} \\ 0 \leq \alpha_i \leq C \quad i = 0, \dots, l \\ \text{et } \sum_{i=1}^{l} \alpha_i\, y_i = 0 \end{cases} \tag{5.14}$$

Et la nouvelle fonction de décision est alors :

$$f(x)=sign\ (\sum_{i=1}^{l} \alpha_i\, y_i K(x_i,x)+b) \tag{5.15}$$

5.1.3 Méthode d'entraînement des SVM

L'entraînement d'une machine à Vecteurs de Support consiste à résoudre le problème d'optimisation quadratique convexe (l'équation **5.14**). Le choix de la

technique de résolution numérique est critique car les performances de l'implémentation en seront directement tributaires.

En raison de son immense taille, le problème (l'équation **5.14**) qui résulte de l'approche SVM ne peut pas être résolu facilement par l'intermédiaire des techniques standards de programmation quadratique (PQ). La forme quadratique dans (l'équation **5.14**) implique une matrice qui a un nombre d'élément égale au carré du nombre d'exemples d'entraînement. Cette matrice ne peut pas être traitée correctement avec une RAM de 128 méga-octets s'il y a plus de quatre milles (4000) exemples d'entraînement ou chaque élément de la matrice est stocké en double précision (8 octets).

5.1.3.1 Les conditions de KKT

En appliquant les conditions de Kuhn et Tucker au problème (5.9), on aura :

$$\nabla \left(\frac{\mathbf{1}}{2} w^T w + C\sum_{i=1}^{l} \varepsilon_i\right) - \sum_{i=1}^{l} \alpha_i \nabla(\mathbf{y_i} (w^T x_i - b) + \varepsilon_i - 1) - \sum_{i=1}^{l} \varepsilon_i \mu_i = 0$$

$$\alpha_i (y_i(w x_i + b) + \varepsilon_i - 1) = 0 \qquad i = 0, \dots, l \qquad (*1)$$

$$\mu_i \varepsilon_i = 0 \quad i = 0, \dots, l$$

$$\varepsilon_i \geq 0 \ \alpha_i \geq 0. \qquad (*2)$$

Les conditions (*1) et (*2) dépendent de α_i on aura 3 cas :

1. Si $\alpha_i = 0$ alors $\mu_i = C - \alpha_i = C$, de (*2) on aura $\varepsilon_i = 0$ donc :
$$y_i(w x_i + b) - 1 \geq 0$$
2. Si $0 < \alpha_i < C$ alors $y_i(w x_i + b) + \varepsilon_i - 1 = 0$ et $\mu_i = C - \alpha_i > 0 \Longrightarrow \varepsilon_i = 0$ donc :
$$y_i(w x_i + b) - 1 = 0$$
3. Si $\alpha_i = C$ alors $y_i(w x_i + b) + \varepsilon_i - 1 = 0$ et $\mu_i = C - \alpha_i > 0 \Longrightarrow \varepsilon_i \geq 0$ donc :
$$y_i(w x_i + b) - 1 \leq 0$$

On pose :

$$R_i = y_i(w x_i + b) - 1 = y_i(w x_i + b) - y_i^2 = y_i(w x_i + b - y_i) = y_i E_i \qquad (5.16)$$

Si les conditions KKT sont vérifiées alors :

$$\begin{cases} \alpha_i = 0 \implies R_i \geq 0 \\ 0 < \alpha_i < C \implies R_i = 0 \\ \alpha_i = C \implies R_i \leq 0 \end{cases} \quad (5.17)$$

Les conditions de KKT sont violées dans les deux cas suivants :

$$\begin{cases} \alpha_i < C \; et \; R_i < 0 \\ \alpha_i > C \; et \; R_i > 0 \end{cases}$$

5.1.3.2 La méthode d'Optimisation Séquentielle Minimale (SMO)

A cause de sa taille considérable, le problème de programmation quadratique *PQ* provenant d'une *SVM* ne peut pas être facilement résolu avec les techniques standards. La matrice de gram contient un nombre d'éléments égal au carré du nombre total des données de la base d'apprentissage. Depuis plusieurs années, deux algorithmes ont été proposés pour l'optimisation du *PQ*. Il s'agit de l'algorithme procédant par tronçons (*Chunking)* et l'algorithme de décomposition. Plus récemment, Platt et Cristianini ont proposé un algorithme beaucoup plus rapide que les deux autres. Cet algorithme est appelé *Sequential Minimal Optimisation SMO* ou Optimisation minimale et séquentielle **[41] [42]**. L'algorithme *SMO* est l'algorithme adopté dans notre travail.

L'algorithme SMO

L'optimisation minimale et séquentielle du risque est un algorithme simple et rapide proposé pour résoudre le problème de programmation quadratique des *SVM* sans la nécessité de stocker une grande matrice en mémoire et sans une routine numérique itérative pour chaque sous-problème. *SMO* décompose le problème de *PQ* en plusieurs sous-problèmes aussi mais contrairement aux autres méthodes, à chaque étape *SMO* optimise le plus petit problème possible. Ainsi à chaque étape il optimise deux multiplieurs de Lagrangien (on ne peut pas utiliser un seul multiplieur car on a une contrainte d'égalité linéaire). A chaque étape d'optimisation *SMO* cherche les valeurs optimales des deux multiplieurs et effectue une mise à jour de la *SVM* pour donner le reflet des nouvelles valeurs optimisées.

L'avantage principal de *SMO* réside dans le fait que la résolution du *PQ* pour deux multiplieurs peut être effectuée analytiquement. En plus, il ne nécessite pas un espace mémoire important (On stocke à chaque fois une matrice de taille 2 × 2). L'algorithme *SMO* comporte trois éléments :

1. Une méthode analytique pour résoudre le problème de *PQ*.
2. Des heuristiques pour choisir les multiplieurs à optimiser.
3. Une méthode pour calculer le seuil *b*.

Méthode de résolution pour les deux multiplicateurs de Lagrange

Afin de résoudre le problème de PQ (l'équation **5.14**) pour les deux multiplicateurs de Lagrange, SMO calcule d'abord les contraintes sur ces multiplicateurs et les résout ensuite, en optimisant la fonction objectif à chaque étape, toutes les variables (x.y,E) se référant au premier multiplicateur auront l'indice 1, de même pour le second, ils auront l'indice 2. Puisqu'il y a seulement deux multiplicateurs, les contraintes peuvent facilement être représentées dans deux dimensions (voir la figure **5-5**). Ainsi, le maximum contraint de la fonction objectif doit se trouver sur le segment de ligne droite (comme représenté sur la figure **5-5**). Cette contrainte explique pourquoi deux est le nombre minimum de multiplicateurs de Lagrange qui peuvent être optimisés : si SMO optimisait seulement un multiplicateur, il ne pourrait pas respecter la contrainte d'égalité à chaque étape **[42]**.

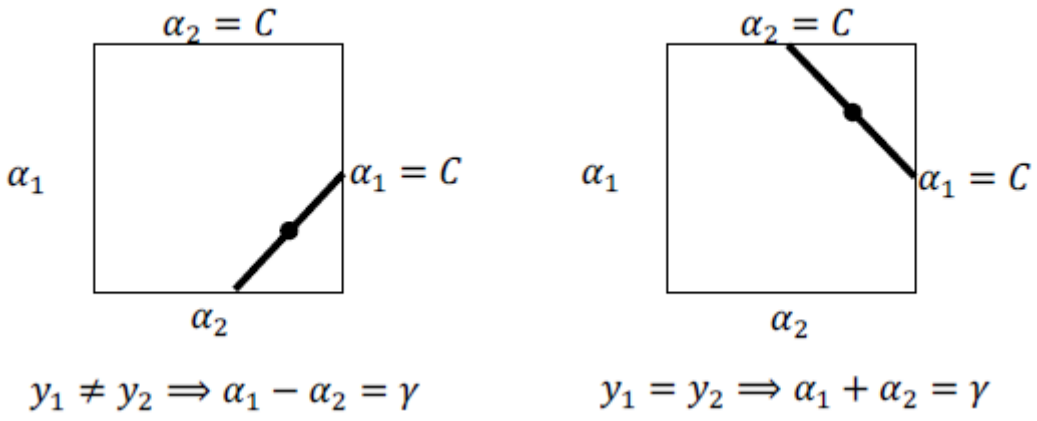


FIG 5-5. Deux cas d'optimisation.

γ est une contrainte, puisque les autres multiplicateurs de Lagrange qui ne vont pas être optimisés sont considérés comme des constantes.

Les deux multiplicateurs de Lagrange doivent satisfaire toutes les contraintes du problème (l'équation **5.14**). Les contraintes d'inégalité font situer les multiplicateurs de Lagrange à l'intérieur du carré. La contrainte d'égalité les situe sur une ligne diagonale. Par conséquent, une étape de SMO consiste à trouver un optimum de la fonction objectif sur le segment de droite délimité par le carré.

Avec :

$$\gamma = \alpha_1^{old} + s\alpha_2^{old} \text{ qui dépend de } \alpha_1, \alpha_2 \text{ et } s = y_1 y_2$$

L'algorithme détermine d'abord l'intervalle où peut varier s_, ensuite il calcule sa vraie valeur.

Si $y_1 \neq y_2$ alors les limites de α_2 sont :

$$L = \max(0, \alpha_2^{old} - \alpha_1^{old}), H = \min(C, C + \alpha_2^{old} - \alpha_1^{old}) \quad (5.18)$$

Si $y_1 = y_2$ alors les limites de α_2 sont :

$$L = \max(0, \alpha_1^{old} + \alpha_2^{old} - C), H = \min(C, \alpha_1^{old} + \alpha_2^{old}) \quad (5.19)$$

En ne permettant qu'à deux multiplicateurs de Lagrange de varier, la fonction objective sera :

$$L_D = \frac{1}{2}\eta\, \alpha_2^2 + (y_2(E_1^{old} + E_2^{old}) - \eta\, \alpha_2^{old})\, \alpha_2 + const$$

Avec :

$$\eta = 2K(x_1, x_2) - K(x_1, x_1) - K(x_2, x_2) \quad (5.20)$$

Et $E_i = f^{old}(x_i) - y_i$ l'erreur d'entrainement du $i^{ème}$ exemple.

Les dérivées première et seconde de la fonction objective L_D par rapport à α_2, peuvent s'exprimer comme suit :

$$\frac{\partial L_D}{\partial \alpha_2} = \eta\, \alpha_2 + (y_2\,(E_1^{old} - E_2^{old}) - \eta\, \alpha_2^{old})$$

$$\frac{\partial^2 L_D}{\partial \alpha_2^2} = \eta$$

On peut facilement démontrer que $\eta \leq 0$.

Pour calculer la valeur de α_2 qui maximise au mieux l'augmentation de la fonction objective, il faut que :

$$\frac{\partial L_D}{\partial \alpha_2} = 0 \text{ et } \frac{\partial^2 L_D}{\partial \alpha_2^2} = \eta < 0$$

Si $\eta < 0$ alors :

$$\alpha_2^{new} = \alpha_2^{old} - \frac{y_2(E_1 - E_2)}{\eta} \quad (5.21)$$

En introduisant α_2^{new} dans les limites de α_2 on aura :

$$\begin{cases} H, \Longrightarrow \alpha_2^{new} \geq H \\ \alpha_2^{new}, \ si\ L < \alpha_2^{new} < H \\ L, \Longrightarrow \alpha_2^{new} < L \end{cases} \quad (5.22)$$

Si $\eta = 0$ alors, on a besoin de calculer la valeur de la fonction objective pour les deux bornes de l'intervalle de variation de α_2 (L et H), la nouvelle valeur de α_2 sera celle qui maximise au mieux l'augmentation de L_D.

A présent la valeur de α_1 peut être calculée par :

$$\alpha_1^{new} = \alpha_1^{old} + s\,(\alpha_2^{old} - \alpha_2^{new,\,clipped}) \quad (5.23)$$

Heuristiques pour le choix du multiplicateur à optimiser

L'algorithme SMO permet d'optimiser deux multiplicateurs de Lagrange à chaque étape, avec un des multiplicateurs ayant violé les conditions de KKT avant cette étape. Par conséquent on maintiendra toujours le vecteur des multiplicateurs de Lagrange de l'étape précédente. La fonction objective globale augmentera à chaque étape et l'algorithme convergera asymptotiquement. Afin d'accélérer la convergence, des heuristiques pour le choix des deux multiplicateurs de Lagrange à optimiser conjointement sont utilisées.

Deux heuristiques sont employées : une pour le premier multiplicateur de Lagrange et l'autre pour le second. La première heuristique détermine les exemples qui violent les conditions de KKT. Si un exemple viole les conditions de KKT, il est alors candidat à l'optimisation immédiate.
Une fois qu'un exemple violant les conditions de KKT est trouvé, un deuxième multiplicateur est choisi en utilisant la deuxième heuristique. Les deux multiplicateurs sont alors conjointement optimisés.

La deuxième heuristique essaye de maximiser $|E_1 - E_2|$ dans (l'équation **5.21**). Si E_1 est positif, on choisit un exemple avec une erreur minimale E_2. Si E_1 est négative, on choisit un exemple avec l'erreur maximale E_2. On peut tomber dans des cas ou l'algorithme SMO ne permet pas d'avoir une progression positive en utilisant la deuxième heuristique. Par exemple, le premier et le deuxième exemple liés à α_1 et α_2 respectivement partagent des vecteurs d'entrée x identiques.

Pour éviter ce problème, on cherche séquentiellement un α_2 qui viole les conditions de KKT et qui est compris entre 0 et C ($0 < \alpha_i < C$) en partant d'un indice choisi au hasard. Si l'on ne trouve pas un α_2 qui satisfait ces deux conditions, on recommence la recherche séquentielle d'un α_2 qui viole les conditions de KKT seulement en commençant par un indice pris au hasard.
Après avoir trouvé les deux multiplicateurs à optimiser conjointement, on calcule leurs valeurs grâce aux formules (**4.22**) et (**4.23**) et on met à jour la valeur de la fonction objective.
L'algorithme SMO s'arrête lorsqu'il ne reste aucun multiplicateur de Lagrange qui viole les conditions de KKT.

Calcul du paramètre b

Le calcul des multiplicateurs de Lagrange ne donne pas le b directement. Ainsi, le paramètre b doit être calculé séparément.

$E_i\, f^{old}(x_i) - y_i \Rightarrow \Delta E_i = \Delta\alpha_1 y_1 K(x_1,x_i) + \Delta\alpha_2 y_2 K(x_2,x_i) - \Delta b$

On va forcer : E_1^{new} à zéro si $0< \alpha_1< C$

Ou

E_2^{new} à zéro si $0< \alpha_2< C$

$0 = E_i^{new} \Rightarrow 0 = E_i^{old} + \Delta E_i \Rightarrow b^{new} = E_i + \Delta\alpha_1 y_1 K(x_1,x_i) + \Delta\alpha_2 y_2 K(x_2,x_i) - b^{old}$

La valeur de b^{new} aura comme formule :

si $0< \alpha_1< C$

$b^{new} = b_1 = E_1 + y_1 (\alpha_1^{new} - \alpha_1^{old}) K(x_1,x_1) + y_2(\alpha_2^{new, clipped} - \alpha_2^{old}) K(x_1,x_2) + b^{old}$

si $0< \alpha_2< C$

$b^{new} = b_2 = E_2 + y_1 (\alpha_1^{new} - \alpha_1^{old}) K(x_1,x_2) + y_2(\alpha_2^{new, clipped} - \alpha_2^{old}) K(x_2,x_2) + b^{old}$

Si α_1 et α_2 sont tous les deux égaux à 0 ou C alors :

$$b^{new} = \frac{b_1 + b_2}{2} \qquad (5.24)$$

Il faut noter que l'algorithme SMO ne donne pas nécessairement le même résultat s'il est exécuté plusieurs fois avec les mêmes entrées. En effet, pour choisir le α_2 à optimiser, si la première condition qui maximise $|E_1 - E_2|$ n'est pas vérifiée, la deuxième heuristique entame une recherche séquentielle de ce multiplicateur de Lagrange en partant d'un indice choisi aléatoirement. Le fait de commencer cette recherche d'une position aléatoire signifie qu'on va trouver des α_2 différents ce qui explique ces résultats différents.

5.2. SVM pour la vérification du locuteur en mode indépendant du texte

Comme indiqué auparavant, plusieurs travaux utilisant les SVM sont apparus. Ils sont réalisés principalement sur la reconnaissance de lettres et de chiffres manuscrits et sur la détection du visage. D'autres travaux ont utilisé les SVM pour faire de la fusion des données de différents experts pour l'identification biométrique. Les résultats intéressants obtenus par ces travaux ont encouragé les chercheurs à penser aux d'autres disciplines comme la reconnaissance du locuteur.

Les SVM exigent des vecteurs d'entrée de taille fixe, et pour les adapter à toute application utilisant le signal de parole (qui est non délimité dans le temps), il faudrait trouver une nouvelle représentation de données qui permette de fournir un vecteur de taille fixe quelle que soit la longueur du signal de parole à traiter.
Appliquée à la RAL, la SVM est entraînée en utilisant des données provenant du client (pour lesquelles (f (x) = 1), mais aussi des données appartenant à d'autres locuteurs (f (X) = −1). Les données fournies aux SVM peuvent être de deux types : paramètres **[43]** ou scores **[44]**.

5.2.1 Approche GMM/SVM

Notre approche concernant l'utilisation des SVM en vérification du locuteur en mode indépendant du texte utilise une représentation des données basée sur une modélisation des clients par des modèles GMM. Cette représentation des données va nous permettre d'utiliser les SVM dans la phase de décision.
Notons que la différence entre le système de référence (système GMM-UBM) et ce nouveau système GMM-SVM est le module de décision. Les modules (paramétrisation et modélisation) sont les mêmes pour tous les deux systèmes.

La première étape de système GMM-SVM consiste à diviser les données d'apprentissage de chaque client en deux parties. La première partie va être utilisée pour construire le modèle GMM du client et la deuxième partie sera utilisée pour construire le modèle SVM de chaque client. Les modèles GMM des clients sont construits par adaptation du modèle du monde en utilisant l'algorithme MAP (chapitre 4).

Pour apprendre le modèle SVM de chaque client, nous avons besoin du modèle GMM du client, du modèle du monde, quelques segments de parole du client pour construire les vecteurs SVM de la classe client et de quelques segments de parole provenant des imposteurs pour construire les vecteurs SVM de la classe non-client.

5.2.2 Construction de vecteurs d'entrée des SVM

Supposons que le modèle du client et le modèle du monde ont ***n*** gaussiennes chacun, la taille des vecteurs d'entrée des SVM est de **2 * *n***. Les ***n*** premières composantes vont correspondre à des mesures de vraisemblances par rapport aux ***n*** gaussiennes du modèle de client et les ***n*** dernières composantes vont correspondre à des mesures de vraisemblances par rapport aux ***n*** gaussiennes du modèle du monde. Soit *X* un segment de parole (client où imposteur), $\boldsymbol{\lambda}$ le modèle de l'identité proclamée et $\boldsymbol{\bar{\lambda}}$ le modèle du monde. Pour construire un vecteur d'entrée des SVM $\mathbf{V_X^{\lambda}}$, nous allons construire deux vecteurs $\mathbf{V_X^{\lambda}}$ (λ) et $\mathbf{V_X^{\lambda}}$ ($\bar{\lambda}$) de dimension ***n*** chacun dont la concaténation fournira le vecteur final **[38]**. La construction des vecteurs d'entrée des SVM est réalisée comme suit :

D'abord toutes les composantes des deux vecteurs sont initialisées à 0. Ensuite pour chaque trame *t* du segment *X*, le score S_t est calculé par l'équation suivante :

$$S_t = \max_{\text{gi} \in \lambda\bar{\lambda}} \text{Log}\, p(t|g_i)$$

Si le score S_t est maximisé par l'ième gaussienne du modèle du client, la ième composante de notre vecteur $\mathbf{V_X^{\lambda}}$ (λ) sera incrémentée par le score S_t. Si ce score est maximisé par l'ième gaussienne du modèle du monde, c'est la ième composante du vecteur $\mathbf{V_X^{\lambda}}$ ($\bar{\lambda}$) qui sera incrémentée par le score S_t. Enfin et après avoir traité toutes les trames du segment *X*, les composantes des deux vecteurs $\mathbf{V_X^{\lambda}}$ (λ) et $\mathbf{V_X^{\lambda}}$ ($\bar{\lambda}$) sont normalisées par le nombre de trames du segment *X*. Ainsi le vecteur d'entrée des SVM est obtenu par concaténation des deux vecteurs $\mathbf{V_X^{\lambda}}$ (λ) et $\mathbf{V_X^{\lambda}}$ ($\bar{\lambda}$) (figure **5-6**).

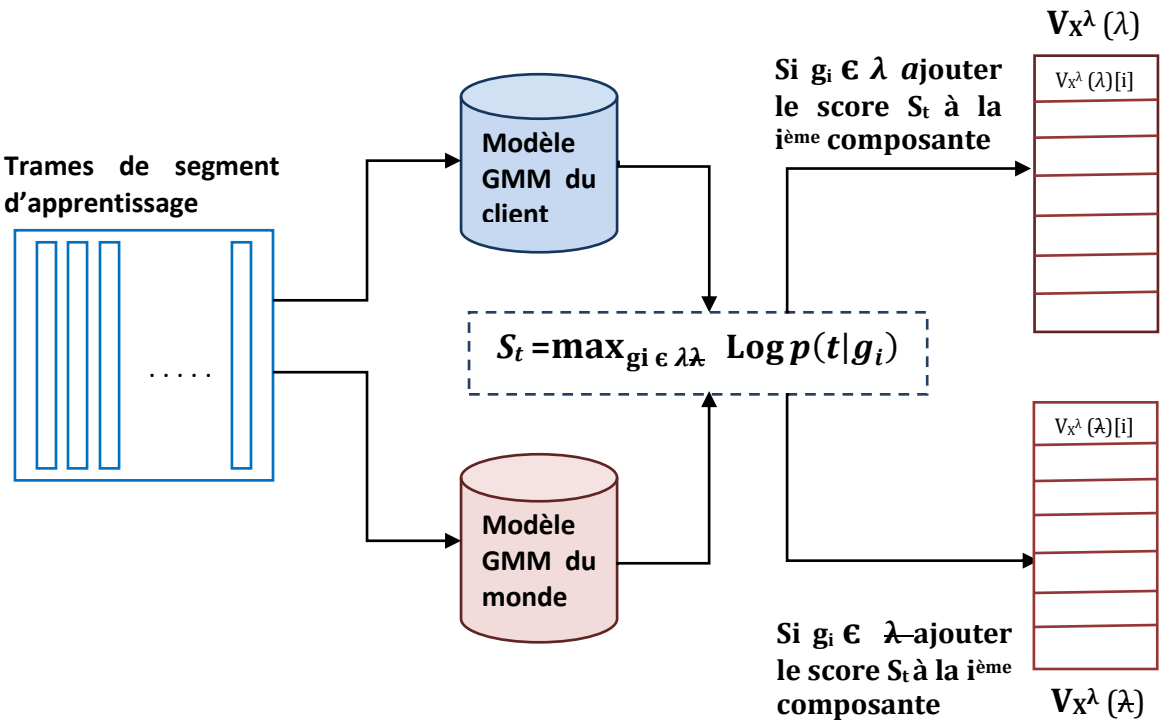


FIG 5-6. Construction des vecteurs d'entrée pour les SVM **[38]**

Dans la phase de test, pour chaque segment de test Y et une identité $\boldsymbol{\beta}$ un vecteur $\mathbf{V_Y^{\beta}}$ est construit de la même façon que dans l'apprentissage. Un score de décision est obtenu par la fonction de classement des SVM suivante :

$$Class\,(Y) = \sum_{z_i \in VS\,(\beta)} \alpha_i y_i K(z_i, V_Y^{\beta}) + b$$

Où z_i est un vecteur support du modèle SVM du client β, α_i est le coefficient de Lagrange correspondant au vecteur support z_i, y_i est la classe de z_i, $K(z_i, V_Y^{\beta})$ représente le noyau utilisé pour apprendre le modèle SVM du client β appliqué au couple (z_i, V_Y^{β}) et b est le biais du modèle SVM du client β.

Supposons que dans l'apprentissage des modèles SVM des clients, les vecteurs d'entrée représentant la classe client ont été étiquetés par 1 et les vecteurs représentant la classe non-clients ont été étiquetés par -1. Si *class*(Y) est positif alors le système décide que le segment Y est prononcé par le client β sinon le système décide que le segment y provient d'un imposteur.

Chapitre 6

Expérimentation et évaluation des performances

Sommaire

Ce chapitre présente le contexte expérimental ainsi l'évaluation des performances de notre système de reconnaissance du locuteur à base de **GMM** en mode indépendant du texte. Le système est validé sous l'environnement **C**.

En premier lieu, nous décrivons la base de données utilisées. Ensuite, nous décrivons la chaîne de traitement acoustique qui assure l'extraction des paramètres pertinents pour la reconnaissance du locuteur. En troisième lieu, nous présentons l'algorithme d'apprentissage des modèles clients et monde (modèle UBM). Enfin, nous présentons le protocole d'évaluation en identification et vérification du locuteur.

Différentes étapes sont nécessaires à la construction et à la validation d'un système de reconnaissance du locuteur à base des GMM:

1. Une base de données, utile à l'apprentissage du système: pour cela, on dispose de fichiers sons (format wav).

2. Une étape de **segmentation parole/non-parole** sera nécessaire afin de ne conserver que les zones de parole.
3. Puis, on générera les vecteurs acoustiques correspondants : phase de **paramétrisation**.
4. Ensuite, on fera l'**apprentissage** des modèles de mélanges de lois gaussiennes GMM : il s'agira de construire le modèle du « monde » et les modèles des clients.
5. Enfin, on passera à la phase de **reconnaissance** (décision) :

   VAL : le locuteur testé, est-il le locuteur cible ?

   IAL : à qui, parmi N locuteurs, l'identité proclamé ?

6.1 Construction du système de RAL

6.1.1 Base de donnée

Nos expériences ont été effectuées sur la base **VoxForge [46]**, une base de données d'enregistrements de textes lus en anglais. Les signaux sont enregistrés à une fréquence d'échantillonnage de 48 kHz.
Un modèle du monde, indépendant du genre, est créé avec un sous ensemble de 40 locuteurs (hommes et femmes), environ une heure de parole.
23 locuteurs sont utilisés comme clients du système de reconnaissance (hommes et femmes) et 17 autres locuteurs sont utilisés comme imposteurs.
Nous utilisons des enregistrements de 100 secondes pour l'apprentissage des modèles client, et de 7 à 10 secondes pour la phase de test.

6.1.2 Analyse acoustique

L'analyse acoustique consiste à déterminer les paramètres pertinents à la reconnaissance du locuteur du signal parole. En utilisant la méthode d'analyse cepstrale, nous calculons les MFCC représentant le signal de parole en suivant les étapes présentées en paragraphe **4.2.1**. Une fenêtre de Hamming de 22 ms avec un décalage de 11 ms est utilisée. Le nombre de coefficients de MFCC dans chaque vecteur est 12. On ajoute à ce vecteur l**es dérivés** premières et secondes des coefficients MFCC (12 Δ+ 12

ΔΔ), le log d'énergie de la trame, et la dérivé première et seconde de log d'énergie (log E+ Δ log E+ ΔΔ logE). Au total, le vecteur contient 39 coefficients.

6.1.3 Apprentissage des modèles

Les paramètres du modèle du monde(UBM) sont estimés par l'algorithme **EM** en maximisant le critère du Maximum de Vraisemblance**.** Les moyennes du modèle initial sont obtenues en utilisant l'algorithme K-means, la matrice de variance (diagonale) est la matrice unité, et les poids suivent la loi d'équiprobabilité.
La valeur minimale des variances du modèle de monde (Paragraphe **4.3.2**) est obtenue par la formule suivante :

$$\sigma^2 = \beta \times \sigma^2_{Global}$$

On prend β (flooring variance) égal à 0.5 (système LIA).
σ^2_{Global} , la variance globale, est calculée à partir des données d'apprentissage du modèle UBM.
En notant **Y** la séquence de trames acoustiques caractérisant le modèle UBM, chacune de ces trames étant de dimension d, la variance global de cette séquence s'écrit par :
VG (Y) = $[\, v^1 , v^2 \ldots\ v^d \,]$.
$\boldsymbol{v^d} = \frac{1}{T} \times \sum_{t=1}^{T} \left(\boldsymbol{y_t^d} - \frac{1}{T} \times \sum_{t=1}^{T} \boldsymbol{y_t^d} \right)^2$, $\boldsymbol{y_t^d}$ est la composante d de la trame t de la séquence **Y.**

Les modèles client sont dérivés par adaptation MAP du modèle du monde avec un relevance factor de 14. Seules les moyennes des modèles sont adaptées.

6.2 Evaluation des performances

En **identification de locuteur**, nous allons évaluer les performances d'identification de 23 clients, i.e. il s'agit d'identifier un client parmi 23 clients (ensemble fermé) et de calculer le taux d'identification correcte, **TIC** (Paragraphe **3.4.1**).

En **vérification de locuteur**, nous allons déterminer le taux d'égale erreur **EER**, qui est obtenu quand TFA=TFR = EER (paragraphe **3.4.3**), pour un seuil commun à tous les locuteurs. Pour cela, nous effectuons 506 tests d'imposteurs et 46 tests clients.

On calcul aussi le DCF avec les paramètres suivants (utilisées par la compagne NIST SRE 2008) **[45]** :

- **Coût** de faut rejet τ_{FR}= 10
- probabilité a priori d'apparition d'un locuteur cible P_{loc} = 0.01
- **Coût** de fausse alarme τ_{FA} = 1
- probabilité a priori d'apparition d'un imposteur P_{imp} = 1 − P_{loc} = 0.99

La DCF considéré s'exprime alors en fonction des taux de faux rejets TFR% et de fausses alarmes TFA % selon :

$$\mathrm{DCF} = \underbrace{\tau_{\mathrm{FR}} \times \mathrm{P}_{\mathrm{loc}}}_{0.1} \times \mathrm{TFR} + \underbrace{\tau_{FA} \times P_{imp}}_{0.99} \times \mathrm{TFA}$$

Les expériences décrites par la suite ont pour objectif de présenter le comportement général du notre système en utilisant les résultats sur les protocoles définis au paragraphe précédent. Dans un premier temps, nous allons évaluer le système GMM-UBM, par la suite nous allons évaluer le système GMM-SVM et enfin, une comparaison entre ces deux systèmes est effectuée.

6.2.1 Évaluation du système GMM-UBM

Cette partie a pour but d'évaluer les performances de système de reconnaissance du locuteur en utilisant l'approche GMM-UBM, en mode indépendant du texte. En phase de décision, nous avons utilisé la technique de logarithme de maximum de vraisemblance (LLR, **Logarithm** of the **Likelihood Ratio)**.

6.2.1.1 Évaluation du système d'IAL

Pour une quantité de données d'apprentissage de 100 secondes, et pour 256 composantes gaussiennes, on a obtenu un taux d'identification correcte de 91.1%.

- **Influence de l'ordre des modèles**

Dans cette expérience, nous étudions l'impact de l'ordre des modèles **K** (nombre de composantes dans le mélange gaussien) sur les performances d'identification**,** dans le cas où la quantité de données d'apprentissage des modèles des locuteurs est très faible (4 secondes de parole). Le tableau **6-1** présente les différents taux d'identification correcte (**TIC**) obtenus.

4 s de parole	
K	**TIC(%)**
2	42.2
4	46.7
8	55.5
16	51.5
32	62.6
64	53.5
128	66.6
256	71.7

Tab 6-1. Influence de l'ordre de modèle sur les performances d'identification.

Le tableau **6-1** montre que l'augmentation du nombre de composantes gaussiennes dans le mélange (K) apporte une augmentation de taux d'identification correcte(TIC), donc une amélioration des performances d'identification. Cependant, le gain est peu significatif au de là de K égal à 32, et le temps de calcul augmente considérablement.

- **Influence de la quantité d'apprentissage**

On cherche ici à évaluer les performances d'identification en fonction de la quantité de données d'apprentissage. Le tableau **6.2** présente les taux d'identification correcte en fonction de la quantité de données d'apprentissage :

K = 256	
durée	**TIC(%)**
3 s	60
6 s	64.4
9 s	80
18 s	80
36 s	82.2
100 s	91.1

Tab 6-2. Influence de la quantité d'apprentissage sur les performances d'identification.

Le tableau **6-2** montre les variations des taux d'identification correcte obtenus en fonction de la quantité de données d'apprentissage. Les TIC croissent significativement jusqu'à 30 secondes de données de parole. Au-delà de cette valeur les performances d'identification ont tendance à saturer. Pour avoir un taux d'identification correcte de 90%, il faut disposer d'au moins une quarantaine de secondes de données d'apprentissage.

6.2.1.2 Évaluation du système VAL

- **Influence de l'ordre des modèles**

Dans cette expérience, nous étudions l'impacte de l'ordre de modèle sur les performances de système de vérification du locuteur dans le cas où la quantité de données d'apprentissage est de 4 secondes de parole (les moyennes et les variances sont adaptés). Le tableau **6-3** résume les variations de taux d'erreur ERR obtenus:

4 s de parole		
K	**EER(%)**	**DCFmin × 100**
16	18	1
32	17.75	1
64	16.5	0.98
128	15	0.98
256	21.5	0.98

Tab 6-3. Influence de l'ordre de modèle sur les performances de vérification.

Le tableau **6-3** montre que l'augmentation du nombre de gaussiennes **K** dans la représentation des locuteurs apporte une amélioration des performances. Cependant, au de là de K égal à 128 on remarque une dégradation de performances de 5%. Nous pouvons justifier cette dégradation par la faible quantité de données d'apprentissage devant le nombre élevé de composantes gaussiennes dans le modèle GMM. la représentation des locuteurs par GMM avec un nombre de composantes élevé nécessite beaucoup de données d'apprentissage. Donc, nous conclurons que le nombre élevé de composantes gaussiennes va nuire à la précision du modèle du locuteur disposant de très peu de données d'apprentissage. Au niveau des valeurs du DCFminn, nous remarquons que l'amélioration est peu significative.

- **Influence de la quantité d'apprentissage**

Dans cette expérience, nous étudions l'impact de la quantité d'apprentissage sur les performances du système de vérification du locuteur. Le tableau **6-4** présente les taux d'erreur **EER** en fonction de la quantité de données d'apprentissage :

K = 256		
Durée	**EER(%)**	**DCFmin × 100**
4 s	21.5	0.98
8 s	11.5	0.98
16 s	11.25	1
32 s	9.5	0.95
64 s	8.3	0.95
100 s	7.4	1

Tab 6-4. Influence de la quantité d'apprentissage sur les performances de vérification.

Cette expérience illustre le bon comportement de notre système lorsque la quantité de données pour l'apprentissage est importante. Un gain (absolu) de 14% à l'EER est observé, avec un gain peu significatif à la DCFmin.

- **Influence de la quantité de données des tests**

Dans cette expérience, nous étudions l'impact de la quantité de données des tests (signal d'entré) sur les performances du système de vérification du locuteur. La quantité d'apprentissage pour cette expérience est de 100s de parole. Le tableau **6-5** présente les taux d'erreur **EER** en fonction de la quantité de données des tests :

	K=256	
Durée de test	**10s**	**30s**
EER	7.4	3.75
DCFmin × 100	1	0.98

Tab 6-5. Influence de la quantité de données de tests sur les performances de vérification.

Nous remarquons que les performances du système de VAL sont meilleures lorsque la quantité de données des tests augmente les performances obtenues avec 10 secondes par test sont plus réelles, car dans une application réelle, nous disposons que de quelques secondes lors d'un accès d'un client.

- **Comparaison entre l'algorithme EM-ML et l'algorithme EM-MAP**

Dans cette expérience, nous étudions les performances de vérification en utilisant deux critères d'estimation des modèles des clients, afin de comparer leurs performances.

Nous appliquons l'algorithme **EM** une fois avec le critère de maximum de vraisemblance **ML** (maximum likelihood), et une autre fois avec le critère **MAP** (Maximum à posteriori). En choisissant un nombre de composantes gaussiennes égale à 256 et un peu de quantité de données d'apprentissage (environ 8 secondes de parole), nous obtenons les taux d'erreur EER représentés dans le tableau ci-dessous:

	8 s de parole	
	EM-ML	**EM-MAP**
EER	16.5	11.5
DCFmin × 100	1	0.98

Tab 6-6. La distance en terme performance entre l'algorithme EM-ML et l'algorithme EM-MAP.

Le tableau **6-6** montre que les performances obtenues par le critère **MAP** comme critère d'estimation des modèles des clients sont meilleures que celles obtenues avec le critère **ML**, surtout en cas où nous disposons une faible quantité de données d'apprentissage.

- **Influence de la fréquence d'échantillonnage**

Les données disponibles pour évaluer notre système de reconnaissance sont à 48 KHZ. Cependant, les taux de reconnaissance donnés dans la littérature sont obtenus généralement avec des données à 16 KHZ. Cette expérience nous permet de voir comment se dégradent les résultats lorsque les données sont ré-échantillonnées à 16 KHZ.

Pour cette expérience, la quantité de données d'apprentissage est de 100s de parole. Nous utilisons un test par 30 secondes.

	K=256	
	Fe=48KHZ	**Fe=16KHZ**
EER	3.75	6.4
DCFmin × 100	0.98	0.96

Tab 6-7. Influence de la fréquence d'échantillonnage sur les performances de vérification.

Le tableau **6-7** nous montre que la diminution de la fréquence d'échantillonnage dégrade les performances de vérification.

6.2.2 Évaluation du système GMM-SVM

Cette partie a pour but d'évaluer les performances de système de vérification du locuteur en utilisant l'approche GMM-SVM proposé dans la section **5.2.1**, en mode indépendant du texte. Cette approche diffère de l'approche GMM-UBM au niveau du module de décision, elle utilise la technique des SVM au lieu de la technique LLR utilisé dans le système de l'état de l'art.

Pour bien évaluer notre approche, nous avons comparé les performances obtenues par cette approche avec celles obtenues par le système évalué en section **6.2.1**.

6.2.2.1 Modélisation GMM-SVM

Nous avons utilisé pour chaque client une session de 50 à 60 secondes de parole chacune pour apprendre le modèle GMM avec 256 composantes gaussiennes, alors que l'autre session (40 secondes de parole environ) a été divisée en 5 segments de 8 secondes chacune pour construire 5 vecteurs d'entrée représentant la classe client. La classe non-client a été représentée par 6 vecteurs construits en utilisant 6 segments de test de 12 secondes chacun provenant de 6 imposteurs (choisis au hasard).
Les vecteurs de test sont construit à partir d'un segment de parole de 10 secondes (issu soit des clients ou d'imposteurs) par la même façon de construction des vecteurs représentant la classe client et la classe non client (paragraphe **5.2.2**)

6.2.2.2 Résultats et interprétations

Nous avons appliqué trois types des noyaux, donc trois machines SVM : une SVM linéaire ; une SVM avec un noyau polynômial et une SVM à noyau RBF. Sauf les résultats obtenus par l'application du noyau RBF sont intéressantes et on peut les comparer avec les résultats obtenus par le système de référence qui utilise la technique LLR.

- **Avec le noyau RBF**

La forme générique de ce noyau est :

$$K(x,y)= \exp\left(-\gamma \|x-y\|^2\right)$$

Où γ est un paramètre de régulation.

Nous avons étudié l'effet des paramètres de contrôle C et γ sur les performances du système de vérification, C étant la tolérance.

- On prend C=10 et on varie γ :

γ	**EER(%)**	**DCFmin × 100**
0.01	23	0.91
0.02	16.5	0.78
0.025	13.5	0.83
0.04	18.5	0.87
0.08	18.5	0.7

Tab 6-8. Influence de valeur de **γ** sur les performances de vérification.

Le tableau **6-8** montre que les meilleures résultats sont obtenus par le système GMM-SVM utilisant le noyau RBF avec **γ = 0.025,** avec un gain de 0.2% à la DCFmin est observé.

- Maintenant, on prend γ =0.025 et on varie la tolérance C :

C	**EER(%)**	**DCFmin × 100**
0.01	51.5	1
1	45	1
5	13	0.83
10	13.5	0.83
100	36.5	0.83

Tab 6-9. Influence de valeur de la tolérance C sur les performances de vérification.

Le tableau **6-9** montre que les meilleures résultats sont obtenus par le système GMM-SVM utilisant le noyau RBF avec **γ = 0.025 et C=5,** avec un gain de 0.17% à la DCFmin est observé.

6.2.3 Comparaison des résultats

Afin de bien évaluer notre approche GMM-SVM, nous avons comparé les performances obtenues par cette approche avec celles obtenues par le système GMM-UBM, la durée des données d'apprentissage des modèles des clients est d'environ 64 secondes. Les résultats de comparaison sont dans le tableau **6-10**.

64 s	EER(%)	DCFmin × 100
GMM-UBM	8.3	0.95
GMM-SVM	13	0.83

Tab 6-10. Comparaison des résultats obtenus par l'approche GMM-UBM et l'approche GMM-SVM.

Le tableau **6.10** montre que les performances du système de référence (GMM-UBM) basé sur la technique LLR sont meilleures que les performances obtenues par le système GMM-SVM. L'EER (Taux d' Égale Erreur) est de 8.3% pour le système de référence contre 13% obtenu par le système GMM-SVM. Cette dégradation de performances peut être expliquée par le petit nombre de vecteurs utilisés pour apprendre le modèle SVM de chaque client (11 vecteurs en tout dont 5 représentant la classe client et 6 représentant la classe non-client).

6.3 Conclusion

Dans ce chapitre, nous avons présenté les évaluations GMM sur la base de données VoxForge. Dans les évaluations de l'approche GMM-UBM, les expériences réalisées avec des données d'apprentissage de l'ordre de 10 secondes ont montré que l'augmentation du nombre de gaussiennes dans le mélange apporte une amélioration des performances peu significative au delà de 32 gaussiennes. Cela est du au fait que la représentation des locuteurs par GMM avec un nombre de gaussiennes élevé nécessite beaucoup de données d'apprentissage. L'augmentation parallèle des données d'apprentissage et du nombre de gaussiennes dans le mélange GMM améliore les performances. Les expériences d'évaluation ont montré qu'avec une modélisation de 256 gaussiennes et

avec 100 secondes d'apprentissage, le système de reconnaissance par GMM-UBM atteint un taux d'erreur d'identification de 8.9% et un EER égale à 7.4%.

En ce qui concerne l'approche GMM-SVM, les expériences d'évaluation ont montré qu'avec un paramètre de régulation $\gamma = 0.025$ et une tolérance C=5, le système de VAL atteint un EER = 13%.

Conclusion Générale

Ce travail thèse s'inscrit dans le domaine de la Reconnaissance Automatique du Locuteur (RAL). Un système de RAL consiste à vérifier l'identité d'une personne à partir de sa voix. On peut également utiliser la VAL associée à d'autre modalités (ex : vérification de signatures, analyse du visage, des empreintes digitales et de la forme de la main) dans des systèmes de vérification et d'authentification multimodale de l'identité.

Notre travail se base sur la reconnaissance automatique du locuteur indépendant du locuteur. Donc, nous avons commencé par étudier les différentes techniques afin de choisir la technique qui répond le mieux aux contraintes de robustesse. Nous avons choisi une approche statique basé sur les mélanges de gaussiennes GMM, approche la plus performante. Le critère le plus utilisé pour l'apprentissage des modèles GMM, est le critère de maximum de vraisemblance **ML** (maximum likelihood). Mais le critère **MAP (**Maximum A Posteriori) est très utilisé dans le cas où nous disposons de très peu de données.

Aussi, nous avons implémenté l'algorithme EM, algorithme sert à l'apprentissage des modèles des locuteurs en maximisant l'un des deux critères (EM ou MAP). Pour la phase de décision, la technique du rapport du maximum de vraisemblance a été utilisée.
La méthode GMM que nous avons implémenté a été testée sur des signaux de parole de la base de données VoxForge, avec deux minutes de parole pour chaque locuteur pour l'apprentissage et 10 à 30 secondes de chaque locuteur pour le test.
Les résultats obtenus montrent que le critère MAP apporte une amélioration de performance de 5% par rapport à celles obtenues en utilisant le critère ML.

Enfin, nous nous sommes intéressés au processus de décision pour la tâche de vérification du locuteur. Pour cela nous avons appliqué la technique des machines à vecteurs de support (SVM) au lieu de la technique du rapport du maximum de vraisemblance. Pour réaliser cette tâche, nous avons implémenté les SVM pour la vérification du locuteur. Pour cela, nous avons construit un modèle GMM-SVM pour

chaque client. L'algorithme SMO a été choisi comme algorithme d'optimisation, parmi les différents algorithmes disponibles. Cet algorithme est simple et rapide pour résoudre le problème de programmation quadratique des SVM sans la nécessité de stocker une grande matrice en mémoire et sans une routine numérique itérative pour chaque sous-problème.

Les résultats obtenus montrent que les performances sont inférieures de 4% par rapport à celles obtenues par la technique LLR. Cette différence entre les performances peut être expliquée par le petit nombre de vecteurs utilisés pour apprendre le modèle GMM-SVM de chaque client. Ce travail pourra être poursuivre par l'évaluation sur d'autres bases de données plus riches en terme de nombre d'individus et de quantité de parole.

RÉFÉRENCES

[1] R. Bolle, S. Pankanti et Jain, R.. Biometrics, Personal Identification in Networked Society . Library of Congress Cataloging-in-Publication Data, USA, 1998.

[2] S. Liu, et M. Silverman, A practical guide to biometric security technology, Actes de IEEE, vol. 3, n.1 , Janvier 2001.

[3] International Biometric Group. Which is the Best Biometric Technology? http://www.biometricgroup.com/reports/public/reports/best_biometric.html

[4] G. Jobard. Traitement des sons du langage. GINLANG CI-NAPS UMR 6232 CNRS CEA Universités de Caen & Paris Descartes.

[5] G. Fant. Acoustic Theory of Speech Production, with Calculations based on X-Ray Studies of Russian Articulations, Mouton & Co, 1960.

[6] M. Ben. Approches robustes pour la vérification du locuteur par normalisation et adaptation hiérarchique, Thèse de Doctorat, Université de Rennes 1. 2004

[7] L.R. Rabiner et R. W. Schafer, 2007, Introduction to Digital Speech Processing.

[8] S.Furui, Toward the Ultimate Synthesis/Recognition System. In Voice communication Between Humans and Machines, pp. 450-466.

[9] J.Rachedi, Reconnaissance et classification de phonèmes, Mémoire de Master Sciences et Technologie de l' UPMC, Laboratoire de l' IRCAM, Paris 2005.

[10] P. Delacourt et C.J.Wellekens, La segmentation et le regroupement pour l'indexation des documents audio, Thèse de doctorat, Institut Eurecom, Septembre 2000.

[11] G.Baudoin, P.Jardin, G.Chollet et G.Gross, Comparaison de techniques de paramétrisation spectrales pour la reconnaissance vocale en milieu bruité, GRETSI, Groupe d'Etudes du Traitement du Signal et des Images , Septembre 1993.

[12] K. Aizawa, Y. Nakamura et S. Satoh, Advances in Multimedia Information Processing PCM 2004, 5th Pacific Rim Conference on Multimedia.

[13] J.A.Campbell. 'Speaker Recognition: A tutorial', Actes de l'IEEE, vol. 85, n.9, pp. 1437-1462, 1997.

[14] L.R. Rabiner and R.W. Schafer, Introduction to Digital Speech Processing. Foundations and Trends® in Signal Processing: Vol. 1: No 1–2, pp 1-194.

[15] E. Osuna, R. Freund, et F. Girosi, « Training Support Vector Machine : An application to Face Detection », IEEE Proc, Int. Conf. Computer Vision and Pattern Recognition, 6, 1997.

[16] P.J.Philips, Support vector machines applied to face recognition, Adv. Neural Inform. Process, Syst. 11, pp.803-809, 1998.

[17] F. Bimbot, I. Magrin-Chagnolleau, et L. Mathan, Second-order statistical measures for

text-independent speaker identification, Speech Communication 17(1-2), pp.177–192, 1995

[18] D.A. Reynolds et C.R.Richard, Robust Speaker identification and verification using Gaussian mixture speaker models, Speech Communication, IEEE Transaction on speech and audio processing, vol .3 n.1. Janvier 1995.

[19] A.Martin et M.Przybocki, NIST's Assessment of Text-Independent Speaker Recognition Performance, The COST 275 Workshop – The Advent of Biometrics on the Internet, pp.25-32.

[20] R.Boite, H.Bourlard, T.Dutoit, J.hancq et H.Leich, Traitement de la parole. Presses Polytechniques et Universitaires Romandes - Collection Electricité , 2000.

[21] D.A.Reynolds, F.T. Quatieri, et R.B. Dunn, Speaker Verification Using Adapted Gaussian Mixture Models, Digital Signal Processing 10, 19–41, 2000.

[22] J. P. Egan, "Signal Detection Theory and ROC analysis", Academic Press, New York, 1975.

[23] A. Martin, G. Doddington, T. Kamm, M. Ordowski, M. Przybocki, "The DET Curve in Assessment of Detection Task Performance", Proc. EUROSPEECH'97, Vol. 4, pp.1895-1898, Rhodes Greece, 1997.

[24] A. Martin et M. Przybocki, The NIST 1999 speaker recognition evaluation - An overview, National Institute of Standards and Technology, 2000.

[25] D.Jurafsky, Speech Recognition, Synthesis and Dialog. CS 224S / LINGUIST 281. Lecture 9: Feature Extraction and start of Acoustic Modeling (VQ).

[26] A. P. Dempster, N. M. Laird et D. B. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm**,** Journal of the Royal Statistical Society. Series B (Methodological), Vol. 39, No. 1. pp.1-38. 1977.

[27] A.Ghoshal, P.Ircing et S.Khudanpur, Hidden Markov Models for Automatic Annotation and Content Based Retrieval of Images and Video, ACM SIGIR Conference on Research and Development in Information Retrieval, 2005.

[28] Y. Zhang, A.I. Fahmy et M. S. Scordilis. Speaker Verification Using Speaker-Specific Prompts, Department of Electrical and Computer Engineering, University of Miami, Coral Gables, Florida 33124.

[29] V. Vapnik, The nature of statistical learning theory, Spring-Verlag, New York, 1995.

[30] C. Cortes et V. Vapnik, Support-Vector Networks, Machine Learning, 20, 1995.

[31] B. Schölkopf, O. Burges, et V. Vapnik, Incorporating invariances in support vector learning achines, Vol. 1112, pages 47-52, 1996.

[32] V. Blanz, B. Schölkopf, H. Bulthoff, C. Burges, V. Vapnik, et T. Vetter, Comparision of view-based object recognition algorithms using realistic 3D models, International Conference on Artificial Neural Networks, 1996.

[33] E. Osuna, R. Freund et F. Girosi, Training support vector machines: an application to face detection, 1997.

[34] T. Joachims, Text categorization with support vector machines: learning with many relevant features,1998.

[35] F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys, vol. 34(1), 2002.

[36] Y. Ben Ayed, Détection de mots clés dans un flux de parole, Thèse de l'Ecole Nationale Supérieure des Télécommunications, 2003.

[37] W. M. Campbell, E. Singer, P. A. Torres-Carrasquillo et D. A. Reynolds, Language Recognition with Support Vector Machines, In Proc. Odyssey: The Speaker and Language Recognition Workshop in Toledo, Spain, ISCA, pages 41–44, 2004.

[38] J.Kharroubi, Etude de techniques de classement Machines à vecteurs supports pour la vérification automatique du locuteur, PhD thesis, Signal et Images, ENST, 2000.

[39] S.Guillaume, Apprentissage de classifieurs à noyau sur des données bruitées, Mémoire du Stage de Master 2 , Laboratoire d'Informatique Fondamentale de Marseille (LIF), Juin 2006.

[40] R. Courant et D. Hilbert, Methods of mathematical physics. Vol.I, New York, Interscience, 1953.

[41] J.C. Platt, Fast Training of support Vector Machines using Sequential Minimal Optimization, Microsoft Research, Microsoft Way, Redmond, WA 98052, USA.

[42] G. Ratzer, H. Vangheluwe, The Implementation of Support Vector Machines Using The Sequential Minimal Optimization Algorithm, School of Computer Science, McGill University, Montreal, Canada, April 2006.

[43] T. S. Jaakkola et D. Haussler, Exploiting generative models in discriminative classifiers, Advances in Neural Informations Processing Systems 11, pp.487–493, 1999.

[44] W. Campbell, D. Sturim, et D. Reynolds, Support Vector Machines Using GMM Support vectors for Speaker Verification. IEEE Signal Processing Letters 13(5), 308. 60, 2006

[45] The NIST Year 2008 Speaker Recognition Evaluation Plan.

[46] http://www.repository.voxforge1.org/downloads/SpeechCorpus/Trunk/Audio/Original/48kHz_16bit