UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTÉ DES SCIENCES ET TECHNIQUES FÈS
DÉPARTEMENT D'INFORMATIQUE

# PROJET DE FIN D'ETUDES

## MASTER SCIENCES ET TECHNIQUES
## SYSTÈMES INTELLIGENTS & RÉSEAUX

# APPRENTISSAGE AUTOMATIQUE DES MODÈLES DE MÉLANGES GAUSSIENNES BASÉ SUR LES ALGORITHMES DE FRACTION ET FUSION (SPLIT AND MERGE ALGORITHMS) DES GAUSSIENNES POUR LA RECONNAISSANCE DE LOCUTEURS

**Lieu de stage : Laboratoire systèmes intelligents et applications**

**Réalisé par : Safhi Hicham Moad**

**Soutenu le 24/06/2015**

**Encadré par :**

Mr. Kharroubi Jamal
Mme. Majda Aicha

**Devant le jury composé de :**

Mr. Kharroubi Jamal
Mme Majda Aicha
Mr. Zenkouar Khalid
Mme Lamrini Loubna

**Année Universitaire 2014-2015**

# Remerciement

Quatre mois pour écrire ces 69 pages, 1971 lignes, 15647 mots et 105367 Caractères. La probabilité qu'un singe écrive une manuscrite identique en tapant au hasard sur un clavier est donc sous hypothèse que l'on arrête le singe au bout des 105367 Caractères et que celui-ci utilise les touches de son clavier à 26 lettres, 10 chiffres et quelques 33 symboles supplémentaires de manière équiprobable, la probabilité est ainsi de $p_1$ = [1/(26 + 10 + 33)]^105367 , cette probabilité est somme tout assez faible.

J'espère que le contenu de ces 105367 caractères intéressera les lecteurs avec une probabilité $p_2$ plus grande que $p_1$ . L'écart entre $p_1$ et $p_2$ pouvant être vu comme une mesure de l'apport de cette mémoire à la communauté scientifique.

Après Dieu, je souhaite sincèrement remercier toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce travail de recherche, ainsi que toutes les personnes qui m'ont apporté leur soutien.

Je souhaite tout d'abord adresser mes premiers remerciements à **Pr. Kharroubi Jamal**, responsable du Master SIR, et membre du Laboratoire Systèmes Intelligents et Applications (LSIA), pour la confiance qu'il m'a accordé en acceptant d'encadrer ce mémoire. Durant toute la période de mes études, j'ai été extrêmement sensible à ses qualités humaines d'écoute et de compréhension, sa droiture et son sérieux, je suis très honoré qu'il ait accepté de me confier ces travaux.

Je remercie **Pr. Aicha Majda**, Professeur à la FST, et membre du Laboratoire Systèmes Intelligents et Applications (LSIA), pour l'aide qu'elle m'a apporté, ses précieuses recommandations ainsi son encouragement qui m'a été adressé durant tout le déroulement de ce travail.

Je remercie également les membres de jury, **Pr. Zenkouar Khalid** et **Pr. Lamrini Loubna**, d'avoir pris le temps d'évaluer mon travail.

J'adresse ma plus profonde gratitude au Monsieur Bouziane Ayoub, doctorant au laboratoire SIA, pour son aide et ses conseils de grande qualité scientifique durant mon stage, sans lui l'avancement de ce projet n'aurait pas été possible.

J'aimerais dédier ce travail à mes parents, mes frère Amine EL Mahdi, Ahmed Marouane, mes sœurs Sarah et Soukaina, ainsi qu'à tous mes amis(es) et mes proches.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

# Résumé

La reconnaissance automatique de locuteur désigne l'utilisation d'une machine pour identifier une personne à partir les caractéristiques de sa voix. Au cours des dernières années, la RAL est devenu un domaine d'étude actif. En effet, l'évolution des technologies et des appareils de télécommunication, des nouveaux produits et des nouveaux moyens de communication sont disponibles (VoIP, smartphones,…) rend l'utilisation des systèmes RAL utile et importante dans nombreuses applications. C'est un domaine riche d'applications allant de la sécurisation d'accès à l'indexation des documents audio. Pour laisser le champ à un large éventail d'applications, nous nous intéresserons à la reconnaissance automatique du locuteur en mode indépendant du texte.

Divers systèmes de RAL ont été développés, mais trouver un système performant reste un problème ouvert. Une parmi les approches de modélisation qui semble prometteuse et qui est utilisé dans la majorité des systèmes actuels, est la modélisation par mélanges de gaussiens malgré certaines contraintes et limites dans les algorithmes utilisés dans cette approche.

Nous présentons dans ce travail une étude bibliographique sur le système de RAL, en présentons les résultats d'une étude comparative des variantes de l'algorithme d'apprentissage division-fusion (split and merge) pour l'auto classification des gaussiennes.

**Mots clés :** reconnaissance automatique du locuteur, Identification automatique du locuteur, autoclassification , modèle de mélanges de gaussiens, algorithme division-fusion.

# Abstract

Speaker recognition is the task of identifying a speaker based on the characteristics of her/his voice. ASR has become an active research area over the last few years. In fact, with the evolution of technology and telecommunication devices, some new products and new tools of communication are available (VOIP, smartphones ...) which makes the use of the ASR system useful in many applications, sometimes important. It is a rich field of applications ranging either in securing access or indexing of audio. To leave the field to a wide range of applications, we will focus on the independent text speaker recognition system.

Various ASR systems have been developed, but finding an efficient system remains an open problem. Among the modeling approaches that seems promising and is used in most current systems is Gaussian mixture models, despite some constraints and limitations in the algorithms used in this approach.

In this work we present a bibliographic study on the ASR system, presenting a comparative study of the results of several variants of the split and merge algorithm for self-Gaussian classification.

**Keyworkds :** speaker recognition, speaker identification, self-organisation mixture models, unsupervised learning, split and merge algorithm.

# Table des matières

# List des figures



# Liste des tableaux

# Liste des abréviations

**RAL :** Reconnaissance automatique du locuteur.

**RAP :** Reconnaissance automatique de la parole.

**ASR :** Automatic speaker recognition.

**EM :** Expectation maximization.

**SMEM :** Split and merge expectation maximization.

**TFD :** Transformé de Fourrier.

**DCT :** Discrete cosine Trasnform.

**VQ :** Vector Quantisation.

**MMG :** modèle de mélange gaussienne.

**GMM:** Gaussian mixture models.

**HMM:** Hidden Markov model.

**EM:** Expectation Maximisation.

**MDL:** Minimum Description Length.

**VoIP:** Voice over IP.

**MFCC:** Mel-Frequency Cepstral Coefficients.

**MFSC:** Mel Frequency Spectral Coefficients.

**LPCC:** Linear Predictive Cepstral Coefficients.

**LPC:** Linear Predictive Coefficients.

**LFSC:** Linear Frequency Spectral Coefficients.

**LFCC:** Linear Frequency Cepstral Coefficients.

# Introduction générale

La reconnaissance automatique de locuteur (RAL) est le processus de détecter automatiquement l'identité de celui qui parle en se basant sur les informations incluses dans son signal vocal, c'est une tache particulière de la communication homme-machine vocale.
Lorsqu'un locuteur prononce une phrase, notre cerveau est capable d'analyser une gamme d'autres informations que le message prononcé, tel que : l'âge de locuteur, son sexe, son humeur, …etc., simplement en l'écoutant. De plus, si on a déjà entendu ce locuteur, il est possible qu'on reconnaître son identité. Depuis des années les chercheurs ont intéressé par le sujet de rendre les machines capables de faire cette tâche de reconnaissance d'une manière automatique, du fait qu'il est un sujet très utile dans de nombreuses applications et environnements de notre vie, allant des applications domestiques aux applications militaires. On peut utiliser par exemple cette technique : comme moyen d'authentification biométrique, pour contrôler des machines à distant, pour archiver les documents multimédias,…etc.

Le domaine général de la reconnaissance du locuteur comprend deux tâches fondamentales: identification du locuteur, qui consiste à déterminer l'identité de locuteur parmi un ensemble de locuteurs possibles, et la vérification du locuteur, qui consiste à accepter ou refuser l'identité proclamée par un locuteur, ces deux taches peuvent êtres en mode dépendant ou indépendant du texte.

La construction d'un système de reconnaissance du locuteur implique les taches d'extraction des échantillons de chaque locuteur, la modélisation qui signifie la représentation de ces échantillons dans un espace pour arriver à les comparer, et finalement prendre une décision. Plusieurs approches de modélisation ont été proposées dans la littérature : approche vectorielle, approche connexionniste, approche statistique et approche prédictive. De ce large panel, l'approche statistique demeure au premier plan des systèmes de RAL des récentes années, offrant d'excellents résultats. Plus précisément en ce qui concerne l'identification en mode indépendant du texte, la modélisation par mélanges de gaussiennes fournit des bonnes performances, et constitue l'état de l'art en la matière.
Il s'agit de modéliser un locuteur par un mélange gaussien, c.-à-d. par une somme pondérée de gaussienne. La détermination des paramètres de ce mélange s'effectue à l'aide de l'algorithme EM (Expectation Maximisation). C'est un algorithme performant capable de bien estimer les paramètres du modèle, malheureusement il présente certaines limites :

- Le nombre de gaussien doit être fixé à l'avance, ce qui est difficile dans notre cas.
- Mauvaise initialisation conduite à des maximums locaux.

En fait, ces problèmes sont liés à la majorité des méthodes de classification non supervisées. Dans ce travail, nous avons étudié l'algorithme SMEM (Split and Merge EM) qui permet de combler ces limitations dans le cas où les données sont représentées par des densités gaussiennes. Cet algorithme est présenté en deux versions : la première permet d'éviter les maximums locaux pour un nombre de gaussiennes données, la deuxième permet à la fois de déterminer automatiquement le nombre de gaussien, et d'éviter les maximums locaux. Ces algorithmes sont basés sur la méthode de division fusion appliquée aux mélanges gaussiens. L'idée est d'effectuer un ensemble des opérations de division et fusion des

gaussiennes en se basant sur certains critères jusqu'à trouver la bonne modélisation c.-à-d. trouver les paramètres qui décrivent mieux le mélange.

Ce rapport est organisé comme suit : dans le chapitre I nous introduisons l'architecture du système RAL, puis nous présentons l'étape d'extraction des paramètres en chapitre II, et en chapitre III nous détaillons la partie de modélisation, puis nous passons aux algorithmes d'autoclassification en chapitre IV. Le chapitre V est dédié aux expériences et résultats et nous terminons par une conclusion.

# Chapitre I

# Principe de reconnaissance automatique de locuteur

# 1. Introduction

Dans ce chapitre, nous proposons une présentation générale sur le système de reconnaissance automatique de locuteur, ses différentes taches, passant par les domaines d'utilisations et les difficultés, nous terminons par le mode de fonctionnement.

# 2. Taches de reconnaissance automatique du locuteur

Dans notre vie quotidienne, nous utilisons plusieurs moyens pour communiquer avec les autres humains: les textes écrits, les gestes du corps, les dessins, et la voix. Mais sûrement la communication vocale est la plus utilisée. Depuis plusieurs années, les chercheurs étudient la possibilité d'utiliser ce moyen non seulement pour communiquer avec les êtres vivants, mais aussi avec les machines.

La communication homme-machine parlée représente un domaine vaste qui regroupe plusieurs taches (Figure 1), les deux grands axes de cette communication sont : la synthèse de la parole qui s'intéresse à la génération automatique de la parole artificielle, et l'analyse de la parole qui est le domaine qui s'occupait d'analyser les informations contenant dans un signal de voix parlé : soit on cherche à reconnaître la parole soit le locuteur.

La tâche de reconnaissance de locuteur comporte deux modes : identification et vérification de locuteur. Cette reconnaissance peut être en mode dépendant du texte ou en mode indépendant du texte.

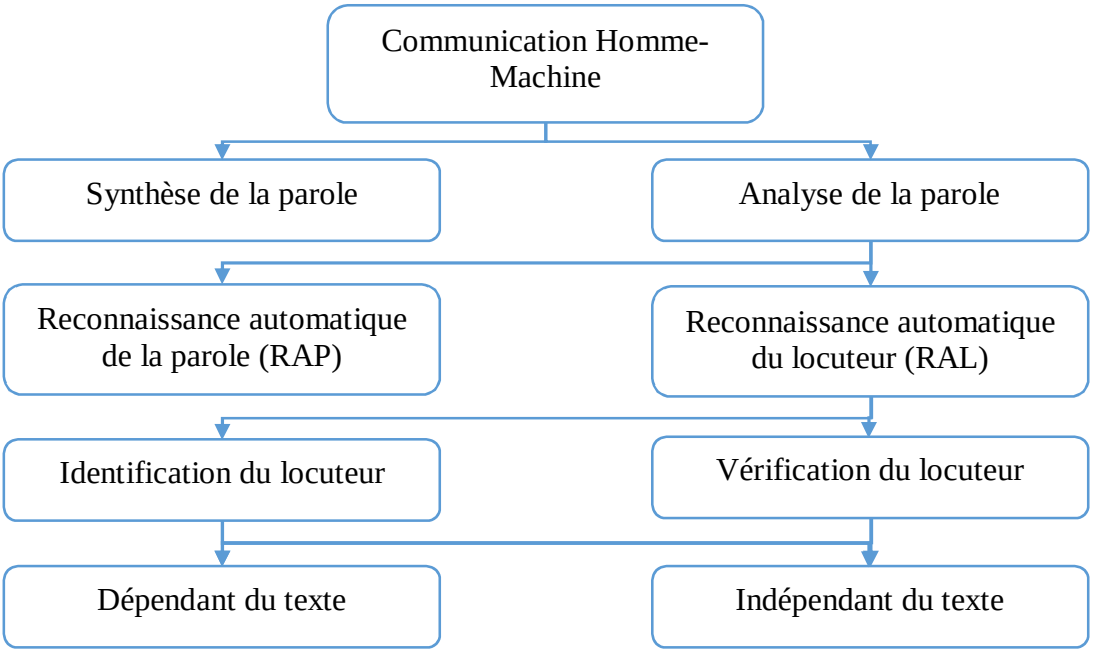



*Figure 1 : Schéma de la communication homme-machine*

## 2.1. Identification et vérification du locuteur

Les systèmes de reconnaissance automatique de locuteur peuvent être classé selon leur mode de fonctionnement, on distingue deux grandes modes : identification et vérification.

### 2.1.1. Identification du locuteur :

L'identification du locuteur consiste à identifier une personne à l'intérieur d'une population déterminée, en comparant son signal vocal à tous les signaux dans la base. Ici le locuteur ne déclare pas son identité, mais le système qui doit la trouver. Sa tâche est de chercher parmi les N locuteurs celui qui a parlé. Le système effectue N comparaison, chaque comparaison donne un score de vraisemblance puis le système choisit l'identité qui a le score maximum. La décision sera une identité d'un locuteur $L(i)$ avec $i = 1, \dots, N$. La figure (2) résume le processus d'identification.

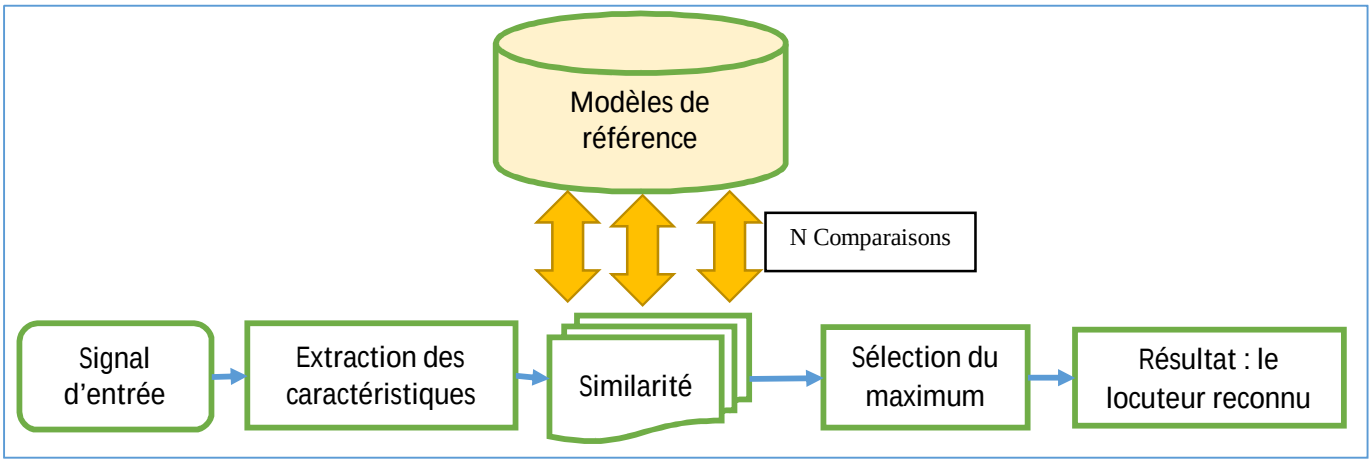



*Figure 2: Schéma d'identification du locuteur*

On distingue encore deux sous taches dans le problème d'identification, selon la possibilité que le système soit utilisé par tout le monde (ensemble ouvert) ou par un ensemble limité de locuteurs (ensemble fermé) :

- ***Ensemble fermé :***

Un système à ensemble fermé (figure 3) a un nombre fixe d'utilisateurs connus qui vont utiliser le système. Dans ce cas, la décision prise par le système sera le locuteur le plus probable de générer la parole d'entrée.

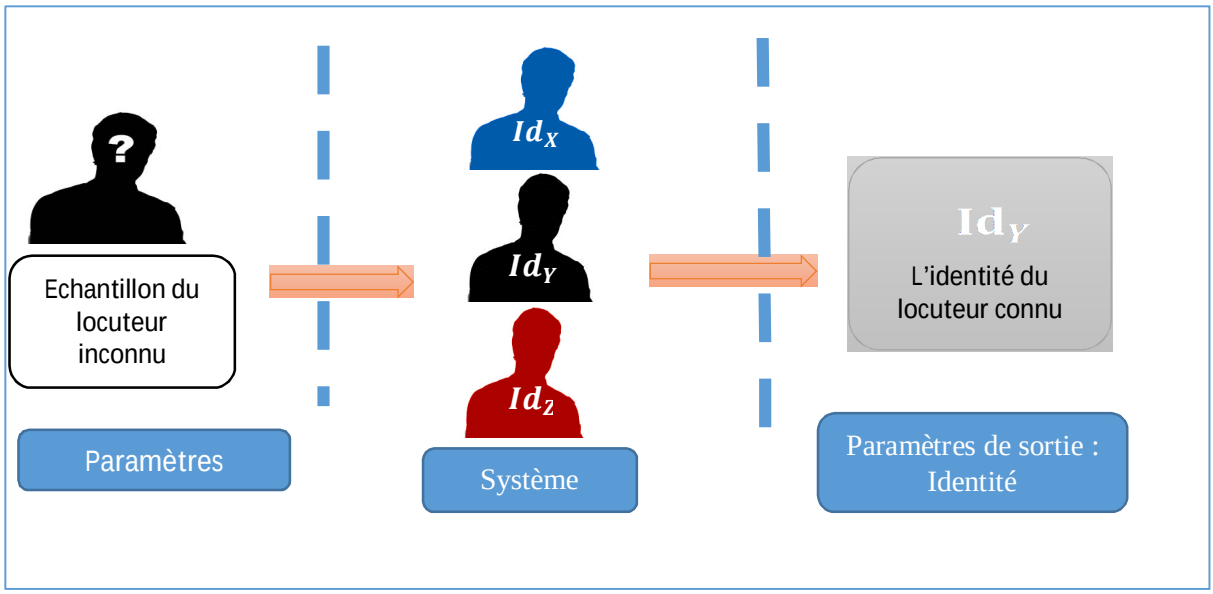



*Figure 3: Schéma d'identification du locuteur en un ensemble fermé*

- ***Ensemble ouvert :***

Un système à ensemble ouvert (figure 4) examinera la possibilité que l'utilisateur qui tente d'entrer dans le système soit inconnu. Cela signifie qu'il n'y a pas un modèle associé à cet utilisateur. Par conséquent la décision prise par le système est que l'utilisateur est inconnu ou un imposteur.

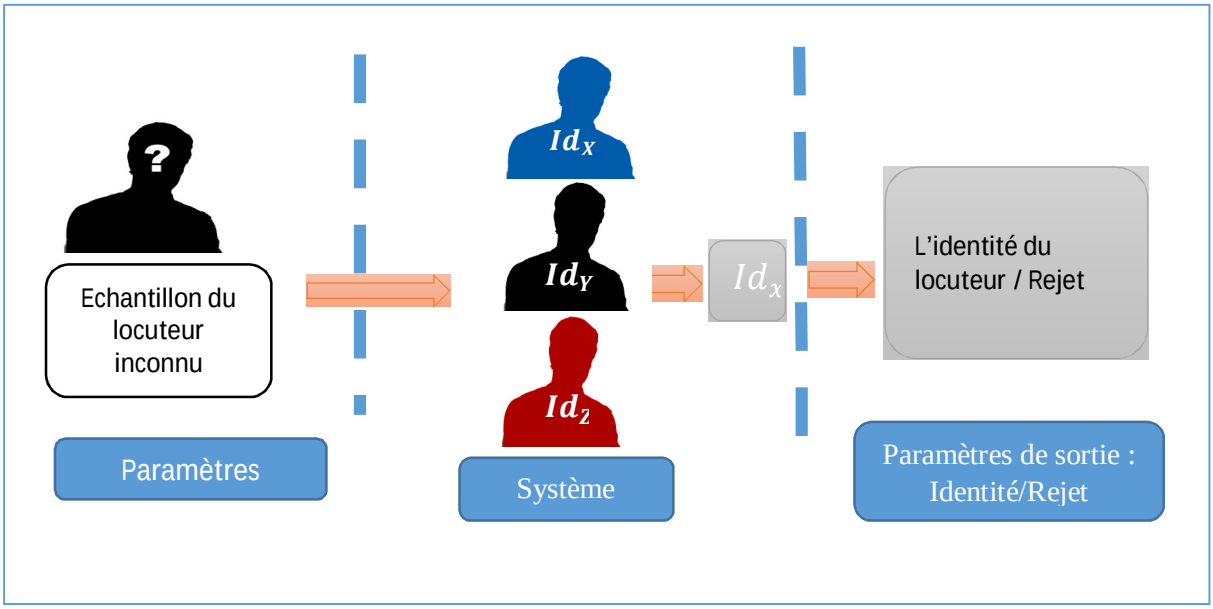



*Figure 4: Schéma d'identification du locuteur en un ensemble ouvert*

## 2.1.2. Vérification du locuteur :

La vérification de locuteur ou authentification est le processus de décision d'accepter ou de rejeter la demande d'une personne. Il s'agit d'une décision binaire. Le locuteur déclare son identité, puis le système va vérifier si cette identité est correcte en analysant sa voix et la comparant avec la voix de la personne qu'il prétend d'être. La figure (5) résume le processus de vérification.

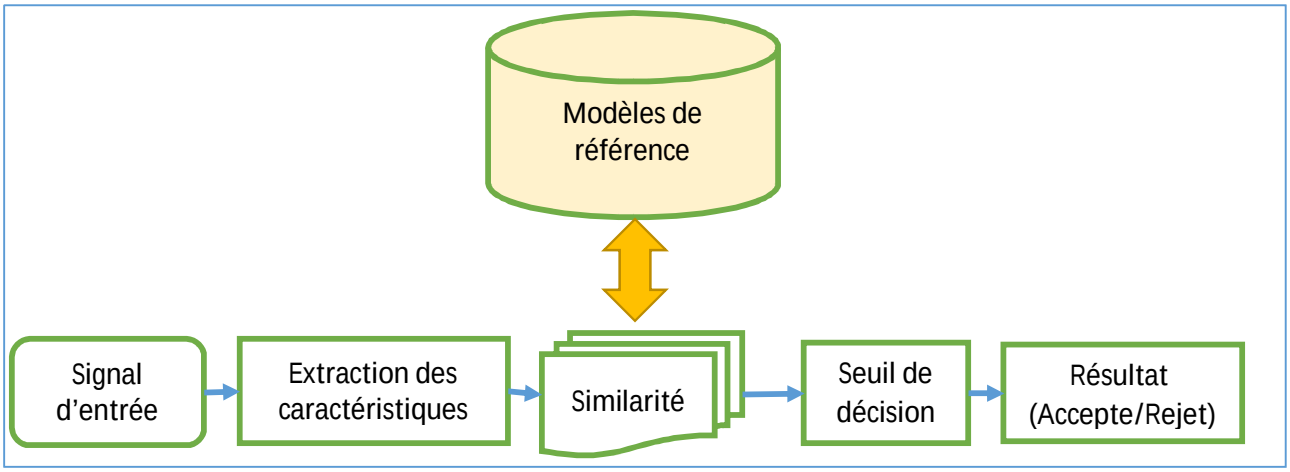



*Figure 5 : Schéma de vérification de locuteur*

Par exemple, dans le cas d'un service téléphonique, un utilisateur veut écouter ses messages de boîte vocale il appelle le service. Le numéro de la carte SIM contient l'identité de l'utilisateur. Si le service contient un système de RAL, il peut vérifier si le téléphone est utilisé par son propriétaire ou par un imposteur. Le système fait donc une seule comparaison du

signale vocale de l'utilisateur avec le modèle de base du vrai utilisateur puis il répond par une décision oui /non basé sur le résultat de comparaison.

Il est clair que le problème d'identification est plus complexe que la vérification. Et par conséquent les résultats obtenus dans les systèmes d'identification seront moins bons que dans les systèmes de vérification.

## 2.2. Dépendance et indépendance du texte :

Une autre méthode de classifier les systèmes de RAL est basée sur l'utilisation de la phrase prononcée par le locuteur. On distingue deux modes : dépendant et indépendant du texte.

### 2.2.1. Mode dépendant du texte :

Dans le mode dépendant de texte, la phrase utilisée dans la phase de test est identique à la phrase utilisée dans la phase d'apprentissage. Dans ce cas une autre couche de sécurité est ajoutée : la voix doit être produite par un utilisateur autorisé, et d'autre part, l'utilisateur doit fournir la phrase correspondante. Donc, dans ce mode le locuteur doit avoir une connaissance préalable du système.

### 2.2.2. Mode indépendant du texte :

Les systèmes en mode indépendant du texte identifient le locuteur indépendamment de l'énoncé. Le locuteur peut n'avoir aucune connaissance au préalable du contenu de la phrase utilisée dans l'apprentissage, car dans cette phase le système assiste seulement aux attributs dépendants du locuteur et ne repose pas sur la séquence de mots prononcés par le locuteur.

Il est clair que le mode indépendant du texte est beaucoup intéressant, car son fonctionnement ne nécessite pas un mot spécifique à dire pour reconnaître le locuteur. Ce qui implique que les applications de cette méthode ne sont pas limitées à la sécurité biométrique, mais aussi en d'autres domaines qui besoins juste de savoir qui parle indépendamment de ce qu'il dit.

# 3. Domaines d'utilisation :

Les domaines d'applications d'un système de reconnaissance automatique de locuteur sont très variés et en croissance. L'intégration de ce système dans des applications permet d'ajouter une couche de sécurité et de gagner du temps en facilitant plusieurs tâches dans nos vies. Les exemples suivants illustrent certains cas d'utilisations d'un système RAL :

En sécurité : ces jours plusieurs entreprises permets aux ses clients d'utiliser des opérations à distant par téléphone. Or ceci pose des problèmes de sécurité. Un imposteur peut facilement usurper l'identité d'un client et effectue des transactions éligibles. L'intégration d'un système RAL permet à l'entreprise de vérifier l'identité de l'utilisateur puis faire la distinction entre un vrai client et l'imposteur.

Le contrôle des machines à distant : contrôle des machines à distant en utilisant la voix, ce qui est très utile surtout pour les handicapés. Par exemple en médecine, lorsqu'un chirurgien

a les deux mains occupées, il peut parler pour demander une information technique au lieu de taper sur un clavier.

La recherche des informations dans des documents multimédia. Par exemple : trouver les appels d'un client à un centre d'appels. La segmentation des documents : par exemple le regroupement des interventions dans une émission par locuteur. L'indexation des documents, détection d'un menteur,….etc.

## 4. Principe de fonctionnement d'un système RAL :

Le problème de reconnaissance du locuteur peut être vu comme problème de classification ou reconnaissance de forme. Comme tout système de reconnaissance de forme, la reconnaissance de locuteur fonctionne en deux phases : la première est la phase d'apprentissage, la deuxième est la phase de reconnaissance ou de test. Le schéma global du système est illustré dans la figure (6).

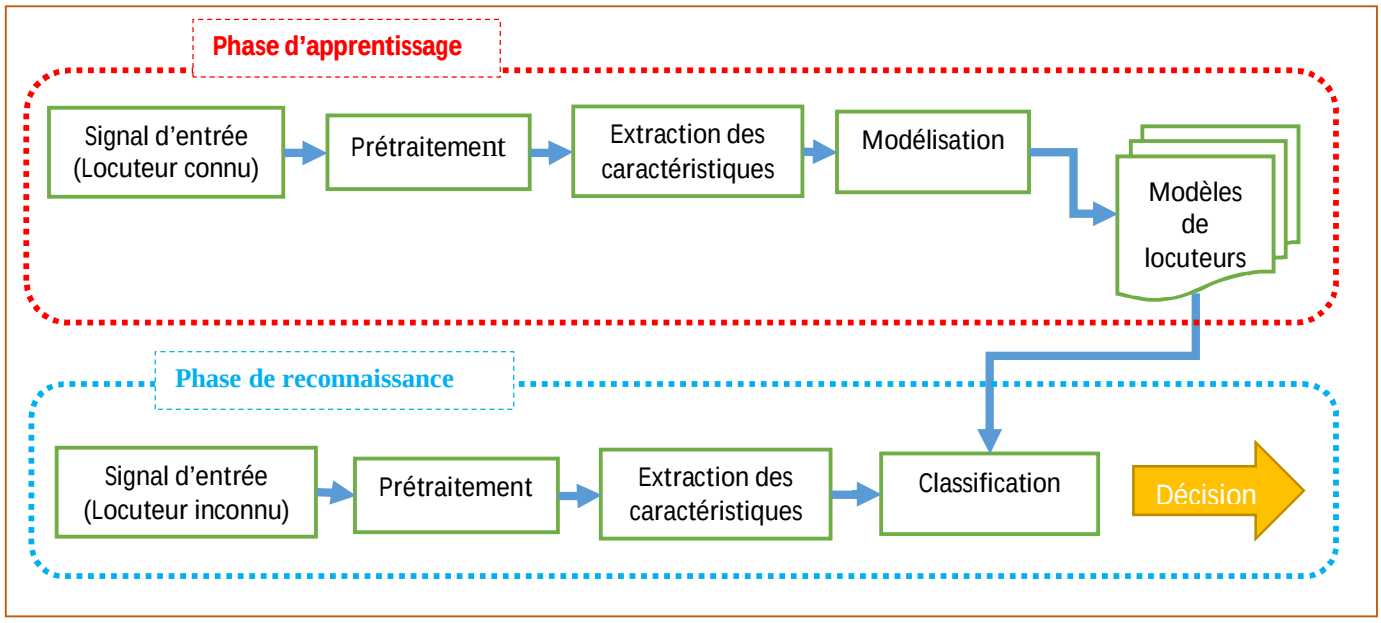



*Figure 6: Architecture du système de reconnaissance de locuteur*

Dans la **phase d'apprentissage**, chaque locuteur enregistré doit fournir des échantillons de son parole afin que le système puisse extraire leurs caractéristiques. Ces caractéristiques sont utilisées pour construire un modèle de référence ou une template pour chaque locuteur. La figure (7) montre le fonctionnement du cycle d'apprentissage.

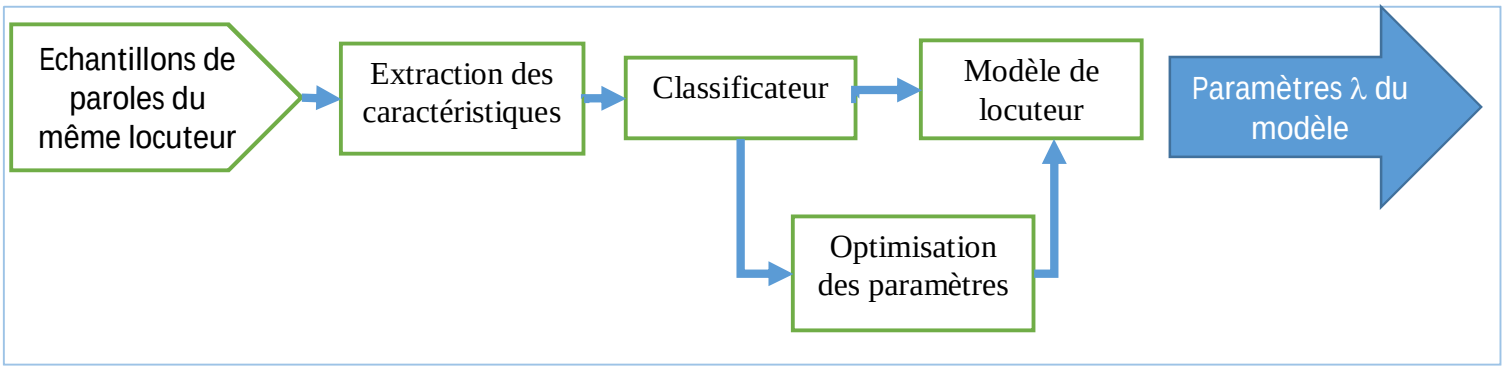



*Figure 7: fonctionnement du cycle d'apprentissage*

Dans la **phase de reconnaissance**, les caractéristiques du signal d'entrée sont calculées, est mises en correspondance avec les modèles de référence mémorisées lors de l'apprentissage. Puis, une décision de reconnaissance est faite : soit une identification de locuteur susceptible de générer ce signale, soit une vérification d'identité (Accepte/Rejet de la demande). La figure (8) montre le schéma de reconnaissance pour la tâche d'identification de locuteur, la figure (9) montre le schéma de reconnaissance pour la tâche de vérification.

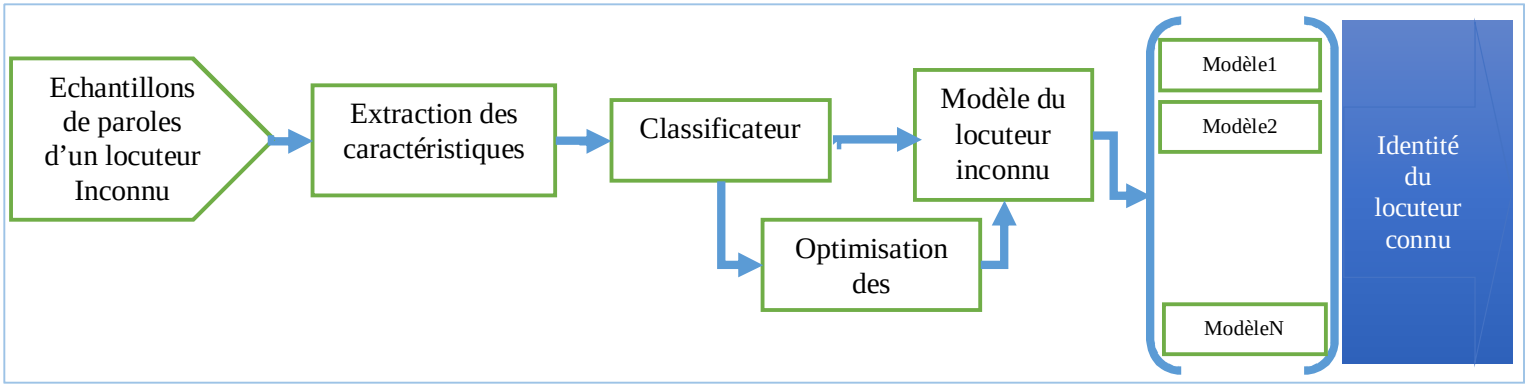



*Figure 8: Schéma de reconnaissance pour la tâche d'identification*

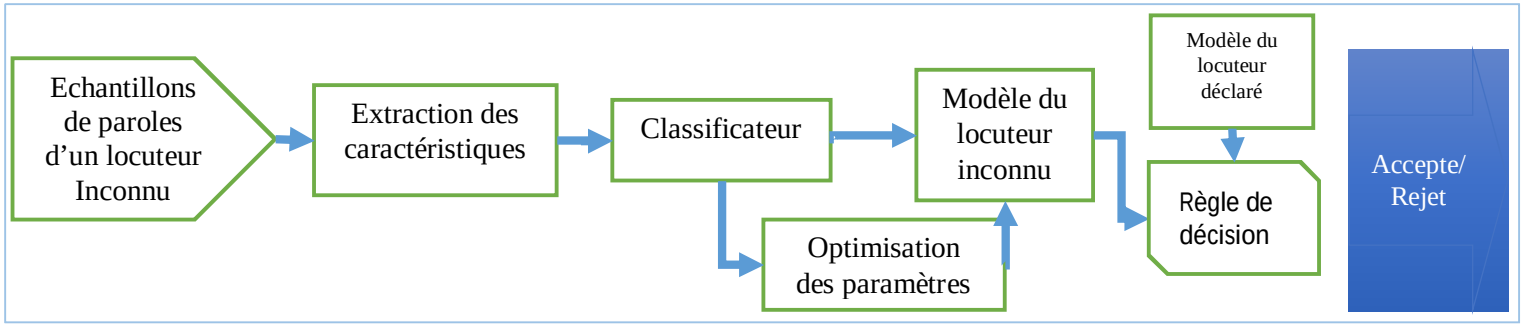



*Figure 9: Schéma de reconnaissance pour la tâche de vérification*

Pour les deux taches : identification ou vérification, on trouve les sous éléments suivants :

## 4.1. Prétraitement :

L'étape de prétraitement du signal de la parole implique:

- la conversion analogique numérique du signal.
- L'élimination du bruit : ce sont les signaux non nécessaires qui dégradent la performance du système. Ils sont dus au milieu et au dispositif d'acquisition.
- La suppression du silence, car la plupart des caractéristiques de parole sont dans la partie de la voix.

## 4.2. Extraction des caractéristiques :

La parole humaine contient de nombreuses caractéristiques qui peuvent être utilisées pour identifier les locuteurs, or sa complexité (multitudes d'informations et redondance) ne permet pas de l'exploiter directement. Une représentation simplifiée du signal de parole est par conséquent nécessaire. Cette représentation repose généralement sur des vecteurs de

paramètres acoustiques calculés périodiquement sur le signal de parole. Le but de cette étape est d'extraire une petite quantité de données à partir du signal vocal qui peut être utilisé par la suite pour représenter chaque locuteur.
Les attributs d'un extracteur des paramètres idéal comprennent [1][2]:

- Les attributs devraient être résistants contre les bruits dans l'environnement et les distorsions de canal.
- Les variations de la voix causées par l'état de santé de locuteur ne doivent pas dégrader les performances de la méthode d'extraction de caractéristiques.
- La méthode d'extraction de caractéristiques devrait maintenir une grande discrimination interlocuteur, et le plus petit possible de la variabilité intra-locuteur.
- Les caractéristiques devraient être relativement faciles à calculer.
- La méthode d'extraction de caractéristiques doit être difficile à imiter ou reproduire en utilisant la parole des imposteurs.

Difficile de vérifier toutes ces conditions en une seule méthode d'extraction des caractéristiques, parce que si certaines conditions sont améliorées les autres sont détériorés. La littérature propose un grand nombre de traitements selon la nature des informations à extraire du signal de parole. On considère généralement trois grandes classes de paramètres: les paramètres de l'analyse spectrale, les paramètres prosodiques et les paramètres dynamiques [3][4], nous détaillons par la suite les caractéristique de chacune de ces paramètres.

## 4.3. La modélisation :

L'objectif de la modélisation est de trouver un espace de représentation pour modéliser les caractéristiques extraites de chaque locuteur connu du système. Cette modélisation est réalisée à partir des données d'apprentissage collectées au cours des sessions d'enrôlement. Une mesure de similarité est ensuite calculée entre un modèle client et un signal de parole, puis transmis au processus de décision.

On distingue quartes grandes approches pour la construction des modèles clients :

- L'approche vectorielle : le locuteur est représenté par un ensemble de vecteurs de paramètres dans l'espace acoustique. Ses principales techniques sont la reconnaissance à base de DTW et par quantification vectorielle.
- L'approche statistique : consiste à représenter chaque locuteur par une densité de probabilité dans l'espace des paramètres acoustiques. Elle couvre les techniques de modélisation par les modèles de Markov cachés, par les mélanges de gaussiennes et par des mesures statistiques du second ordre.
- L'approche connexionniste : consiste principalement à modéliser les locuteurs par des réseaux de neurones.
- L'approche relative : il s'agit de modéliser un locuteur relativement par rapport à d'autres locuteurs de référence dont les modèles sont bien appris.

Nous détaillons en chapitre 4 les fondements de chacune de ces approches, les techniques qui leur sont associées ainsi que les mesures de similarité utilisées.

## 4.4. La mise en correspondance :

La mise en correspondance implique la procédure de reconnaître le locuteur inconnu en comparant les caractéristiques extraites du signal vocal d'entrée avec celles d'un ensemble de locuteurs connus.

Soit un signal de parole inconnue $S_x$ ,et un ensemble de N locuteurs. Chaque locuteur $S_i$ , $i$ = 1, … , $N$ correspond à un modèle $\lambda_i$ , i = 1, … , N . Soit $P_i$ une mesure de similarité entre $S_x$ et les locuteurs $S_i$. Dans la tâche d'identification, on cherche le locuteur qui a le modèle le plus proche parmi les locuteurs de base. C.-à-d. celui qui maximise la mesure $P_i$. Le résultat donc est l'identité de locuteur le plus susceptible d'être $S_x$. Pour la tâche de vérification, on cherche à vérifier la demande d'authentification, si la mesure $P_i$ est supérieure à un seuil la demande est acceptée sinon la demande est rejetée;

## 4.5. Évaluation :

Les mesures de performance prennent généralement la forme de taux :

- Les performances du système d'identification sont données en termes de taux d'identification correcte $I_c$ ou incorrecte $I_i$ :

$$I_c = \frac{Nombre\ de\ tests\ correctement\ identifiés}{Nombre\ total\ de\ tentatives} \qquad (1.1)$$

Et

$$I_i = \frac{Nombre\ de\ tests\ mal\ identifiés}{Nombre\ total\ de\ tentatives} \qquad (1.2)$$

Avec $I_c + I_i = 100\%$

- Les performances de vérification de locuteur sont données en termes des faux rejets$f_r$, et de fausses acceptations $f_a$.

$$f_r = \frac{Nombre\ de\ tentatives\ d'abonnés\ rejetées}{Nombre\ total\ de\ tentatives\ d'abonnés} \qquad (1.3)$$

$$f_a = \frac{Nombre\ de\ tentatives\ d'abonnés\ acceptés}{Nombre\ total\ de\ tentatives\ d'imposteurs} \qquad (1.4)$$

# 5. Difficultés

Le signal de parole est un signal très complexe où se mêlent informations linguistiques, informations caractéristiques du locuteur, informations relatives au matériel utilisé pour la transmission ou l'enregistrement du signal, etc. En outre, le signal de parole est très redondant. Cette caractéristique est d'ailleurs reconnue pour faciliter la communication entre deux personnes dans un environnement très bruyant. Par ces différents aspects, le signal de parole présente une très grande variabilité.

La capacité des systèmes de RAL à différencier plusieurs individus repose essentiellement sur la variabilité interlocuteur i.e. la disposition du signal de parole à varier entre différents individus. Néanmoins, le signal de parole renferme d'autres types de variabilité qui rendent problématique la tache de reconnaissance, telles que la variabilité interlocuteur ou la variabilité due au matériel. Par ailleurs, les systèmes de RAL doivent faire face à d'autres difficultés liées davantage au domaine applicatif, comme l'utilisation des systèmes dans des conditions difficiles, les tentatives d'imposture, etc.

## 5.1. Variabilité due au locuteur :

Si le signal de parole est variable entre deux individus, il varie également pour un même individu. Cette variabilité intra-locuteur est induite par l'évolution naturelle ou volontaire de la voix d'une personne. Cette évolution peut être :

- Ponctuelle ou à très coute terme. L'état pathologique (fatigue, rhume, etc.) ou émotionnel (stress) d'une personne provoque des altérations momentanées dans sa voix. Dans ce sens, la voix d'une personne peut évoluer entre le début et la fin de la journée (fatigue, irritation due à la pollution). D'autre part, il est impossible pour un individu de répéter consécutivement deux phrases identiques et de produire un même signal de parole pour ces deux phrases. Une légère variation est toujours observée. Finalement, une personne a la possibilité de modifier volontairement sa voix.
- A moyen terme : En RAL, le comportement d'un individu se modifie au fur et à mesure de son utilisation du système. L'individu devient de plus en plus confiant et sa voix évolue dans ce sens.
- A long terme. La voix change au fur et à mesure du vieillissement d'une personne.

La variabilité intra-locuteur pose le problème de la représentativité des signaux de parole collectés lors des sessions d'enrôlement (et des modèles des locuteurs correspondants) au sein des systèmes de RAL. Des travaux ont montré que les performances d'un système sont très fortement corrélées au temps qui sépare les sessions d'enrôlement et les tests [5]. Plus ce temps augmente, plus les performances se dégradent. Néanmoins, même les variations à court terme (émotion, état pathologique) peuvent être très préjudiciables aux systèmes de RAL.

## 5.2. Variabilité due au matériel :

Le signal de parole est porteur d'informations caractérisant le matériel utilisé lors de sa capture (ex : microphone, combine téléphonique), de sa transmission (ex : lignes téléphoniques, air ambiant) et de son enregistrement (ex : microphones, convertisseurs). Ces informations apparaissent le plus souvent sous la forme de déformations/dégradations du signal de parole. Ces déformations sont différentes selon le type de matériel utilisé. Si la bande téléphonique est reconnue pour dégrader les performances des systèmes de RAL, elle n'est pas la seule responsable. En effet, de nombreux travaux expérimentaux ont montré que des variations de matériel entre les phases d'apprentissage et de test sont à l'origine de graves dégradations des performances [6]. Par exemple, dans [7] , il est démontré que des différences de types de combines téléphoniques entre l'apprentissage et le test sont une des causes de ces dégradations.

## 5.3. Tentatives d'imposture - locuteurs non coopératifs

Selon l'application visée, un système de RAL peut faire l'objet d'attaques d'individus usurpant l'identité de quelqu'un d'autre. Ces attaques (ou tentatives d'imposture) peuvent, par exemple, avoir pour dessein des transactions frauduleuses sur le compte bancaire d'un client ou accès à des données confidentielles. Un système de RAL doit par conséquent être robuste face à de telles attaques.

Dans un contexte judiciaire, le système de RAL peut être soumis à des locuteurs non-coopératifs i.e. des locuteurs qui ne désirent pas être reconnus par le système. Dans ce cas les locuteurs tentent fréquemment de transformer leur voix.

# 6. Conclusion

Dans ce chapitre, nous avons présenté l'architecture globale d'un système de reconnaissance automatique de locuteur, ses différentes modes de fonctionnement, et les étapes de construction d'un système de RAL. Pour ce qui suit, nous détaillons chacune de ces étapes. Nous avons aussi vu certains problèmes rencontrés qui présentent un défi à la technologie de RAL. Trouver une méthode de RAL simple et performante reste un problème ouvert.

# Chapitre II

# Extraction des caractéristiques

# 1. Introduction

Le signal de la parole contient des informations concernant les mots prononcés, ainsi que l'identité de locuteur, telles que son dialecte, son âge, son état émotionnel, etc. Lors de la production de parole, une grande quantité des données est générée. La tâche de garder que les informations nécessaires parmi cette grande quantité de données, par exemple garder ceux qui concernent le locuteur pour le reconnaître, est appelée : extraction des caractéristiques ou extraction des paramètres. Pour extraire les caractéristiques d'un signal, ce dernier passe par plusieurs transformations, nous présentons dans ce chapitre en détail ces transformations.

# 2. Extraction des caractéristiques

L'extraction des paramètres d'un signal de parole, est la transformation de ce signal en une représentation compacte, mais efficace, qui est plus stable et plus discriminante que le signal original. En d'autres termes, c'est un processus de réduction de dimension qui tente de capturer l'essentiel des caractéristiques du signal analysé avec peu de données.
Étant la première étape de la chaîne de reconnaissance, les performances des prochaines étapes (modélisation, apprentissage et reconnaissance) sont fortement déterminées par la qualité de l'étape d'extraction des paramètres. Donc ces caractéristiques doivent être robustes par rapport aux conditions d'enregistrement et fournir le plus de renseignements possibles concernant l'identité du locuteur.
Pour la tâche de reconnaissance de locuteur, les caractéristiques idéales doivent présenter les fonctionnalités suivantes :

- Stable dans le temps
- Ne devrait pas être sensible à la mimique
- Faciles à mesurer
- Discrimination entre le locuteur
- Peu de variabilité intra locuteur (santé, l'émotion, le temps ...)
- Résistant contre les bruits dus aux conditions d'enregistrement.
- Devrait se produire fréquemment et naturellement dans la parole

Dans la pratique, il est très difficile de respecter toutes ces propriétés simultanément, c'est la raison pour laquelle il n'existe pas une seule meilleure méthode d'extraction qui marche partout, mais le choix et un compromis entre la discrimination des locuteurs, la robustesse et la faisabilité.

## 2.1. Les méthodes d'extraction caractéristiques

La littérature propose un grand nombre de paramétrisations selon la nature des informations à extraire du signal de parole. Ils sont classifiés en trois grandes catégories : les paramètres de l'analyse spectrale, les paramètres prosodiques et les paramètres dynamiques. Néanmoins, d'autres classifications sont envisageables.

### 2.1.1. Paramètres de l'analyse spectrale :

L'analyse spectrale est l'analyse la plus employée en RAL. Les paramètres qui en découlent sont généralement représentatifs des caractéristiques physiques de l'appareil

phonatoire (forme du conduit vocal) de chaque individu. De multiples paramètres ont été étudiés dans la littérature, nous citons ici les plus pertinents en RAL :

- Coefficients issus d'une analyse par prédiction linéaire : LPCC ou LPC.
- Coefficients spectraux issus d'une analyse en banc de filtres 3 : LFSC ou MFSC;
- Coefficients cepstraux issus d'une analyse en banc de filtres : LFCC ou MFCC.

### 2.1.2. Paramètres prosodiques :

Les paramètres prosodiques illustrent en grande partie le style d'élocution d'un locuteur : vitesse élocution (débit), durée et fréquence des pauses ainsi que les caractéristiques de la source glottale (fréquence fondamentale, énergie, taux de voisement,...). Néanmoins, ces paramètres caractéristiques du locuteur, notamment la fréquence fondamentale et ses variations, ne sont pas suffisamment discriminants pour être utilisés seuls dans un système de RAL. Ils sont généralement associés aux paramètres de l'analyse spectrale pour améliorer les performances des systèmes de RAL.

### 2.1.3. Paramètres dynamiques :

Le vecteur de paramètres issus des paramétrisations précédentes peut être complété par le vecteur correspondant aux dérivées du premier et second ordre de ces paramètres. Ces dérivées sont calculées à partir de plusieurs trames adjacentes. Elles permettent d'introduire une information concernant le contexte temporel d'une trame courante**.**

## 2.2. Le choix de la méthode :

Les caractéristiques spectrales à court terme sont les plus simples et les plus discriminants, faciles à calculer, et donnent de bonnes performances, ils sont les plus couramment utilisés dans la reconnaissance du locuteur. L'état de l'art des systèmes de reconnaissance du locuteur combine souvent ces caractéristiques, en essayant d'obtenir des résultats de reconnaissance plus précis.
Les caractéristiques prosodiques et les caractéristiques de haut niveau sont censées être plus robustes, mais moins discriminantes et faciles à imiter; par exemple, il est relativement bien connu que les imitateurs professionnels ont tendance à modifier le contour global du pitch du locuteur imité. Les caractéristiques de haut niveau ont également besoin d'un front-end plus complexe, nécessitant par exemple un système de reconnaissance de la parole.
Dans cette étude nous utilisons des paramètres issus de l'analyse spectrale qui sont les coefficients spectraux MFCC.

# 3. Caractéristiques MFCC :

Les paramètres MFCC (Mel-Fréquence Cepstral Coefficients), également appelés couramment coefficients cepstraux, sont les caractéristiques les plus utilisées en reconnaissance de locuteur. Ils sont basés sur les perceptions de l'ouïe humaine qui ne peut pas percevoir des fréquences plus 1Khz.
Les MFCC ont la propriété extrêmement intéressante de transformer un produit de convolution en une somme. Ainsi, si on considère que la voix est le produit de convolution entre une source harmonique (les cordes vocales) et un filtre (le conduit vocal), ils permettent de séparer la source du conduit.

## 3.1. Etapes d'extraction des paramètres MFCC

Les étapes d'extraction des paramètres acoustiques MFCC sont les suivantes : L'entrée de la parole est généralement enregistrée à une fréquence d'échantillonnage ci-dessus de 12500 Hz. Cette fréquence d'échantillonnage a été choisie pour minimiser les effets de crénelage dans la conversion analogique-numérique.

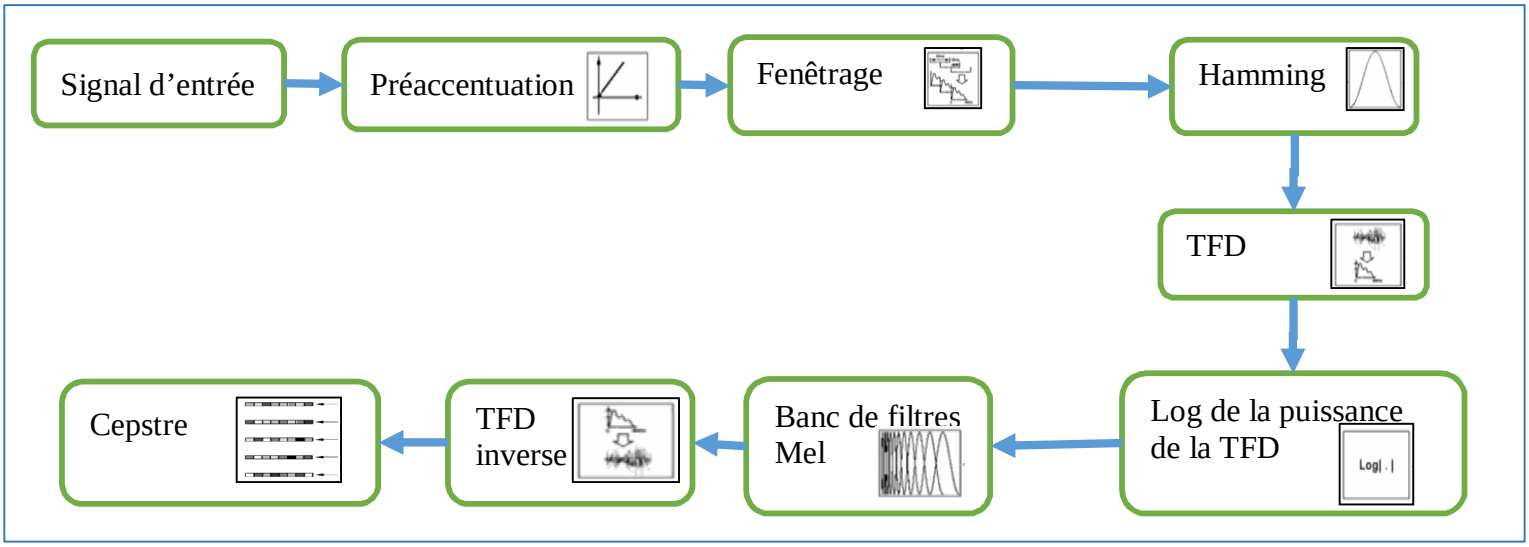



*Figure 10: Etapes d'extraction des paramètres MFCC*

### 3.1.1. Prétraitement :

- ***Isolation parole/non parole :***

Chaque enregistrement n'est pas uniquement composé de parole, en effet, des zones de silence et de bruit sont également présentes et peuvent affecter les résultats du système de RAL. Afin améliorer les performances du système il est nécessaire d'isoler les zones de parole : garder les zones de parole et éliminer le silence et les bruits.
L'élimination des zones de silence qui existent dans un signal de parole est une tâche très importante. Cette tâche semble relativement triviale, mais elle présente quelques difficultés dans la pratique. Les mesures les plus utilisées pour trouver et éliminer le silence sont : l'énergie du signal, la puissance du signal et le rapport de passage par zéro.

- ***Préaccentuation :***

La préaccentuation du signal consiste à faire ressortir les hautes fréquences avec un filtre passe-haut. Ce processus permettra d'accroître l'énergie du signal à plus haute fréquence, c'est une étape nécessaire parce que les composants de haute fréquence du signal de parole ont une faible amplitude par rapport aux composants de basse fréquence. La représentation temps-fréquence d'un signal de parole est considérée comme spectrogramme.
Le signal de parole s (n) est envoyé à un filtre passe-haut:

$$Y[n] = X[n] - a * X[n-1] \qquad (2.1)$$

Où $Y[n]$ désigne le signal de sortie, et la valeur de $a$ est habituellement compris entre 0,9 et 1,0.
La transformée en z du filtre est :

$$H(z) = 1 - 0.9z^{-1} \qquad (2.2)$$

- ***Fenêtrage:***

Dans cette étape, le signal continu de parole est découpé en fenêtres ou trames de N échantillons, avec chevauchement de M (M<N). La première trame est constituée de N premiers échantillons. La deuxième trame commence à M échantillons après la première, avec chevauchement de N-M échantillons. De même, la troisième trame commence par 2M échantillons après la première trame (ou M échantillons après la deuxième trame) et chevauche par N-2M échantillons. Ce processus se poursuit jusqu'à ce que tout le signal de la parole est représenté à l'intérieur d'une ou plusieurs trames.

Le fenêtrage du signal de parole est effectué parce que quand on examine le signal sur une période de temps assez court (entre 5 et 100 ms), ses caractéristiques sont assez stationnaires. Cependant, sur les longues périodes de temps (de l'ordre de 1/5 secondes ou plus) le signal change ses caractéristiques. Des Cadres de chevauchent sont prises pour ne pas avoir beaucoup de pertes d'informations et de maintenir la corrélation entre les trames adjacentes.

Habituellement, la taille de fenêtre (en termes de points d'échantillonnage) est égale à la puissance de deux afin de faciliter l'utilisation de la FFT.
En pratique N=256 (ce qui équivaut à un fenêtrage de ~30ms et facilite la transformer de fourrier), et M en général est prise comme 1/3 ~ 1/2 de la taille de la fenêtre.

Par exemple pour N=128, nous avons une haute résolution de temps, en outre chaque trame dure pendant une très courte période de temps. Ce résultat montre que le signal d'une trame ne change pas sa nature ? D'autre part, il n'y a que 65 (=N/2) échantillons de fréquences distinctes, cela signifie que nous avons une résolution de fréquence pauvre.
Pour N=512 nous avons une excellente résolution de fréquence (256 valeurs différentes) mais il y a des trames très petites, ce qui implique que la résolution de temps est fortement réduite. Il apparut que la valeur 256 pour N est un compromis acceptable. En outre, le nombre de trames est relativement faible, ce qui permettra de réduire le temps de calcul.

Exemple : Si le taux d'échantillonnage est de 16 kHz et la taille de la fenêtre est de 320 points d'échantillonnage, puis la durée de trame est 320/16000 = 0,02 sec = 20 ms. Aussi, si le chevauchement est de 160 points, alors le taux de trame est 16000 / (320-160) = 100 fenêtres par seconde.

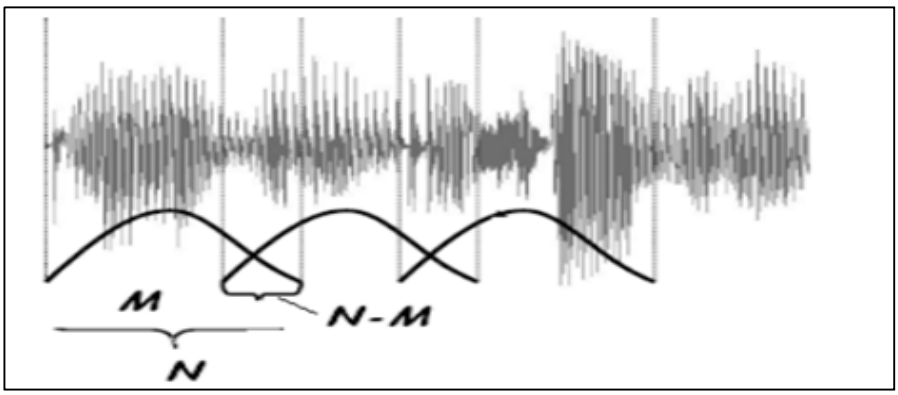



*Figure 11: Fenêtrage d'un signal par fenêtre de N échantillons avec M chevauchement*

### 3.1.2. Hammingr(windowing) :

Application d'une fenêtre de Hamming est utilisée sur ces trames pour lisser le signal ce qui minimise la distorsion spectrale avant de calculer la TFD. Chaque trame doit être

multipliée par une fenêtre de Hamming afin de maintenir la continuité de la première et les derniers points dans la fenêtre.

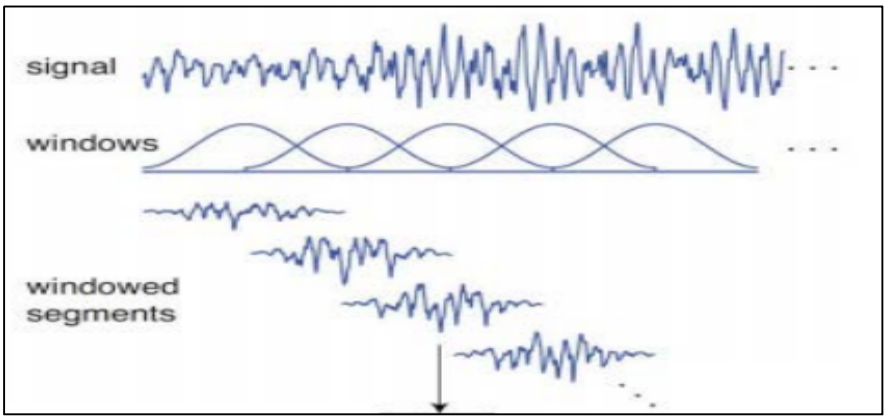



*Figure 12: multiplication d'un signal par une fenêtre de Hamming*

Si le signal dans une trame est indiqué par $(n), n = 0, \dots, N-1$ , alors le signal après Hamming fenêtrage est $s(n) * w(n)$, où $w(n)$ est la fenêtre de Hamming défini par:

$$w(n,\alpha) = (1-\alpha) - \alpha\cos\left(\frac{2\pi n}{N-1}\right) \quad ,0 \leqq n \leqq N-1 \qquad (2.3)$$

Différentes valeurs de $\alpha$ correspond à différentes courbes pour les fenêtres de Hamming comme montre la figure 13:

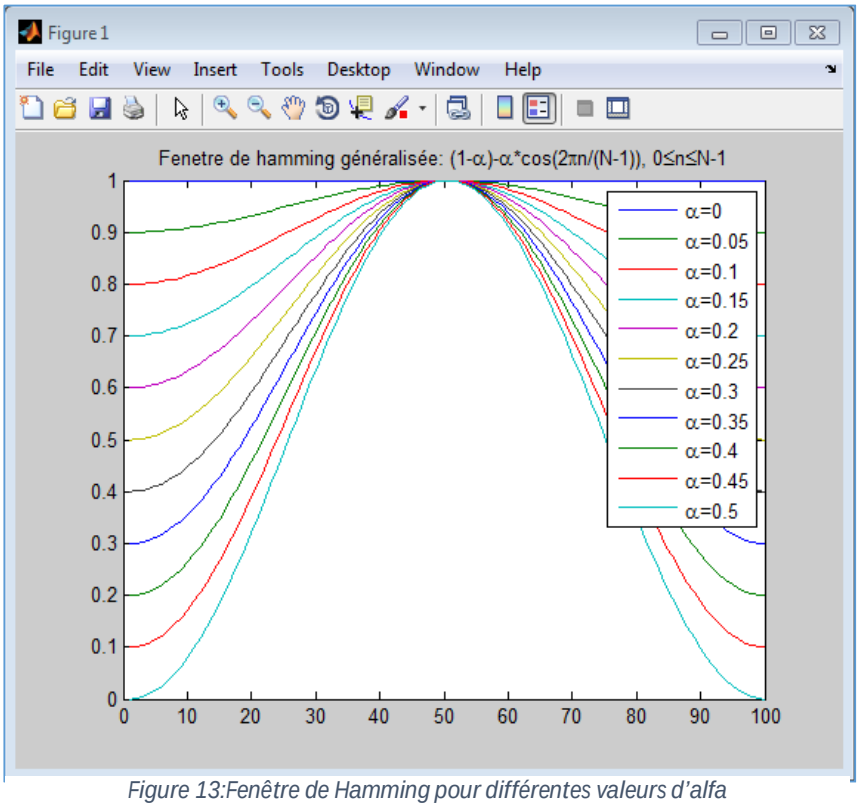



*Figure 13:Fenêtre de Hamming pour différentes valeurs d'alfa*

### 3.1.3. TFD :

L'étape de la transformée de Fourier rapide, permet de convertir chaque trame de N échantillons du domaine temporel en domaine fréquentiel. Par application d'une transformée de Fourrier sur chacune des trames, on obtient le spectre. L'analyse spectrale montre que différents timbres en signaux de parole correspondent à la distribution de l'énergie sur des fréquences différentes. Par conséquent, nous effectuons habituellement FFT, pour obtenir le spectre, qui est la réponse en fréquence de magnitude de chaque trame.

La transformée de Fourier est un algorithme rapide pour appliquer la transformée de Fourier discrète (DFT). Par définition de FFT et DFT c'est la même chose, ce qui signifie que la sortie pour la transformation sera la même, mais ils diffèrent dans leur complexité algorithmique.

### 3.1.4. Banc de filtre Mel :

La gamme des fréquences dans le spectre de TFD est très large et le signal de la voix ne suit pas une échelle linéaire. Chaque sortie du filtre est la somme de ses composantes spectrales filtrées. Cette étape se charge de la création du banc de filtres, il s'agit de plusieurs filtres triangulaires qui vont chacun couvrir une fréquence, ils permettent de mieux simuler le fonctionnement de l'oreille humaine.

### 3.1.5. TFD inverse ou DCT:

L'inverse du transformé est utilisé pour assurer un retour au domaine temporel.
Pour effectuer cette transformation soit DFT soit DCT les deux peuvent être utilisés pour le calcul des coefficients du log d'un spectre donné, comme ils divisent une séquence de données de longueur finie en un vecteur discret.
En général c'est la TFD qui est utilisée, la sortie après l'application de DCT est connue comme MFCC (Mel Frequency Cepstre Coefficient ) :

$$C_n = \sum_{k-1}^{k} (\log D_k)\cos[m(k-1/2)\pi/k] \qquad (2.4)$$

Ou : $m = 0,1,\dots,k-1$

$C_n$ Représente l'MFCC, et $m$ le nombre de coefficients.

## 4. Conclusion :

En appliquant la procédure décrite ci-dessus, pour chaque trame de parole avec chevauchement, un ensemble des MFCC sont calculées. ceux-ci sont le résultat d'un ensemble de transformations du signal. Cet ensemble de coefficients est appelé un vecteur acoustique. Par conséquent, chaque signal d'entrée est transformé en une séquence de vecteurs acoustiques. Dans le chapitre suivant, nous verrons comment ces vecteurs acoustiques peuvent être utilisés pour représenter et reconnaître la caractéristique de la voix de locuteur.

# Chapitre III

# Modélisation des paramètres

# 1. Introduction

Après l'étape d'extraction des paramètres, nous devons comparer les caractéristiques d'un locuteur inconnu avec les caractéristiques des modèles. Pour arriver à cet objectif nous devons trouver un espace de représentation adéquat qui permet mieux de visualiser la différence et la similarité entre ces caractéristiques.

Au cours des dernières années, les modèles de mélange gaussien (MMG) sont devenus l'approche dominante pour la modélisation dans les applications de reconnaissance du locuteur indépendante du texte. Ainsi l'utilisation de ce type de modèle semble être bien prometteuse. L'utilisation des mélanges gaussiens pour la modélisation des locuteurs est motivée par l'interprétation que les composantes gaussiennes représentent certaines caractéristiques des locuteurs. Lorsque vecteurs caractéristiques sont d-dimensionnel après le regroupement, ils ressemblent en quelque sorte à des distributions gaussiennes. Cela signifie que chaque cluster peut être considéré comme une distribution de probabilité gaussienne, et les caractéristiques appartenant aux groupes peuvent représentées par leurs valeurs de probabilité. La seule difficulté réside dans la classification efficace des vecteurs de caractéristiques.

Dans ce chapitre nous commençons par une description des méthodes de modélisation existent dans la littérature, puis nous présentons en détails le modèle des mélanges gaussiens finis, par la suite nous exposons l'algorithme EM utilisé pour résoudre ce type de problème statistique, puis nous voyons comment s'effectue la classification à l'aide de cette technique.

# 2. Les approches de modélisation

On distingue quatre grandes approches pour la construction des modèles clients : les approches vectorielles, statistiques, prédictives et connexionnistes.

## 2.1. L'approche vectorielle :

Le locuteur est présenté par un ensemble de vecteurs de paramètres dans l'espace acoustique. Lors de la reconnaissance, une distance entre cet ensemble de vecteurs et les vecteurs de paramètres issus des signaux de test sont calculés. Ses principales techniques sont la reconnaissance à base de la programmation dynamique(DTW) et par quantification vectorielle.

### 2.1.1. Déformation temporelle dynamique :

La déformation temporelle dynamique (Dynamic Time Warping : DTW) consiste à aligner temporellement une séquence de vecteurs de paramètres de test avec une séquence de vecteurs d'apprentissage. Dans ce cas, le modèle de locuteur est tout simplement l'ensemble des vecteurs de paramètres obtenus après paramétrisation des signaux d'apprentissage. Une distance est calculée entre vecteurs d'apprentissage et de test et moyennée sur l'ensemble de la séquence.

De par son principe, la déformation temporelle dynamique est utilisée exclusivement en mode dépendant du texte, elle est très rapide et montrant des performances relativement

bonnes, mais elle est très sensible à la qualité d'alignement et notamment au choix du point de départ.

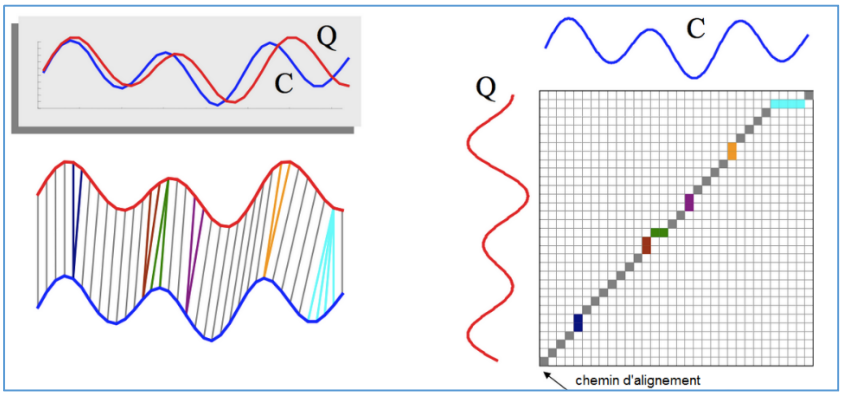



*Figure 14:Chemin d'alignement entre deux signaux*

### 2.1.2. Quantification vectorielle :

La quantification vectorielle (Vector Quantisation : VQ) repose sur un partitionnement de l'espace acoustique en sous-espaces. Chaque sous-espace est associé à leur vecteur centroïde (i.e. à un vecteur de paramètres représentant l'ensemble des vecteurs composant le sous-espace). Dans ces conditions, un modèle de locuteur est composé d'un ensemble de vecteurs centroïdes, et est appelé dictionnaire de quantification (codebook).

Lors de la phase de reconnaissance, une distance est calculée entre un vecteur de test et chaque vecteur centroïde du dictionnaire. La distance minimale est assignée au vecteur de test. La distance d'une séquence de vecteurs de test est obtenue par moyenne des distances minimales attribuées à chacun des vecteurs de test.

La quantification vectorielle s'applique en mode dépendant ou indépendant du texte.
La rapidité et les performances de cette technique dépendent fortement de la taille du dictionnaire : plus la taille du dictionnaire augmente, meilleures sont les performances ; néanmoins, le processus devient d'autant plus lent. Le codage par quantification vectorielle est simple, or la conception d'un dictionnaire est plus compliquée et sensible l'initialisation.

## 2.2. L'approche statistique :

Consiste à représenter chaque locuteur par une densité de probabilité dans l'espace des paramètres acoustiques. Elle couvre les techniques de modélisation par les modèles de Markov cachés, par les mélanges gaussiens.

### 2.2.1. Modèles de Markov cachées :

Les modèles de Markov (ou HMM pour Hidden Markov Models) ont été initialement introduits en reconnaissance de la parole. Puis leur utilisation s'est étendue peu à peu au domaine de la reconnaissance du locuteur.
Dans cette approche, il ne s'agit plus d'une mesure de distance d'une forme acoustique à une référence, mais de la probabilité que la forme acoustique ait été engendrée par le modèle de référence du locuteur. Le modèle d'un locuteur est constitué de l'association d'une chaîne de Markov, une succession d'états avec des probabilités de transition d'un état à l'autre, et de lois de probabilités (probabilités d'observation d'un vecteur acoustique dans un état).

### 2.2.2. Modèles de mélanges :

La reconnaissance du locuteur par mélanges de gaussiennes consiste à modéliser un locuteur par une somme pondérée de composantes gaussiennes. Ainsi une large gamme de distributions peut être parfaitement représentée. Chaque composante des gaussiennes est supposée modéliser un ensemble de classes acoustiques. L'utilisation de ce type de modèle semble être bien prometteuse. Il semble bien modéliser les caractéristiques spectrales des voix des locuteurs, et il est relativement simple à mettre en œuvre.

Les mélanges de gaussiennes sont considérés comme un cas particulier des HMM et une extension de la quantification vectorielle. Nous allons aborder cette méthode avec plus de détails par la suite.

## 2.3. L'approche connexionniste :

Les réseaux de neurones ont été assez largement utilisés en reconnaissance du locuteur. Ils offrent en effet une bonne alternative au problème de la discrimination entre les locuteurs. Ces outils de classification permettent 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). On peut aussi utiliser les réseaux de neurones en les couplant à d'autres techniques, par exemple les modèles de Markov cachés. On parle alors de méthodes hybrides.

## 2.4. L'approche relative :

Il s'agit de modéliser un locuteur relativement par rapport à d'autres locuteurs de références dont les modèles sont bien appris.

# 3. Les mélanges gaussiens :

Le MMG utilise un mélange de plusieurs gaussiennes pour modéliser la densité de probabilité des variables observées (figure 15). On s'intéresse à N individus pour lesquels on dispose d'observations pour une variable $x$ de D-dimension qui sont notées $x_1, \ldots, x_n$. Ces individus sont issus de K populations. Dans un premier temps, supposons connu ce nombre de populations. Le modèle de mélange gaussien s'écrit comme une somme pondérée des densités gaussiennes :

$$P(x|\lambda) = \sum_{i=1}^{K} \pi_i p(x|\theta_i) \quad \forall x \in R^D \qquad (3.1)$$

Avec : $\pi_i \in [0,1] \; \forall \; i = 1,2,\ldots,K$ sont les coefficients de mélange vérifient $\sum_{i=1}^{K} \pi_i = 1$ elles représentent la probabilité qu'un $x_k$ sélectionné d'une façon aléatoire est généré par la $i^{ème}$ gaussienne.

Et $p(x|\theta_i)$ , $i = 1,\ldots,K$ sont les densités gaussiennes, Avec $\theta_i = (u, \Sigma)$ désigne les paramètres de la $i^{ème}$ densité : $u$ c'est le vecteur de moyenne, $\Sigma$ est la matrice de covariance.

Chaque composante est de la forme:

$$p(x|\theta_i) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_i|^{\frac{1}{2}}} exp\left\{-\frac{1}{2}(x-u_i)'\Sigma_i^{-1}(x-u_i)\right\} \qquad (3.2)$$

Le MMG est noté: $\lambda = \{\pi_i, u_i, \Sigma_i\} \quad i = 1, \dots, K$.

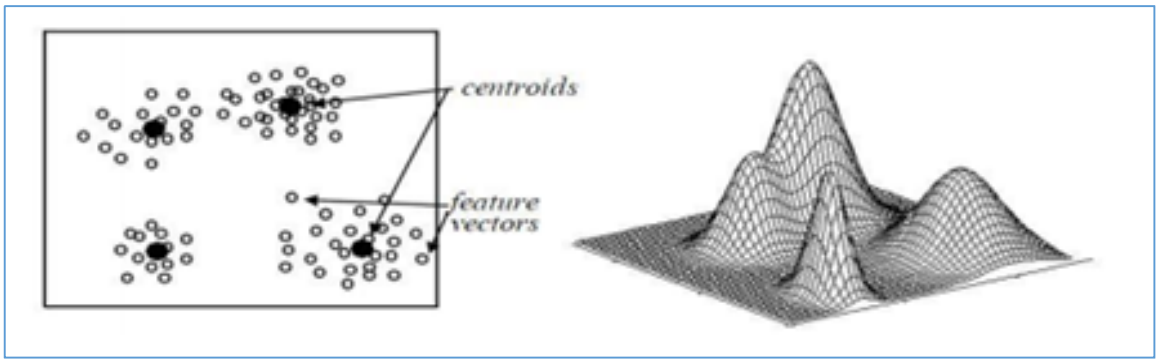



*Figure 15: Représentation d'une distribution par un mélange des gaussiennes*

Les matrices pleines et les matrices diagonales sont les deux formes de matrice de covariance les plus utilisées dans la modélisation avec les MMG :

- Le modèle de mélange avec une matrice de covariance pleine théoriquement est le modèle le plus puissant, car il permet de mieux ajuster les données, l'inconvénient de ce type de matrice de covariance est qu'il nécessite un grand volume de données pour l'estimation des paramètres.
- D'autre part, en pratique, le nombre des paramètres du modèle est limité par la quantité des données d'apprentissage disponible, la matrice de covariance diagonale permet d'obtenir des performances semblables aux matrices de covariance pleines en utilisant un nombre plus élevé de mélanges de gaussien [8].

En faisant varier le nombre de K gaussiens, les poids $\pi_i$, et les paramètres μk et Σ de chaque fonction gaussienne de la densité, des mélanges gaussiens peuvent être utilisés pour décrire une fonction complexe de densité de probabilité. Nous voyons par la suite comment se fait le choix de modèle parmi ces diverses possibilités.

## 4. La vraisemblance :

Étant donnée un ensemble de données $X = (x1, \dots, x_N)^T$, générées à partir d'une distribution inconnue, on cherche à trouver les paramètres $\theta = \{\mu, \sigma, \Sigma\}$ du modèle qui le plus susceptible de générer ces données.
On suppose qu'un échantillon $X = (x1, \dots, x_N)^T$ est indépendamment et identiquement distribué (iid.) la probabilité jointe de cet échantillon est:

$$P(X|\theta) = \prod_{n=1}^{N} p(x_n|\theta) \qquad (3.3)$$

Ici $x_n = (x_1, \dots, x_n)$ désigne les valeurs observées de $X_1, \dots, X_n$.
$P(X|\theta)$ est la probabilité jointe (PDF/PMF) de l'échantillon X= $(X_1, \dots, X_N)$.
La fonction de vraisemblance est définit par :
$L(\theta|x) = L(\theta_1, \dots, \theta_k|x_1, \dots, x_n) = P(x|\theta) = \prod_{i=1}^{n} p(x_i|\theta_1, \dots, \theta_k)|$

La différence entre les fonctions $P(x|\theta)$ et $L(\theta|x)$ est la variable qui est considérée comme fixe et qui est autorisée à varier. $P(x|\theta)$ Suppose que $\theta$ a une valeur fixe tandis que $x$ varie sur toutes les valeurs possibles dans l'espace d'échantillons, et $P(\theta|x)$ traite x comme une valeur connue et $\theta$ est variable dans l'espace des paramètres θ.
Nous pouvons utilisée la fonction de vraisemblance pour déterminer parmi deux valeurs de paramètres $\theta'$ et $\theta''$ ∈ Θ données celle qui plus susceptibles de générer l'échantillon x observé. Supposons que $L(\theta'|x) > L(\theta''|x)$ , cela signifie que pour l'échantillon x, le paramètre $\theta$' est plus probable que, $\theta''$ car elle a donné une grande valeur de la fonction de vraisemblance. Par conséquent, nous devrions trouver les valeurs des paramètres qui donnent les plus grandes valeurs possibles de notre fonction de vraisemblance.
Pour des raisons analytiques, il est plus facile de travailler avec le logarithme de la vraisemblance qu'avec la vraisemblance elle-même. Étant donné que le logarithme est croissant et monotone, la valeur de $\theta$ qui maximise le logarithme de la vraisemblance maximise également la vraisemblance.
On utilise souvent une fonction auxiliaire Q définie comme suit :

$$Q(\theta, \hat{\theta}) = \sum_{Y} P(X|\hat{\theta}) * \log(P(X|\theta)) \qquad (3.4)$$

Maximiser la fonction $Q(\theta, \hat{\theta})$ est équivalent à maximiser la vraisemblance des données observées, étant donné que :
$Q(\theta, \hat{\theta}) \geq Q(\theta, \hat{\theta})$ => $\log P(X|\hat{\theta}) \geq \log P(X|\theta)$

## 5. L'algorithme Expectation Maximisation :

Expectation-Maximisation(EM) est une méthode itérative permet d'estimer les paramètres du modèle qui maximisent la vraisemblance ou le maximum a posteriori dans les modèles probabilistes et statistiques faisant intervenir des données manquantes, latentes ou partiellement observées . La simplicité et la modularité de l'algorithme EM ont fini de la populariser à travers diverses problématiques d'inférence statistique où la vraisemblance du modèle en étude est difficile, voire impossible à mettre en œuvre par les outils habituels de maximisation. L'idée est de faciliter la recherche d'optimum de la fonction objectif en complétant celle-ci des variables inobservables. En d'autres termes, on augmente l'espace des observations afin que celui-ci incorpore ces dites données manquantes tout en sachant que l'inférence ne peut se faire que sur les données effectivement observées.

L'Algorithme EM

1. **Initialisation :** Choisir une estimation initiale $\theta$.
2. **Étape estimation :** Calculer la fonction auxiliaire $Q(\theta, \hat{\theta})$ qui représente une estimation du $logP(X|\theta)$, en se basant sur les données observables.
3. **Étape Maximisation :** Calculer $\hat{\theta} = \arg\max\left(Q(\theta, \hat{\theta})\right)$ afin de maximiser la fonction auxiliaire $Q$ sur $\theta$.
4. **Itération :** Mettre $\widehat{\theta} = \theta$, répéter étape 2 et 3 jusqu'à ce qu'il y ait convergence.

L'utilisation d'EM dans les mélanges gaussiennes suppose qu'on connaît à l'avance le nombre de gaussiennes qui a généré certaines observations, mais on ne connaît pas l'appartenance (l'affectation) de chaque point, l'algorithme EM permet de trouver les paramètres des modèles qui sont plus susceptibles de générer ces observations, en maximisant la

vraisemblance, de sorte que le modèle peut décrire chaque échantillon des données et estimer les autres.

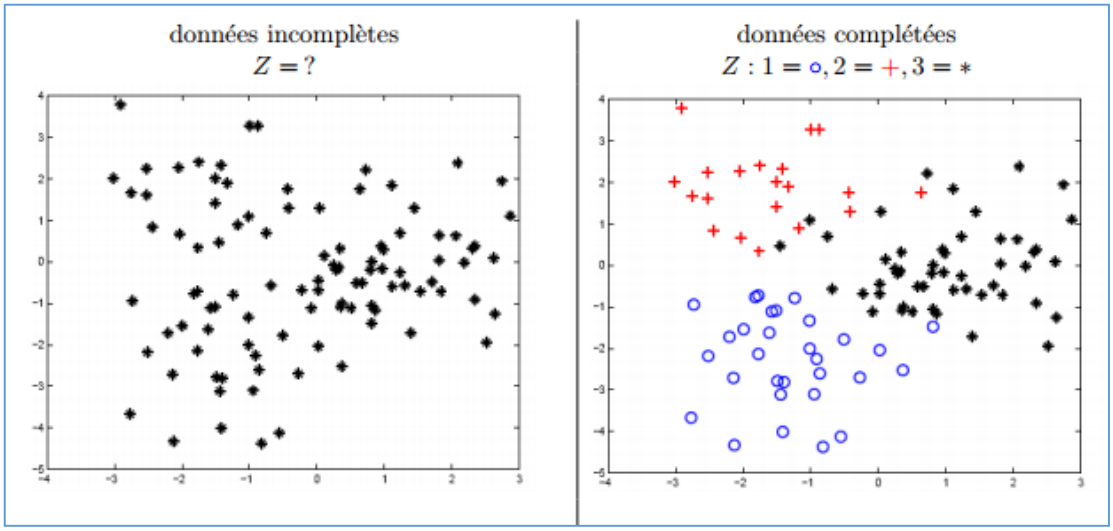



*Figure 16: Exemple de classification par l'algorithme EM*

EM itère entre en deux étapes: étape d'estimation qui calcule l'espérance du log de vraisemblance évaluée en utilisant l'estimation actuelle des paramètres, et étape de maximisation : qui calcule les paramètres en maximisant le log de vraisemblance estimée dans l'étape d'estimation, ces paramètres vont être utilisés dans la prochaine étape d'estimation. Les étapes de cet algorithme sont les suivantes :

**1-Initialisation :**
Initialise les paramètres : la moyenne u, la variance sigma, et la matrice de covariance$\sum$, puis évaluer la vraisemblance.

**2-Estimation:**
Étape d'estimation des paramètres:

$$\gamma\left(z_k^{(i)}\right) = \frac{\pi_k \mathcal{N}\left(x^{(i)} \middle| \mu_k, \Sigma_k\right)}{\sum_{i=1}^{K} \pi_i \mathcal{N}\left(x^{(i)} \middle| \mu_k, \Sigma_i\right)} \qquad (3.5)$$

**3-Maximization :**
Mettre à jour des paramètres en se basant sur le maximum de vraisemblance :

$$\mu_k = \frac{\sum_{i=1}^{n} p(x_i|\theta) x^{(i)}}{\sum_{i=1}^{n} p(x_i|\theta)} \qquad (3.6)$$

$$\Sigma_K = \frac{\sum_{i=1}^{n} p(x_i|\theta_k)\left(x^{(i)} - \mu_k\right)\left(x^{(i)} - \mu_k\right)^T}{\sum_{i=1}^{n} p(x_i|\theta)} \qquad (3.7)$$

$$\pi_k = \frac{\sum_{i=1}^{n} p\left(x_i|\theta_j\right)}{K} \qquad (3.8)$$

**4-Évaluation :**
Évalue la nouvelle valeur de la vraisemblance, et vérifier la convergence soit de paramètres ou de vraisemblance, Si elle n'est pas satisfaite, retourner à l'étape 2:

$$\ln p(X|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k N(x_n|\mu_k,\Sigma_k)\right\} \qquad (3.9)$$

A chaque itération EM garantit que le maximum de vraisemblance ne sera pas diminué.

# 6. Classification par modèles de mélanges :

La classification par les MMG ne se base pas sur des considérations géométriques, mais s'appuie sur l'analyse de la distribution de probabilité de la population. La notion d'homogénéité se traduit par le fait que les observations qui sont dans un même groupe sont issues d'une même distribution. A la fin du processus de modélisation de mélange gaussienne, toutes les données d'apprentissage peuvent être regroupées en groupes en affectant chaque objet au modèle de mélange qui est plus susceptible de générer (figure 16).

Pour le système d'identification automatique du locuteur, chaque locuteur est représenté par une somme pondérée de gaussiennes, en d'autres termes il est représenté par une matrice de moyennes μ, une matrice de variance σ, et un vecteur des poids de mélange.
Étant donné un ensemble de locuteurs $S = \{1, 2 \ldots . M\}$, représenté par MMG $\lambda_1, \lambda_1, \ldots . , \lambda_M$, l'objectif est de trouver le modèle de locuteur qui maximise la probabilité a posteriori pour une séquence de test donnée.

$$\hat{S} = \arg \max_{1 \le k \le K} p(\lambda_k)$$

# 7. Conclusion :

Dans ce chapitre nous avons vu les différentes approches dans la littérature. La modélisation par mélanges gaussiens fournit des bonnes performances et constitue l'état de l'art. L'algorithme EM permet d'approcher les paramètres de ces gaussiennes en fonction des données observées. Cet algorithme est stable numériquement et la vraisemblance croit à chaque itération, malheureusement elle présente certaines limitations a noté : EM peut converger lentement même pour les problèmes qui semblent inoffensifs. Il peut converger lentement aussi lorsqu'il y a beaucoup d'information manquante, et il n'est pas certain que l'algorithme EM convergera à un maximum global ou local lorsqu'il y a plusieurs maxima.

# Chapitre IV

# Les algorithmes d'auto-classification

# 1. Introduction

Nous avons vu dans le chapitre précédent la puissance de l'algorithme EM pour l'estimation des paramètres du modèle de mélange. Or avant l'utilisation de l'algorithme EM, il est nécessaire de déterminer deux facteurs importants pour l'apprentissage des MMG, qui sont le nombre des gaussiens K et les valeurs initiales des paramètres du modèle. Le nombre de composantes est un paramètre important pour modéliser le mélange gaussien. Avec trop de composantes, le modèle de mélange serait sur-adapter les données, d'autre part, avec peu de composants, il ne suffirait pas pour décrire la distribution des données. Aussi l'initialisation est un facteur important dans l'estimation des paramètres, une mauvaise initialisation, signifie une grande probabilité de tomber en un maximum local.

Ces deux problèmes sont communs à plusieurs autres algorithmes qui dépendent d'une façon cruciale de certains paramètres qui doivent être réglés pour chaque nouveau problème, le plus important de ces paramètres en classification non supervisée est le nombre de classes, la détermination de ce nombre est généralement une tâche difficile.
Nous allons voir dans ce chapitre les algorithmes de classification dans la littérature, aussi que certaines techniques d'auto classification, et nous présentons la méthode division-fusion puis nous détaillons les approches d'auto-classification EM basée sur cette méthode.

# 2. La classification non supervisé :

La classification non supervisée est la partie du cycle de datamining (d'analyse de données) qui vise à diviser un ensemble de données en des classes homogènes, les membres de même groupe partagent les mêmes caractéristiques communes. Elle devient un domaine qui intéresse beaucoup de chercheurs dans divers domaines tels que : la reconnaissance des formes, l'apprentissage, la bio-informatique,...

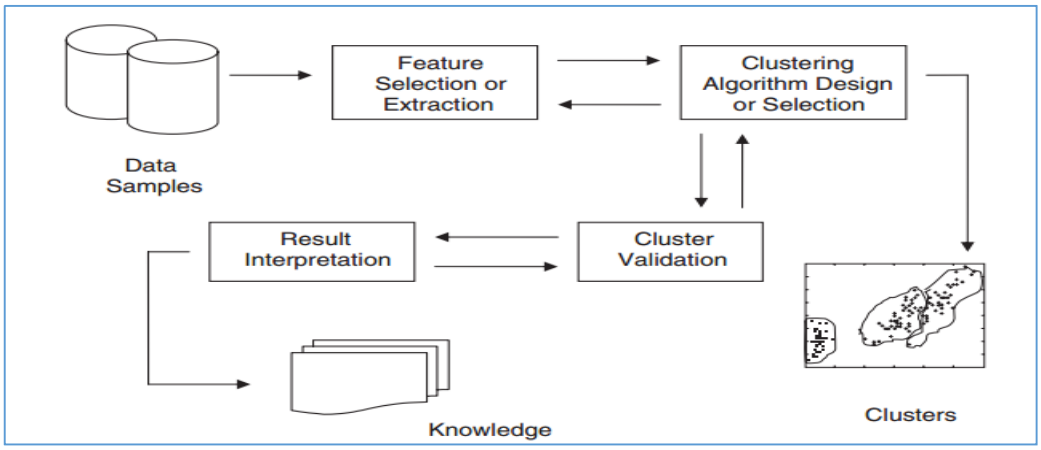



*Figure 17: Cycle du datamining*

## 2.1. Les méthodes de classification non supervisé :

On peut classer les méthodes de classification non supervisée en se basant sur leurs méthodologies utilisées pour partitionner les objets en classes, c.-à-d. selon :

- Le type de données d'entrée à l'algorithme de classification.

- Le critère de regroupement définissant la similarité entre les objets.
- Les théories et les concepts fondamentaux sur lesquels les techniques de regroupement sont basées (par exemple la théorie floue, statistique).

On trouve :

### 2.1.1. Agglomératifs vs divisives

Les méthodes agglomératifs : partent d'une seule entité, et successivement fusionnent les objets jusqu'à ce qu'un critère d'arrêt soit rencontré. Les méthodes divisives: commence avec tout la totalité des objets, et effectue le fractionnement jusqu'à ce qu'un critère d'arrêt soit satisfait.

### 2.1.2. Monothéiques vs polythétiques

Méthodes monothéiques : découpage est effectuée d'une manière individuelle. Méthodes polythétiques : découpage est effectuée globalement, en tenant compte des interactions entre les variables prédictives(les caractéristiques des objets).

### 2.1.3. Dure vs floue

Les méthodes dures (Hard) : assignent chaque objet à une et un seul class durant l'exécution et aussi en résultat. Dans les méthodes floues (Fuzzy) chaque objet a un degré d'appartenance à plusieurs classes. On peut transformer une classification floue en dure on affectant chaque objet à la classe laquelle il possède le degré d'appartenance maximale.

### 2.1.4. Déterministe vs stochastique

Cette question est plus pertinente aux approches de classification par partition conçue pour optimiser une fonction d'erreur quadratique. Cette optimisation peut être réalisée en utilisant des techniques traditionnelles ou par une recherche aléatoire de l'espace d'état composé de toutes les étiquettes possibles.

### 2.1.5. Incrémentale vs non incrémental

Ce problème se pose lorsque la base de données à regrouper est volumineuse, et que les contraintes du temps et espace d'exécution affectent l'architecture de l'algorithme. L'histoire des méthodologies de regroupement ne contient pas de nombreux exemples d'algorithmes de clustering conçus pour fonctionner avec des ensembles de données volumineux, mais l'avènement du datamining a favorisé le développement d'algorithmes de clustering qui minimisent le nombre de balayages sur l'ensemble de données, et réduit par conséquent, le nombre d'objets examinés au cours de l'exécution.

### 2.1.6. Monothétique vs polythétique

Cet aspect concerne l'utilisation séquentielle ou simultanée des caractéristiques dans le processus de regroupement. La plupart des algorithmes sont polythétiques, c'est-à-dire, toutes les caractéristiques sont impliquées dans le calcul des distances entre les exemples et les clusters. Un algorithme monothétique simple considère les caractéristiques de façon séquentielle lors de la classification de données.

Nombreuses méthodes de classification non supervisées sont utilisées dans la littérature:

**Méthodes de classification par Partitionnement:** le but est de construire k partitions, ces partitions doivent optimiser une fonction objective, exemples :

- Global optimal: prend en considérer toutes les k-partitions
- Heuristic methods: Algorithmes *k-means* et *k-medoids*
  - ✓ *k-means*: Chaque cluster est représenté par son centre
  - ✓ *k-medoids* or PAM (Partition around medoids): Chaque cluster est représenté par un de ses objets

**Méthodes de classification hiérarchique:** Créer une décomposition hiérarchique des objets selon certains critères, deux approches :

- l'approche ascendante : part de l'entité la plus petite, et effectue des fusions successives. Exemple : Clustering Using Representatives(CURE), Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), Robust Clustering using links (ROCK).
- l'approche descendante : part de l'ensemble de données, et effectue des décompositions successives. Exemple : Williames et Lambert, Tree Structured Vector Quantization (TSVQ)

**Méthodes basées sur la densité:** basées sur des notions de connectivité et de densité, l'idée est de regrouper les objets les plus proches, exemple : DBSCAN.

**Méthodes de grille:** basées sur une structure à multiniveaux de granularité, l'idée est de diviser l'espace en un nombre fini de cellules formant une grille.

**Méthodes à modèles:** Un modèle est supposé pour chaque cluster, ensuite vérifie le modèle de chaque groupe pour choisir le meilleur. Exemple : Expectation-Maximisation.

## 2.2. Méthodes d'auto-classification :

On peut résumer la classification non supervisée comme la tâche de construire une matrice $A(X)$ à partir un ensemble de données $X$ d $n$ 'objets notés : $X = \{X_1, \dots, X_n\}$ , tel que :

$$\sum_{i=1}^{n} u_{ki} \geq 1 \qquad pour\ k = 1, \dots, K \qquad (4.1)$$

$$\sum_{k=1}^{K} u_{ki} = 1 \qquad pour\ i = 1, \dots, n \qquad (4.2)$$

$$\sum_{k=1}^{K} \sum_{i=1}^{n} n \qquad (4.3)$$

La matrice A(X) de taille $K * n$ peut être représenté comme $A = [u_{ki}]$ , $k = 1, \dots, K\ et\ i = 1, \dots, n$ , avec $u_{ki}$ est le degré d'appartenance de l'objet $x_i$ au groupe $C_k$.

Dans la plupart des situations réelles, le nombre de clusters K est difficile de déterminer à l'avance, ce qui complique la tâche de classification non supervisée. Plusieurs études pour déterminer automatiquement la valeur correcte de K ont été effectué, certaines dites

algorithmes gloutons : part d'un nombre minimal de class K1, puis réunit à chaque itération les clusters qui sont proches, se basant sur une distance adéquate [9][10][11], certaines méthodes sont basées sur les algorithmes génétiques [12][13][14], une autre approche est de calculer un indice de validité pour plusieurs valeurs Ki, puis choisir le meilleur [15][16][17], d'autres approches sont basées sur l'algorithme Division-Fusion : après initialiser les clusters, réunir deux à deux les clusters en respectant un critère de fusion et autre de division jusqu'à convergence.

Nous nous intéressons par cette dernière approche qui est la division-fusion, plus précisément l'application de cette méthode à des mélanges gaussiens. L'idée générale de cet algorithme est d'effectuer un ensemble des opérations division-fusion sur l'ensemble de données jusqu'à stabilisation.

# 3. Algorithmes de division fusion des gaussiennes :

L'algorithme division-fusion est une méthode itérative introduite en 1974 par Horowitz et Pavlidis[18], comme son nom l'indique elle se réalise en deux étapes: Division et Fusion. Son objectif est de découper un ensemble de données initiales en sous-ensembles homogènes. Un indice d'homogénéité $H(x)$ est utilisé pour mesurer la similarité entre les composantes de l'ensemble de données. En général les prédicats de fusion $Hs(k)$ et de division $Hm(i,j)$ sont différents.

- L'étape de division :

On commence par un ensemble de données initialement contient $k$ sous-ensembles, $k = k \geq 1$, pour chaque sous-ensemble $k$, Si $Hs(k)$ est vérifiée pour un ensemble k, cela signifie qu'elle contient des éléments similaires alors on arrête la décomposition, sinon on la découpe en deux sous-ensembles I et J. et on examine le contenu de chaque sous-ensemble.

- L'étape de Fusion :

Cette étape a pour objectif de regrouper les ensembles qui sont similaires, l'indice d'homogénéité de fusion est évalué pour chacun des deux ensembles I et J, si cet indice est vérifié alors les deux ensembles seront regroupés en un seul ensemble.
L'algorithme division-fusion donc exécute ces deux opérations jusqu'à arriver à un critère d'arrêt, à chaque itération un indice de validation est mesuré pour vérifier l'efficacité de l'étape exécutée.

De nombreux travaux de recherches ont intéressé par l'application de l'algorithme division-fusion sur des densités gaussiennes, dont le but est de résoudre les problèmes liés à EM par diviser et fusionner les composantes incorrectes. En général, on distingue deux variantes de cet algorithme, une permet d'optimiser le résultat d'EM, et l'autre permet de trouver le nombre de composantes qui est suffisant pour mieux décrire l'ensemble de données.

## 3.1. Algorithme général de division fusion des gaussiennes :

En 2000 N.Ueda et al.[19] ont introduisent SMEM qui est la première application de l'algorithme division fusion sur les mélanges de gaussiennes après une optimisation avec

l'algorithme EM, ce qui permet d'éviter les maximums locaux et d'améliorer la vraisemblance du modèle. La figure (18) montre l'algorithme général de SMEM.
Après la convergence d'EM, la fonction Q sera écrite comme ceci :

$$Q^* = q_i^* + q_j^* + q_k^* + \sum_{m,m\neq i,j,k} q_m^* \qquad (4.4)$$

Deux groupes i et j sont sélectionnées pour être regroupés en un seul groupe i′, et le groupe k sera fractionné en deux j′ et k′. La sélection de ces groupes est effectuée à l'aide des deux critères suivants :

- Le critère de fusion de deux groupes :

$$J_{merge}(i,j;\theta^*) = P_i(\theta^*)^T P_j(\theta^*) \qquad (4.5)$$

Deux classes vont être regroupées s'ils ont une probabilité presque proche, donc deux classes ont un grand $J_{merge}$ vaut mieux les regroupées.

- Le critère utilisé pour mesurer l'homogénéité d'une classe pour la diviser est :

$$J_{split}(k;\theta^*) = \int f_k(x;\theta^*) log \frac{f_k(x;\theta^*)}{p_k(x;\theta^*)} dx \qquad (4.6)$$

C'est le Local Kullback-Leiber divergence, qui est utilisée pour mesurer la distance entre deux distributions.

Dans SMEM on calcule ces deux critères pour tous les groupes, nous obtenons deux listes, une contient les éléments avec le critère de fusion $L_m = \{i,j,J_{merge}\}$ et une autre liste contenant les éléments avec le critère de division à l'exclusion des éléments déjà sélectionnés pour la fusion $L_s = \{k,J_{split}\}$ (C.-à-d.$k$ ! = $i,j$). les éléments de chaque liste sont triés en se basant sur les critères de division et fusion, puis on sélectionne $Cmax$ éléments à modifier de chaque liste, avec $Cmax = M(M-1)(M-2)/2$, selon [19] la valeur 5 pour C est suffisante.

Pour chaque triplet $\{i,j,k\}_c$ modifié, l'initialisation après la modification est effectuée comme suivant:
L'initialisation après fusionner deux ensembles $i$ et $j$ en un ensemble $i$' est: $\pi_i' = \pi_i^* + \pi_j^*$ et $\theta_i' = \frac{\pi_i^*\theta_i' + \pi_j^*\theta_j'}{\pi_i^* + \pi_j^*}$ .
L'initialisation des paramètres après diviser un ensemble $k$ en deux sous-ensembles $j$' et $k$': $\pi_j^i = \pi_k' = \frac{\pi_k^*}{2}$ et $\theta_j' = \theta_k^* + \varepsilon$ et $\theta_k' = \theta_k^* + \varepsilon'$ (avec $\varepsilon$ et $\varepsilon$' sont deux matrices ou vecteur de perturbation). La matrice de covariance doit être toujours positive, pour cela ils ont introduit la formule suivante :

$$\Sigma_{j'} = \Sigma_{k'} = \det(\Sigma_k)^{1/D} I_d$$

L'EM partielle à la même structure d'EM sauf qu'il est appliqué seulement aux groupes sélectionnés sans affecter les autres, la formule de la probabilité a posteriori devient de la forme:

$$P(m'|x;\theta^{(t)}) = \frac{\pi_{m'}^{(t)} p_{m'}(x;\theta_{m'}^{(t)})}{\sum_{l=i',j',k'} \alpha_l^{(t)} p_l(x;\theta_l^{(t)})} \sum_{m=i,j,k} P(m|x;\theta^*), \qquad pour\ m' = i',j',k' \qquad (4.7)$$

**Algorithme SMEM:**
Initialisation des paramètres comme pour EM
Itération de l'algorithme EM jusqu'à convergence
**Faire**
Sélection des candidats à la fusion et division en calculant les critères de fusion et division
Tri cette liste, on note $\{i, j, k\}_c$ le $C^{ème}$ candidat.
**Pour** c = 1, . . . , $Cmax$ **faire**
Effectue les opérations des divisions-fusion
Initialiser les paramètres affectés (EM partielle)
Applique l'algorithme EM sur l'ensemble des groupes jusqu'à convergence
**si** le nouveau vraisemblance est mieux que celui avant les opérations de division-fusion **alors**
ignorer les modifications et retourner à l'étape des sélections des nouveaux candidats.
**Sinon**
Restaurer les paramètres
**Jusqu'à** ce qu'aucun candidat n'améliore l'ancienne vraisemblance

*Figure 18: Algorithme général SMEM*

En SMEM le nombre de composantes est constant : à chaque itération, une composante est divisée et deux autres sont fusionnées.

## 3.2. Algorithmes d'optimisation par division fusion des gaussiennes :

### *a. SMEM (Zhihua Zhang[20]) :*

Du fait que EM est très sensible à l'initialisation, et les méthodes d'initialisation pratique sont eux-mêmes convergent vers les optimaux locaux, ces algorithmes peuvent accélérer la convergence d'EM, mais ne garantit par la convergence vers l'optimum global, [20] ont discuté ce problème d'initialisation des paramètres, tandis que l'opération de division est la procédure inverse de la fusion, ils admettent que si la $k^{ième}$ composante est divisée en deux composantes $i$ et $j$, alors les vecteurs moyens et les matrices de covariance devraient satisfaire les équations répartition suivante:

$$\pi_k = \pi_{j'} + \pi_{k'}$$
$$\pi_k \mu_k = \pi_{j'} \mu_{j'} + \pi_{k'} \mu_{k'}$$
$$\pi_k (\Sigma_k + \pi_k \mu_k{}^T) = \pi_{j'} (\Sigma_{j'} + \pi_{j'} \mu_{j'}{}^T) + \pi_{k'} (\Sigma_{k'} + \pi_{k'} \mu_{k'}{}^T)$$

Ce qui n'était pas le cas pour SMEM, en SMEM que la matrice de covariance n'était pas prise en charge, elle est considérée comme qu'elle n'influence pas sur les résultats. Pour arriver à ce résultat, ils ont proposé deux méthodes basées sur la décomposition de matrice de covariance : une se base sur la décomposition en valeurs singulières et l'autre sur la décomposition de Cholesky, dans le cas où la matrice de covariance est diagonale le résultat de ces deux décompositions est la même.
Pour chaque matrice Σ il existe une matrice $A = [a_1, a_2, \dots, a_n]$ telle que:

$$\Sigma = A * A^T = \sum_{j=1}^{n} a_j * a_j^T$$
$$a_j^T * a_j = \begin{cases} 0 & i \neq j \\ \lambda_i & i = j \end{cases}$$

Donc après l'opération de division, au lieu de cherche les matrices $\Sigma_{j'}$ et $\Sigma_{k'}$, il suffit de trouver les matrices $A_{j'} = a_1^{(j')}, a_2^{(j')}, \dots, a_n^{(j')}$ et $A_{k'} = a_1^{(k')}, a_2^{(k')}, \dots, a_n^{(k')}$ ,
Avec $\Sigma_{j'} = A_{j'} * A_{j'}^T$ et $\Sigma_{k'} = A_{k'} * A_{k'}^T$
Les éléments de ces matrices sont déterminés à l'aide de formules suivantes :

$$a_m^{(j')} = \begin{cases} \sqrt{\beta(1-u^2)\frac{\pi_k}{\pi_{j'}}}\, a_m^{(k)} & m = l \\ \sqrt{\frac{\pi_{k'}}{\pi_{j'}}}\, a_m^{(k)} & m \neq l \end{cases} \quad (4.8)$$

$$a_m^{(k')} = \begin{cases} \sqrt{\beta(1-u^2)\frac{\pi_k}{\pi_{k'}}}\, a_m^{(k)} & m = l \\ \sqrt{\frac{\pi_{j'}}{\pi_{k'}}}\, a_m^{(k)} & m \neq l \end{cases} \quad (4.9)$$

L'initialisation des autres paramètres est :

$$\pi_{j'} = \pi_k \alpha \quad et \quad \pi_{k'} = \pi_k(1-\alpha)$$

$$u_{j'} = u_k - \sqrt{\frac{\pi_{k'}}{\pi_{j'}}} u a_l^{(k)} \quad et \quad u_{k'} = u_k - \sqrt{\frac{\pi_{j'}}{\pi_{k'}}} u a_l^{(k)}$$

Avec $l \in \{1,2,\dots,n\}$ $choisit\ aléatoirement, et\ \alpha, \beta, u \in (0,1)$
Le critère de fusion est modifié comme suit:

$$J_{merge}(i,j;\theta^*) = \frac{[P_i(\theta^*)^T P_j(\theta^*)]}{\|P_i(\theta^*)\| * \|P_i(\theta^*)\|} \quad (4.10)$$

### *b. MSMEM [21]:*

Dans le cadre de la vérification de locuteur [21] ont présenté une méthode MSMEM qui est une amélioration de SMEM, en modifiant les critères de division fusion :

$$J_{merge}(i,j;\theta^*) = \min\left\{\frac{|P_i(\theta)|}{|P_j(\theta)|}, \frac{|P_j(\theta)|}{|P_i(\theta)|}\right\} * [P_i(\theta^*)^T P_j(\theta^*)] \quad (4.11)$$

$$J_{split}(k;\theta^*) = (1/D_{norm}) \int p_k(x;\theta^*) log \frac{p_k(x;\theta)}{P_k(x;\theta)} dx \quad (4.12)$$

Avec $D_{norm}$ est le nombre d'observations dans le $k^{ième}$ groupe normalisées.

### *c. SMILE [22] :*

[22] ont présenté un nouvel algorithme appelé : SMILE (Split and Merge Incremental Learning), qui est une méthode incrémentale pour construire le modèle de mélange. Étant donné le nombre de classes désiré $k > 2$, leurs algorithmes commencent par 2 classes, et il effectue une suite d'opérations division-fusion successives jusqu'à l'optimisation du maximum de vraisemblance. Chaque opération de division ou fusion est suivie d'une optimisation, qui est l'EM partiel.
La sélection du groupe à diviser est facilitée à l'aide d'une parmi les méthodes suivantes :

- ✓ L'entropie maximale:

$$H(j) = -\int p(x|\theta_j) \log p(x|\theta_j) dx \qquad (4.13)$$

- ✓ la moyenne minimale du maximum de vraisemblance locale :

$$L(j) = \frac{\sum_{i=1}^{N} p(j|x_i\theta_j) \log\left(p(x_i|\theta_j)\right)}{\sum_{i=1}^{N} p(j|x_i, \theta_j)} \qquad (4.14)$$

- ✓ Le maximum de divergence locale de Kullback :

$$J(j) = \int f(x|\theta) log \frac{f(x|\theta)}{p(x|\theta_j)} dx \qquad (4.15)$$

Pour la fusion, deux critères sont utilisés :

- ✓ La distance de distribution minimale ( Symmertric Kullback Leibler) :

$$\int p(x|\theta_{k_1}) log \frac{p(x|\theta_{k_1})}{p(x|\theta_{k_2})} dx + \int p(x|\theta_{k_2}) log \frac{p(x|\theta_{k_2})}{p(x|\theta_{k_1})} dx \qquad (4.16)$$

- ✓ ou la distribution maximale Overlap *(utilisé en* [19]*)*:

$$\sum_{i=1}^{N} p(k_1|x_i, \theta_{k_1}) p(k_2|x_i, \theta_{k_2}) \qquad (4.17)$$

### *d. SMEM Finnian [23] :*

[23] ont présenté une méthode proche à **MSMEM**[21]**,** avec modification du critère de fusion comme suit :

$$J_{merge}(i, j; \theta^*) = \frac{[P_i(\theta^*)^T P_j(\theta^*)]}{\|P_i(\theta^*)\| * \|P_i(\theta^*)\|} \qquad (4.18)$$

### *e. SMEM Ran Xin [24] :*

[24] ont remarqués que les algorithmes basés sur SMEM nécessitent les $K$ échantillons de données ou groupes avant de démarrer l'algorithme, ce qui rend les applications en temps réel trop lent. Ils ont présenté une méthode incrémentale basée sur l'algorithme MSMEM présenté en [21]**,** les critères de division fusion sont les mêmes. Cet algorithme essaye de modifier les paramètres du modèle de mélange immédiatement après l'ajout de chaque nouvelle composante, ce qui est demandé dans les applications en temps réel.

## 3.3. Algorithmes d'auto-classification par la méthode division fusion des gaussiennes :

Contrairement aux algorithmes d'optimisation d'EM, qui permettent juste de trouver les meilleurs paramètres du modèle, les algorithmes présentés dans cette section permettent au fur et à mesure de trouver le nombre de gaussienne nécessaire pour décrire l'ensemble des données et d'optimiser les paramètres.

### *f. FSMEM [25] :*

Daniel [25] a proposé son algorithme FSMEM, ce dernier permet de modifier le nombre de clusters d'une manière dynamique, il exécute les opérations de division d'une manière indépendante : il commence par fusionner les ensembles qui vérifient le critère de fusion parmi tous les ensembles de données, et continue de le faire jusqu'à ce qu'il ne trouve aucun ensemble à fusionner. Puis, il cherche des ensembles à diviser, et il continue jusqu'à ce qu'il ne trouve pas. Il répète ce processus jusqu'aucune opération de division ou fusion n'améliore le MDL. Le MDL (Minimal Description Length) est donné par :

$$L_{MDL} = L - \frac{1}{2}\log(N)\, K(1 + D + \frac{1}{2}D(D + 1) \qquad (4.19)$$

### *g. SMEM Wang [26] :*

[26] ont proposé l'algorithme SSMEM, c'est une nouvelle formulation de SMEM par introduction de deux seuils de division fusion : $T_{split}$ et $T_{Merge}$, le choix des éléments à modifier n'est plus basé sur le $Cmax$ mais sur ces seuils :
Les groupes à fusionner sont ceux qui ont : $J_{merge}(i, j; \theta^*) > T_{Merge}$
Les groupes à diviser sont ceux qui ont : $J_{split}(k; \theta^*) > T_{Split}$
Le critère de fusion en SSMEM est :

$$J_{merge}(i, j; \theta^*) = \frac{\left(P_i(\theta) - \overline{P}_i(\theta)\right)^T (P_j(\theta) - \overline{P}_j(\theta))}{\|P_i(\theta) - \overline{P}_i(\theta)\| * \left\|P_j(\theta) - \overline{P}_j(\theta)\right\|} \qquad (4.20)$$

### *h. SMEM Shih-Sian [27] :*

[27] ont proposé un algorithme SGML, cet algorithme commence par un seul composant, puis il effectue des divisions successives, jusqu'au trouver le nombre d'éléments le plus approprié. Le BIC qui est utilisé pour sélectionner le composant à diviser, aussi pour la détermination du nombre appropriée de composantes. L'idée est de calculer le $BIC_1(X)$ , qui est le BIC de l'ensemble $X$ supposant que cet ensemble présente un cluster, puis le $BIC_2(X)$ qui est le BIC de l'ensemble $X$ supposant que cet ensemble présente deux cluster, si $\Delta BIC = BIC_2 - BIC_1 > 0$ alors il est nécessaire de diviser cet ensemble en deux.

### *i. SMEM Yan Li [28] :*

L'idée de la méthode présentée par [28] est d'effectuer les deux opérations fusion et division pour les ensembles sélectionnés, mais n'accepter que celle qui donne une valeur maximale de la vraisemblance, si aucune des opérations n'améliore la vraisemblance alors on arrête avec les paramètres actuels. Ils ont présenté des nouveaux critères de division et de fusion basée sur le taux entre l'entropie et l'entropie maximale :

$$. H^i(X) = -\frac{1}{N_i}\sum_{t=1}^{N_i} log\emptyset(x_t|\theta_i) \qquad (4.21)$$

$$H^i_{max}(Y) = -\frac{1}{2}\log[(2\pi e)^d\, |\Sigma_i|] \qquad (4.22)$$

Si le taux $M_{e_{ij}}$ entre deux ensembles $i$ et $j$ est le plus grand, alors on doit fusionner ces ensembles, si le taux $Sp_i$ est le plus petit alors on doit diviser cet ensemble. Le MDL utilisé est :

$$L(S|\theta_k) = \log p(S|\theta_k) - \frac{n}{2}\sum_{i=1}^{k}\log\left(\frac{N\alpha_i}{12}\right) - \frac{k}{2}\log\left(\frac{N}{12}\right) - \frac{k(n+1)}{2} \quad (4.23)$$

### *j. SMEM GuoQing [29] :*

[29] ont donné un seuil de division et une autre pour la fusion, comme suivant :
Condition de division : Si $\sigma_{A,s} > \sigma_{seuil}$ alors crée deux groupes $s$ et $S$.
Condition de fusion : Si $\left|\mu_{A,s'} - \mu_{A,s''}\right| < \mu_{seuil}$ $et$ $\left|\sigma_{A,s'} - \sigma_{A,s''}\right| < \sigma_{seuil2}$
(avec $\sigma_{seuil2} \leq \sigma_{seuil}$) alors fusionner les deux groupes en une seule groupe.

## 4. Conclusion :

Dans ce chapitre nous avons vu les différentes méthodes de classification non supervisé, ainsi que les différents essais pour faire une auto-classification, nous avons focalisé notre étude sur la méthode basés sur l'algorithme division–fusion, en présentent les différentes variantes de cette méthode.

# Chapitre V

# Expériences et résultats

# 1. Introduction

Après avoir présenté les différentes variantes de l'algorithme division fusion des gaussiennes, nous allons comparer dans ce chapitre les résultats de ces algorithmes dans le cadre d'identification du locuteur en mode indépendant du texte. Afin d'atteindre cet objectif, nous présentons le dispositif expérimental ainsi que les résultats obtenus.

# 2. Le dispositif expérimental:

## 2.1. Implémentation des paramètres MFCC :

Plusieurs implémentations pour extraire les paramètres MFCC ont été développées [30], elles se diffèrent dans le nombre de filtres nécessaires, la forme des filtres, la façon dont ces filtres sont espacés, et la bande passante des filtres, on distingue : MFCC_FB20, HTK_MFCC_FB24, HTK_MFCC_EB26, MFCC_FB40, HFCC_E_FB29, Skowronski_MFCC_FB20.
Il existe diverses approximations de la perception non linéaire du pitch par le système auditif humain, l'approximation de Koenig(1949) considère que le pitch est linière au-dessous de 1000 Hz, et logarithmique au-delà, cette approximation fournit une représentation de calcul peu coûteuse de l'échelle de Mel, mais elle ne donne pas une meilleure précision.
Une autre approximation plus précise est celle de Fant(1949):

$$f_{mel} = k_{const}.\log_n(1 + \frac{f_{lin}}{F_b}) \qquad (5.1)$$

Avec $F_b = 1000$. L'équation devient :

$$f_{mel} = \frac{1000}{\log_n 2}.\log_n(1 + \frac{f_{lin}}{1000}) \qquad (5.2)$$

Cette approximation par rapport à celle de Koeing fournit une représentation très proche de l'échelle de mel. En outre, la formulation (5.2) est particulièrement intéressante vu que les valeurs de $f_{mel}$ sont pas affecté par le choix de la base du logarithme. Ce choix a conduit à l'apparition de ces deux représentations :

$$f_{mel} = 2595.\log_{10}(1 + \frac{f_{lin}}{700}) \qquad (5.3)$$

Et

$$f_{mel} = 1127.\log_e(1 + \frac{f_{lin}}{700}) \qquad (5.4)$$

Qui sont largement utilisé dans les différentes implémentations de MFCC, et fournir une approximation plus proche de l'échelle de Mel en comparaison avec (5.2).
Dans cette étude nous utilisons les paramètres HTK_MFCC_FB24 issus à partir de la plateforme HMM de (HTK), qui est originalement mise au point à l'université de Cambridge , décrit par Young(1995), la désignation HTK_MFCC_FB24 reflète le nombre de filtres (24) recommandé par HTK, pour une bande passante de 8000 Hz du signal, échantillonné par une fréquence >=16Khz.
Cette variante utilise aussi la représentation fréquentielle selon l'échelle de mel donné dans l'équation (4.3), les limites de la gamme de fréquences sont les paramètres qui définissent la base de la conception du banc de filtres. Pour cela, il est nécessaire de déterminer d'abord les fréquences minimales $\hat{f}_{low}$ et maximales $\hat{f}_{high}$ qui limitent la gamme de fréquence utilisée,

puis on calcule $\Delta\hat{f}$ , qui est considérée comme une empreinte des fréquences centrales de chaque filtre, il est exprimé par :

$$\Delta\hat{f} = \frac{\hat{f}_{high} - \hat{f}_{low}}{M + 1} \qquad (5.7)$$

Avec M le nombre total des filtres.

La fréquence centrale $\hat{f}_{c_i}$ du $i^{ème}$ filtre est donnée par l'équation :

$$\hat{f}_{c_i} = \hat{f}_{low} + i.\Delta\hat{f}\,, \qquad i = 1, \dots, M-1 \qquad (5.6)$$

La conversion des fréquences centrales des filtres en fréquence linéaire (Hz) est donnée par:

$$f_{c_i} = 700\left(10^{\frac{\hat{f}_{c_i}}{2595}} - 1\right) \qquad (5.7)$$

Les paramètres HTK_MFCC_FB24 sont calculés comme suit : la DFT $X(k)$ du signal d'entrée discrète $x(n)$ est calculée, puis elle est utilisée pour calculé l'amplitude du signale : $|X(k)|$, qui agit comme entrée pour la banque de filtres $H_i(k)$ . Ensuite, la sortie de la banque de filtre est donnée par:

$$X_i = \ln\left(\sum_{k=0}^{N-1} |X(k)|.H_i(k)\right)$$

Finalement les paramètres HTK_MFCC_FB24 sont extraits par l'application de la transformée en cosinus inverse DCT.

## 2.2. Description des paramètres MFCC extraits :

Les paramètres acoustiques utilisés dans ce travail sont basés sur la méthode HTK_MFCC_FB24 en utilisant la configuration expérimentale suivante : La parole numérisée est préaccentuée en utilisant le filtre avec la fonction de transfert : $H(z) = 1 - 0.9z^{-1}$ , puis un fenêtrage de Hamming de taille 25ms (400 échantillons) avec chevauchement de 10ms (160 échantillons) entre deux trames adjacentes. Chaque fenêtre a été paramétrée par un vecteur composé de 19 MFCCs. Finalement, les vecteurs de caractéristiques globales extraites des différentes fenêtres de chaque locuteur sont modélisés par un mélange de gaussiennes multivariées diagonales.

Pendant la phase d'apprentissage, environ de 60 secondes de la parole par locuteur ont été utilisés pour construire des modèles de locuteur, alors que dans la phase de test, les données de test se composaient de 10 sessions d'identifications. (10 tests par locuteur, chacune de durée de 8s).

## 2.3. Description de la base de données :

La base de données utilisée dans nos expériences est composée des enregistrements de 40 locuteurs [Marocains] : 17 femmes et 23 hommes, dont leurs âges sont entre 18 et 30 ans.

Pour réduire les variations entre les signaux d'un même locuteur, les enregistrements de chaque locuteur sont pris en deux sessions séparées par deux ou trois semaines, ces enregistrements contentant des paroles en : dialecte marocain, Arabe, français et en anglais.

Les enregistrements audio ont été recueillis via Skype, l'un des outils de communication VOIP les plus utilisés dans le monde. Pour couvrir les différentes situations présentes dans la vie réelle, les locuteurs font leurs enregistrements depuis plusieurs places : maison, bureau, …, et en utilisant différents dispositifs d'enregistrements : (ordinateur, tablette, téléphone,…).

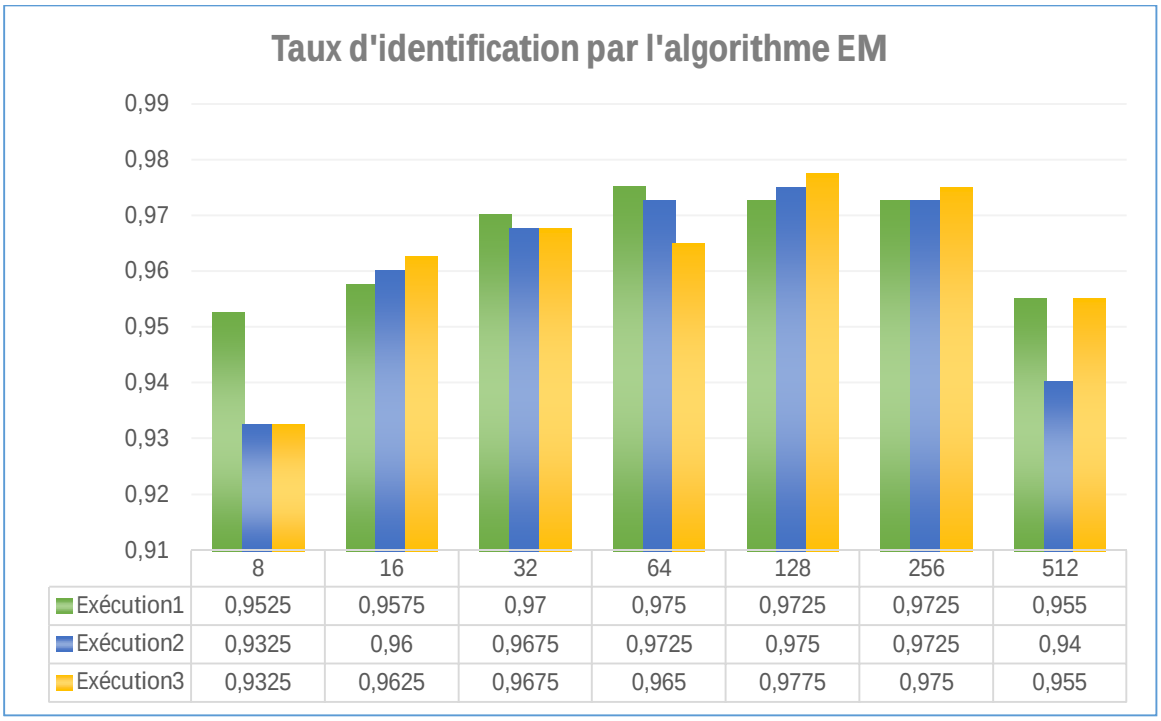
Les voix sont numérisées à 16 kHz avec une résolution de 16 bits (mono, PCM), et stockés en des fichiers de format wav, qui est considérée comme le format d'audio la plus utilisée.

# 3. Les expériences et résultats:

Le même protocole expérimental a été adapté dans toutes les expériences effectué, dans cette partie nous décrivons les résultats obtenus :

## 3.1. Résultats de l'algorithme EM :

Dans ce premier test, nous avons exécuté l'algorithme EM pour différentes classes K de mélange de gaussiennes, K={8,16, 32, 64, 128, 256, 512}, chaque exécution a été répétée trois fois. Les résultats sont exprimés en termes de taux d'identification (équation (1.1) ):

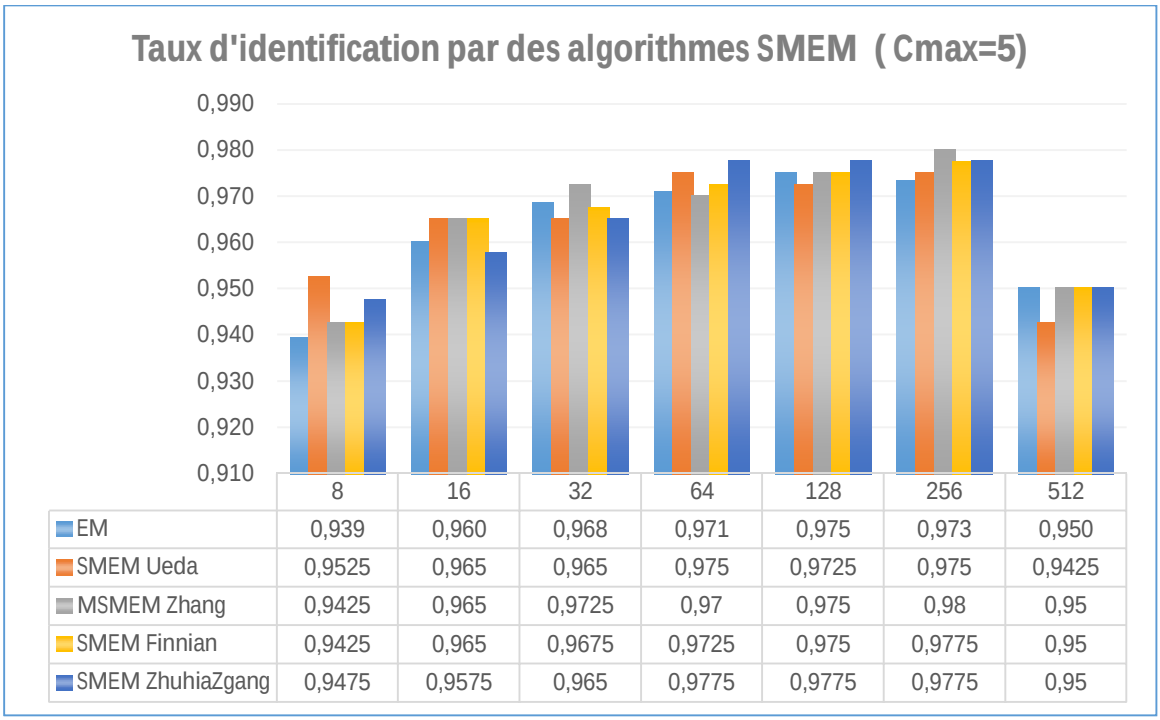



| | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
|---|---|---|---|---|---|---|---|
| Exécution1 | 0,9525 | 0,9575 | 0,97 | 0,975 | 0,9725 | 0,9725 | 0,955 |
| Exécution2 | 0,9325 | 0,96 | 0,9675 | 0,9725 | 0,975 | 0,9725 | 0,94 |
| Exécution3 | 0,9325 | 0,9625 | 0,9675 | 0,965 | 0,9775 | 0,975 | 0,955 |

*Figure 19: Taux d'identification de l'algorithme EM*

## 3.2. Résultats des algorithmes SMEM :

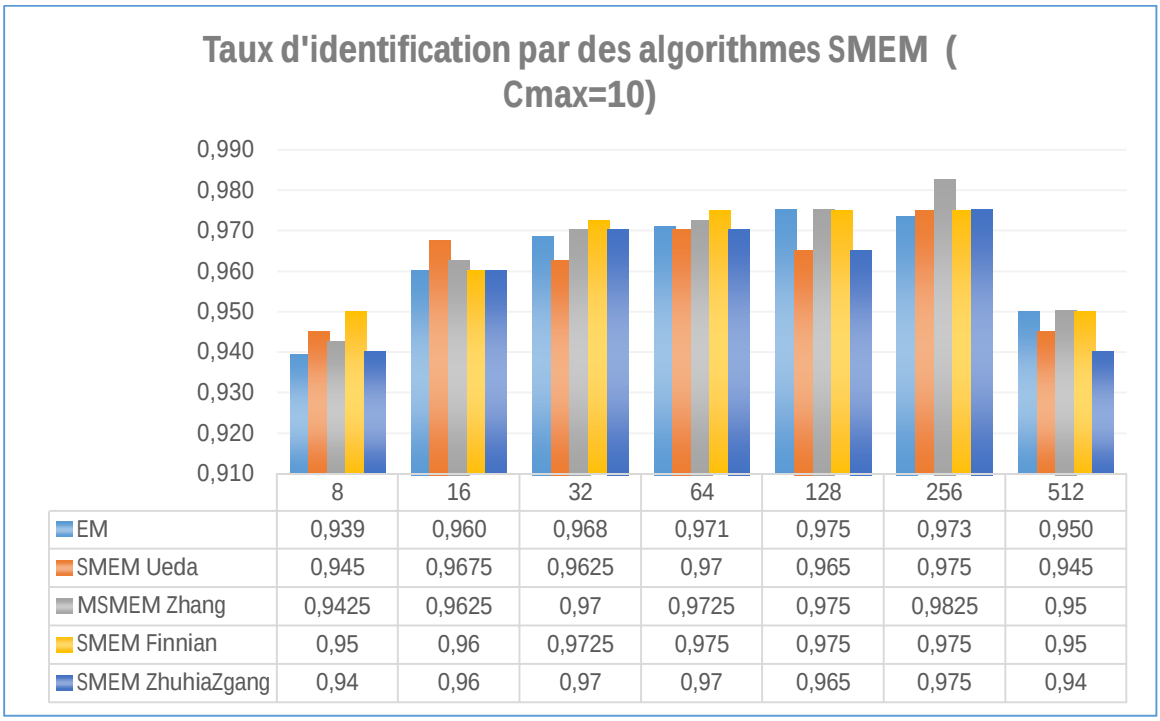
Dans cette partie nous présentons une comparaison des algorithmes SMEM avec EM. Dans un premier temps nous avons choisi un nombre de candidat Cmax=5 et nous avons exécuté les algorithmes SMEM pour différentes valeurs de K, le figure 20 montre les résultats obtenus :




| | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
|---|---|---|---|---|---|---|---|
| EM | 0,939 | 0,960 | 0,968 | 0,971 | 0,975 | 0,973 | 0,950 |
| SMEM Ueda | 0,9525 | 0,965 | 0,965 | 0,975 | 0,9725 | 0,975 | 0,9425 |
| MSMEM Zhang | 0,9425 | 0,965 | 0,9725 | 0,97 | 0,975 | 0,98 | 0,95 |
| SMEM Finnian | 0,9425 | 0,965 | 0,9675 | 0,9725 | 0,975 | 0,9775 | 0,95 |
| SMEM ZhuhiaZgang | 0,9475 | 0,9575 | 0,965 | 0,9775 | 0,9775 | 0,9775 | 0,95 |

*Figure 20: Taux d'identification par des algorithmes SMEM ( Cmax=5)*

Après nous avons changé le nombre de candidats à Cmax=10. La figure 21 montre les résultats obtenus :




| | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
|---|---|---|---|---|---|---|---|
| EM | 0,939 | 0,960 | 0,968 | 0,971 | 0,975 | 0,973 | 0,950 |
| SMEM Ueda | 0,945 | 0,9675 | 0,9625 | 0,97 | 0,965 | 0,975 | 0,945 |
| MSMEM Zhang | 0,9425 | 0,9625 | 0,97 | 0,9725 | 0,975 | 0,9825 | 0,95 |
| SMEM Finnian | 0,95 | 0,96 | 0,9725 | 0,975 | 0,975 | 0,975 | 0,95 |
| SMEM ZhuhiaZgang | 0,94 | 0,96 | 0,97 | 0,97 | 0,965 | 0,975 | 0,94 |

*Figure 21: Taux d'identification par des algorithmes SMEM ( Cmax=10)*

Nous avons aussi tester l'algorithme FSMEM[25] pour différentes valeurs d'initialisation de K={8,32,128} et Cmax={5,10,20,40}, le figure 22 montre les résultats obtenus :

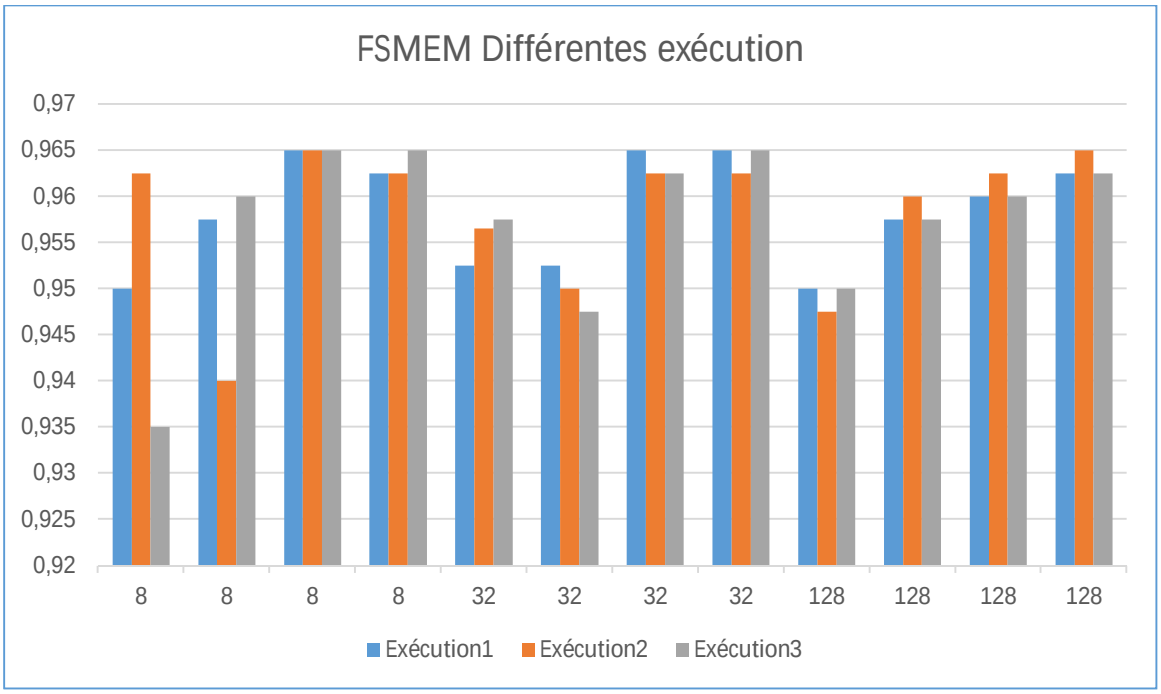



*Figure 22: Taux d'identification FSMEM*

Pour faciliter la visualisation, nous avons pris les valeurs moyennes de valeurs issues des différentes exécutions :

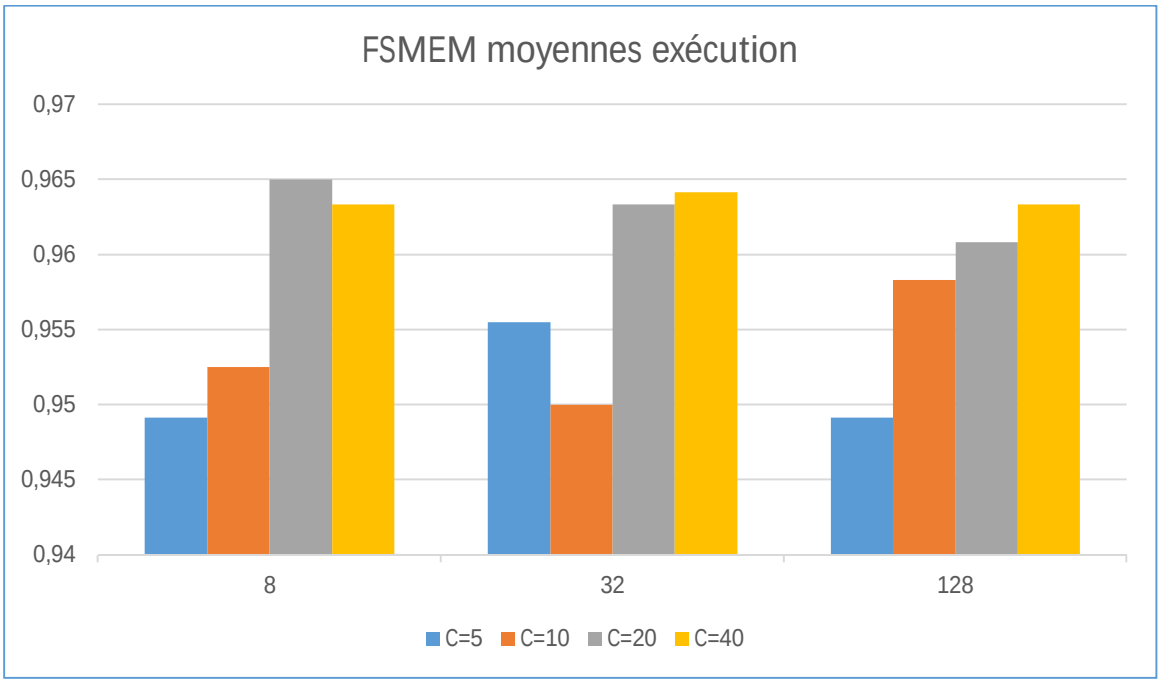



*Figure 23: Taux d'identification FSMEM*

## 3.3. Intérprétation des résultats:

Les résultats affichées en figure 19 confirme le fait qu'EM tombe parfois dans un minimum local. C'est la raison pour laquelle nous obtenons des taux d'identification faible en

quelques tests pour le même nombre de classe. Ceci s'affiche clairement dans la modélisation par 8 gaussiennes.

Après les résultats de comparaison présentée en figures 20 et 21, nous remarquons que les algorithmes SMEM [19]–[21], [23] donnent des taux plus grands que celui d'EM, sauf pour l'algorithme SMEM de Ueda[19], ses performances sont dégradée pour certains nombres de K. MSMEM [21] et SMEM de Finnian [23] ont présenté toujours des taux bonne que les autres algorithmes, ceci est expliqué par le choix des critères de division-fusion. En ce qui concerne l'influence du Cmax: l'augmentation du Cmax apporte une amélioration des performances.

## 4. Conclusion :

Dans ce chapitre, nous avons présenté les évaluations des algorithmes SMEM sur la base de données de la FSTF. Certaines méthodes ont montré leurs efficacités, d'autres ont montrés que leurs performances se dégradent en fonction des paramètres initiaux.

# Conclusion et perspectives

Le principal thème d'étude de cette mémoire a été l'apprentissage des modèles de mélanges par la méthode de division fusion. Nous avons débuté par un chapitre introductif expliquant l'architecture et le fonctionnement général d'un système RAL. Au chapitre 2 nous avons décrit en détail l'étape de paramétrisation qui explique comment nous obtenons les paramètres caractéristiques d'un locuteur. En chapitre 3 nous avons dressé un panorama des différentes approches de modélisation existantes, puis. Pour finir, nous nous sommes intéressés à la modélisation des paramètres par les mélanges gaussiens par la suite en chapitre 4 nous avons présenté les algorithmes d'auto-classification, et nous focalisons notre étude sur les algorithmes basés sur la méthode de division-fusion.

Les algorithmes de division fusion peuvent être classées en deux variantes : ceux qui laissent le nombre de gaussiens fixe, elles sont utilisées comme méthodes d'optimisation des paramètres des gaussiens. Et les autres qui modifient le nombre de gaussiens, ils sont considérés comme des algorithmes d'auto-classification, elles permettent à la fois de chercher le nombre optimal de gaussiens et d'optimiser les paramètres.

L'étude expérimentale porte sur une base de données de 40 locuteurs (17 femmes et 23 hommes), il s'agit d'un test d'identification de locuteurs. Nous avons comparé les taux d'identification données par les algorithmes de division fusion des gaussiens avec celui de l'algorithme de base EM. Ces expériences ont permis de se rendre compte que les algorithmes permettent d'augmenter le taux de reconnaissance, mais certains algorithmes ont été sensibles aux paramètres initiaux. Il faut mentionner aussi que l'apprentissage de ces algorithmes prend beaucoup de temps par rapport à EM.

Les travaux de recherche présentés dans ce document peuvent être poursuivis de nombreuses façons, parmi les perspectives : trouver des bons critères de division et de fusion, trouver une valeur acceptable (en terme de temps de calcules et des performances du système) de $Cmax$ , trouver la bonne initialisation après chaque opération de division/fusion, aussi combiner cette méthode avec des méthodes incrémentales dans le but de réduire la complexité temporelle d'apprentissage.

# Annexe :
# Les indices de validation

## 1. Introduction :

La recherche d'un bon critère de division et de fusion nous a amené effectuer une étude sur les techniques d'évaluation de classification. L'idée était de chercher d'abord la mesure qui donne une bonne estimation pour nos données, puis utiliser cette mesure comme critère de validation de classe, pour prendre la décision de division ou fusion de cette classe.

Différents critères ou indices de validité ont été proposés et étudiés, dans le but d'évaluer le résultat de classification, en général ils sont classés en trois catégories : critères de validité externes, internes ou relatifs. Dans cette partie nous détaillons les propriétés de chacun de ces indices.

## 2. Indices de validité internes:

Ici l'évaluation est basée sur l'ensemble de données et sa structure, aucune connaissance externe n'est utilisée pour la validation, mais uniquement les données d'entrée comme référence, on prend en considération que les objets de la même classe doivent être homogènes, les groupes doivent êtres isolées.

Les indices de validité internes prennent en considération deux critères :

*La compacité* : les membres de la même classe doivent être les plus proches les uns des autres. (Exemple mesure de variance).

*La séparation* : les groupes doivent être séparés le plus que possible. Trois approches pour mesurer la distance entre deux groupes:

- Distance entre les éléments les plus proches des deux groupes (Single linkage).
- Distance entre les éléments les plus éloignées (Complete linkage).
- Distance entre les centres des groupes (Comparison of centroids).

Soit :

$$SSW_M = \sum_{i=1}^{N} \left\| x_i - c_{p_i} \right\|^2$$

$$SSB_M = \sum_{i=1}^{N} n_i \| c_i - \bar{X} \|^2$$

Le SSW est utilisé pour mesurer la compacité, tandis que le SSB est utilisé pour mesurer la séparation.

- L'indice de *Dunn* [Dunn,1974] est basé sur la distance inter classes, il définit le rapport entre la distance minimale interclasse notée $d_{min}$ et la distance maximale intra-classes noté $d_{max}$ , la formule est donnée par :

$$D = \frac{d_{min}}{d_{max}}$$

autre formulation :

$$d(c_i, c_j) = min_{x \in c_i, x' \in c_j} \|x - x'\|^2$$
$$diam(c_k) = max_{x,x' \in c_k} \|x - x'\|^2$$
$$Dunn = \frac{min_{i=1}^{M} min_{j=i+1}^{M} d(c_i, c_j)}{max_{k=1}^{M} diam(c_k)}$$

- L'indice de *Davis-Bouldin* [Davies et Bouldin, 1979] est donné par :

$$DB = \frac{1}{n} \sum_{i=1, i \neq j}^{n} \max \left( \frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

avec n : le nombre de clusters, $\sigma_i$ la distance moyenne de tous les objets de cluster par rapport au centre de groupe $c_i$, $\sigma_j$ est la distance moyenne de tous les objets de cluster par rapport au centre de groupe $c_j$, $d(c_i, c_j)$ est la distance entre les centres de clusters $c_i$ et $c_j$.
Plus que le DB est petit, plus les clusters sont compacts, et leurs centres sont loin les uns des autres. Donc le nombre de classes qui minimise le DB correspond au nombre optimal de clusters.
Autre formulation :

$$Soit\ R_{ij} = \frac{S_i + S_j}{d_{ij}}, i \neq j$$
$$avec\ d_{ij} = \|c_i - c_j\|^2$$
$$et\ S_i = \frac{1}{n} \sum_{j=1}^{n_i} \|x_j - c_i\|^2$$
$$R_i = max_{j=1,\dots,M} R_{ij}\ , i = 1, \dots, M$$
$$DBI = \frac{1}{M} \sum_{i=1}^{M} R_i$$

- *C-index* [Hubert, 1976] est défini par:

$$C = \frac{S - S_{min}}{S_{max} - S_{min}}$$

Avec S est la somme des distances entres chaque paire d'objets du même cluster, n est le nombre de ces paires, $S_{min}$ est la somme des n petits distances de toutes les paires, du même $S_{max}$ est la somme des n grands distances de toutes les paires.
Cindex $\epsilon$ [0,1] Et dois être minimisée.

- *Xie-Beni index* : mesure la compacité moyenne globale et la séparation d'un ensemble floue, la formule est donnée par:

$$XB = \frac{\sum_{i=1}^{N}\sum_{k=1}^{M} {u_{ik}}^2 \|x_i - C_k\|^2}{N\, min_{t\neq s}\{\|C_t - C_s\|^2\}}$$

Avec $u_{ik}$: est le degré d'appartenance de l'objet $i$ au groupe $k$ , $C_k$ est le centre du groupe $k$, $M$ : nombre de groupes, $N$ le nombre de points de l'ensemble de données.

- L'indice de *Calinski-Harabasz*:

$$CH = \frac{SSB_M / (M-1)}{SSW_M / (N-M)} = \frac{SSB_M (N-M)}{SSW_M (M-1)}$$

Avec : $M$ nombre de clusters, $N$ nombre des objets.

- *Ball&Hall* : $BH = \frac{SSW_M}{M}$

- *Xu-index* : $Xu = Dlog_2\left(\sqrt{SSW_M / (DN^2)}\right) + logM$

- *Krzanowski-Lai :* $KL = \frac{|diff_M|}{|diff_{M+1}|}$

avec $diff_M = (M-1)^{\frac{2}{D}} SSW_{M-1} - M^{\frac{2}{D}} SSW_M$

- *Hartigan* : $H = \left(\frac{SSW_M}{SSW_{M+1}} - 1\right)(N - M - 1)$

ou $H = log_2\left(\frac{SSB_M}{SSW_M}\right)$

- *R-square*: $RS = \frac{SST - SSW}{SST}$

avec :

$SSW = \sum_{\substack{k=1,\dots,M \\ d=1,\dots,D}} \sum_{i=1}^{n_{kd}} (x_i - \bar{x}^d)^2$ et $SST = \sum_{d=1,\dots,D} \sum_{i=1}^{n_{kd}} (x_i - \bar{x}^d)^2$

- *RMSSTD :* $RMSSTD = \frac{\sum_{\substack{k=1,\dots,M \\ d=1,\dots,D}} \sum_{i=1}^{n_{kd}} (x_i - \bar{x}^d)^2}{\sum_{\substack{k=1,\dots,M \\ d=1,\dots,D}} (n_{kd} - 1)}$

- *BIC :* Schwarz (1978)

Permet de comparer à la fois plusieurs modèles:

$$BIC = L * N - \frac{1}{2} M(D+1) \sum_{i=1}^{M} log(n_i)$$

- *AIC :* Akaike Information Criterion (Akaike 1974) :

$$AIC = -2 \ln L(\theta) + 2k$$

k est le nombre de paramètres, L(θ) est la vraisemblance.

Le modèle à retenir est celui qui montre l'AIC le plus faible.

Il existe plusieurs versions du critère AIC « corrigé », notamment pour s'ajuster à de petits échantillons (quand le nombre de paramètres est grand par rapport aux observations n/k<40). L'une d'elles est la suivante :

$$AIC_c = AIC + \frac{2k(k+1)}{n-k-1}$$

AIC par Wolfe(1971) :

$$AIC_3 = -2L(M) + 3k$$

## 3. Indices de validité externes:

L'évaluation est appliquée à un ensemble de données dont on connaît déjà la répartition de ses membres, c.-à-d. lorsque les étiquettes sur les classes sont disponibles. En général on compare le résultat donné par deux méthodes, dont l'une est prise comme référence.

Soit $\{X_1, X_2, \dots, X_n\}$ un ensemble de données, supposons que nous avons deux répartitions de ces données : $C = \{P_1, P_2, \dots, P_M\}$ de $M$ classes, et $P = \{G_1, G_2, \dots, G_R\}$ de R groups.

Pour paire de points $X_i$ et $X_j$ , on aura donc 4 états :

- ➔ $X_i$ et $X_j$ appartiennent au même classe de C, et au même groupe de P.
- ➔ $X_i$ et $X_j$ appartiennent au même classe de C, mais au différentes groupes de P.
- ➔ $X_i$ et $X_j$ appartiennent aux différentes classes de C, mais au même groupe de P.
- ➔ $X_i$ et $X_j$ appartiennent aux différentes classes de C, et au différentes groupes de P.

Soit $M_{11}, M_{10}, M_{01}$, et $M_{00}$ le nombre des paires des points correspondants à chaque état. Nous avons :

$$M_{11} + M_{10} + M_{01} + M_{00} = \binom{n}{2} = \frac{n(n-1)}{2}$$

Ces différentes paires sont souvent obtenues à l'aide du tableau de contingence suivant: (Un tableau de contingence est un type de table dans un format de matrice M * R qui affiche tous les possibles chevauchements entre chaque paire de grappes en C et P).

*Tableau 1: contingence entre deux partitions C et P*

| classe\ groupe | $P_1$ | $P_2$ | … | $P_R$ | Somme |
|---|---|---|---|---|---|
| $C_1$ | $n_{11}$ | $n_{12}$ | … | $n_{1R}$ | $n_{1.}$ |
| $C_2$ | $n_{21}$ | $n_{22}$ | … | $n_{2R}$ | $n_{2.}$ |
| … | … | … | … | … | … |
| $C_M$ | $n_{M1}$ | $n_{M2}$ | … | $n_{MR}$ | $n_{M.}$ |
| Somme | $n_{.1}$ | $n_{.2}$ | … | $n_{.R}$ | $n_{..}$ =n |

La dernière ligne et dernière colonne désignent les sommes marginales : $n_{i.} = \sum_j n_{ij}$ et $n_{.j} = \sum_i n_{ij}$ . Dans ce cas nous avons aussi : $n_{i.} = |C_i|$ , et $n_{.j} = |P_j|$ .

Le nombre de paires peut être obtenu à l'aide des formules suivantes:

$$M_{11} = \sum_{i=1}^{M}\sum_{j=1}^{R}\binom{n_{ij}}{2} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{R} n_{ij}^2 - \frac{n}{2}$$

$$M_{10} = \sum_{i=1}^{M}\binom{n_{i.}}{2} - \sum_{i=1}^{M}\sum_{j=1}^{R}\binom{n_{ij}}{2} = \frac{1}{2}\sum_{i=1}^{M} n_{i.}^2 - \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{R} n_{ij}^2$$

$$M_{01} = \sum_{j=1}^{R}\binom{n_{.j}}{2} - \sum_{i=1}^{M}\sum_{j=1}^{R}\binom{n_{ij}}{2} = \frac{1}{2}\sum_{j=1}^{R} n_{.j}^2 - \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{R} n_{ij}^2$$

$$M_{00} = \binom{n}{2} + \sum_{i=1}^{M}\sum_{j=1}^{R}\binom{n_{ij}}{2} - \sum_{i=1}^{M}\binom{n_{i.}}{2} - \sum_{j=1}^{R}\binom{n_{.j}}{2}$$

$$= \frac{1}{2}n^2 - \frac{1}{2}\left(\sum_{i=1}^{M} n_{i.}^2 + \frac{1}{2}\sum_{j=1}^{R} n_{.j}^2\right) + \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{R} n_{ij}^2$$

- *L'indice de Rand est défini par* :

$$RI = \frac{M_{00} + M_{11}}{M_{00} + M_{01} + M_{10} + M_{11}}$$

Qui donne:

$$RI = \binom{n}{2} + \sum_{i=1}^{M}\sum_{j=1}^{R}\binom{n_{ij}}{2} - \sum_{i=1}^{M}\binom{n_{i.}}{2} - \sum_{j=1}^{R}\binom{n_{.j}}{2}$$

La valeur de RI est dans l'intervalle [0,1] , plus que RI est proche de 1, plus la classification est bonne.

- *La mesure F :*

$$P = \frac{M_{11}}{M_{11} + M_{10}}$$

$$R = \frac{M_{11}}{M_{11} + M_{01}}$$

$$F_\alpha = \frac{1 + \alpha}{\frac{1}{P} + \frac{\alpha}{R}} = \frac{(\alpha^2 + 1)PR}{\alpha^2 P + R}$$

$\alpha = 1$ *moyenne harmonique*
$\alpha \epsilon (0;1)$ *favor precision over recall*
$\alpha > 1$ *favor recall over precision*

*Les deux valeurs communes de $\alpha$ sont 2 et 5.*

Les autres indices sont regroupés dans le tableau 2:

*Tableau 2: Indices de validités internes*

| **Référence** | **formule** |
|---|---|
| Jaccard | $J = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}$ |
| Dice (ou coefficient SD) | $SD = \frac{M_{10} + M_{01}}{2 * M_{11} + M_{01} + M_{10}}$ |
| Minkowski Score (MS) (Jardine et Sibson, 1971): | $MS = \frac{2 * (M_{01} + M_{10})}{n^2}$ |
| Hamann (1961) et Hubert (1977) | $H = \frac{(M_{11} + M_{00}) - (M_{01} + M_{10})}{M_{11} + M_{00} + M_{01} + M_{10}}$ |
| Czekanowski (1932), Dice (1945), Gower et Legendre (1986) | $Cze = \frac{2 * M_{11}}{2 * M_{11} + M_{01} + M_{10}}$ |
| Kulczynski index (1927) | $Kul = \frac{1}{2} * \left(\frac{M_{11}}{(M_{11} + M_{10})} + \frac{M_{11}}{(M_{11} + M_{01})}\right)$ |
| McConnaughey (1964) | $mcconn = \frac{M_{11}^2 - M_{10} * M_{01}}{(M_{11} + M_{01})(M_{11} + M_{10})}$ |
| Peirce (1884) | $peirce = \frac{M_{11} * M_{00} - M_{01} * M_{10}}{(M_{11} + M_{01})(M_{10} + M_{00})}$ |
| Wallace (1) (1983) | $wall1 = \frac{M_{11}}{(M_{11} + M_{01})}$ |
| Wallace (2) (1983) | $wall2 = \frac{M_{11}}{(M_{11} + M_{10})}$ |
| Gamma | $Gamma = \frac{M_{11} * M_{00} + M_{10} * M_{01}}{\sqrt{(M_{11} + M_{10}) * (M_{11} + M_{01}) * (M_{00} + M_{10}) * (M_{00} + M_{01})}}$ |

| | |
|---|---|
| Sokal et Sneath (1) (1963) | $sokal1 = 0.25 * \left(\frac{M_{11}}{(M_{11}+M_{10})} + \frac{M_{11}}{(M_{11}+M_{01})} + \frac{M_{00}}{(M_{00}+M_{10})} + \frac{M_{00}}{(M_{00}+M_{01})}\right)$ |
| Sokal et Sneath (2) (1963) | $sokal2 = \frac{M_{11}}{M_{11} + 2 * (M_{10}+M_{01})}$ |
| Sokal et Sneath (3) (1963), Ochiai (1957) | $\frac{M_{11} * M_{00}}{\sqrt{(M_{11}+M_{10}) * (M_{11}+M_{01}) * (M_{00}+M_{10}) * (M_{00}+M_{01})}}$ |
| Fager et McGowan (1963) | $fager = \frac{M_{11}}{\sqrt{(M_{11}+M_{10}) * (M_{11}+M_{01})}} + \frac{0.5}{(M_{11}+M_{10})}$ |
| Gowor et Legendre (1986), Sokal et Sneath (1963) | $gowor = \frac{M_{11}+M_{00}}{M_{11} + 0.5 * (M_{10}+M_{01}) + M_{00}}$ |
| Roger et Tanimoto (1960) | $roger = \frac{M_{11}+M_{00}}{M_{11} + 2 * (M_{10}+M_{01}) + M_{00}}$ |
| Goodman et Kruskal (1954), Yule (1927) | $goodman = \frac{M_{11} * M_{00} - M_{01} * M_{10}}{M_{11} * M_{00} + M_{01} * M_{10}}$ |
| Kruskal (1954), Yule (1927) | $kruskal = \frac{M_{11} * M_{00} + M_{01} * M_{10}}{(M_{11}+M_{10}) * (M_{11}+M_{01}) * (M_{01}+M_{00}) * (M_{10}+M_{00})}$ |
| Fowlkes et Mallows (1983), Ochiai (1957) | $FM = \sqrt{(M_{11}+M_{01}) * (M_{11}+M_{10})}$ |
| Russel et Rao (1940) | $RR = \frac{M_{11}}{M_{00}+M_{01}+M_{10}+M_{11}}$ |
| Baulieu(1989) | $B = \frac{M_{11} * M_{00} - M_{01} * M_{10}}{\binom{n}{2}^2}$ |

# 4. Conclusion:

Les indices de validation externes sont basés sur la connaissance préalable des informations sur les données, ils sont utilisés pour choisir de la bonne méthode de classification pour un ensemble donné. Tandis que les indices internes peuvent être utilisés

pour choisir le bon algorithme ainsi que le bon nombre de clusters, sans avoir besoin d'informations supplémentaires. D'autres critères aussi doivent être pris en considération, tel que :

- La stabilité : qui signifie qu'une petite modification sur les éléments d'une classe ou sur la méthode ne doit pas affecter le résultat de classification.
- L'avis d'un expert.
- Le nombre de classes.

Dans notre cas, nous ne disposons pas des labels qui indiquent l'appartenance des vecteurs aux groupes ni de nombre de groupes nécessaires, donc les indices de validité internes ne peuvent pas être utilisés. Aussi la nature des données traitées (des gaussiennes) rendent l'utilisation de certains indices internes un mauvais choix. Or d'autres mesures peuvent être utilisées pour évaluer la distance entre deux distributions de probabilités (distance de Patrick-Fischer, distance de Bhattacharyya, les divergences, …).

# Bibliographie


[1] M. Jessen, “Review article: *Forensic Speaker Identification* by P. Rose,” *Int. J. Speech Lang. Law*, vol. 10, no. 1, pp. 138–151, Jan. 2007.

[2] T. H. Kinnunen, “Optimizing spectral feature based text-independent speaker recognition /.”

[3] A. K. Jain, A. Ross, and S. Prabhakar, “An introduction to biometric recognition,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 4–20, Jan. 2004.

[4] T. D. Ganchev, “Speaker recognition,” University of Patras, 2005.

[5] J. J. Wolf, “Efficient Acoustic Parameters for Speaker Recognition,” *J. Acoust. Soc. Am. - J ACOUST SOC AMER*, vol. 51, 1972.

[6] S. van Vuuren, *Comparison of Text-Independent Speaker Recognition Methods on Telephone Speech with Acoustic Mismatch*. .

[7] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, “Score Normalization for Text-Independent Speaker Verification Systems,” *Digit. Signal Process.*, vol. 10, no. 1–3, pp. 42–54, Jan. 2000.

[8] D. A. Reynolds and R. C. Rose, “Robust text-independent speaker identification using Gaussian mixture speaker models,” *IEEE Trans. Speech Audio Process.*, vol. 3, no. 1, pp. 72–83, Jan. 1995.

[9] D. Zhang, H. Guo, and B. Luo, “An algorithm for estimating number of components of Gaussian mixture model based on penalized distance,” in *2008 International Conference on Neural Networks and Signal Processing*, 2008, pp. 482–487.

[10] N. Vlassis and A. Likas, “A greedy EM algorithm for Gaussian mixture learning,” *Neural Process. Lett.*, vol. 15, no. 1, pp. 77–87, 2002.

[11] C. Xie, “Estimating the Number of Components in Gaussian Mixture Models Adaptively,” *J. Inf. Comput. Sci.*, vol. 10, no. 14, pp. 4453–4460, Sep. 2013.

[12] S. Bandyopadhyay and U. Maulik, “Nonparametric genetic clustering: comparison of validity indices,” *Syst. Man Cybern. Part C Appl. Rev. IEEE Trans. On*, vol. 31, no. 1, pp. 120–125, 2001.

[13] R. H. Sheikh, M. M. Raghuwanshi, and A. N. Jaiswal, “Genetic Algorithm Based Clustering: A Survey,” in *First International Conference on Emerging Trends in Engineering and Technology, 2008. ICETET ’08*, 2008, pp. 314–319.

[14] S. Bandyopadhyay and U. Maulik, “Genetic clustering for automatic evolution of clusters and application to image classification,” *Pattern Recognit.*, vol. 35, no. 6, pp. 1197–1208, Jun. 2002.

[15] A. LACHKAR, O. AMMOR, and N. RAIS, “Détermination du nombre de classes par le principe du maximum d’entropie pour des classes en chevauchement.”

[16] W. VII, “IDENTIFYING THE NUMBER OF CLUSTERS AND OBTAINING OPTIMAL CLASSIFICATION RESULT.”

[17] U. Maulik and S. Bandyopadhyay, “Performance evaluation of some clustering algorithms and validity indices,” *Pattern Anal. Mach. Intell. IEEE Trans. On*, vol. 24, no. 12, pp. 1650–1654, 2002.

[18] S. L. Horowitz and T. Pavlidis, “Picture segmentation by a directed split-and-merge procedure,” *Proc. Second Int. Jt. Conf. Pattern Recognit.*, vol. 424, p. 433, 1974.

[19] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, “SMEM algorithm for mixture models,” *Neural Comput.*, vol. 12, no. 9, pp. 2109–2128, Sep. 2000.

[20] Z. Zhang, C. Chen, J. Sun, and K. Luk Chan, “EM algorithms for Gaussian mixtures with split-and-merge operation,” *Pattern Recognit.*, vol. 36, no. 9, pp. 1973–1983, Sep. 2003.

[21] Y. Zhang and M. S. Scordilis, *Optimization of gmm training for speaker verification*. 2004.

[22] K. Blekas and I. E. Lagaris, “Split-merge Incremental Learning (SMILE) of Mixture Models,” in *Proceedings of the 17th International Conference on Artificial Neural Networks*, Berlin, Heidelberg, 2007, pp. 291–300.

[23] F. Kelly and N. Harte, “Training GMMs for speaker verification,” Jun. 2010.

[24] Y. Zhang, L. Chen, and X. Ran, “Online incremental EM training of GMM and its application to speech processing applications,” in *2010 IEEE 10th International Conference on Signal Processing (ICSP)*, 2010, pp. 1309–1312.

[25] Daniel A. Wagenaar, “FSMEM for MoG.” 2000.

[26] H. X. Wang, B. Luo, Q. B. Zhang, and S. Wei, “Estimation for the Number of Components in a Mixture Model Using Stepwise Split-and-merge EM Algorithm,” *Pattern Recogn Lett*, vol. 25, no. 16, pp. 1799–1809, Dec. 2004.

[27] S.-S. Cheng, H.-M. Wang, and H.-C. Fu, “A Model-Selection-Based Self-Splitting Gaussian Mixture Learning with Application to Speaker Identification,” *EURASIP J. Adv. Signal Process.*, vol. 2004, no. 17, p. 312192, Dec. 2004.

[28] Y. Li and L. Li, “A Novel Split and Merge EM Algorithm for Gaussian Mixture Model,” in *Fifth International Conference on Natural Computation, 2009. ICNC ’09*, 2009, vol. 6, pp. 479–483.

[29] G. Yin and D. Bruckner, “Gaussian Mixture Models and Split-Merge Algorithm for parameter analysis of tracked video objects,” in *Industrial Electronics, 2009. IECON’09. 35th Annual Conference of IEEE*, 2009, pp. 4155–4158.

[30] T. Ganchev, N. Fakotakis, and G. Kokkinakis, “Comparative evaluation of various MFCC implementations on the speaker verification task,” in *in Proc. of the SPECOM-2005*, 2005, pp. 191–194.