الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 20 août 1955 – Skikda-

وزارة التعليم العالي والبحث العلمي

جامعة 20 أوت 1955 سكيكدة

N° d'ordre : .....

Faculté des Sciences
Département d'Informatique

THESE

Présentée en vue de l'obtention du diplôme de
Doctorat en sciences

Spécialité : Informatique

CONTRIBUTIONS A LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

par

Mme Samira HAZMOUNE (Ep. BOUGAMOUZA)

Soutenue publiquement le 26 Janvier 2021 devant le jury composé de :

Président :	Mr Mohammed REDJIMI	Professeur à l'Université 20 août 1955-Skikda
Rapporteur :	Mr Mohamed BENMOHAMMED	Professeur à l'Université Abdelhamid Mehri- Constantine 2
Co-rapporteur :	Mr Smaine MAZOUZI	Professeur à l'Université 20 août 1955-Skikda
Examinateurs :	Mr Bachir BOUCHEHAM	Professeur à l'Université 20 août 1955-Skikda
	Mr Ramdane MAMRI	Professeur à l'Université Abdelhamid Mehri- Constantine 2
	Mr Said LABED	MCA à l'Université Abdelhamid Mehri- Constantine 2

2020/2021

À l'âme de mon père que je n'oublierai jamais.

اللهم اغفر له وارحمه وأدخله جنات النعيم

Remerciements

Je tiens à remercier chaleureusement Mr *Mohamed BENMOHAMMED*, Professeur à l'université Abdelhamid Mehri de constantine et Mr *Smaine MAZOUZI*, Professeur à l'université 20 août 1955 de Skikda, respectivement mon directeur et co-directeur de thèse, pour leur qualité d'encadrement. Je leur exprime ma reconnaissance pour leurs précieux avis et leurs conseils judicieux sur les directions à prendre ou à éviter dans ce travail.

Je tiens à remercier vivement Mr *Mohammed REDJIMI*, Professeur à l'université 20 août 1955 de Skikda qui m'a fait l'honneur de présider le jury de soutenance. Je remercie également Mr *Bachir BOUCHEHAM*, Professeur à l'université 20 août 1955 de Skikda, Mr *Ramdane MAMRI*, Professeur à l'université Abdelhamid Mehri de constantine, et Mr *Said LABED*, Maitre de conférences à l'université Abdelhamid Mehri de constantine, d'avoir accepté la lourde tâche d'examinateur de mon travail.

Un grand merci à ma grande famille, particulièrement à ma mère, à mes frères et sœurs pour leurs soutiens et encouragements.

Enfin, j'addresse mes très profonds remerciements à ma petite famille et plus précisément à mon cher mari pour sa compréhension, ses conseils et ses encouragements, et à mes adorables enfants qui ont dû souvent subir le coût de mes absences, qu'elles soient physiques ou d'attention.

مـلـخص

هذه الأطروحة تندرج في الإطار العام للتعرف الآلي على الكلام (RAP) الذي، على الرغم من تطوره المذهل على مدى العقد الماضي، لا يزال يجذب انتباه المجتمع العلمي، فتصميم نظام SRAP يتسم بالكفاءة والثبات في آن واحد يظل إشكالية. الهدف النهائي من هذا العمل هو اقتراح حلول لتحسين دقة SRAP وثباتها في مواجهة تغير البيانات، وخاصة في حالة التطبيقات محدودة المفردات.

ترتكز مساهمتنا في هذا السياق على نقطتين رئيسيتين: أولاً، نقترح مقاربة هجينة جديدة تعتمد على نمذجة متعددة بنماذج ماركوف المخفية (HMM). في هذه المقاربة، يتم دمج HMM في بنية k-NN على مستوى التمثيل وعلى مستوى التعرف. ويبقى الهدف هو تصميم مصنِّف يرث كلاً من ثبات k-NN وكفاءة خوارزميات HMM مع تجنب عيوب كل منهما. النقطة الثانية من مساهمتنا هي اقتراح مقاربة مجموعاتية جديدة والتي هي، مثل المقاربة الأولى، قائمة على نمذجة ماركوف المتعددة. تكمن الفكرة في إنشاء العديد من النماذج، لنفس فئة البيانات، انطلاقا من إعدادات أولية مختلفة. بعد ذلك يتم تجميع هذه النماذج في مصنفات ليتم دمجها لاحقا في مرحلة التعرف. بالإضافة إلى ذلك، قمنا بإعداد دراسة تجريبية لتأثير الإعدادات الأولية المختلفة لخوارزمية التعلم لماركوف على إنشاء مجموعات المصنفات، حيث نقوم بإجراء تحليل عميق للعلاقة بين كل إعداد ومقاييس التنوع المستخدمة بشكل شائع في أدبيات هذا المجال. حيث، وعلى حد علمنا، لم يتم التطرق لهذه المشكلة بهذه الطريقة من قبل. نهدف من خلال النمذجة المتعددة المقترحة، من ناحية، إلى تخفيف تأثير الإعدادات الأولية على النتائج، ومن ناحية أخرى، إلى تحسين الثبات مهما تغيرت البيانات.

يتم تقييم مساهماتنا باستخدام قاعدة الأرقام العربية المنطوقة " Spoken Arabic Digits ". تظهر نتائج المقارنة تفوق مقارباتنا المقترحة من حيث الأداء والثبات، من ناحية، على HMM و-k NNالأساسية، ومن ناحية أخرى، على أعمال السابقة.

يمكن تطبيق المقاربات المقترحة مباشرة في مجال الأوامر الصوتية (الطلبات الآلية الهاتفية على سبيل المثال) حيث تكون المفردات المحدودة كافية، أو تكييفها بسهولة مع الكلام المستمر بمفردات كثيرة باستخدام منهج تحليلي قائم على الصوتيات السياقية كوحدات نمذجة صوتية، والاستفادة من التجزئة الضمنية التي تقدمها نماذج HMM.

الكلمات المفتاحية: التعرف على الكلام، تغيرات البيانات، الثبات، HMM ، EM، النمذجة المتعددة، مجموعات المصنفات، التنوع، الأنظمة الهجينة، k-NN / HMM

Abstract

This thesis is part of the general framework of automatic speech recognition (ASR) which, despite its striking development over the past decade, continues to attract the attention of the scientific community. Thus, the design of a SRAP (RAP system) that is both accurate and robust remains an open issue. The ultimate objective of this work is to propose solutions to improve the performance of ASR systems and boost their robustness in the face of data variability, in particular for limited vocabulary application.

Our contribution, in this context, focuses on two main points: First, we propose a novel hybrid approach based on multiple modeling by hidden Markov models (HMM). In this approach, HMM are integrated into a k-NN architecture in both the representation and the recognition level. The aim is to design a classifier inheriting both the robustness of k-NN and the efficiency of HMM, while avoiding their respective drawbacks. The second point of our contribution is the proposal of an ensemble approach which, like the first approach, based on Markovian multiple modeling. The idea is to train, for the same data class, several models coming from different initial configurations. These models must then be grouped together into classifiers which will be combined during the recognition phase. In addition, we carry out an experimental study that aims to show the impact of the different initial parameters of Markovian learning on the creation of classifiers' ensembles, where we make a deep analysis of the relationship between each parameter and the diversity measures, commonly used in the literature. To the best of our knowledge, this problem has never been explored previously in the same way that we introduce in this work. Through the proposed multiple modeling we aim, on the one hand, to reduce the influence of the initial configuration of the training parameters on the results, and on the other hand, to improve the robustness against data variability.

Our contributions are evaluated using the standard dataset "Spoken Arabic Digits". The comparative results in terms of performance and robustness show the superiority of our approaches, on the one hand, over a basic HMM and k-NN, and on the other hand, over previous works in the literature.

The proposed approaches can be applied directly in the field of voice commands (a phone dialer for example) where, a limited vocabulary is sufficient. Also, they can easily be adapted to continuous speech with large vocabulary using, in this case, an analytical approach based on contextual phonemes as acoustic modeling units, and taking advantage of the implicit segmentation provided by HMM.

Keywords: speech recognition, data variability, robustness, HMM, EM, multiple modeling, ensembles of classifiers, diversity, hybrid systems, HMM / k-NN.

Resume

Cette thèse s'inscrit dans le cadre général de la reconnaissance automatique de la parole (RAP) qui, malgré son évolution frappante durant la dernière décennie, continue à attirer l'attention de la communauté scientifique, car la conception d'un SRAP (système de RAP), à la fois performant et robuste, reste toujours une problématique. L'ultime objectif de ce travail est de proposer des solutions pour améliorer les performances des SRAP, et de booster leur robustesse face à la variabilité des données, et ce, dans le cas particulier d'une application à vocabulaire limité.

Notre contribution, dans ce contexte, s'axe sur deux points principaux : En premier lieu, nous proposons une nouvelle approche hybride basée sur une modélisation multiple par les modèles de Markov cachés (HMM). Dans cette approche, les HMM sont intégrés au sein d'une architecture k-NN (k-Nearest Neighbors) au niveau représentation et au niveau reconnaissance. L'objectif est de concevoir un classifieur héritant à la fois de la robustesse du k-NN et de l'efficacité des HMM tout en écartant leurs inconvénients respectifs. Le second point de notre contribution est la proposition d'une approche ensembliste qui, comme la première approche, basée sur une modélisation markovienne multiple. L'idée est de faire apprendre, pour la même classe de données, plusieurs modèles, obtenus à partir de différentes configurations initiales. Ces modèles doivent ensuite être regroupés dans des classifieurs qui seront combinés durant la phase de reconnaissance. En plus, nous mettons en place une étude expérimentale visant à montrer l'impact des différents paramètres initiaux de l'apprentissage markovien sur la création des ensembles de classifieurs, où nous faisons une analyse profonde de la relation entre chaque paramètre et les mesures de diversité utilisées couramment dans la littérature. Ce problème, à notre connaissance, n'a jamais été exploré de la façon avec laquelle nous l'avons abordé. A travers la modélisation multiple proposée, nous visons, d'une part, à réduire l'influence de la configuration initiale des paramètres de l'apprentissage, et d'autre part, à améliorer la robustesse face à la variabilité des données.

Nos contributions sont évaluées en utilisant la base des chiffres arabes « Spoken Arabic Digits ». Les résultats comparatifs en termes de performances et de robustesse montrent la supériorité de nos approches, d'une part, sur un HMM et un k-NN de base, et d'autre part, sur des travaux précédents de la littérature.

Les approches proposées peuvent être appliquées directement dans le domaine de la commande vocale (un composeur téléphonique par exemple) où un vocabulaire limité est suffisant. Comme elles peuvent être adaptées facilement à la parole continue à grand vocabulaire en utilisant, dans ce cas, une approche analytique basée sur les phonèmes contextuels comme unités de modélisation acoustique, et en bénéficiant de la segmentation implicite, assurée par les HMM.

Mots clés : reconnaissance de la parole, variabilité de données, robustesse, HMM, EM, modélisation multiple, ensembles de classifieurs, diversité, systèmes hybrides, HMM/k-NN.

TABLE DES MATIERES

REMERCIEMENTS

ملخص

ABSTRACT

RÉSUMÉ

LISTE DES FIGURES

CHAPITRE 1. INTRODUCTION GENERALE

CHAPITRE 2. LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE : PRESENTATION GENERALE

CHAPITRE 3. LES APPROCHES DE CLASSIFICATION : UN ETAT DE L'ART

CHAPITRE 4. LES MODELES DE MARKOV CACHES ET LE PROBLEME DU REGLAGE INITIAL DES PARAMETRES DE L'ALGORITHME EM

CHAPITRE 5. UNE APPROCHE HYBRIDE HMM/K-NN POUR LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

LISTE DES TABLEAUX

CHAPITRE 3. LES APPROCHES DE CLASSIFICATION : UN ETAT DE L'ART

CHAPITRE 4. LES MODELES DE MARKOV CACHES ET LE PROBLEME DU REGLAGE INITIAL DES PARAMETRES DE L'ALGORITHME EM

CHAPITRE 5. UNE APPROCHE HYBRIDE HMM/K-NN POUR LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

CHAPITRE 6. UNE APPROCHE ENSEMBLISTE BASEE SUR UNE MODELISATION MARKOVIENNE MULTIPLE

LISTE DES ABRIVEATIONS ET SYMBOLES

RAP	Reconnaissance Automatique de la Parole
SRAP	Système de Reconnaissance Automatique de la Parole
HMM	Hidden Markov Models
k-NN	k-Nearest Neighbors
DTW	Dynamique Time Warping
ANN	Artificial Neural Network
NN	Neural Network
MLP	Multi-Layer Perceptron
DNN	Deep Neural Network
CNN	Convolutional Neural Network
RNN	Recurent Neural Network
LSTM	Long Short-Term Memory networks
SVM	Support Vector Machines
GMM	Gaussian Mixture Model
EM	Expectation-Maximisation
MFCC	Mel Frequency Cepstral Coefficients
FFT	Fast Fourier Transform
DCT	Discret Cosinus Transform
LPC	Linear Prediction Coefficients
LPCC	Linear Prediction Cepstral Coefficients
PLP	Perceptual Linear Prediction
RASTA-PLP	Relative Spectral Transform-Linear Prediction Coefficients
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
WER	Word Error Rate
SNR	Signal to Noise Rate
RSM	Random Subspace Method

ATP	Ambiguity Treatment by minimizing models' Positions
ATL	Ambiguity Treatment by maximizing models' Likelihoods
Q	*Q-statistic*
ρ	*Correlation*
D	Disagreement
DF	Double Fault
Ent	Entropy of the votes
θ	Difficulty Index
kw	Kohavi-Wolpert variance
k	Interrater Agreement
GD	Generalized Diversity
CFD	Coincident Failure Diversity
DIC	Discriminative Information Creterion
BIC	Bayesian Information Criterion
AIC	Akaike Information Criterion
CV	Coefficient de Variation

CHAPITRE 1

INTRODUCTION GENERALE

SOMMAIRE

1.1. CONTEXTE

La parole est le mode le plus rapide et le plus naturel pour la communication entre humains. Ce fait a motivé les chercheurs à penser à la parole comme un moyen rapide et efficace pour l'interaction Homme/Machine, ce qui a donné naissance à ce qu'on appelle la reconnaissance automatique de la parole (RAP). Il s'agit d'un des domaines les plus actifs de la reconnaissance des formes. Sa finalité est de transformer automatiquement un signal acoustique de la parole en texte ou en action. L'avantage de la RAP est de faciliter l'interaction Homme/Machine, et de soulager l'homme des tâches ennuyeuses, telles que la saisie par le clavier et la traduction classique en une langue étrangère. La RAP peut se retrouver dans plusieurs applications, les plus marquantes sont : la dictée, la compréhension automatique et la commande vocale.

La conception d'un système de RAP nécessite la précision de certaines caractéristiques permettant de mesurer la capacité du système à reconnaitre la parole, et sa complexité de développent. Selon le domaine d'application, un système peut se caractériser par un vocabulaire ouvert ou limité, un discours continue ou isolé, et il peut être dédié à un seul utilisateur (mono-locuteur), un nombre restreint d'utilisateurs (multi-locuteurs) ou à n'importe qui (indépendant du locuteur). Selon l'unité de modélisation acoustique utilisée, une distinction est faite entre l'approche globale et l'approche analytique. L'approche globale est basée sur le mot comme unité de base, le mot est donc considéré dans sa globalité sans essayer de le décomposer en unités élémentaires, telles que les phonèmes ou les syllabes. Dans ce cas, chaque mot est modélisé par un modèle différent. L'approche analytique quant à elle, se base sur les phonèmes comme unité de modélisation. La reconnaissance d'un mot revient alors à reconnaitre les phonèmes le constituant. L'approche globale est généralement utilisée pour les applications dont le vocabulaire est limité, tels que la commande vocale. Son avantage est qu'elle garde le phonème dans son contexte de voisinage sans avoir besoin de segmenter le mot en unités élémentaires, ce qui permet d'éviter les erreurs de segmentation et par conséquent, d'améliorer le résultat de la reconnaissance.

Cette thèse s'insère dans le cadre de la RAP en général, et des systèmes indépendants du locuteur, à vocabulaire limité, en particulier. Elle peut trouver ses applications dans le domaine de la commande vocale, telles qu'un composeur téléphonique, une chaise roulante, contrôle vocal d'un robot, etc. Dans ce type de systèmes, un vocabulaire restreint est suffisant et parfois même nécessaire pour des fins ergonomiques. En effet, les systèmes de commande vocale sont souvent dédiés aux personnes ayant des déficiences physiques. L'utilisation d'un vocabulaire limité dans ce cas est indispensable pour deux raisons principales. Premièrement, de telles personnes ont souvent des aptitudes limitées (ex, personnes âgées). Ce n'est donc pas commode de leur demander d'apprendre une longue liste de commandes. La deuxième raison, est que ce genre de systèmes est le plus souvent critique. Cette contrainte exige des performances très élevées, et il est intuitif que l'utilisation d'un vocabulaire limité (moins d'erreurs d'ambigüité) avec une approche de modélisation globale (moins d'erreurs de segmentation) permet d'améliorer les performances d'un tel système. De nombreux travaux dédiés à la reconnaissance de mots

isolés à vocabulaire limité sont récemment publiés, notamment dans le cas des langues peu dotées telles que l'Arabe. Nous citons à titre d'exemples, les travaux (Wazir et al., 2019), (Guerid, et al., 2018), (Bharali & Kalita, 2015), (Debnath & Roy, 2019), (Karthikeyan et al., 2019), (Kacur & Urbancikova, 2018), (Touazi & Debyeche, 2017), (Guerid & Houacine, 2019).

1.2. PROBLEMATIQUE ET OBJECTIF

Depuis la fin des années cinquante, plusieurs recherches en RAP ont été menées tant au niveau analyse acoustique qu'au niveau classification, dans le but est principalement l'amélioration des performances et la robustesse des systèmes pour se rapprocher le plus possible d'un système idéal. Malheureusement, malgré les grands progrès accomplis dans le domaine, nous sommes encore loin d'avoir une interaction naturelle entre l'homme et la machine, et de nombreux problèmes ne sont résolus que partiellement. Ces problèmes que les principales applications de la RAP en souffrent, peuvent être classés en deux grandes familles : ceux liés aux caractéristiques du signal de la parole et aux conditions de prise de mesures (tels que la variabilité, la redondance, les effets de coarticulation, le bruit ambiant, etc.), et ceux liés aux techniques d'analyse et de classification utilisées (tels que la sensibilité des classifieurs à la configuration initiale des paramètres d'apprentissage).

Parmi les problèmes de la première famille, et qui affectent considérablement la qualité des systèmes de reconnaissance en termes de performance et de robustesse, étant la variabilité des données intra et inter-locuteur. Cette variabilité provient de sources, liées principalement à l'état du locuteur (physiologique, psychologique, social ou encore culturel) et à son environnement (bruit, perturbations, matériel et conditions de saisie). Le degré de la variabilité varie essentiellement selon l'indépendance ou non du locuteur. En effet, un système mono-locuteur est moins sensible à la variabilité de données, comparé à un système multi-locuteurs qui est, à son tour, moins sensible qu'un système indépendant du locuteur, où, n'importe qui peut l'utiliser.

Au-delà de la variabilité du signal de la parole, un autre problème lié, cette fois-ci, aux techniques de classification, étant le problème de sensibilité des classifieurs à la configuration initiale des paramètres de l'algorithme d'apprentissage. Ceci est essentiellement dû à l'incohérence entre les données d'apprentissage et des données de test et à la variabilité intra-classe. La majorité des méthodes de classification proposées dans la littérature, telles que les réseaux de neurones artificiels, les machines à vecteurs support et les modèles de Markov cachés, dont nous préférerons employer l'acronyme anglais HMM pour Hidden Markov Models, sont très affectées par ce problème.

Les HMM, qui nous intéressent dans le cadre de cette thèse, sont l'une des méthodes les plus efficaces et les plus populaires en reconnaissance de la parole. Ils sont devenus largement acceptés comme technique standard de reconnaissance de la parole dans la communauté de RAP (Mustafa et al., 2019). Les raisons pour lesquelles cette méthode est devenue si populaire sont la disponibilité d'algorithmes d'apprentissage efficaces pour

estimer les paramètres des modèles à partir d'ensembles finis de données vocales (Rabiner et Juang 1992), leur base mathématique solide et leur capacité à modéliser des séries temporelles de longueurs variables. L'apprentissage des HMM est généralement basé sur la maximisation des fonctions objectives (vraisemblance, vraisemblance conditionnelle, etc.) en utilisant l'algorithme de gradient (Levinson et al., 1983) ou celui d'Expectation-Maximisation EM (Dempster et al., 1977). Ce dernier est une procédure itérative qui permet de réestimer les paramètres du modèle en fonction de leurs estimations actuelles de manière à améliorer la vraisemblance de données d'apprentissage après chaque itération. Le problème est que cet algorithme est trop sensible aux valeurs des paramètres initiaux qui doivent être réglés avec prudence. Aussi, il ne garantit pas une solution globalement optimale, car les fonctions objectives sont de nature non convexe et donc sujettes au problème des maxima locaux. La recherche exhaustive est par ailleurs impossible, car le nombre de maxima locaux est inconnu. La configuration des paramètres initiaux est généralement optimisée expérimentalement, et elle dépend fortement des données d'apprentissage et de test utilisées, ce qui affecte la robustesse et la stabilité du classifieur. La question qui se pose est donc, quelle configuration initiale ou quel modèle final doit on choisir afin de se rapprocher le plus possible du maximum global ? Pour aborder ce problème, certains travaux de la littérature se sont concentrés sur l'optimisation de la configuration initiale de façon expérimentale ou en faisant appel à des techniques d'optimisation combinatoire telles que les algorithmes génétiques. D'autres se sont focalisés plutôt sur l'optimisation du modèle final en se basant, généralement, sur les critères de sélection de modèles. Dans les deux cas, le modèle HMM appris est unique. Ce modèle ne pourra pas modéliser efficacement une classe de données vu la grande variabilité intra-classe, notamment dans les systèmes indépendants du locuteur.

L'objectif de la présente thèse est de proposer de nouvelles approches de classification permettant de concevoir des systèmes de RAP (SRAP) à la fois performants et robustes face à la variabilité des données en se basant, principalement, sur les HMM comme classifieurs de base, et en essayant d'alléger les problèmes de la variabilité intra-classe et de la sensibilité des HMM au réglage initial des paramètres de l'algorithme d'apprentissage.

1.3. CONTRIBUTIONS

Nous nous sommes attaqués à la problématique de sensibilité des classifieurs markoviens au choix des paramètres initiaux de l'apprentissage, et de l'incohérence entre les données d'apprentissage et les données de test, due à la variabilité intra-classe, avec deux approches différentes ayant, en commun, une première étape basée sur une modélisation markovienne multiple partant de différentes configurations initiales.

Les contributions majeures de cette thèse peuvent être résumées dans les points suivants :

- Une approche hybride basée sur une modélisation multiple (Hazmoune et al., 2018), où les HMM sont intégrés au sein d'une architecture k-NN au niveau représentation et au niveau reconnaissance. Ceci permet de faire rejoindre la souplesse et l'efficacité de la modélisation acoustique des HMM, et la robustesse du k-NN.

- Une approche ensembliste homogène qui combine un certain nombre de classifieurs markoviens (Hazmoune et al., 2013a ; Hazmoune et al., 2013b). La différence entre les classifieurs de base est créée en faisant varier les valeurs de l'un des paramètres de la configuration initiale de l'apprentissage, tels que le nombre d'états, le modèle initial de l'algorithme EM, le nombre de densités gaussiennes associées aux états, et le nombre d'itérations de l'algorithme d'apprentissage.

- Une étude expérimentale profonde de l'impact des paramètres de la configuration initiale de l'algorithme d'apprentissage des HMM sur les mesures de diversité les plus utilisées, dans la littérature, pour évaluer la qualité de l'ensemble des classifieurs. Le but principal de cette étude est de choisir parmi ces mesures les plus adaptées à notre approche ensembliste.

Le principal avantage des deux approches est qu'elles permettent d'améliorer les performances et la robustesse face à la variabilité de données, et de contourner le délicat problème d'optimisation de la configuration initiale de l'algorithme d'apprentissage markovien, et ce, grâce à la modélisation multiple proposée.

1.4. ORGANISATION DU DOCUMENT

Outre cette introduction générale et la conclusion générale, ce document est organisé en 5 chapitres regroupés en deux parties (Fig.1.1). Dans la 1ière partie, nous présentons le domaine de la reconnaissance automatique de la parole et les approches de classification couramment utilisées dans la littérature, notamment, celles basées sur les modèles de Markov cachés. La seconde partie est consacrée à notre contribution à l'amélioration des performances et la robustesse des SRAP.

Partie I. Cadre théorique et état de l'art

- Le chapitre 2, a pour objet l'introduction du domaine de la RAP en général. Certaines notions élémentaires et techniques de base utilisées couramment pour le traitement et l'analyse du signal vocal sont également abordées.

- Le troisième chapitre est consacré à l'état de l'art des approches de classification de la RAP, où nous mettons l'accent sur les approches ensemblistes et les approches hybrides.

- Dans le chapitre 4, nous nous intéressons aux modèles de Markov cachés qui sont à la base des recherches menées dans le cadre de cette thèse. Nous consacrons une bonne partie du chapitre, au problème de la sensibilité du classifieur

markovien à la configuration initiale des paramètres d'apprentissage, tout en exposant une revue bibliographique des travaux utilisant les HMM pour la RAP.

Partie II. Contributions à la reconnaissance automatique de la parole

- Le chapitre 5 porte sur la description et l'évaluation de notre première contribution qui consiste en une approche hybride intégrant les HMMs dans une architecture k-NN.
- Le sixième chapitre est consacré à notre deuxième contribution et qui consiste en la proposition d'une approche ensembliste combinant plusieurs classifieurs markoviens.

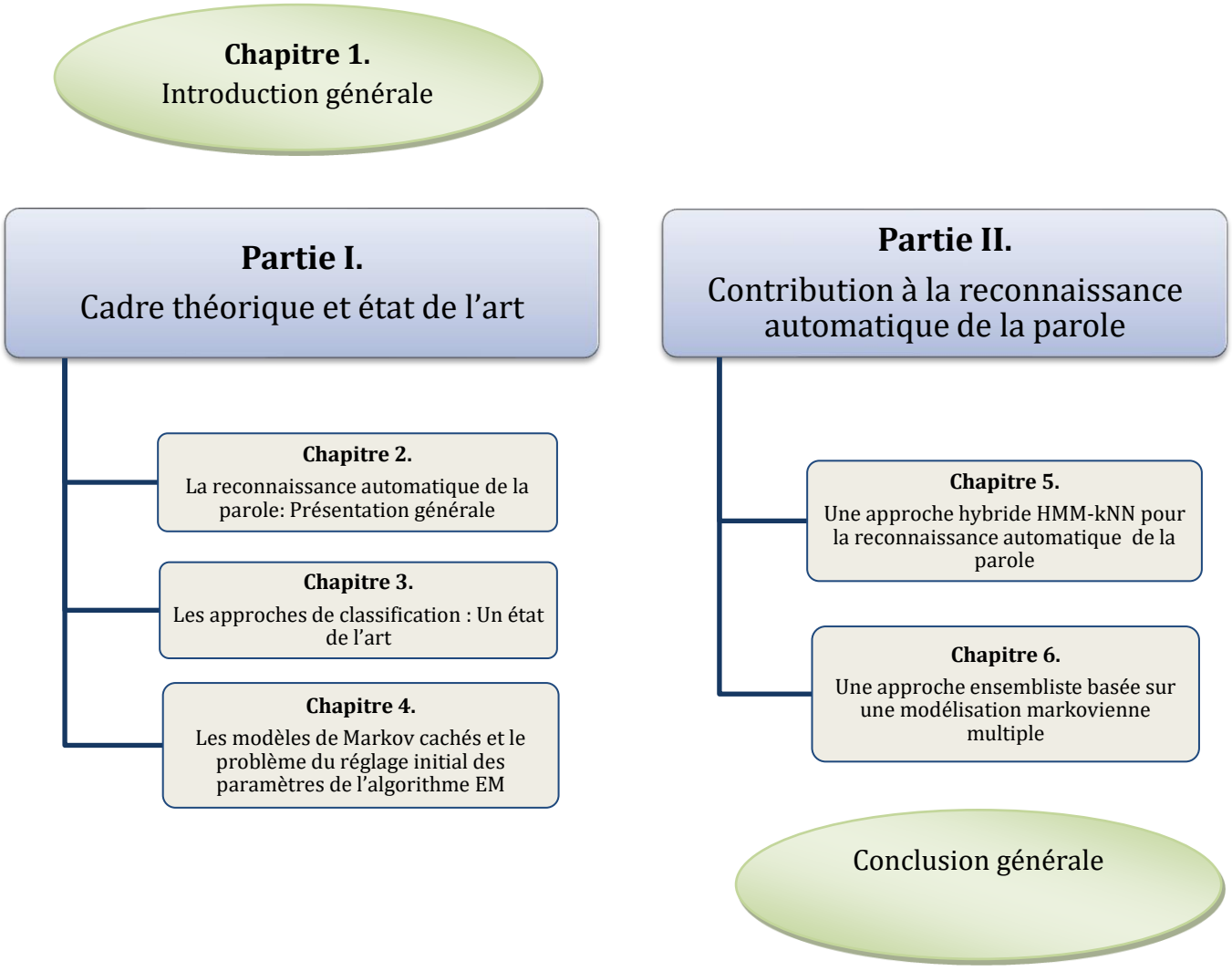


Fig.1.1. Organisation du document

" There is nothing more practical than a good theory"

Kurt Lewin

Partie I.

Cadre théorique et état de l'art

Chapitre 2.

La reconnaissance automatique de la parole: Présentation générale

Chapitre 3.

Les approches de classification : Un état de l'art

Chapitre 4.

Les modèles de Markov cachés et le problème du réglage initial des paramètres de l'algorithme EM

CHAPITRE 2

LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE : PRESENTATION GENERALE

SOMMAIRE

2.1. INTRODUCTION

Grâce au progrès des méthodes et techniques de reconnaissance et grâce à l'évolution rapide des processeurs, la RAP a connu un grand progrès ces dernières années, tant au niveau performance que complexité. En effet, les SRAP sont actuellement intégrés dans différents domaines de la vie, dont nous citons à titre d'exemples :

- L'aide aux personnes ayant des déficiences physiques : Handicapés (chaise roulante), aveugles (gestion d'un menu dans une application), malentendants (transcription de la parole de leur interlocuteur en un texte écrit), personnes âgées (assistant vocal).
- La commande vocale des robots, le contrôle des objets domotiques et l'indexation de documents multimédias.
- La dictée automatique (ordinateur sans clavier ni souris), la transcription automatique de la parole (sous-titrage), et la traduction automatique des conversations téléphoniques avec un interlocuteur de langue étrangère.

Au cours de ce chapitre, qui est une présentation générale du domaine de la RAP, nous essayons d'introduire les éléments nécessaires pour la conception d'un SRAP. Nous allons tout d'abord, présenter les caractéristiques inhérentes au signal de la parole et montrer pourquoi ces caractéristiques, notamment la variabilité intra et inter-locuteur, peuvent influencer les performances, la robustesse et la complexité de développement des SRAP. Puis, nous donnerons une catégorisation des SRAP selon plusieurs critères, et nous discuterons les facteurs de complexité liés à chaque catégorie de système. Ensuite, nous présenterons les différentes étapes du processus général de la RAP, ainsi que les techniques utilisées dans chacune d'elles. Pour finir, les différents aspects à prendre en compte durant l'évaluation de la qualité d'un SRAP seront abordés, en mettant l'accent sur l'aspect robustesse, où nous exposerons les principales solutions proposées dans la littérature pour améliorer cet aspect.

2.2. LA PAROLE ET SES CARACTERISTIQUES

La parole est un mode d'expression de la langue. C'est le moyen le plus rapide et le plus naturel pour la communication entre humains. En termes fonctionnels (Clairet, 2004), la production de la parole implique l'utilisation des systèmes respiratoire et le larynx qui fonctionnent ensemble pour générer de la parole. L'air expiré des poumons est transformé en vibrations audibles par les différents organes du conduit vocal. La source de vibration la plus importante est la partie inférieure du conduit vocal, le larynx, dans lequel les cordes vocales sont logées, dont les mouvements d'ouverture et de fermeture dépendent des vibrations de l'air. Dès que le flux d'air traverse le larynx, il pénètre dans les cavités supra glottiques, où il est affecté par l'action de plusieurs articulateurs, tels que la langue, le palais, les lèvres et la mâchoire.

Le signal de la parole produit se caractérise par plusieurs propriétés inhérentes, qui doivent être prises en considération durant l'analyse et le traitement automatique de la parole, car comme nous le montrerons ultérieurement, ce sont des facteurs de complexité qui influent grandement les performances et surtout la robustesse des SRAP. Nous allons découvrir, dans les paragraphes qui suivent, les plus importantes, à savoir : la redondance, les effets de coarticulation et la variabilité intra et inter-locuteur.

2.2.1. La redondance

La représentation des signaux acoustiques numérisés, dans le domaine temporel, est caractérisée par des informations redondantes, qui ne sont nécessairement pas utiles à la reconnaissance correcte du message lexical. En plus du message lui-même, la communication orale contient également de nombreuses autres informations paralinguistiques, telles que le genre du locuteur, son identité, son état émotionnel, son état de santé, etc. Pour un SRAP, une énorme quantité de données peut être exploitée à partir de ce flux d'informations. Par exemple, un signal échantillonné à 16 kHz sur 16 bits (paramètres qui sont généralement utilisés pour numériser un son) représente un débit de 256 KBits/s. Cela signifie que le SRAP doit gérer 32000 octets de données par seconde. Pour des fins de rapidité d'exécution, tout SRAP cherchera alors à minimiser cet important flux de données en recourant à une étape de prétraitement du signal, afin d'éliminer les informations redondantes et inutiles pour reconnaitre un message (Lauri, 2004).

2.2.2. Les effets de coarticulation

La parole est un processus séquentiel, dans lequel les unités de base se succèdent. En effet, tout message parlé peut être décomposé en une série de mots, qui à leur tour peuvent être décrits comme une séquence d'unités acoustiques plus petites. Cependant, même si certains événements acoustiques spécifiques peuvent être détectés, il est parfois difficile, même pour les phonéticiens, d'identifier individuellement les sons caractéristiques du langage dans un signal vocal. La parole est en fait un continuum sonore, dans lequel il n'y a pas de pause évidente entre les mots qui pourrait faciliter leur localisation automatique par un SRAP. De plus, durant la production d'un message, l'inertie de l'appareil phonatoire et l'anticipation du geste articulatoire influencent la production de chaque son, de sorte que la réalisation acoustique d'un son est fortement perturbée par les sons qui le précèdent mais aussi par ceux qui le suivent. Ces effets s'étendent sur la durée d'une syllabe ou même au-delà, et sont renforcés par un rythme de parole soutenu (LAURI, 2004). Évidemment, dans un système de reconnaissance de mots isolés, dans lequel les mots sont séparés par un silence, ces effets de coarticulation sont beaucoup moins importants que dans un système de parole continue où les mots s'enchainent sans pause.

2.2.3. La variabilité intra et inter-locuteur

Le signal de la parole se caractérise par une grande variabilité. En effet, le même locuteur ne prononce jamais la même élocution de la même façon, même dans des conditions identiques (cf, Fig.2.1). C'est ce qu'on appelle "variabilité intra-locuteur". Plusieurs

sources peuvent causer ce genre de variabilité, telles que, l'état physique (rhume, grippe, fatigue, etc.), psychologique et émotionnel (stress, peur, colère, etc.) du locuteur, ainsi la voix est affectée par d'autres facteurs liés à l'environnement. Par exemple, on a tendance à hausser sa voix quand on parle dans un environnement bruité, ce qui change les caractéristiques spectrales du signal acoustique.

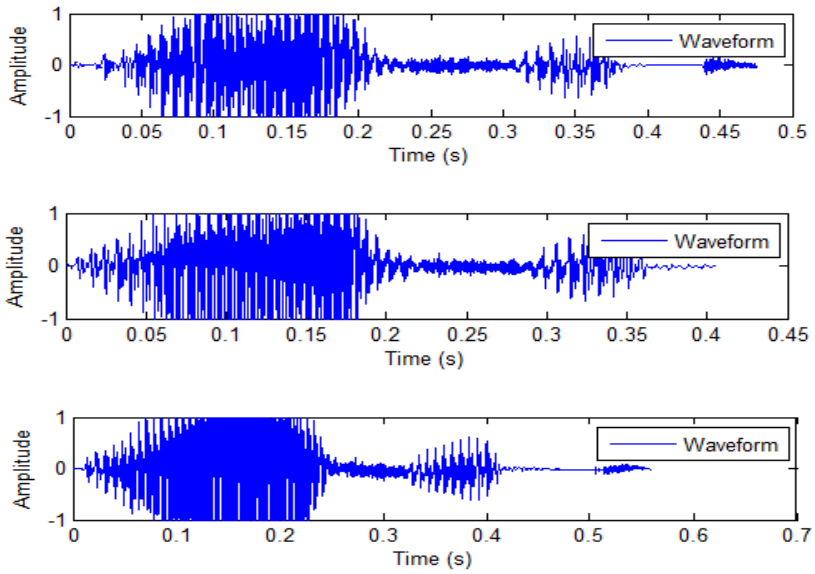


Fig.2.1. Forme d'ondes (waveform) du chiffre arabe « 1 » prononcé trois fois par le même locuteur dans les mêmes conditions d'enregistrement.

Il existe aussi la variabilité inter-locuteur, qui est la variation de prononciation d'un locuteur à un autre. Elle est principalement liée aux caractéristiques physiologiques (longueur du conduit vocal, genre, âge, etc.) ou sociales (niveau culturel, accent régional, etc.).

Le genre du locuteur (Speaker Gender) est une source importante de variabilité dans les systèmes de reconnaissance multi-locuteurs et indépendants du locuteur, car il existe une grande différence entre les productions des locuteurs féminins et masculins. Cette différence à des conséquences importantes sur la qualité de la reconnaissance. La figure 2.2 montre les formes d'ondes (représentation temporelle) de deux occurrences du chiffre arabe « 1 » prononcées par deux locuteurs différents (femme & homme). On remarque une différence hautement significative en amplitude et en durée entre les deux occurrences.

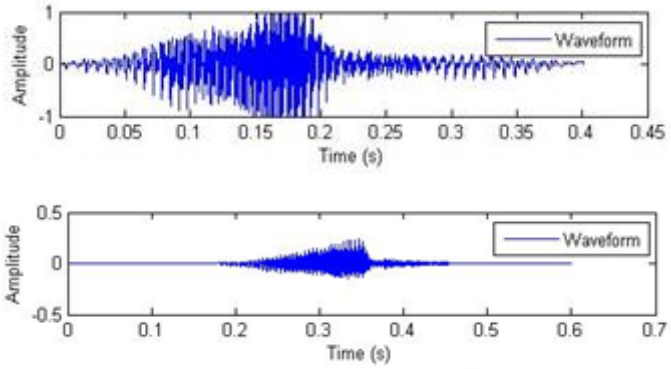


Fig.2.2. Forme d'ondes (waveform) du chiffres arabe « 1 » prononcé par deux locuteurs différents dans les mêmes conditions d'enregistrement (haut : male, bas : femelle).

Outre la durée et l'amplitude du signal, les différences acoustiques les plus significatives entre les productions des locuteurs masculins et féminins, selon (Pépiot, 2013) peuvent être résumées dans les points suivants :

- La fréquence fondamentale (Pitch ou F_0) : C'est le nombre de fois que la corde vocale vibre dans une seconde. F_0 est communément considérée comme la différence la plus importante entre les deux types de voix. Elle serait de l'ordre de 200 Hz chez les femmes, et de 120 Hz chez les locuteurs masculins. Cela justifie l'utilisation de F_0 comme caractéristique discriminante et robuste dans les systèmes d'identification automatique du genre du locuteur. Néanmoins, comme il a été montré par Pépiot (Pépio, 2013), ces variations peuvent aussi être dues aux comportements sociaux et pas uniquement dues aux différences physiologiques des locuteurs.

- Les fréquences secondaires (Formants vocaliques) : On désigne par formant l'un des maxima d'énergie du spectre sonore du son de parole. Les formants des voyelles produites par les locuteurs masculins tendent à être situés dans des fréquences globalement plus basses que celles des voyelles prononcées par les locutrices.

- La durée des voyelles : Les voyelles produites par les locuteurs masculins sont significativement plus courtes que celles produites par les locutrices.

Par ailleurs, l'âge du locuteur est une autre source de variabilité. En effet, il y a une grande différence entre les productions d'un adulte et celles d'un enfant, la fréquence fondamentale par exemple est plus élevée chez les enfants que celles des locuteurs adultes.

Pour finir, la variabilité intra et inte-rlocuteur rend la tâche de reconnaissance plus compliquée, notamment dans le cas des systèmes multi et indépendants du locuteur. Ainsi, afin d'améliorer les résultats des SRAP en termes de performance et de robustesse

vis-à-vis de la variabilité des données, les informations caractérisant le message lexical doivent être séparées de celles caractérisant le locuteur.

2.3. LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

A partir du signal de la parole, des informations de différentes natures peuvent être extraites, telles que le message qu'on veut communiquer, la langue parlée, le genre, l'identité et l'émotion du locuteur (l'état psychologique). Selon le but visé, on peut diviser le domaine de traitement automatique de la parole en cinq axes qui sont illustrées dans la figure 2.3. La RAP, l'un de ces axes, permet de reproduire le comportement audible et compréhensible de l'homme. Sa finalité est de transformer un signal de la parole acquis au moyen d'un microphone en un texte (information lexicale) ou en action (cf., Fig.2.4).

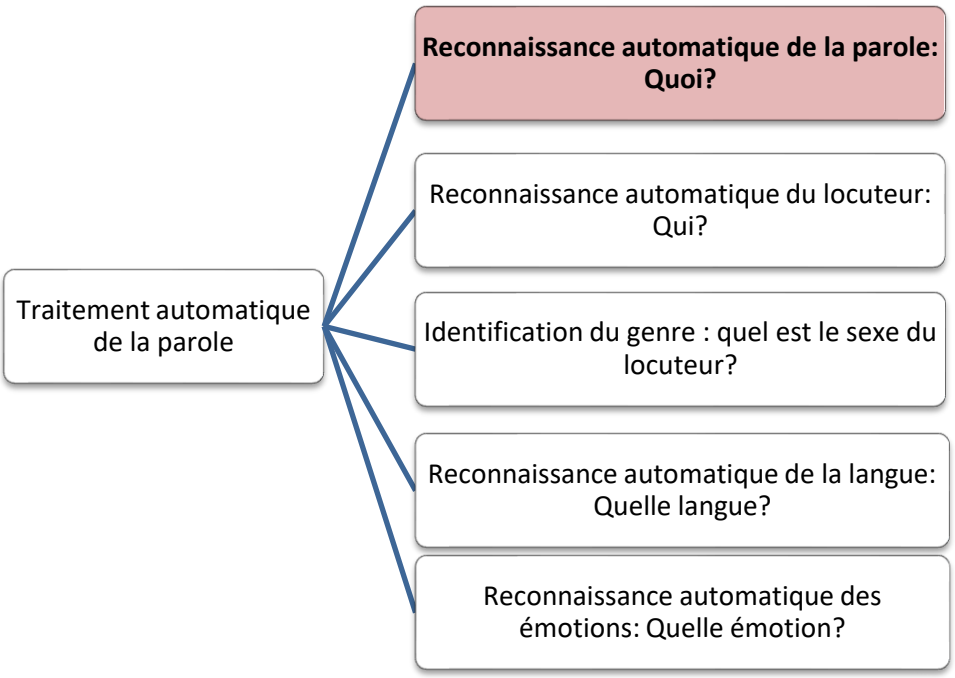


Fig.2.3. Les différents axes du traitement automatique de la parole

Fig.2.4. Finalité d'un SRAP

De nombreux facteurs, pouvant significativement affecter les performances et la robustesse des SRAP et rendre la tâche de reconnaissance très difficile, ont été marqués. Ces facteurs peuvent être classés en trois groupes :

- Facteurs liés aux caractéristiques inhérentes à la parole, telles que la redondance, la variabilité et les effets de coarticulation.
- Facteurs liés à l'environnement, tels que les perturbations qui troublent l'acquisition du signal de la parole, comme le bruit additif (dans la rue, dans une voiture, etc.), les distorsions et les bruits de lèvres ou de respiration.
- Facteurs liés au matériel d'acquisition, par exemple, une mauvaise qualité du capteur, un mauvais réglage des paramètres de numérisation (fréquence d'échantillonnage et nombre de bits de quantification) ou le positionnement du microphone par rapport à la bouche du locuteur.

Outre ces facteurs affectant la robustesse des SRAP, ainsi que la complexité de leur développement, il existe d'autres facteurs qui rendent la conception d'un SRAP une tâche extrêmement ardue. Nous revenons sur ces derniers dans les sections qui suivent du présent chapitre.

2.3.1. Catégorisation des systèmes de RAP

Les SRAP peuvent être catégorisés selon plusieurs critères que nous résumons dans le schéma présenté dans la figure 2.5. Les rectangles correspondent aux systèmes et les ellipses aux critères de classification.

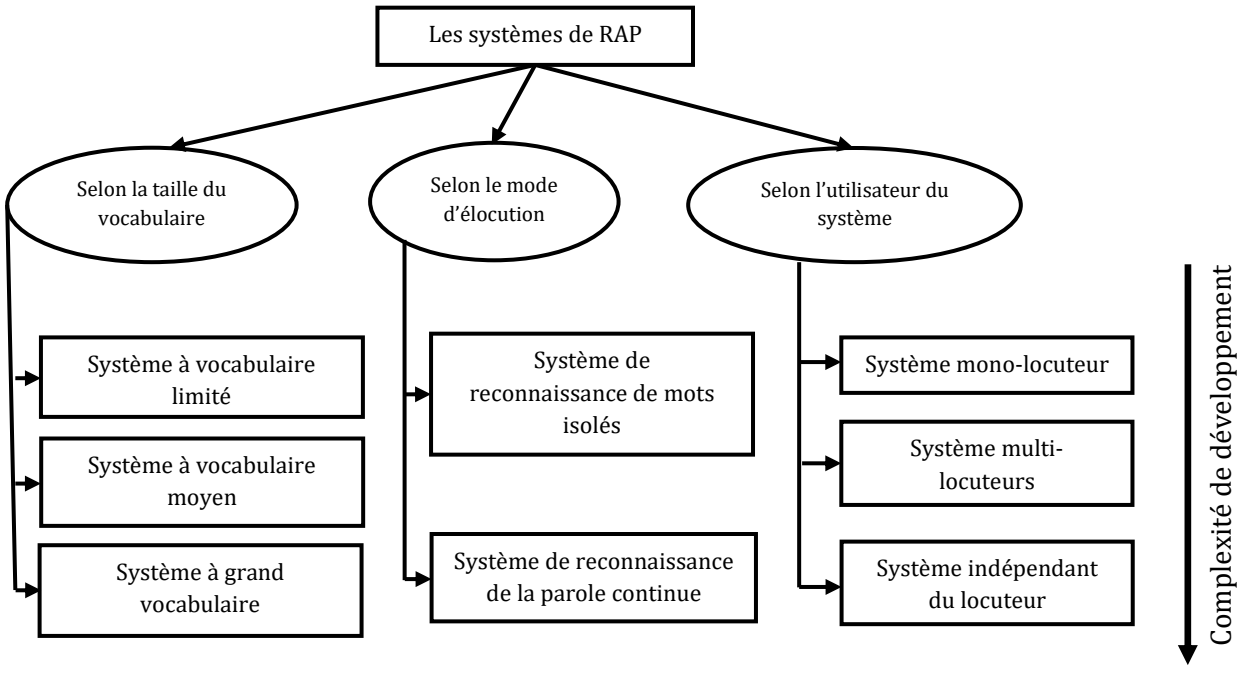


Fig.2.5. Catégorisation des systèmes de RAP

2.3.1.1. Selon la taille du vocabulaire

Les SRAP peuvent être partitionnés, selon la taille du vocabulaire considéré, en trois catégories : systèmes à vocabulaire restreint, systèmes à vocabulaire moyen et systèmes à grand vocabulaire.

Les systèmes à vocabulaire restreint, sont utilisables surtout dans les applications de commande vocale qui, par leur nature, requièrent un nombre limité de commandes. Parmi ces systèmes nous citons le contrôle d'un robot par la voix, le composeur téléphonique, l'aide aux handicapés (contrôle d'une chaise roulante), l'aide aux non-voyants (gestion d'un menu dans une application), etc. Ce genre de systèmes qui, dans la plupart des cas, sont basés sur une approche de modélisation globale, se caractérisent par un taux d'erreur relativement faible, car plus la taille du vocabulaire est réduite, plus le nombre de confusions entre mots est réduit.

Un SRAP à grand vocabulaire, nécessite une approche de modélisation analytique, c.à.d. les modèles générés sont des modèles de sous-mots (phonèmes ou syllabes), car, comme nous le montrerons plus loin dans ce chapitre, il n'est plus possible de modéliser chaque mot du vocabulaire par un modèle différent à cause des contraintes de calcul et de stockage. Le taux d'erreur des systèmes à grand vocabulaire est généralement plus important que celle des systèmes à vocabulaire moyen ou restreint. Ceci est justifié, d'une part, par le fait que les modèles des sous-mots ne peuvent pas capturer les effets de coarticulation, et d'autre part, par le fait que le taux de confusion entre mots augmente avec l'augmentation de la taille du vocabulaire. Les systèmes à grand vocabulaire sont nécessaires dans de nombreuses applications, telles que la dictée automatique, les systèmes de dialogue et les systèmes de traduction vocale.

2.3.1.2. Selon le mode d'élocution

Selon ce critère, un SRAP peut être l'un des deux types suivants : Un système de reconnaissance de *mots isolés* ou un système de reconnaissance de la *parole continue*. Dans le premier type, les mots sont séparés par un silence, tandis que dans le deuxième, les mots sont connectés sans pause. Bien évidemment, la reconnaissance de la parole continue est beaucoup plus difficile que la reconnaissance de mots isolés. Cette complexité est le résultat naturel de deux propriétés de la parole continue. Premièrement, les effets de coarticulation sont beaucoup plus forts dans la parole continue, faisant apparaître le même son différemment dans divers contextes. Deuxièmement, les frontières entre mots sont difficiles à localiser. De plus, la reconnaissance de la parole continue nécessite la segmentation de la parole en unités de base avant de faire la reconnaissance, ce qui conduit à des erreurs de segmentation influant ainsi le taux d'erreur global du système.

2.3.1.3. Selon l'utilisateur du système

Un SRAP peut être *mono-locuteur*, *multi-locuteurs* ou *indépendant* du locuteur. Un système mono-locuteur est destiné à être utilisé par un seul locuteur, son apprentissage est donc réalisé chez l'utilisateur sur des échantillons enregistrés par lui-même. Un système multi-

locuteurs est conçu pour être capable de reconnaitre la parole d'un nombre restreint de locuteurs, qui doivent participer à la réalisation de la base d'apprentissage. Un système indépendant du locuteur est destiné à être utilisé par n'importe quel locuteur et est naturellement plus difficile à réaliser. Ceci est logique, car, d'une part, les utilisateurs du système ne peuvent pas tous participer à l'enregistrement de la base d'apprentissage, et d'autre part, la variabilité de données est beaucoup plus importante que celle des systèmes *dépendants* du locuteur (mono et multi-locuteurs). De plus, la plupart des représentations paramétriques de la parole dépendent fortement du locuteur et les modèles de référence adaptés à un locuteur peuvent mal fonctionner pour un autre (Lee et al., 1990). En effet, selon une étude (Tebelskis, 1995), les systèmes indépendants du locuteur, ont tendance à avoir des taux d'erreur 3 à 5 fois supérieurs à ceux des systèmes dépendants du locuteur.

Nous abordons, dans la section suivante, les différentes étapes nécessaires pour l'identification du message lexical véhiculé à partir de la représentation temporelle d'un signal vocal.

2.3.2. Le processus général de la RAP

Bien que les SRAP se diffèrent souvent selon l'application et les méthodes d'analyse et de classification utilisées, ils partagent tous un minimum d'étapes obligatoires qui s'enchainent pour transformer un signal de parole en texte ou en action. La figure 2.6 schématise globalement le processus général de la RAP.

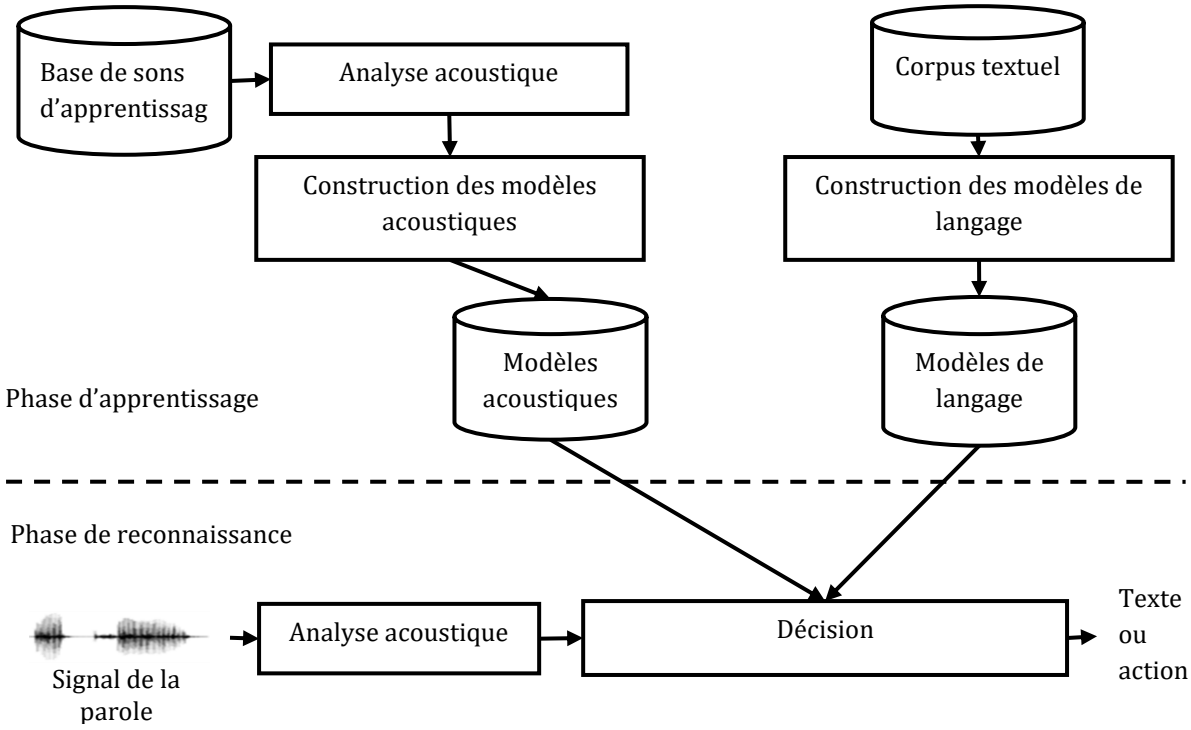


Fig.2.6. Le processus général de la reconnaissance automatique de la parole

Nous nous sommes contentés dans cette présentation du décodage lexical du signal dans un cadre statistique. Cependant, si on considère aussi la compréhension de la parole (ex, traduction automatique ou sous-titrage), d'autres composants doivent être ajoutés au système tels que, le module sémantique qui permet d'offrir des connaissances sur le sens des mots afin d'éviter par exemple, la confusion entre des mots différents mais qui se prononcent de la même façon (homophones), en favorisant l'un ou l'autre selon le contexte (ex. voix et voie, vers et verre).

Par la suite, nous allons présenter en détails chaque étape du processus de reconnaissance.

2.3.2.1. Analyse acoustique de la parole

L'analyse acoustique est la première étape dans tout système de RAP. Elle se donne comme mission de transformer le signal de parole en une séquence de vecteurs acoustiques représentant le trait caractéristique du signal. Cette transformation se fait en deux étapes : Prétraitement du signal et extraction des caractéristiques. Celles-ci doivent être pertinentes, discriminantes et robustes.

a. Prétraitement ou mise en forme du signal de la parole

La première opération de prétraitement consiste à faire passer le signal numérisé à travers un filtre de préaccentuation afin de compenser la partie haute fréquence qui a été supprimée lors de la production du son. Le filtre de préaccentuation donné par la fonction de transfert suivante est couramment utilisé :

$$H(Z) = 1 - \alpha . Z^{-1} \tag{2.1}$$

Où α est le facteur de préaccentuation permettant de contrôler la pente du filtre. Typiquement, α prend la valeur entre 0.9 et 0.97.

Comme le signal de parole est fortement non stationnaire, après préaccentuation, le signal est décomposé en une séquence de tranches élémentaires de faible durée (appelées trames) et qui seront supposées stationnaires. La durée typique des tranches est entre 20 et 30 ms avec un chevauchement de moitié. Chaque trame est par la suite multipliée par une fenêtre de Hamming (fenêtrage) pour réduire les effets de bord.

La figure 2.7 montre le résultat de l'application de la fenêtre de Hamming sur une trame de 200 échantillons d'un signal de parole (Këpuska & Elharati, 2015).

Cette première étape est commune à la majorité des méthodes d'analyse.

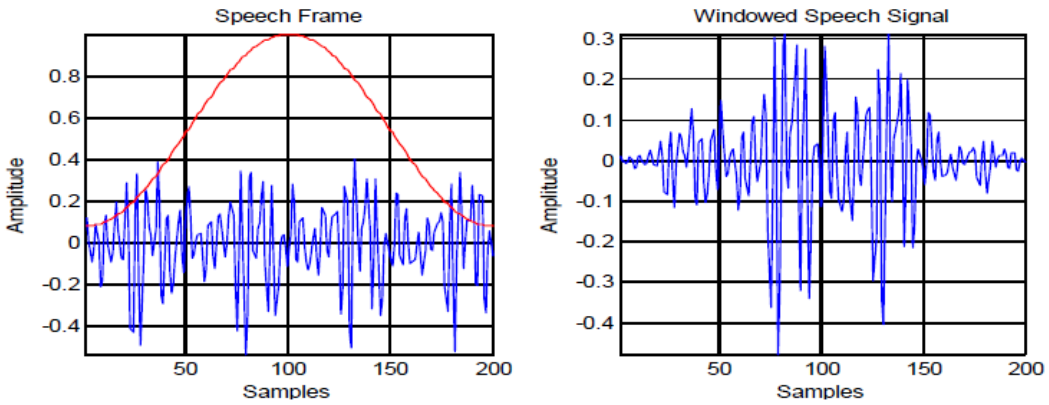


Fig.2.7. La fenêtre de Hamming (Këpuska & Elharati, 2015)

b. Extraction des caractéristiques

Après la mise en forme du signal de parole, une transformée de Fourier à court terme, appelée aussi, transformée de Fourier rapide FFT (Fast Fourier Transform) est appliquée au signal pour passer au domaine fréquentiel et calculer le spectre du signal. Ensuite, chaque méthode d'analyse a ses particularités pour calculer les vecteurs de caractéristiques. Parmi les nombreuses méthodes d'analyse qui existent dans la littérature, nous allons ici présenter brièvement les plus utilisées en s'intéressant spécifiquement à la méthode MFCC, celle que nous avons utilisée dans cette thèse.

- **La méthode LPCC (Linear Prediction Cepstral Coefficients)**

Le codage prédictif linéaire (LPC) est une méthode de représentation de l'enveloppe spectrale d'un signal numérique de parole sous forme compressée, avec les informations fournies par un modèle prédictif linéaire (Alessandro et al., 2006).

Des paramètres cepstraux LPCC peuvent être dérivés à partir des coefficients LPC calculés par la prédiction linéaire. Dans ce cadre d'analyse, le signal de parole est considéré comme la conséquence de l'excitation du conduit vocal par un signal provenant des cordes vocales. La prédiction est basée sur le fait que les échantillons de parole adjacents sont fortement corrélés, et qu'un échantillon peut être estimé en fonction des p échantillons qui le précèdent (Barrault, 2008).

Après avoir mis en forme le signal de parole, et avoir calculé la FFT, les coefficients LPCC sont obtenus en procédant selon les étapes suivantes :

- Analyse d'auto-corrélation : Chaque trame d'échantillons du signal de parole est auto-corrélée pour donner un vecteur de $(p + 1)$ coefficients, où p est l'ordre de l'analyse LPC désiré (Rabiner, 1989). L'ordre d'analyse détermine le nombre de formants que l'analyse est capable de prendre en compte (Lazli, 2007).

- Analyse LPC/ Cepstral : Pour chaque trame, un vecteur de coefficients LPC est calculé à partir du vecteur d'auto-corrélation à l'aide d'une méthode de récursion

de Levinson-Durbin (Levinson, 1947 ; Durbin, 1960). Un vecteur cepstral dérivé est ensuite calculé jusqu'à la $Q^{ième}$ composante, où $Q > p$.

- Pondération : Le vecteur de Q coefficients cepstraux $c_t(m)$ pour une trame t est pondéré par une fenêtre $W_c(t)$ de la forme (Tohkura, 1987 ; Juang et al., 1986 ; Rabiner, 1989) :

$$W_c(m) = 1 + \frac{Q}{2}\sin\left(\frac{\pi m}{Q}\right), 1 \leq m \leq Q \quad (2.2)$$

Ce qui donne :

$$c'_t(m) = c_t(m) * W_c(m).$$

L'avantage des paramètres LPCC est qu'ils sont peu corrélés entre eux. Ce qui permet d'utiliser des matrices de covariances diagonales pour leur moment du deuxième ordre, et gagner beaucoup de temps lors du décodage (Barrault, 2008).

La figure 2.8 schématise les différentes étapes suivies pour calculer les coefficients cepstraux de prédiction linéaire.

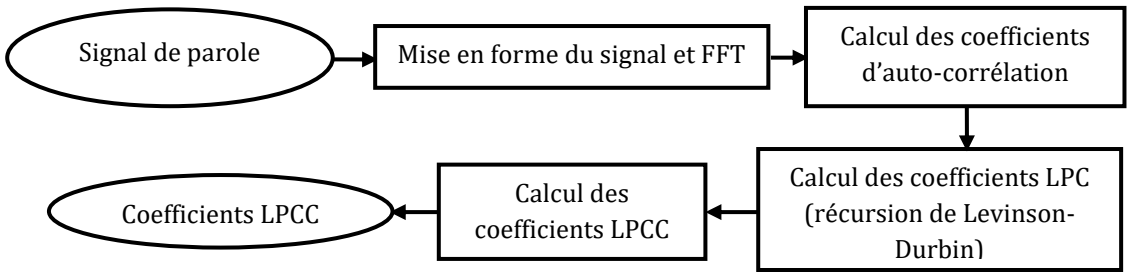


Fig.2.8. La méthode LPC/LPCC

- **La méthode PLP (Perceptual Linear Prediction)**

Initialement proposée par Hermansky dans (Hermansky, 1990). Il l'a défini comme étant une technique d'analyse de la parole utilisant trois concepts issus de la psychophysique de l'audition pour dériver une estimation du spectre auditif : la résolution spectrale de bande critique, la courbe d'égalité des sons, et la loi de puissance d'intensité des sons.

Elle est comme la méthode LPC, basée sur le spectre à court terme du signal de parole, qui nécessite une analyse du signal sur une fenêtre glissante à courte durée. Les différentes étapes de la méthode PLP sont résumées dans la figure 2.10 (partie au centre) (Barrault, 2008).

- **La méthode MFCC (Mel Frequency Cepstral Coefficients)**

La méthode MFCC, parfois appelée l'analyse en banc de filtres, a été proposée par (Davis et Mermelstein, 1980). C'est la méthode la plus utilisée dans la littérature et pour laquelle nous avons opté dans notre travail. Une fois le signal de parole est mis en forme, il est

passé à travers un ensemble de filtres passe-bande triangulaires, appelé banc de filtres Mel. Celui-ci se charge à estimer l'enveloppe spectrale en se basant sur le calcul de l'énergie dans les bandes de fréquences considérées. Après avoir représenté l'enveloppe spectrale dans l'échelle logarithmique, la DCT (transformation discrète en cosinus) est appliquée pour obtenir N coefficients cepstraux. La formule de DCT est montrée dans l'équation :

$$C(n) = \sum_{m=0}^{M-1} \log\big(s(m)\big) \cos\left(\frac{\pi n(m-0.5)}{m}\right);\ n = 0,1,2,\dots,N-1 \qquad (2.3)$$

Où, s est le signal de parole, M est le nombre total de filtres triangulaires de Mel, $C\ (n)$ sont les coefficients cepstraux et N le nombre de coefficients qui est typiquement égal à 12 ou 13.

Nous résumons les différentes étapes à suivre pour la mise en forme et l'analyse du signal acoustique par la méthode MFCC dans la figure 2.9.

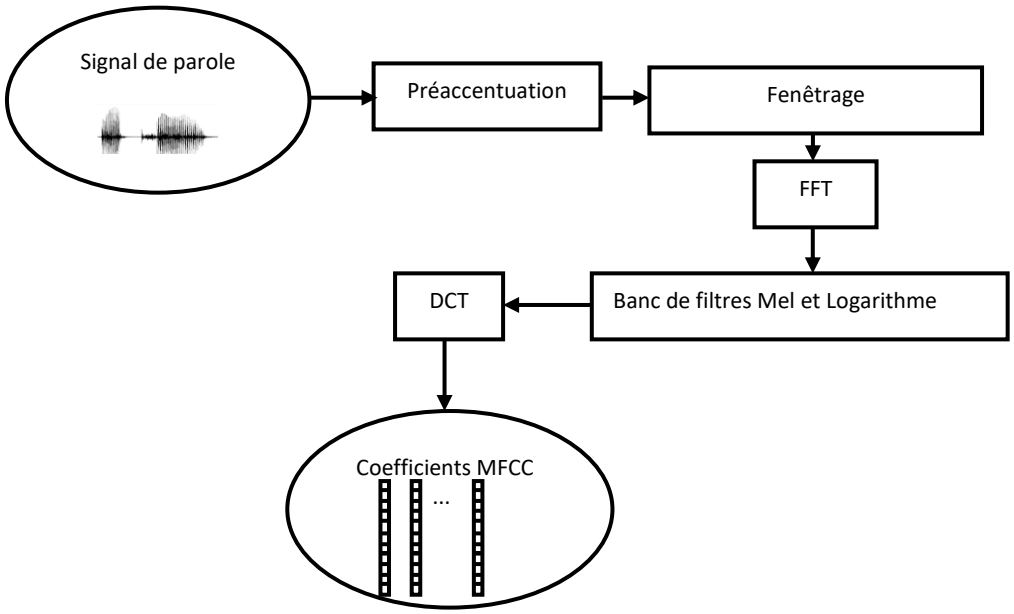


Fig.2.9. La méthode d'analyse MFCC

Pour une présentation plus détaillée de la méthode MFCC, le lecteur peut se référencer à (Rao et al. 2015).

La figure 2.10 (Barrault, 2008) donne une comparaison entre les trois méthodes d'analyse les plus répandues, à savoir LPCC, PLP et MFCC.

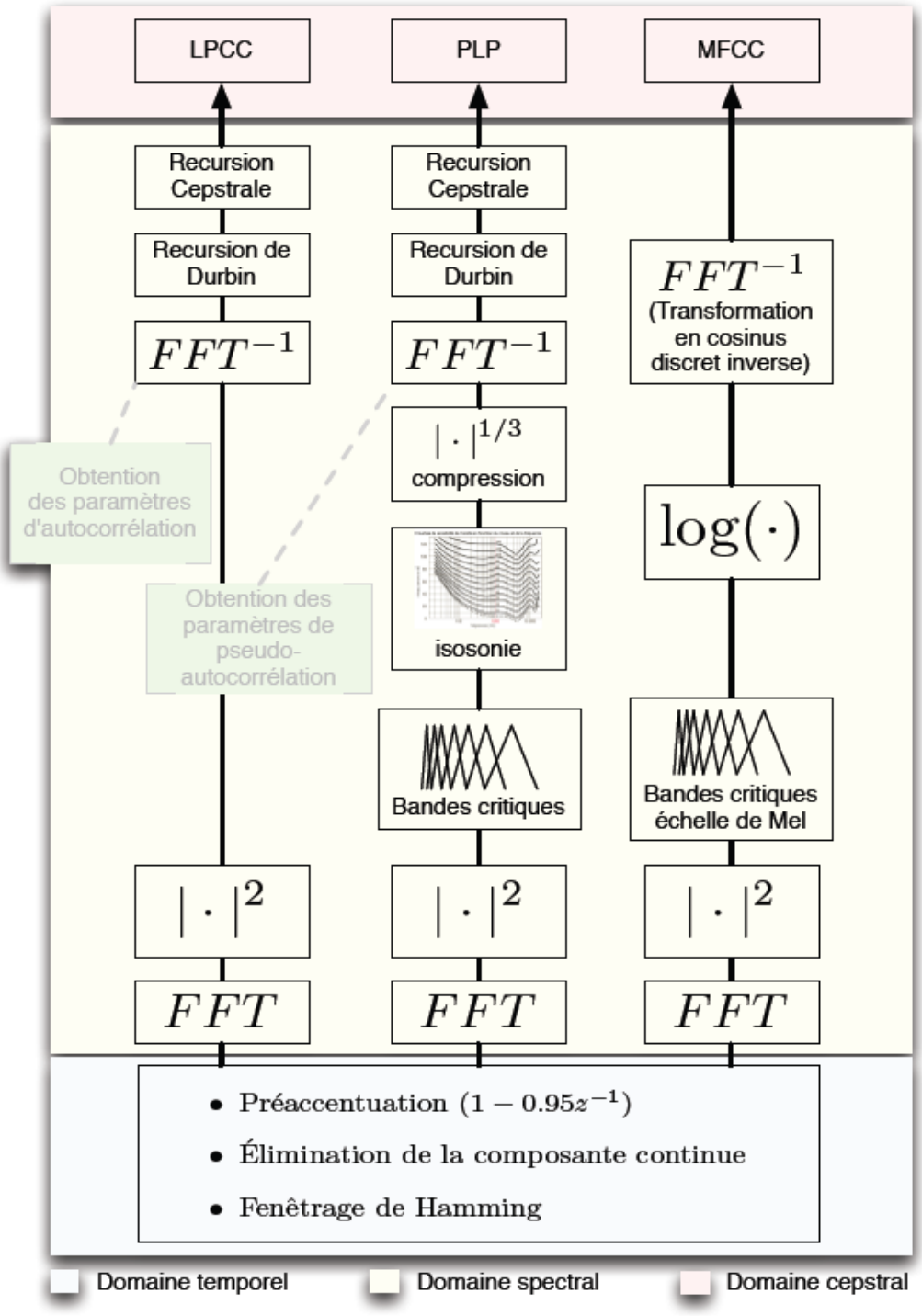


Fig.2.10. Comparaison de trois techniques d'analyse du signal : LPCC, PLP et MFCC (Barrault, 2008)

Pour améliorer la qualité des systèmes de RAP en termes de performance et de robustesse, il est très courant d'ajouter les dérivées premières (△) et secondes (△△) des paramètres acoustiques (voir par exemple, Misra et al., 2003), ou de combiner plusieurs

méthodes d'analyse au sein du même système, tel que le travail de (Ellis, 2000 ; Këpuska & Elharati, 2015) à titre d'exemple. Ce genre de combinaisons s'insère dans ce qu'on appelle communément « approches multi-flux (multi-stream) ».

Bien que la combinaison de plus d'une méthode d'analyse permette souvent d'améliorer les performances et la robustesse, elle augmente le nombre de paramètres acoustiques, ce qui exige une quantité de ressources importante. Pour pallier cette limitation, une solution consiste à utiliser des techniques de réduction de dimensionnalité, telles que l'analyse en composantes principales (PCA : Principal Component Analysis) et l'analyse linéaire discriminante (LDA : Linear Discriminant Analysis). Une autre solution consiste à sélectionner les paramètres les plus pertinents pour concevoir un système performant, mais aussi acceptable en termes de temps de calcul et de ressources mémoire. Cette dernière solution est connue dans la littérature sous le nom de sélection de primitives ou *feature selection*.

2.3.2.2. L'apprentissage

Le rôle de l'apprentissage est de construire les modèles nécessaires pour effectuer la reconnaissance d'un nouvel exemple à savoir, les modèles acoustiques, les modèles de langage et les modèles de prononciation.

a. Modélisation acoustique

L'objectif principal de cette étape est de construire des modèles acoustiques représentant des unités élémentaires de la parole. Ces modèles sont appris préalablement sur une grande base d'enregistrements appelée base d'apprentissage. Celle-ci doit contenir plusieurs élocutions répétées par plusieurs locuteurs afin de permettre au système de prendre en compte la variabilité intra et interlocuteurs. Les techniques de modélisation acoustique les plus utilisées dans la littérature sont les modèles de Markov cachés (HMM), les réseaux de neurones artificiels (ANN) et les systèmes à vaste marge (SVM). Ces techniques et notamment les HMM qui sont à la base des travaux présentés dans cette thèse, seront développées dans les deux chapitres suivants.

Le choix de l'unité de modélisation est crucial. Les unités acoustiques sont généralement divisées en trois catégories : Les phonèmes, les unités courtes (ou phones) et les unités longues (diphones, triphones, semi-syllabes, syllabes, mots). Comme il a été affirmé dans (Lauri, 2004), les unités courtes sont généralement mieux identifc iées. Cependant leur concaténation, pour former des unités plus longues, est problématique en raison de l'absence d'un statut linguistique particulier. L'utilisation des unités longues permet de mieux simuler l'effet de coarticulation interne. Néanmoins, à cause de leur nombre important, la mise en œuvre du SRAP n'est pas une tâche aisée. Concernant les phonèmes, leur utilisation souffre d'une mauvaise modélisation des effets de coarticulation et d'une difficulté à les localiser. Toutefois, leur nombre est relativement faible, ce qui facilite la mise en œuvre du SRAP.

Selon l'unité de modélisation choisie, on peut distinguer deux approches différentes : Globale et analytique. L'approche globale est basée sur des unités longues (typiquement

le mot). Le mot est donc considéré dans sa globalité sans le décomposer en unités élémentaires, telles que les phonèmes ou les syllabes. Dans ce cas, chaque mot est modélisé par un modèle différent. L'approche analytique quant à elle, se base sur les phonèmes comme unité de modélisation (les modèles acoustiques sont des modèles de phonèmes). La reconnaissance d'un mot revient alors à identifier les phonèmes le constituant.

Le choix d'une approche de modélisation est principalement justifié par la taille du vocabulaire utilisé. L'approche globale est généralement utilisée pour les applications dont le vocabulaire est limité. Son avantage est qu'elle garde le phonème dans son contexte de voisinage sans avoir besoin de segmenter le mot en unités élémentaires. Ceci, permet d'éviter les erreurs de segmentation et par conséquent, d'améliorer le résultat de la reconnaissance. Parmi les applications de l'approche globale on peut citer la commande vocale où un vocabulaire restreint est suffisant et parfois même nécessaire pour des fins ergonomiques.

Lorsque le vocabulaire est illimité (ouvert), l'approche globale ne convient plus, car il est impensable de construire et stocker en mémoire autant de modèles que le nombre de mots du vocabulaire. Ainsi, l'approche analytique devient sans doute indispensable. L'inconvénient majeur de cette dernière est que la reconnaissance est fortement liée à la qualité des résultats de la segmentation. Les applications qui nécessitent une approche analytique sont nombreuses, telles que la dictée automatique, le sous-titrage, etc.

b. Modélisation linguistique

En plus des modèles acoustiques, d'autres modèles peuvent être établis dans la phase d'apprentissage, tels que le modèle de langage. Celui-ci est estimé de façon statistique à partir de grande quantité de textes sélectionnés, annotés et normalisés dans le but est de calculer la probabilité conditionnelle d'un futur mot, sachant les $(n-1)$ mots qui le précédent, parlant ainsi des modèles *n-gram*. Nous reviendrons, avec plus de détails, sur ce type de modèles dans le chapitre suivant.

Un autre modèle, appelé modèle de prononciation peut aussi être établi. Il sert à prendre en compte la variation de prononciation principalement liée à l'accent et au dialecte du locuteur. Pour une application de compréhension de la parole, un autre module doit être ajouté au système, il s'agit du module sémantique qui se charge à vérifier une hypothèse dans son contexte sémantique, le signal de parole est tout d'abord, transcrit automatiquement en une chaîne lexicale, puis ce module traite cette chaîne lexicale afin d'en extraire une interprétation sémantique.

2.3.2.3. Décision (décodage ou pattern matching)

A cette étape, le système commence d'abord par le prétraitement et l'extraction des caractéristiques à partir du signal de la parole, en se basant sur la même technique d'analyse utilisée pour représenter les données d'apprentissage. Une fois les vecteurs de caractéristiques sont établés, le système les compare avec les modèles acoustiques préalablement appris. L'algorithme de décodage dépend de l'approche de classification

utilisée. Dans les approches stochastiques par exemple, la décision consiste à choisir le modèle acoustique maximisant la vraisemblance de l'exemple à reconnaitre : La comparaison est de type exemple/modèles. Dans les approches basées distance en l'occurrence l'algorithme *k*-NN qui n'exige pas une phase d'apprentissage, la décision consiste à choisir la classe majoritaire dans le *k* voisinage dans l'espace de représentation de l'exemple à reconnaitre : C'est une comparaison de type exemple/exemples. Nous reviendrons sur ce point dans le chapitre 3.

Outre les modèles acoustiques, représentants des unités de modélisation, le système peut utiliser le vocabulaire des mots à reconnaitre et le modèle de langage qui apporte des connaissances sur la manière dont les mots s'enchaînent dans une phrase.

Le résultat de cette étape peut être l'une des décisions suivantes :

- Réponse correcte : le système classifie correctement l'exemple à reconnaitre.
- Fausse classification (substitution) : le système affecte à l'exemple à reconnaitre une classe à laquelle il n'appartient pas.
- Ambiguïté (confusion) : le système trouve plus d'une solution. Pour enlever l'ambiguïté, une étape de post-traitement est nécessaire. Celle-ci consiste, par exemple, à calculer un poids pour chaque classe.
- Rejet : le système ne peut pas décider, car aucune classe n'est suffisamment proche à l'exemple à reconnaitre. C'est le cas d'un mot qui n'appartient pas au vocabulaire considéré. Une telle décision est possible grâce à un ou plusieurs seuils de rejet, généralement fixés de façon empirique dans la phase d'apprentissage.

L'erreur de substitution est l'erreur la plus grave comparée à l'erreur de rejet. En effet, pour un système critique par exemple, il est préférable de ne pas décider que de faire une fausse décision, qui pourrait causer de graves conséquences.

2.3.3. Evaluation ou test des SRAP

La dernière phase dans le processus de développement d'un SRAP est l'évaluation de sa qualité. Pour cette fin, une base de test contenant plusieurs exemples de chaque classe est utilisée. Si le système est mono-locuteur ou multi-locuteurs, la base de test et celle d'apprentissage ne doivent contenir que des échantillons prononcés par les locuteurs auxquels est dédié le système. Dans le cas d'un système indépendant du locuteur, la base de test doit contenir des échantillons prononcés par des locuteurs qui n'ont pas participé à l'enregistrement de la base d'apprentissage. Typiquement, environ 80% des données disponibles pour le développent d'un SRAP sont réservées à l'apprentissage, et 20% au test. La base de données utilisée étant limitée, et elle doit être partitionnée judicieusement dans une partie d'apprentissage et une autre de test, car la qualité du système dépond beaucoup de cette partition. En effet, comme il a été affirmé dans (Jain et al., 2000), si la *base d'apprentissage* est de faible taille, le classifieur résultant ne sera pas très robuste et

aura une faible capacité de généralisation. D'autre part, si *la base de test* est de petite taille, la confiance dans le résultat d'évaluation sera faible.

Jain et ses collaborateurs dans (Jain et al., 2000) ont présenté les méthodes de partitionnement les plus utilisées. Ces méthodes diffèrent par la façon dont elles utilisent les échantillons disponibles comme bases d'apprentissage et de test. Si le nombre d'échantillons disponibles est extrêmement important, toutes ces méthodes sont susceptibles de conduire au même résultat d'évaluation. Dans ce qui suit, nous donnons un résumé de ces méthodes :

- **La méthode de resubstitution :** Tous les échantillons disponibles sont utilisés à la fois pour l'apprentissage et pour le test ; la base d'apprentissage et celle de test sont les mêmes. Cette méthode assure un bon apprentissage mais une confiance faible dans le résultat d'évaluation. En effet, pour une bonne évaluation, il faut éviter d'utiliser la base d'apprentissage pour le test, et ce, pour ne pas produire une vue optimiste de l'évaluation.

- **La méthode « Holdout » :** La moitié des données est utilisée pour l'apprentissage et le reste pour le test. Les parties de test et d'apprentissage sont indépendantes. L'inconvénient de cette méthode est que différents partitionnements donnent différents résultats d'évaluation.

- **La méthode « Leave one out » :** Un classifieur est conçu en utilisant $(n-1)$ échantillons et évalué en utilisant l'échantillon qui reste. La méthode permet d'utiliser un maximum de données pour l'apprentissage et elle est très utilisée lorsque les bases de données sont de tailles insuffisantes. Son inconvénient est qu'elle exige un calcul important, impliquant plusieurs classifieurs différents qui doivent être entrainés et testés.

- **La méthode de validation croisée (n-fold cross validation) :** Il s'agit d'un compromis entre « Holdout » et « Leave one out ». Elle divise les données en P , où $(1 \leq P \leq n)$ sous bases différentes, $(P-1)$ sous bases sont utilisées pour l'apprentissage et le reste pour le test.

- **La méthode de rééchantillonnage « Bootstrap » :** Cette méthode rééchantillonne les données disponibles avec remplacement pour générer un certain nombre d'ensembles de données « simulées » (généralement des centaines) de la même taille que l'ensemble d'apprentissage initial. Elle est utilisable si la quantité de données disponible est faible.

Il devient désormais courant d'utiliser trois au lieu de deux bases de données : une pour l'apprentissage, une pour la validation et une pour le test. La base de test est invisible pendant le processus d'apprentissage. La base de validation peut être considérée comme un pseudo-test. Nous continuons le processus d'apprentissage jusqu'à ce que l'amélioration des performances sur la base d'apprentissage ne soit plus accompagnée d'une amélioration des performances sur la base de validation. À ce stade, l'apprentissage doit être arrêté afin d'éviter le sur-apprentissage (Kuncheva, 2014).

Selon les besoins de l'application, trois aspects principaux peuvent être considérés durant l'évaluation d'un SRAP : Ses performances, sa robustesse et sa complexité. Par la suite, nous allons aborder en détails chacun de ces aspects.

2.3.3.1. Les performances des SRAP

L'aspect performance est le plus considéré pour évaluer la qualité d'un SRAP. Les performances des systèmes de reconnaissance de mots isolés sont généralement évaluées à l'aide du taux de reconnaissance, qui est le pourcentage de mots correctement reconnus. Cependant, en reconnaissance de la parole continue, le taux d'erreur de mots (WER, de l'anglais « Word Error Rate ») est la métrique courante pour mesurer les performances. Le WER correspond au pourcentage de mots incorrectement reconnus dans un texte de référence. Il est calculé à l'aide de la formule suivante :

$$WER = \frac{I + D + S}{N} \tag{2.4}$$

Avec N est le nombre de mots de référence, S est le nombre de substitutions (mots incorrectement reconnus), D est le nombre de suppressions (mots omis), et I est le nombre d'insertions (mots ajoutés).

2.3.3.2. La robustesse des SRAP

La notion de robustesse a été introduite par Box (Box, 1979). Elle est définie comme étant la capacité du système de demeurer stable face à la variabilité des données et aux perturbations de l'environnement. Dans un système grand public (destiné à être utilisé dans une voiture, dans une rue, etc.), la robustesse au bruit est le défi le plus important, car l'environnement est très affecté par plusieurs sources de bruit. De plus, un système indépendant du locuteur ne doit pas être trop sensible à la variabilité interlocuteur, et à l'incohérence entre les données d'apprentissage et de test.

Pour améliorer la robustesse des SRAP, plusieurs solutions ont été proposées dans la littérature, dont la majorité de ces solutions se focalise, essentiellement, sur la phase d'analyse soit par la sélection de caractéristiques jugées robustes afin d'améliorer le rapport signal/bruit SNR, soit par la normalisation du locuteur pour s'affranchir de la variabilité interlocuteur (Giuliani, 2006), ou par la combinaison de plusieurs méthodes d'analyse (Khelifa et al., 2017). D'autres solutions se sont plutôt focalisées sur la phase de classification en combinant plus d'une méthode de classification. C'est dans cette catégorie de solutions que s'insèrent les travaux réalisés dans le cadre de cette thèse.

Une autre solution au problème de robustesse étant la multi-modalité, qui consiste à intégrer d'autres sources d'information autres que celles du signal acoustique, comme dans la reconnaissance audiovisuelle de la parole (Noda et al., 2015 ; Feng et al., 2017), qui combine des informations acoustiques et des informations visuelles, telles que les expressions faciales et le mouvement des lèvres. La combinaison audiovisuelle peut être réalisée au niveau représentation, mais aussi au niveau décision.

Une manière différente pour accroître la robustesse des SRAP, comme il a été mentionné dans (Spalanzani, 1999), consiste à augmenter la taille de la base d'apprentissage afin d'avoir des systèmes performants dans de multiples situations de test. Toutefois, malgré la taille de plus en plus immense des bases d'apprentissage, celles-ci ne sont qu'un échantillon très limité de l'ensemble des variabilités possibles du signal de parole. Par conséquent, les modèles appris même sur des bases de données de taille importante ne peuvent pas modéliser efficacement tous les types de variabilité possibles.

Dans (Zelinka et al., 2012), une architecture multi-modèles a été proposée pour s'affranchir de la variabilité de l'effort vocal du locuteur, cinq modes de parole ont été considérés : Chuchotements, paroles douces, paroles normales, paroles fortes et cris. L'apprentissage consiste, pour chaque classe de données, à apprendre un modèle différent pour chaque mode de parole, la reconnaissance consiste d'abord à identifier l'effort vocal, puis à décoder le signal parole en sélectionnant les modèles de classes qui correspondent le mieux à l'effort vocal identifié, puis à reconnaitre l'information lexicale en utilisant le décodage de Viterbi (Viterbi, 1967 ; Forney, 1973). Les résultats obtenus sur une base de mots isolés ont montré, selon les auteurs, une réduction de 50% du taux d'erreur de mots par rapport au système de base.

Il est à noter que, l'aspect robustesse n'est pas toujours considéré durant l'évaluation d'un SRAP, et même les travaux traitant de ce problème, comme par exemple les travaux que nous venons de citer dans ce chapitre, n'utilisent généralement pas des métriques permettant de mesurer la robustesse des systèmes proposés, mais seulement des mesures de performance (taux d'erreur ou taux de reconnaissance).

2.3.3.3. La complexité des SRAP

Un troisième aspect à évaluer dans un SRAP est sa complexité. Celle-ci peut être mesurée en termes de ressources mémoire ou de temps de calcul. Cet aspect est moins abordé dans la littérature, comparé aux aspects performance et robustesse. Les auteurs dans (Minker & Néel, 2002) affirment que les algorithmes de traitement du signal de parole, gourmands en temps de calcul et de mémoire, fonctionnent aujourd'hui sur des composants numériques de base et ne requièrent plus de carte ou de matériel spécialisé, et ce, grâce au développement rapide de la microélectronique conduisant à une augmentation en puissance des processeurs standards et des mémoires qui, selon la loi de Moore (Moore, 1965), double environ tous les deux ans. Cela peut justifier l'ignorance de cet aspect lors de l'évaluation d'un SRAP dans la majorité des travaux de la littérature. Toutefois, un bon SRAP doit établir un bon compromis entre les trois aspects, et ce selon les besoins de l'application.

2.4. CONCLUSION

Ce chapitre a été consacré aux éléments de base et aux fondements théoriques de la RAP. Nous avons vu que le signal de parole se caractérise par des propriétés spécifiques rendant son traitement automatique très complexe, et les systèmes conçus peu robustes.

Parmi ces caractéristiques, la variabilité intra et interlocuteur provenant de plusieurs sources, à savoir l'état physiologique, physique, psychologique et social du locuteur.

Nous avons présenté globalement le processus de la RAP, ainsi que les techniques utilisées communément pour réaliser chaque étape du processus, notamment celle de l'analyse acoustique. Quant à l'étape de classification, nous avons juste montré son principe général de fonctionnement, car vu son importance pour notre travail de thèse, le chapitre suivant lui sera entièrement consacré, où nous présentons un état de l'art de ses différentes méthodes et approches.

Nous avons également exposé les solutions proposées dans la littérature pour s'affranchir du problème de robustesse des SRAP, après avoir constaté que, malgré son importance, l'aspect robustesse est peu considéré durant l'évaluation de la qualité des SRAP comparé à l'aspect performance.

CHAPITRE 3

LES APPROCHES DE CLASSIFICATION : UN ETAT DE L'ART

SOMMAIRE

3.1. INTRODUCTION A LA CLASSIFICATION ET L'APPRENTISSAGE AUTOMATIQUE

Nous commençons ce chapitre par la présentation de quelques notions de base relatives à la classification et l'apprentissage automatique.

3.1.1. La classification

La classification est la tâche la plus importante de tout système de reconnaissance de formes. Elle comprend deux opérations principales : L'apprentissage et la reconnaissance.

L'**apprentissage** est le processus qui permet d'apprendre les paramètres d'un classifieur à partir des données préalablement enregistrées et éventuellement étiquetées. Il a été définit par Shalev-Shwartz et Ben-David comme étant le processus de conversion de l'expérience en expertise ou en savoir (connaissances) :

« Learning is the process of converting experience into expertise or knowledge. The input to a learning algorithm is training data, representing experience, and the output is some expertise, which usually takes the form of another computer program that can perform some task » (Shalev-Shwartz & Ben-David, 2014, p 1).

Une distinction est faite entre trois types d'apprentissage : supervisé, non supervisé et semi supervisé. Dans le cas de l'apprentissage supervisé (appelé aussi « avec professeur »), les données d'apprentissage sont fournies au système avec leurs étiquettes de classes. A l'inverse, dans l'apprentissage non supervisé ou « sans professeur », on ne dispose pas de données d'apprentissage étiquetées, et le nombre de classes peut ne pas être prédéterminé. C'est la raison pour laquelle, ce genre d'apprentissage est parfois appelé « classification automatique ». Le regroupement (Clustering) d'un ensemble de données en sous-ensembles d'objets similaires est un exemple typique de ce type d'apprentissage. L'apprentissage semi-supervisé, quant à lui, utilise un mélange de données étiquetées et non étiquetées. L'idée est de compléter l'apprentissage supervisé par des données non étiquetées lorsque le nombre de données étiquetées et insuffisant, ou quand, la quantité de données d'apprentissage non étiquetées est trop grande, rendant fastidieuse la tâche d'étiquetage, qui nécessite l'intervention d'un ou plusieurs utilisateurs humains. Il y a aussi, l'apprentissage par renforcement utilisé surtout dans le domaine des jeux et de la robotique. Contrairement aux types précédents où les systèmes apprennent à partir des exemples d'apprentissage, l'apprentissage par renforcement consiste à apprendre les actions à prendre à partir d'expériences. Il est basé sur une interaction itérée du système apprenant avec son environnement. Ce dernier donne une récompense pour une action ou pour une séquence d'actions prises par le système en fonction de son état courant. En maximisant la somme des récompenses au cours du temps, le système cherche une fonction optimale, appelée politique ou stratégie, associant à l'état courant l'action à exécuter.

La deuxième opération de classification est la **reconnaissance** ou **décision**. Elle permet à partir du résultat de l'apprentissage, d'affecter une classe à une nouvelle forme

inconnue, représentée par ses caractéristiques pertinentes, en se basant sur des métriques de similarité de nature géométrique ou probabiliste.

3.1.2. Qu'est-ce qu'un classifieur ?

Résoudre un problème de classification, nécessite la construction d'un classifieur adéquat. Il a été défini par (Zouari, 2004) comme étant :

« *un système de reconnaissance qui travaille dans un certain* ***espace de caractéristiques****, qui utilise une certaine base pour* ***apprendre ses paramètres****, qui* ***prend sa décision*** *à partir d'une certaine* ***règle*** *et qui fournit en sortie un certain type de* ***réponse*** » (Zouari, 2004, p 18).

Pour un problème de classification à N classes $(c_1, c_2, \dots, c_N)$, la réponse R attribuée par un classifieur à une forme à reconnaitre x, peut être une classe, un rang ou une mesure :

- Une classe : $R(x) = c_i, 1 \leq i \leq N$, le classifieur donne à la forme à reconnaitre x une étiquette de classe.
- Un rang : $R(x) = (r_1, r_2, .., r_N)$, tel que, r_i est le rang attribué à la classe i par le classifieur.
- Une mesure : $R(x) = (m_1, m_2, .., m_N)$, tel que, m_i est la mesure attribuée à la classe i par le classifieur.

3.1.3. Méthode de classification

Une méthode de classification peut être définie comme étant l'ensemble des techniques, des règles et/ou des algorithmes permettant d'apprendre les paramètres d'un classifieur et d'attribuer une forme inconnue à une classe. Pour le développement des systèmes de reconnaissance de formes en général, et de la parole en particulier, de nombreuses méthodes de classification ont été utilisées dans la littérature. Ces méthodes sont généralement distinguées selon plusieurs critères :

- **Selon le type d'apprentissage :** Comme nous l'avons montré précédemment, l'apprentissage peut être supervisé, non supervisé, semi-supervisé ou par renforcement.
- **Selon la distribution des données conditionnellement aux classes : On** distingue les méthodes paramétriques qui modélisent les données par un ensemble de paramètres à estimer, et les méthodes non paramétriques qui ne font pas d'hypothèses sur la distribution de ces données (Saint-Jean, 2001).
- **Selon la nature des caractéristiques des formes :** Il existe deux approches différentes. La première approche, dite structurelle, se base sur les caractéristiques physiques des formes. Ainsi, une forme est décomposée en objets élémentaires et les relations entre ces objets sont décrites. Les caractéristiques sont des primitives de type topologique comme un arc, un cercle, etc. La deuxième approche est dite statistique, car les formes se représentent par des

caractéristiques numériques souvent réelles, regroupées dans des vecteurs appelés vecteurs de caractéristiques. Cette dernière approche est celle communément utilisée en RAP.

- **Selon la manière dont les connaissances nécessaires à la réalisation d'une tâche de reconnaissance sont prises en compte :** La reconnaissance peut être guidée par les données ou par les connaissances. Dans une approche statistique, cette connaissance est extraite à partir d'exemples par des techniques d'apprentissage, tels que les modèles de Markov ou les réseaux de neurones. D'autre part, l'approche de l'intelligence artificielle basée sur la connaissance essaye d'exploiter toutes les sources de connaissance disponibles (phonétique, lexique, syntaxique, sémantique, etc.) pour aboutir à une interprétation associée à un ensemble de caractéristiques acoustiques. Les connaissances sont fournies par des experts humains plutôt qu'apprises (Junqua & Haton, 2012).

Dans le reste de ce chapitre, nous allons présenter un survol des différentes approches de classification que nous proposons, comme le montre la figure 3.1, de les regrouper selon que la méthode de classification est utilisée seule ou combinée à d'autres méthodes, en trois catégories principales : Les approches à base d'un classifieur individuel, les approches ensemblistes et les approches hybrides. Les méthodes et approches qui nous intéressent dans le cadre de notre thèse sont nuancées dans la figure 3.1.

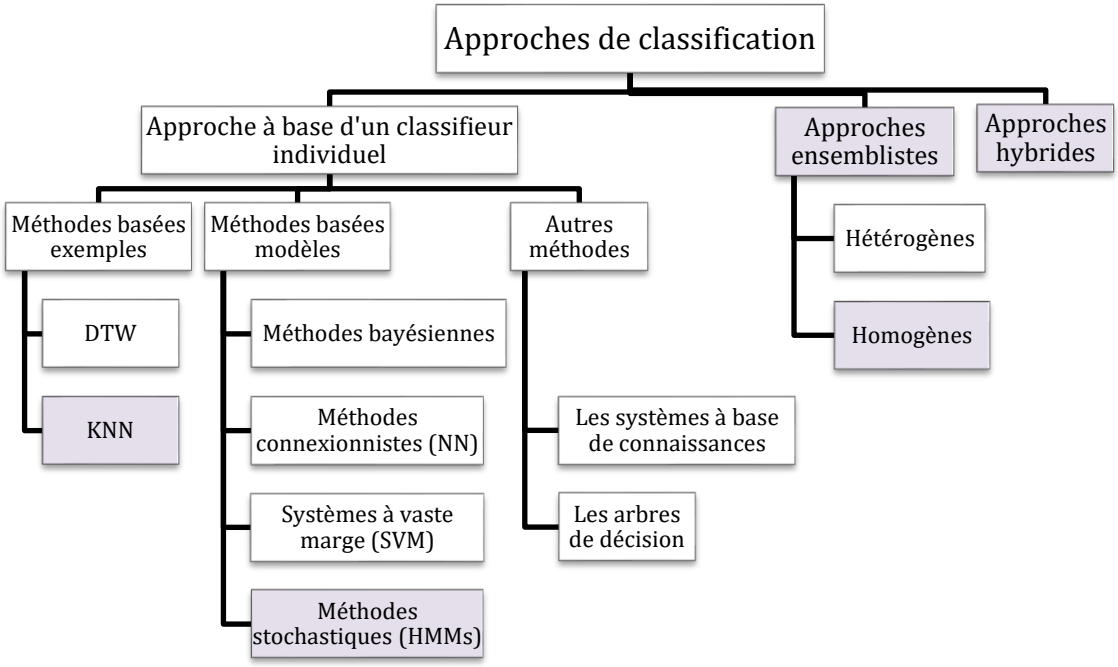


Fig.3.1. Les différentes méthodes et approches de classification

3.2. APPROCHES A BASE D'UN CLASSIFIEUR INDIVIDUEL

Lorsqu'une méthode de classification est utilisée seule dans un système de reconnaissance, on parle d'approche à base d'un classifieur individuel. Selon le type de

comparaison durant l'étape de reconnaissance et la nécessité ou non d'une phase d'apprentissage, nous proposons de regrouper les méthodes de classification de base en deux catégories principales : les méthodes basées exemples et celles basées modèles. Par la suite, nous allons présenter ces deux catégories de méthodes en citant pour chacune d'elles quelques travaux de la littérature du domaine de la RAP.

3.2.1. Les méthodes basées exemples

Cette catégorie regroupe les méthodes dont l'apprentissage consiste, seulement, à stocker les vecteurs de caractéristiques d'un grand nombre d'exemples représentant chaque classe de données. A l'étape de reconnaissance, la comparaison est de type exemple/exemple. Cette dernière est basée le plus souvent sur la distance comme mesure de similarité. Dans ce qui suit, nous allons présenter les deux méthodes basées exemples les plus utilisées en RAP qui sont le DTW (Dynamic Time Warping) et k-NN (k-Nearest Neighbors).

3.2.1.1. DTW (Dynamic Time Warping)

DTW ou déformation temporelle dynamique, est un algorithme utilisé efficacement surtout pour la comparaison des séries temporelles. Il est basé sur le concept de programmation dynamique qui a été proposé en 1957 pour une tâche de contrôle optimal (Bellman, 1957) et a été introduit en reconnaissance de la parole en 1968 (Vintsyuk, 1968).

Le DTW est utilisé pour la comparaison des exemples de la parole représentés par des vecteurs de caractéristiques, nous citons par exemple (Dhingra et al., 2013 ; Muda et al., 2010 ; Marković et al., 2017 ; Mansour et al., 2015). Comme ces vecteurs n'ont généralement pas la même longueur, le problème qui se pose est comment peut-on aligner correctement ces vecteurs pour les comparer en termes de distance. Le DTW est une solution efficace à ce problème. Considérant deux exemples de la parole $X = (x_1, x_2, \dots, x_{T_x})$ et $Y = (y_1, y_2, \dots, y_{T_y})$. Le principe de DTW consiste à déterminer la distance $d(X, Y)$ en transformant les indices des séquences vectorielles en un axe temporel normalisé k, à l'aide de deux fonctions de déformation $\emptyset_x$ et $\emptyset_y$. Ainsi, nous avons,

$$i_x = \emptyset_x(k), 1 \leq k \leq T \tag{3.1}$$

$$i_y = \emptyset_y(k), 1 \leq k \leq T \tag{3.2}$$

Cela nous donne un mappage de $(x_1, x_2, \dots, x_{T_x})$ à $(x_1, x_2, \dots, x_T)$, et de $(y_1, y_2, \dots, y_{T_y})$ à $(y_1, y_2, \dots, y_T)$. Avec un tel mappage, nous pouvons calculer $d_\emptyset(X, Y)$ en utilisant ces fonctions de déformation, ce qui nous donne la distance entre les deux exemples,

$$d_\emptyset(X, Y) = \sum_{k=1}^{T} d\left(\emptyset_x(k), \emptyset_y(k)\right) m_k / M_\emptyset \tag{3.3}$$

où m_k est un poids de chemin et $M_\emptyset$ est un facteur de normalisation (Yfantis & Elison, 1970 ; Yfantis et al., 1999). Voir la figure 3.2 pour un exemple d'un tel alignement temporel.

Ainsi, tout ce qui reste est la détermination du chemin ∅ indiqué dans l'équation 3.3. Le choix le plus courant est de spécifier que ∅ est le minimum de tous les chemins possibles, soumis à certaines contraintes, ce qui donne :

$$d(X,Y) \triangleq \min_{\emptyset} d_{\emptyset}(X,Y) \tag{3.4}$$

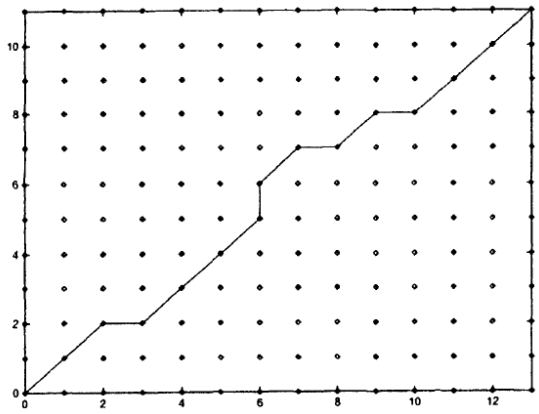


Fig.3.2. Alignement temporel en utilisant une fonction de déformation (Yfantis & Elison, 1970 ; Yfantis et al., 1999).

3.2.1.2. *k*-NN (*k*- Nearest Neighbors)

k-NN ou k-plus proches voisins, initialement proposé par Fix et Hodges (1951) est un algorithme basé-exemple très simple. Il a été utilisé avec succès dans plusieurs domaines de reconnaissance de formes. Son idée, d'après (Dong, 2013) vient du fait que les exemples se situent généralement à proximité des autres exemples ayant les propriétés similaires. Le principe de reconnaissance de k-NN consiste d'abord à calculer la distance entre le vecteur de caractéristiques de la forme à reconnaitre et chacun des vecteurs de caractéristiques de référence représentant les exemples de la base d'apprentissage, puis à affecter à la forme à reconnaitre l'étiquette de la classe majoritaire dans les k voisins les plus proches. Dans le cas particulier où $k = 1$, la forme est attribuée à la classe de son voisin le plus proche. La méthode k-NN est à la base de nombreux travaux récents de la littérature de la RAP, nous citons à titre d'exemples (Aulia et al., 2017 ; Imtiaz & Raja, 2016 ; Xu et al., 2015).

Les principaux avantages de la méthode k-NN sont sa robustesse, sa simplicité de mise en œuvre et le fait qu'aucune phase d'apprentissage n'est nécessaire, qui ne consiste qu'à stocker les vecteurs des caractéristiques et les étiquettes des classes de tous les exemples de la base d'apprentissage. Cependant, son inconvénient est qu'elle est très couteuse en termes de ressources mémoire et de temps de calcul, car on doit calculer la distance entre la forme à reconnaitre et chacun des exemples de la base d'apprentissage. Ce problème limite considérablement son utilité dans certaines applications (telles que la reconnaissance vocale) où un grand nombre d'exemples d'apprentissage est indispensable pour obtenir des performances acceptables.

Malgré leur popularité au début des recherches en reconnaissance de la parole, les méthodes basées exemples sont actuellement rarement utilisées, et décriées pour leur gourmandise en ressources et leur grande complexité de calcul durant l'étape de reconnaissance.

3.2.2. Les méthodes basées modèles

L'apprentissage dans ce type de méthodes, consiste à exploiter les exemples de la base d'apprentissage pour construire des modèles de référence représentant les différentes classes ou pour décrire les fonctions discriminantes caractérisant les frontières de décision. La reconnaissance est basée sur une comparaison de type exemple/modèle dont le principe de décision dépend de la méthode.

Dans les pages qui suivent, nous allons exposer les différentes méthodes basées modèle utilisées couramment en RAP : les méthodes bayésiennes, les méthodes connexionnistes, les systèmes à vaste marge et les méthodes stochastiques à base des modèles de Markov cachés.

3.2.2.1. Les méthodes bayésiennes

Les méthodes bayésiennes sont des méthodes basées sur le calcul des statistiques sur la répartition des classes et leurs exemples dans l'espace d'observations. Les formes dans ces méthodes sont représentées par des valeurs numériques souvent réelles, et la décision est de type « plus forte probabilité d'appartenance à une classe ». Le calcul de probabilité est basé sur la règle de Bayes qui permet de trouver la probabilité à postériori en fonction de la probabilité à priori et la probabilité conditionnelle. L'application de ces méthodes en reconnaissance de formes, d'après (Hallouli, 2004), a été initialement formalisée en 1965 par (Chow, 1965).

Le principe général de décision bayésienne peut être résumé comme suit : Etant donné un ensemble de classes $\Omega = \{w_1, w_2, \dots, w_N\}$ et une forme inconnue représentée par son vecteur de caractéristiques x. On suppose connaitre la probabilité à priori $P(w_i)$ de chaque classe w_i appartenant à Ω, qui est la probabilité qu'une forme quelconque soit dans la classe w_i indépendamment de toute observation, et les fonctions de densités de probabilités permettant de calculer $P(x/w_i)$, qui est la probabilité de trouver la forme x dans la classe w_i. La décision bayésienne consiste d'abord à calculer pour chaque classe w_i la probabilité à postériori $P(w_i/x)$ en utilisant la règle de Bayes (équation 3.5), puis à choisir la classe w^* qui maximise $P(w/x)$ (équation 3.7).

$$P(w/x) = \frac{P(w) * P(x/w)}{P(x)} \tag{3.5}$$

$$P(x) = \sum_{i=1}^{N} P(w_i) * P(x/w_i) \tag{3.6}$$

Où $P(x)$ est un facteur de normalisation qui peut être ignoré durant la décision puisque c'est le même pour toutes les classes (il ne dépend pas d'une valeur particulière w_i).

$$w^* = argmax\big(P(w_i/x)\big) = argmax(P(w_i) * P(x/w_i)), 1 \leq \text{i} \leq \text{N} \quad (3.7)$$

Le principe de la décision bayésienne peut être illustré sur la figure suivante :

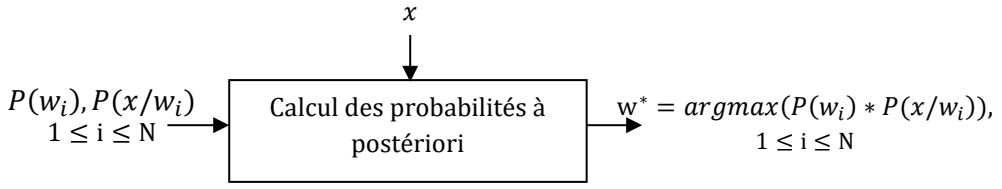


Fig.3.3. Principe de décision bayésienne

Si toutes les classes sont équiprobables, la décision revient à maximiser la probabilité conditionnelle et on parle, dans ce cas, de la décision par maximum de vraisemblance.

Les quantités $P(w_i)$ et $P(x/w_i)$sont les résultats de l'apprentissage bayésien. Ce dernier consiste alors à calculer d'une part, la probabilité à priori $P(w_i)$ caractérisant chaque classe, et d'autre part les paramètres de la fonction de densité de probabilité permettant de calculer les probabilités conditionnelles $P(x/w_i)$. La probabilité à priori $P(w_i)$ est estimée facilement et avec précision sur une base d'apprentissage contenant plusieurs exemples de chaque classe en appliquant la formule $P(w_i) = \frac{nc}{nt}$, tel que nc est le nombre d'exemples de la classe w_i et nt est le nombre total d'exemples dans la base d'apprentissage. En revanche, le calcule de $P(x/w_i)$ est beaucoup plus complexe. En général, on suppose que cette probabilité a une forme gaussienne. Dans ce cas, il suffit de calculer, sur la base d'apprentissage, les paramètres de la loi gaussienne μ et σ caractérisant chaque classe, puis d'appliquer (dans la phase de reconnaissance) la formule suivante :

$$p(x/w) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (3.8)$$

Le classifieur bayésien est souvent décrit comme « naïf », car il est basé sur l'hypothèse d'indépendance simplificatrice qui suppose que les caractéristiques d'une forme sont indépendantes. Le problème est que, en pratique, ce n'est pas toujours le cas. En plus de sa simplicité, l'avantage du classifieur de Bayes naïf, comme il a été mentionné par (Sharma & Kaur, 2013), est qu'il n'exige pas une grande base d'apprentissage pour estimer les paramètres du modèle.

Dans le contexte de la RAP, le classifieur de Bayes a été largement utilisé. Nous citons à titre d'exemple, les travaux (Seltzer, 2004 ; Huo & Lee, 2000 ; Norris & McQueen, 2008 ; Kamper et al., 2015 ; Kamper et al., 2017).

3.2.2.2. Les méthodes connexionnistes (les réseaux de neurones artificiels et profonds)

Un réseau de neurones artificiel (ANN), comme son nom l'indique, est une méthode d'apprentissage automatique, dont le principe est inspiré du fonctionnement du cerveau humain. Les informations sont traitées de façon parallèle par des fonctions appelées neurones.

Les perceptrons multicouches (MLP) sont les réseaux de neurones les plus couramment utilisés en raison de leur rapidité de fonctionnement, de leur facilité de mise en œuvre et de la taille réduite des bases d'apprentissage requises (Orhan et al., 2011). Le MLP (figure 3.4) se compose typiquement de trois couches de neurones dont le nombre de neurones dépond du problème considéré : Une couche d'entrée, une couche intermédiaire (cachée) et une couche de sortie. Les neurones d'une couche produisent des sorties après avoir reçu des entrées. Les sorties des neurones d'une couche sont transmises à la couche suivante, qui les utilise comme entrées de ses propres neurones et produisent d'autres sorties. Ces dernières sont ensuite transmises à la couche suivante de neurones et ainsi de suite jusqu'à la couche de sortie. Les neurones terminaux fournissent alors le résultat final du réseau.

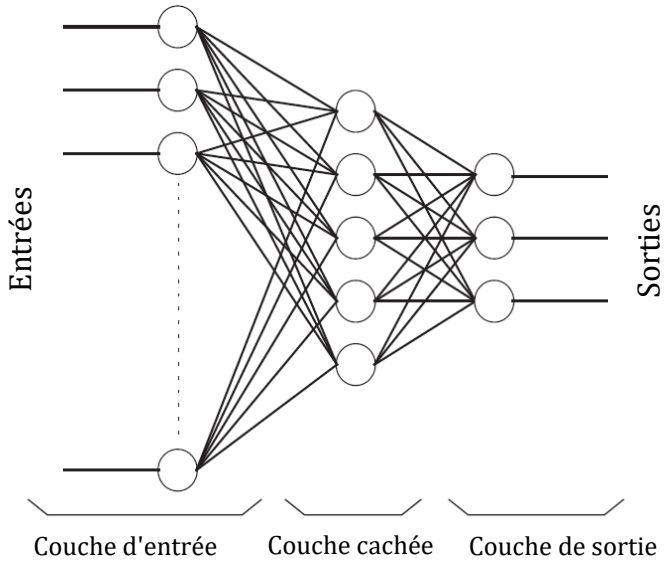


Fig.3.4. Exemple d'architecture d'un Perceptron multicouches

Il est également possible d'avoir plus d'une couche cachée, c'est ce qu'on appelle réseau de neurones profond DNN (Deep Neural Network) ou à apprentissage profond (Deep Learning), qui peut générer automatiquement des représentations dans une configuration multicouches à partir des données brutes. Par conséquent, les caractéristiques brutes, telles que les spectrogrammes de signaux et les pixels d'images, peuvent être directement utilisées dans ce type d'apprentissage, sans avoir besoin de passer par l'étape d'extraction des caractéristiques, comme pour le cas des approches traditionnelles d'apprentissage automatique qui nécessitent un choix judicieux de la représentation des données (les caractéristiques extraites des données brutes durant la phase d'analyse).

Le DNN présente de nombreuses variations qui diffèrent dans leur architecture associée. Différentes architectures ont été développées pour réaliser certaines applications, telles que le réseau de neurones récurrents RNN (Recurent Neural Network) et le réseau de neurones convolutifs CNN (Convolutional Neural Network). Le RNN est conçu pour modéliser des données séquentielles, telles que les phrases et les formes d'onde audio, quant au CNN, il peut être considéré comme un type spécial de perceptron multicouches MLP. Il a été inspiré de la recherche sur les systèmes neuronaux de vision des

mammifères, dans lesquels les neurones corticaux individuels ne répondent à l'activation que dans une zone restreinte du champ visuel. De même, l'architecture CNN introduit des connexions restreintes, pas complètement connectées, les réseaux qui en résultent sont conçus pour traiter des données avec une structure intrinsèque de type grille, telles que les pixels dans une image (Wu, 2018).

Récemment, les DNN ont montré un grand succès par rapport aux réseaux de neurones classiques, et ils sont, actuellement, une tendance de recherche importante dans de nombreux domaines de l'apprentissage automatique et de l'intelligence artificielle. En RAP, plusieurs travaux récents sont basés sur les DNN, nous citons à titre d'exemple, les RNN (Sak et al., 2015 ; Iwana & Uchida, 2017 ; Song et al., 2018,) et les CNN (Abdel-Hamid et al., 2013 ; Palaz et al., 2015 ; Iwana & Uchida, 2017 ; Sumon et al., 2018).

Le succès actuel de l'apprentissage profond peut s'expliquer par plusieurs motivations (Ravanelli, 2017). Nous en citons, ici, deux que nous jugeons les plus importantes :

- **Big Data :** Un élément essentiel du succès de cette technologie est la disponibilité de grandes quantités de données. En effet, l'efficacité des DNN dépend fortement de la capacité du modèle, qui peut être améliorée en adoptant des architectures profondes et étendues. L'amélioration de la capacité du réseau, cependant, augmente le nombre de paramètres, ce qui nécessite plus de données pour les estimer de manière fiable. Heureusement, la diffusion rapide d'Internet et des smartphones permet des collectes de données volumineuses faciles et peu coûteuses.

- **Puissance de calcul :** Pour exploiter correctement les modèles profonds et les bases de données de plus en plus importantes, une puissance de calcul considérable est nécessaire. Au cours des dernières années, des progrès notables ont été réalisés dans le développement de matériel spécialisé pour un apprentissage profond. Les processeurs graphiques modernes (GPU), par exemple, sont actuellement très utilisés pour apprendre efficacement des modèles complexes.

3.2.2.3. Les systèmes à vaste marge

Les systèmes à vaste marge ou les machines à vecteurs de support (SVM : Support Vector Machines) sont des méthodes de classification statistiques à apprentissage supervisé, proposés par Vladimir Vapnik (Vapnik, 1995). Ils permettent de traiter les problèmes de discrimination non linéaire, les données sont classifiées en séparant deux classes à l'aide d'une frontière de décision appelée *hyperplan séparateur* et un sous-ensemble d'échantillons d'apprentissage appelés *vecteurs supports*. Ils sont bien connus pour leurs bases théoriques solides, leur performance de généralisation et leur capacité à gérer des données à grande dimension. Les SVM reposent principalement sur deux notions fondamentales : la marge maximale et la *fonction noyau*. La marge maximale est la distance entre la frontière de décision (hyperplan séparateur) et les vecteurs supports (les échantillons les plus proches). Quant à la fonction noyau (Kernel function), elle est utilisée en cas des problèmes non linéairement séparables où il est impossible de trouver

un hyperplan séparateur, elle permet de changer l'espace de représentation à un autre espace de dimension supérieure, éventuellement infini, appelé *espace de re-description*, dans lequel les données peuvent être séparées par des hyperplans. En effet selon (Hasan & Boris, 2006), plus l'espace de re-description est de dimension grande, plus la chance de pouvoir trouver un hyperplan séparateur entre les données est importante. Ceci est illustré par la figure suivante :

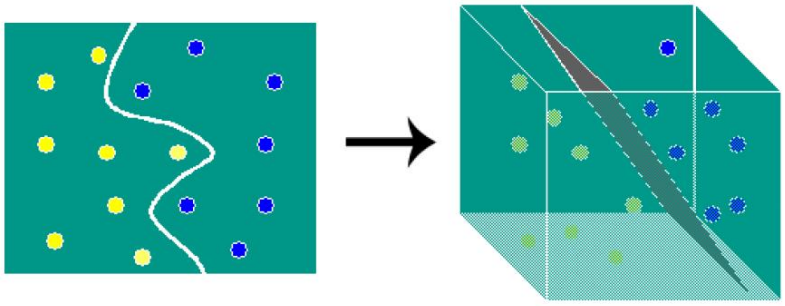


Fig.3.5. Transformation d'un problème non linéairement séparable en un problème linéairement séparable (Hasan & Boris, 2006)

Dans (Pérez-Cruz & Bousquet, 2004), les auteurs ont résumé le principe de fonctionnement des SVM en ce qui suit : Étant donné un ensemble de données séparables, l'objectif est de trouver la fonction de décision optimale (hyperplan séparateur). Nous pouvons facilement voir qu'il existe un nombre infini de solutions optimales (hyperplans valides) pour ce problème, dans le sens où elles peuvent séparer les échantillons d'apprentissage avec zéro erreur. Cependant, comme nous recherchons une fonction de décision capable de généraliser pour des nouveaux échantillons inconnus, nous pouvons penser à un critère supplémentaire pour trouver la meilleure solution parmi celles qui ont zéro erreur sur la base d'apprentissage. Si nous connaissions les densités de probabilité des classes, nous pourrions appliquer le critère du maximum a posteriori (MAP) pour trouver la solution optimale. Malheureusement, dans la plupart des cas pratiques, ces informations ne sont pas disponibles, nous pouvons donc adopter un autre critère plus simple : parmi ces fonctions sans erreurs d'apprentissage, nous choisirons celle *maximisant la marge*, cette marge étant la distance entre l'échantillon le plus proche et la frontière de décision définie par cette fonction. Bien entendu, l'optimalité dans le sens de la marge maximale n'implique pas nécessairement l'optimalité dans le sens de minimiser le nombre d'erreurs dans le test, mais c'est un critère simple qui donne lieu à des solutions qui, en pratique, s'avèrent être les meilleures pour beaucoup de problèmes. La figure 3.6 schématise le principe de fonctionnement des SVM.

Par leur nature, les SVM sont principalement des classifieurs binaires (bi-classe), mais ils peuvent être adoptés pour des problèmes multi-classes (N classes). Pour ce faire, deux stratégies sont couramment utilisées : La stratégie *Un-Contre-Un* (One-Against-One), c'est la plus ancienne et la plus courante, et qui consiste à diviser la base de données en N cas bi-classe. La seconde stratégie, *Un-Contre-Tous* (One-Against-All), consiste à construire une machine pour chaque paire de classes, ce qui donne $N\ (N-1)\ /\ 2$ machines (Anthony et al., 2007).

Les SVM ont été largement utilisés dans le domaine de RAP. Nous citons à titre d'exemples, les travaux (Zhang & Gales, 2012 ; Kanisha, 2018 ; Manikandan & Venkataramani, 2011 ; Solera-Urena et al., 2011 ; Besbes & Lachiri, 2016).

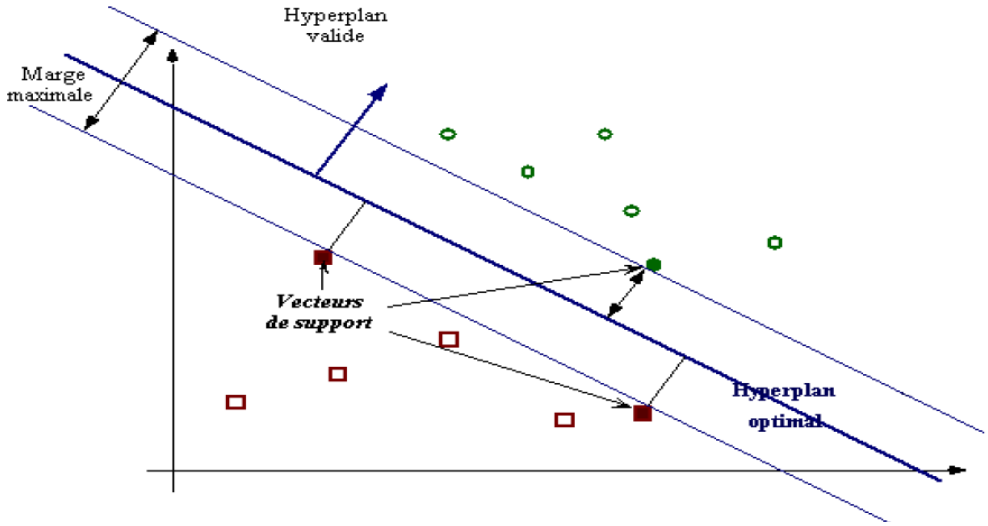


Fig.3.6. Principe de fonctionnement des SVM (Hasan & Boris, 2006).

3.2.2.4. Les méthodes stochastiques à base des modèles de Markov cachés

Les méthodes stochastiques sont des méthodes de modélisation utilisables là où se trouvent le hasard et l'incertitude. Elles permettent l'utilisation des modèles probabilistes basés sur des processus stochastiques évoluant aléatoirement avec le temps pour traiter les problèmes à information incomplète ou incertaine. Un processus stochastique signifie, dans la théorie probabiliste, un processus aléatoire ou un ensemble de variables aléatoires. Si l'état du processus au temps t ne dépend que de son état au temps $t-1$, on l'appelle un processus de Markov. Ce dernier peut être modélisé par un modèle de Markov observable ou caché.

Les modèles de Markov cachés HMM sont des méthodes stochastiques utilisées pour la modélisation et la classification dans de nombreux domaines de reconnaissance de formes et de l'apprentissage automatique, principalement dans le domaine de RAP. Un HMM est un automate probabiliste permettant d'engendrer des séquences d'observations au fil de temps. Il est basé sur deux processus stochastiques. L'un observable permet de modéliser les séquences d'observations, et l'autre caché modélisant les séquences d'états ayant engendré les observations. Il est qualifié de « caché », car, étant donnée une séquence observée, on ne peut pas déterminer de quelle suite d'états provient cette séquence.

Grâce à leurs bases mathématiques solides et leur capacité à modéliser des séquences de longueur variable et à assurer une segmentation implicite de la parole, les HMM restent l'une des techniques les plus performantes dans le domaine de la reconnaissance vocale. Malgré leur grand succès, les HMM soufrent de certaines limites, telles que la nécessité d'une large base d'apprentissage étiquetée afin d'avoir des performances acceptables, et le problème d'initialisation d'un grand nombre de paramètres qui doivent être

minutieusement choisis. Ces paramètres sont généralement fixés expérimentalement et ils dépendent fortement des données d'apprentissage et de test, ce qui affecte la robustesse et la stabilité du système. Nous détaillerons plus largement ce dernier problème dans le chapitre suivant qui sera entièrement consacré aux modèles de Markov cachés.

3.2.3. Autres méthodes de classification

Dans cette sous-section, nous présentons brièvement deux méthodes de classification : Les systèmes à base de connaissances et les arbres de décision. Ces méthodes sont actuellement rarement utilisées en RAP comparées aux méthodes statistiques.

3.2.3.1. Les systèmes à base de connaissances

Ces méthodes sont basées sur les connaissances acquises de l'expérience d'experts en traitement de la parole et incorporent explicitement les connaissances. La principale caractéristique de ces systèmes est la modélisation de l'expertise des experts sous la forme de bases de données reflétant les connaissances et les règles qui reproduisent leur "savoir-faire". Généralement, ces systèmes se composent de trois modules principaux (Jacob, 1995) :

- Un module de prétraitement dans lequel des paramètres complexes et hétérogènes sont extraits ; des paramètres acoustiques tels que l'énergie dans des bandes de fréquences prédéterminées peuvent coexister avec des coefficients de type articulatoire tels que les formants.

- Un module de segmentation du signal. Les données sont segmentées en unités phonétiques ou segments minimaux proches d'unités linguistiques. Celles-ci sont modélisées à l'aide de connaissances d'experts et constituent ainsi une base de faits.

- Un module d'identification de ces unités phonétiques qui se repose sur des règles de production basées sur des connaissances des experts dans les domaines de l'acoustique, de l'articulation, de la phonologie et de la phonétique, etc.

3.2.3.2. Les arbres de décision

Les arbres de décision (en anglais, Decision Trees) sont des méthodes simples de classification supervisée, introduits dès les années 60. Un arbre de décision est défini comme étant un graphe orienté acyclique, se compose d'une racine, d'un ensemble de nœuds intermédiaires et d'un ensemble de nœuds terminaux appelés feuilles. Comme il a été présenté dans (Quinlan, 1987) et (Hawarah, 2008), le principe de construction d'un arbre de décision peut être résumé en ce qui suit. Il s'agit de partitionner récursivement les exemples de l'ensemble d'apprentissage en utilisant des tests décrit en termes de collection d'attributs, jusqu' à l'obtention des feuilles qui ne contiennent que des exemples appartenant à une seule classe. Pour partitionner l'ensemble d'apprentissage, on choisit des attributs qui vont minimiser l'impureté dans les sous-arbres. Cela signifie que pour chaque attribut qui n'a pas encore été utilisé, on calcule l'impureté qui reste

après son utilisation. Celui qui laisse le moins de désordre est choisi comme étant le prochain nœud de l'arbre de décision. Ce processus est répété sur chaque nouveau nœud et s'arrête quand les feuilles de l'arbre obtenu contiennent des exemples d'une seule classe ou quand aucun test n'apporte plus d'amélioration. Un nouvel exemple inconnu est classé en traçant un chemin depuis la racine de l'arbre jusqu'à la feuille appropriée et en assurant que l'exemple appartient à la même classe que les exemples de l'ensemble d'apprentissage associés à cette feuille.

En plus de leur simplicité, les arbres de décision ont de nombreux avantages (Zighed, 2008) :

- Possibilité de décomposer des problèmes complexes en une union de problèmes simples.
- Ils supportent les données catégorielles comme les données numériques.
- La structure hiérarchique des arbres permet d'extraire des règles compréhensibles pour l'utilisateur.
- Très peu d'hypothèses sont faites sur les données comme les arbres sont des méthodes exploratoires et non inférentielles.

Les arbres de décision ont été utilisés pour des problèmes de régression et de classification. Dans le domaine de RAP, ils sont très peu utilisés, on peut néanmoins citer quelques travaux (Akamine & Ajmera, 2012 ; Reichl & Chou, 2000 ; Bahl et al., 1991).

3.3. LES APPROCHES ENSEMBLISTES

"Referring to classification problems, Wolpert's theorem has a specific lecture: there is not a single classifier modeling approach which is optimal for all pattern recognition tasks, since each has its own domain of competence" (Woźniak et al., 2014, p1).

Bien que les méthodes de classification puissent fonctionner individuellement de manière efficace, en particulier les HMM dans le domaine de RAP, plusieurs recherches ont été faites, dans le but est de s'approcher le plus possible des performances du classifieur optimal. Parmi ces recherches celles qui s'intéressent aux méthodes ensemblistes, où, plusieurs classifieurs sont combinés, selon certaines stratégies, afin d'exploiter efficacement la complémentarité entre eux.

Les méthodes ensemblistes, appelées aussi combinaison de classifieurs ou encore systèmes multi-classifieurs, sont des approches très efficaces dans le domaine de classification et d'apprentissage automatique. Elles consistent à entrainer séparément plusieurs classifieurs différents et à combiner leurs prédictions, pour chaque échantillon de test, au niveau de décision afin d'obtenir des performances meilleures que celles des classifieurs de base. En effet, de nombreuses recherches ont montré que l'ensemble permet le plus souvent d'améliorer les performances par rapport à tous les classifieurs de base, tel qu'il a été cité dans les travaux (Dietterich, 2000 ; Bauer et Kohavi, 1999 ; Freund et al., 1996). L'efficacité des approches ensemblistes peut être justifiée par le fait que chaque classifieur peut avoir sa propre portion d'échantillons de données ou de

caractéristiques là où il est le plus performant, c'est-à-dire, différents classifieurs commissent des erreurs de classification différentes.

3.3.1. Architectures de combinaison de classifieurs

Il existe plusieurs types d'architectures ensemblistes qui peuvent être classées, comme l'illustre la figure 3.7, en trois catégories principales : Parallèle, en série et hybride.

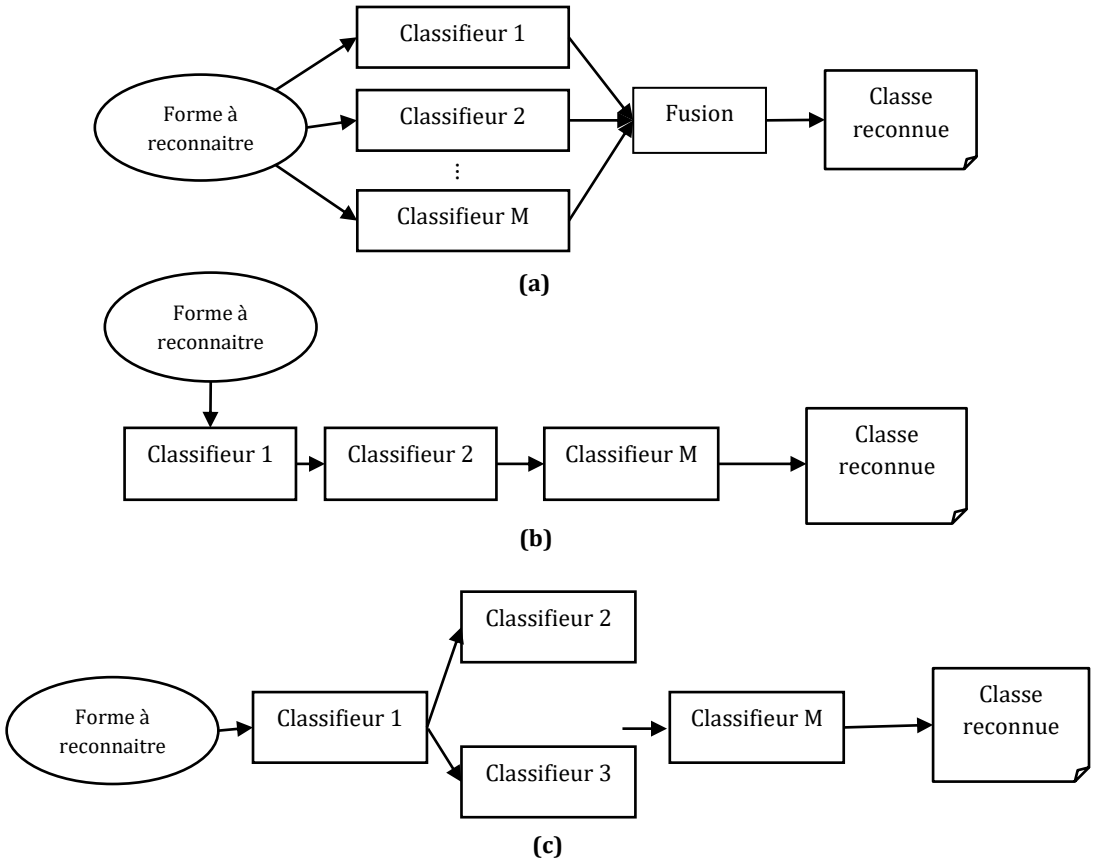


Fig.3.7. Architectures des méthodes ensemblistes (a) parallèle, (b) série et (c) hybride

- **Architecture parallèle :** Chaque classifieur de base, membre de l'ensemble, donne sa prédiction indépendamment des autres. Ensuite, les prédictions individuelles sont fusionnées avant de prendre la décision finale. Un schéma illustratif est montré dans la Fig.3.7 (a). L'inconvénient majeur de l'approche parallèle (Zouari, 2004) est qu'elle nécessite d'activer tous les classifieurs qui doivent participer de manière concurrente et indépendante. Par contre, la décision finale est prise avec le maximum de connaissances mises à disposition par chaque classifieur. Dès lors se posent les problèmes de précision des informations fournies par les classifieurs et de la confiance qu'on peut accorder à chacun d'eux.

- **Architecture en série (en cascade) :** Les classifieurs de base sont appliqués séquentiellement l'un après l'autre. Chaque fois qu'un classifieur est traversé, le nombre de classes cibles est réduit. Chaque classifieur prend en entrée la réponse

du classifieur placé en amont pour traiter les rejets ou confirmer la décision qui lui est fournie. La Fig.3.7 (b) illustre l'architecture en série.

Vu qu'elle permet de diminuer au fur et à mesure l'ambiguïté sur les classes, cette approche, peut être vue comme un filtrage progressif des décisions. Cela permet généralement de diminuer le taux d'erreur globale de la chaîne de reconnaissance. Cependant, une combinaison de ce type demeure particulièrement sensible à l'ordre dans lequel sont placés les classifieurs. Les premiers classifieurs doivent être robustes, même s'ils ne nécessitent pas d'être les plus performants. En cas de fausse décision du premier classifieur, l'erreur va se propager de façon irrévocable. Il faudra donc choisir judicieusement le premier classifieur afin d'éviter l'apparition d'une telle situation. La combinaison séquentielle suppose donc une certaine connaissance a priori du comportement de chacun des classifieurs. Il est à noter que dans cette approche, chaque classifieur est réglé en fonction du classifieur placé en amont de la chaîne. Une légère modification du premier classifieur peut inciter de refaire l'apprentissage des classifieurs suivants (Zouari, 2004).

- **Architecture hybride (hiérarchique) :** Dans ce cas, les deux architectures précédentes sont combinées afin de tirer profit des avantages de chacune d'elles (voir Fig.3.7 (c)).

Peu importe l'architecture de combinaison utilisée, un ensemble de classifieurs peut être hétérogène (Kim et al., 2000 ; Asafuddoula et al., 2017) ou homogène (Al-Hajj et al. 2007 ; Koerich et Poitevin, 2005). Dans le premier cas, l'ensemble est constitué de classifieurs de types différents, par exemple un HMM, un ANN et un SVM. Dans le cas d'un ensemble homogène, les classifieurs de base sont tous de même type. Dans le cadre de cette thèse, nous nous sommes intéressés, plus particulièrement, aux ensembles homogènes dans une architecture de combinaison parallèle qui va être abordée plus en détail dans la section suivante.

3.3.2. Ensembles de classifieurs homogènes dans une architecture de combinaison parallèle

Dans un ensemble de classifieurs homogènes, appelés aussi classifieurs faibles, est utilisée une même méthode de classification, et les mêmes algorithmes sont exploités pour construire tous les classifieurs de base. L'avantage principal de l'ensemble homogène par rapport à l'ensemble hétérogène est qu'elle n'exige pas l'implémentation de plusieurs algorithmes d'apprentissage et de reconnaissance, mais les mêmes algorithmes sont utilisés pour tous les classifieurs de base, ce qui simplifie leur implémentation et réduit l'espace mémoire occupé. Un autre avantage des ensembles homogènes est que la fusion des réponses des classifieurs de base est beaucoup plus simple, car ces réponses sont de même type, et donc ne nécessitent pas une étape de normalisation.

L'utilisation de toute méthode ensembliste dans une architecture de combinaison parallèle nécessite la résolution des trois problèmes suivants :

- Comment créer la différence entre les classifieurs ?
- Comment sélectionner le meilleur ensemble de classifieurs à combiner ?
- Comment fusionner les sorties (réponses) des classifieurs au niveau de décision ?

Dans ce qui suit, nous présenterons ces trois problèmes avec quelques travaux de la littérature de la reconnaissance de formes en général, vu le fait que très peu de travaux en RAP ont opté pour les méthodes ensemblistes.

3.3.2.1. Création de l'ensemble

Il est intuitif, que l'utilisation d'un ensemble de classifieurs identiques n'a aucun impact sur le résultat. Par conséquent, l'utilisation de classifieurs homogènes nécessite de les différencier. Deux classifieurs homogènes sont considérés comme différents, s'ils n'ont pas les mêmes performances, sachant que les performances dépendent de certains paramètres tels que, les données d'apprentissage, l'espace de caractéristiques, le type de réponse et les choix initiaux du classifieur, comme le réglage initial de l'algorithme d'apprentissage. C'est la raison par laquelle, il est courant de jouer sur ces paramètres pour créer la diversité entre les classifieurs de base.

Nous proposons de classifier les méthodes ensemblistes homogènes selon les deux critères suivants :

- **Selon la phase durant laquelle les classifieurs de base sont différenciés :** On peut créer la diversité entre les classifieurs de base au niveau d'extraction des caractéristiques et/ou au niveau d'apprentissage. Au niveau d'extraction des caractéristiques, la différence est généralement faite en utilisant différents sous-ensembles de caractéristiques (Fig.3.8 (a)), comme la méthode des sous-espaces aléatoires RSM (Random Subspace Method) (Ho, 1998)) qui permet d'apprendre des classifieurs sur des caractéristiques différentes. Au niveau d'apprentissage, la différence entre les classifieurs de l'ensemble peut être faite en utilisant différents sous-ensembles d'échantillons d'apprentissage (Fig.3.8 (b)), la sélection des sous-ensembles d'échantillons peut être faite de façon aléatoire tel que la méthode de Bagging (Breiman, 1996), ou non aléatoire telle que la méthode de Boosting (Freund, 1995) où les échantillons difficiles ont une plus grande probabilité d'être sélectionnés, et les plus faciles ont moins de chances d'être utilisés pour l'apprentissage. Il est aussi possible de créer la diversité, au niveau d'apprentissage, en utilisant différentes architectures (Fig.3.8 (c)) comme, par exemple, les approches proposées dans (Hansen et Salamon, 1990 ; Cho et Kim, 1995).

- **Selon la manière par laquelle les classifieurs de base sont différenciés :** Les méthodes ensemblistes homogènes peuvent être classées selon ce critère en trois catégories. La première concerne les méthodes qui utilisent différents sous-ensembles d'échantillons pour créer différents classifieurs, tels que le Bagging et le Boosting. Dans cette catégorie, la base d'apprentissage est divisée en plusieurs sous-ensembles, ce qui n'est pas souhaitable pour les classifieurs génératifs tels que les HMM, où de grandes quantités de données d'apprentissage sont requises

pour obtenir de bonnes performances. La deuxième catégorie comprend les méthodes utilisant différents sous-ensembles de caractéristiques, telles que RSM. Ces méthodes sont bien adaptées aux classifieurs génératifs et aux applications qui doivent traiter un nombre limité d'échantillons d'apprentissage. L'inconvénient de ces méthodes est leur grande complexité, car il est nécessaire de calculer différentes caractéristiques du même échantillon pour chaque classifieur. La dernière catégorie comprend les méthodes utilisant la même base d'apprentissage et les mêmes caractéristiques pour entrainer tous les classifieurs. La différence est simplement faite en faisant varier l'architecture ou certains paramètres pour lesquels le classifieur est instable. L'utilisation des mêmes caractéristiques pour tous les classifieurs conduit à une réduction considérable du temps de calcul et de l'espace des caractéristiques par rapport à la seconde catégorie. Les approches telle que celle proposée dans (Hamdi et Frigui, 2015) peuvent être classées dans cette catégorie.

Dans la figure 3.8, nous illustrons les différentes manières de création d'ensembles homogènes.

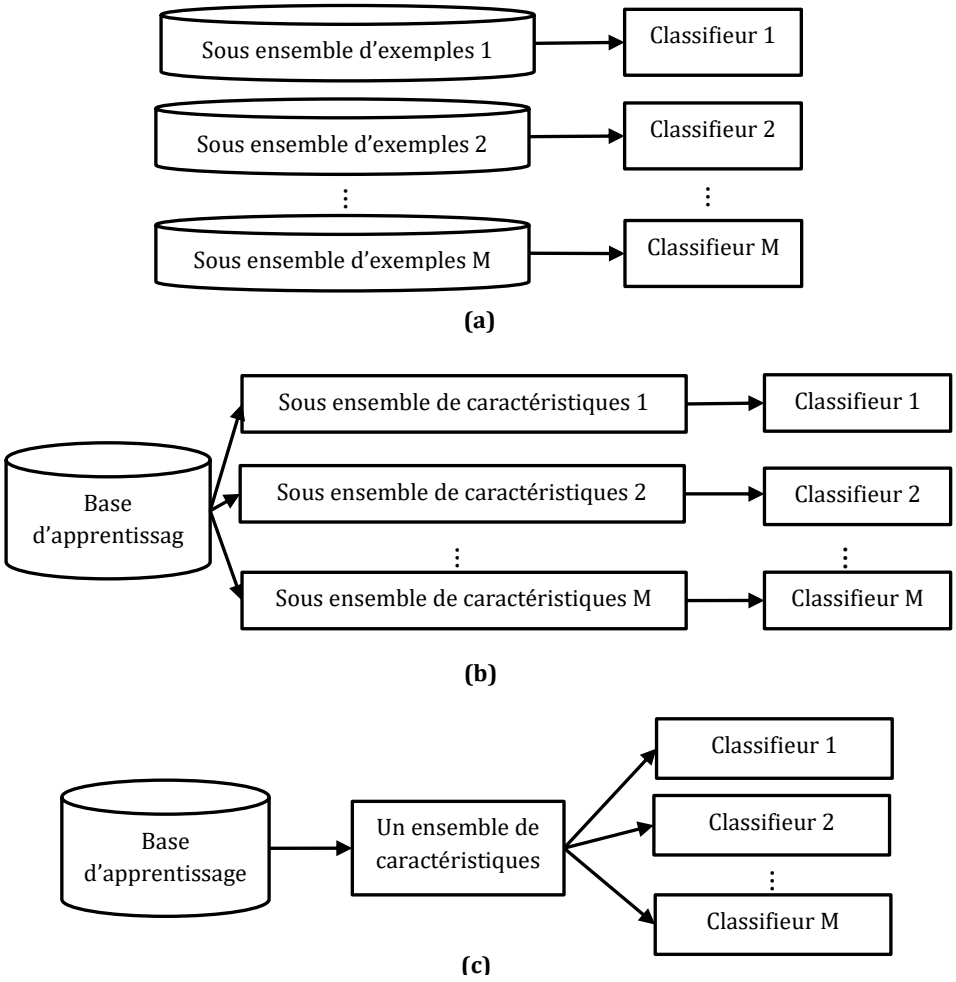


Fig.3.8. Création d'ensemble homogène (a) différents sous-ensembles d'échantillons, (b) différents sous-ensembles de caractéristiques et (c) différentes architectures ou paramètres (mêmes données et mêmes caractéristiques)

La différence entre les classifieurs de base peut être quantifiée en utilisant certaines mesures de *diversité* qui sont en deux types : Les mesures *pairwise* qui sont conçues pour comparer les différences dans les décisions d'une paire de classifieurs, et les mesures *non-pairwise* qui sont conçues pour mesurer la diversité au sein d'un ensemble de plus de deux classifieurs. Les mesures les plus utilisées sont résumées dans le tableau suivant :

Tableau 3.1. Mesures de diversité les plus utilisées dans la littérature

Mesure de diversité	Type	Référence
Q-statistic (Q)	Pairwise	(Yule, 1900)
Coefficient de corrélation : Correlation (ρ)	Pairwise	(Sneath et Sokal, 1973)
Mesure de désacord : Disagreement (D)	Pairwise	(Ho, 1998; Skalak, 1996)
Double Fault (DF)	Pairwise	(Giacinto, et Roli, 2001)
Entropie: Entropy of the votes (Ent)	*Non-pairwise*	(Cunningham et Carney, 2000)
Indice de difficulté : Difficulty Index (θ)	*Non-pairwise*	(Hansen, 1990)
Kohavi-Wolpert variance (kw)	*Non-pairwise*	(Kohavi et Wolpert, 1996)
Interrater Agreement (k)	*Non-pairwise*	(Dietterich, 2000; Fleiss et al., 2013)
Diversité généralisée: Generalized Diversity (GD)	*Non-pairwise*	(Partridge et Krzanowski, 1997)
Coincident Failure Diversity (CFD)	*Non-pairwise*	(Partridge et Krzanowski, 1997)

3.3.2.2. Sélection de l'ensemble

Selon la phase durant laquelle les classifieurs de base sont sélectionnés, nous pouvons distinguer deux catégories : La sélection statique et la sélection dynamique. La sélection statique d'ensemble est effectuée durant la phase d'apprentissage à l'aide d'une base de validation, elle est généralement basée sur l'évaluation de la qualité de l'ensemble en termes de performance, de diversité ou en termes des deux mesures tel que le travail présenté dans (Fern et Lin, 2008).

Il existe un consensus dans la littérature sur le fait que la diversité à travers les classifieurs de base joue un rôle important dans la qualité de l'ensemble, puisqu'il n'y a pas de gain de combinaison si les classifieurs de base fournissent des sorties identiques. Un algorithme d'optimisation est souvent utilisé pour explorer l'espace des classifieurs candidats, tels que les algorithmes génétiques (Hazmoune et al., 2013a ; Ekbal et Saha,

2010 ; Kim et Oh, 2008) et la recherche gourmande ou Greedy Search (Partalas et al., 2008 ; Mao et al., 2011 ; Ye et Dai, 2017). La sélection statique nécessite une base de validation. Ceci est problématique dans certains domaines de reconnaissance des formes, où la quantité d'échantillons disponible est limitée comme par exemple la vérification de signatures numériques. Contrairement à la sélection statique, la sélection dynamique est effectuée pendant la phase de reconnaissance, où un seul classifieur ou un ensemble de classifieurs différents est sélectionné pour chaque nouvel échantillon de test. Nous pouvons citer, à titre d'exemple de sélection dynamique, les travaux de (Batista et al. 2012) et (Ko et al. 2008). Pour un bon et récent état de l'art des méthodes et des critères de sélection dynamique d'ensemble de classifieurs, le lecteur peut se référer à (Cruz et al., 2018) et (Britto et al., 2014).

3.3.2.3. Fusion de l'ensemble

La fusion est l'une des étapes les plus importantes de toute méthode ensembliste. Etant donnés, un ensemble de M classifieurs différents, les classes candidates N, une forme à reconnaitre x et la réponse donnée à chaque classe c_i par chaque classifieur m, soit R_i^m, le problème est de trouver la classe optimale $c*$ que l'ensemble affecte à la forme x. Selon les sorties des classifieurs de base, nous pouvons distinguer trois types de méthodes de fusion. Celles basées sur la combinaison de prédictions (étiquettes de classes) des classifieurs de base comme le vote majoritaire, celles basées sur la fusion de réponses de type rang et celles basées sur la fusion des mesures données par chaque classifieur de base (par exemple, scores, distances, probabilités à postériori, vraisemblances, etc.). De nombreuses règles de fusion ont été utilisées dans la littérature, elles peuvent être résumées essentiellement dans ce qui suit.

a. Cas de réponses de type classe

La fusion dans ce cas consiste à faire voter les classifieurs de base. Il existe plusieurs variantes de vote :

- La règle de vote majoritaire : C'est la règle la plus simple et la plus communément utilisée, la classe ayant le plus de votes est choisie.

- La règle de vote unanime : La classe choisie est celle proposée par tous les classifieurs. S'il n'y a pas de consensus total entre les classifieurs, la décision finale est un rejet.

- La règle de vote notoire : Elle consiste à choisir la classe majoritaire qui se distingue de la deuxième classe par une différence supérieure à un certain seuil.

- La règle de vote pondéré : La réponse de chaque classifieur est multipliée par un poids reflétant le degré de confiance de chaque classifieur.

b. Cas de réponses de type rang

Les méthodes de type rang permettent de résoudre les problèmes des méthodes de vote. Dans une situation d'ambiguïté, où la classe majoritaire n'existe pas en tête de liste (des classes ayant le même nombre de votes), il est raisonnable d'examiner la suite de chaque

liste pour lever cette ambiguïté, comme le montre l'exemple suivant d'un ensemble de 3 classifieurs avec 4 classes (Zouari, 2004) :

Classifieur1 : A|B C D

Classifieur2 : C|A B D

Classifieur3 : B|D C A

Il existe plusieurs variantes de ce type de méthodes, telles que l'intersection, l'union, le meilleur rang, la somme des rangs, etc. Une présentation intensive de ces méthodes se trouve dans (Zouari, 2004 ; Barrault, 2008).

c. Cas de réponses de type mesure

Nous citons ci-après quelques règles de fusion de type mesure :

- La règle de produit : Cette règle permet de choisir la classe maximisant le produit des mesures données par les classifieurs.

$$c^* = arg \max_{1 \le i \le N} \prod_{m=1}^{M} R_i^m \tag{3.9}$$

  L'inconvénient de cette règle est que lorsque l'une des mesures est égale à zéro, ce genre de combinaison devient problématique.

- La règle de somme

$$c^* = arg \max_{1 \le i \le N} \sum_{m=1}^{M} R_i^m \tag{3.10}$$

- La règle de médiane

$$c^* = arg \max_{1 \le i \le N} Med_{m=1}^{M} R_i^m \tag{3.11}$$

- La règle de moyenne

$$c^* = arg \max_{1 \le i \le N} \frac{1}{M} \sum_{m=1}^{M} R_i^m \tag{3.12}$$

  Cette règle est pratiquement équivalente à celle de la somme.

Il y a aussi la règle du maximum et la règle du minimum. Pour plus de détail sur les règles de fusion, le lecteur peut se référer à (Zouari, 2004 ; Barrault, 2008).

Dans le tableau 3.2, nous résumons quelques travaux de la littérature des méthodes ensemblistes homogènes dans différents domaines d'application. Il est à noter que contrairement aux autres domaines de reconnaissance de formes, notamment la

reconnaissance de l'écriture, très peu de méthodes ensemblistes homogènes se sont orientées vers la reconnaissance de la parole, ce qui justifie l'absence des travaux appliqués à la RAP dans le tableau 3.2.

Tableau 3.2. Quelques travaux antérieurs concernant les méthodes ensemblistes homogènes

Référence	Classifieurs de base	Création d'ensemble	Sélection d'ensemble	Fusion des sorties	Application
Gunter et Bunke 2002	10 HMM	Sélection des sous-ensembles différents des caractéristiques	Tous les classifieurs individuels	Vote majoritaire	Reconnaissance hors-ligne des mots manuscrits cursives
Koerich et Poitevin 2005	3 NN	Différents sous-ensembles de caractéristiques extraites de différentes parties (début, milieu et fin)	Tous les classifieurs individuels	Maximum, somme et produit	Classification des types de music
Zelaia et al. 2011	30 k-NN	Sous-ensembles différents d'échantillons	Tous les classifieurs individuels	Vote Bayésien	Catégorisation des documents
Azizi et al. 2013	3 SVM	Sous-ensembles différents des caractéristiques	Tous les classifieurs individuels	Vote majoritaire	Cancer des seins

3.4. LES APPROCHES HYBRIDES

Pour tirer profit des avantages de plusieurs méthodes, leur hybridation semble constituer une approche intéressante pour la robustesse des SRAP (Spalanzani, 1999). On parle d'approche hybride lorsqu'une méthode de classification est intégrée au sein d'une autre. Dans ce cadre, les systèmes les plus répandus sont basés sur une hybridation HMM/NN ou HMM/SVM dans le but est, essentiellement, l'amélioration de la capacité de discrimination et le pouvoir de généralisation des HMM. Ces hybridations sont, le plus souvent, utilisées comme alternatives de l'approche HMM/GMM (GMM pour Gaussian Mixture Models, en anglais). Nous présentons, dans les paragraphes qui suivent, une revue bibliographique de quelques approches hybrides pour la RAP.

Les auteurs dans (Abdel-Hamid et al., 2012), ont proposé d'appliquer les CNN dans un cadre du modèle hybride NN/HMM. Dans cette approche, une paire de couche de filtrage local et de couche de max-pooling est ajoutée à l'extrémité la plus basse du réseau de neurones (NN) pour normaliser les variations spectrales des signaux vocaux. L'architecture CNN proposée a été évaluée pour une tâche de reconnaissance de la parole en mode indépendant du locuteur sur la base de données standard TIMIT. Les résultats expérimentaux ont montré que la méthode CNN proposée peut atteindre une réduction

d'erreur relative supérieure à 10% comparée à un NN standard utilisant le même nombre de poids et de couches cachés.

Dans (Ganapathiraju et al., 2000), un système hybride SVM/HMM a été proposé. Les SVM ont été intégrés dans un cadre de reconnaissance basé sur les HMM, dans le but était d'améliorer les performances des HMM en introduisant de classifieurs puissants, SVM, qui sont entrainés de manière discriminante. L'idée consiste d'abord, étant donné des classifieurs SVM et un système HMM, d'apprendre les classifieurs SVM sur les données au niveau trame et de les utiliser, ensuite, comme classifieurs dans chaque état du HMM. Le système proposé a été évalué sur un corpus nommé « OGI Alphadigits » ayant un vocabulaire de 36 mots, une amélioration des performances du système hybride par rapport au système HMM de base a été marquée (un WER égal à 11,6%, contre 12,7%).

Dans (Zarrouk & Benayed, 2016), un système hybride SVM/HMM a été proposé. Les SVM sont incorporés au sein d'une architecture de reconnaissance markovienne au niveau de l'apprentissage et du test. Le rôle des SVM est d'estimer les probabilités d'émission des observations émises par les états des HMM, qui reformuleront par la suite les probabilités de générer la séquence optimale par l'algorithme de décodage utilisé par HMM. Le système a été évalué sur un corpus de phonèmes arabes. Le taux de reconnaissance obtenu est 75.8% pour le système SVM/HMM, 73.78% pour un système MLP/HMM et 66.98% pour le classifieur HMM de base.

Une autre idée de concevoir des approches hybrides consiste à intégrer la méthode k-NN, au niveau d'états des HMM, comme alternative de modèles de mélange de gaussiennes (GMM). Cette approche a été utilisée dans quelques domaines de reconnaissance de formes avant d'être introduite en RAP par Deselaers et ses collaborateurs dans leur travail (Deselaers et al., 2007). Comme la méthode k-NN standard ne permet pas d'estimer les probabilités d'émission des observations par les états des HMM, les auteurs ont utilisé une version étendue de la méthode basée sur des densités noyau (kernal densities). L'approche proposée a été évaluée sur un corpus allemand « SieTillcorpus » de chaînes de chiffres continues et un corpus anglais à grand vocabulaire nommé « EPPS ». Les résultats obtenus ont montré des performances comparables aux celles du système à base de GMM (un WER de 1.96% pour le système k-NN/HMM, et 1.84% pour le système GMM/HMM).

D'autres travaux à base des approches hybrides ont aussi été publiés, tels que HMM/DNN (Miao et al., 2015), NN/HMM (Xue et al., 2016), SVM/HMM (Zarrouk et al, 2018), etc.

3.5. CHOIX D'UNE METHODE DE CLASSIFICATION

Nous avons présenté plusieurs méthodes de classification utilisées couramment en RAP, chacune d'elles ayant ses avantages et ses inconvénients. Face à une telle gamme de méthodes, le problème du choix de la meilleure méthode se posera à un moment ou à un autre. Bien évidemment, certaines méthodes peuvent être préférées car elles tiennent compte d'une certaine connaissance préalable de la forme des données, telles que les HMM qui sont de leur nature adaptés à la modélisation des séries temporelles de longueurs variables (les signaux de parole en l'occurrence), d'autres peuvent être

préférées en raison de leur faible complexité de calcul, d'autres méthodes peuvent être choisies grâce à leurs performances élevées lorsqu'elles sont appliquées sur les mêmes bases de données, d'autres sont préférées en raison de leur robustesse face au perturbations de l'environnement, d'autres sont choisies car elles ont une bonne capacité de généralisation ou de discrimination, etc. Cependant, il est bien connu qu'une méthode appliquée avec succès dans un domaine d'application peut ne pas l'être dans un autre. De plus, pour le même domaine, les performances d'une méthode peuvent se dégrader considérablement suite à une légère modification dans la base de données ou dans les paramètres d'initialisation de l'apprentissage. Cela rend le choix d'une méthode de classification un problème ouvert. En effet, de nombreuses publications ayant abordé ce problème indiquent qu'il n'existe pas de règles ou critères universels permettant de favoriser une méthode, un classifieur, ou un modèle. Nous citons ci-après quelques conclusions tirées de la littérature :

*"In practice, the choice of a classifier is a difficult problem and it is often based on which classifier(s) happen to be available, or best known, to the user" (*Jain et al., 2000, p37-38).

"On the criterion of generalization performance, there are no context- or problem-independent reasons to favor one learning or classification method over another. The apparent superiority of one algorithm or set of algorithms is due to the nature of the problems investigated and the distribution of data" (Duda et al., 2001, chapitre 9, p4).

"Referring to classification problems, Wolpert's theorem has a specific lecture: there is not a single classifier modeling approach which is optimal for all pattern recognition tasks, since each has its own domain of competence" (Woźniak et al., 2014, p1).

« It is intuitively clear that simple models or stable classifiers are less likely to be overtrained than more sophisticated models. However, simple models might not be versatile enough to fit complex classification boundaries. More complex models have a better flexibility but require more system resources and are prone to overtraining" (Kuncheva, 2014, p 82).

Pour éviter le problème du choix des classifieurs (ou modèles), il est courant d'utiliser les approches hybrides et ensemblistes afin de tirer profit des avantages de plus d'un classifieur (modèle) tout en allégeant leurs inconvénients. Les approches ensemblistes (Kuncheva, 2014) sont basées sur un certain degré d'infraction du principe du *Rasoir d'Occam* disant « *Entities should not be multiplied beyond necessity* ». Néanmoins, dans une situation où un classifieur individuel et un ensemble donnent la même erreur de généralisation, l'ensemble (plus complexe) peut toujours être préférable en raison de sa robustesse espérée.

En résumé, pour choisir une méthode de classification ou une autre, plusieurs aspects peuvent être pris en considération, l'importance de chaque aspect dépend fortement de la problématique adoptée. Les aspects les plus considérés sont les performances (précision), la robustesse, la complexité, le pouvoir de généralisation (la capacité prédictive du classifieur) et la nature des données utilisées (discrètes, continues, chaine,

etc.). Par conséquent, nos choix ne peuvent être validés que par expérimentation sur des bases de données appropriées aux besoins de l'application visée.

3.6. CONCLUSION

L'exposé que nous venons de faire nous a permis de découvrir les méthodes et les approches de classification utilisées couramment en RAP. Nous avons vu que chaque méthode de classification de base a ses avantages, ses inconvénients, ainsi que son domaine de compétence. Cela rend le choix d'une méthode ou d'une autre un problème délicat qui dépend, essentiellement, des besoins de l'application visée. Afin d'exploiter les atouts de différentes méthodes, tout en allégeant leurs inconvénients, les chercheurs ont tendance à adopter plus d'une méthode de classification dans un cadre ensembliste ou hybride.

Parmi les différentes approches de classification présentées, notre choix s'est porté sur les approches hybrides et ensemblistes, basées sur les modèles de Markov cachés. Ces derniers sont également utilisés, comme nous l'avons noté, dans la majorité des systèmes hybrides de la littérature. Leur popularité est justifiée, principalement, par leur capacité à modéliser efficacement les variations temporelles de la parole. Ces modèles seront largement développés dans le chapitre suivant.

CHAPITRE 4

LES MODELES DE MARKOV CACHES ET LE PROBLEME DU REGLAGE INITIAL DES PARAMETRES DE L'ALGORITHME EM

SOMMAIRE

4.1. INTRODUCTION

Nous nous intéressons dans le présent chapitre aux modèles de Markov cachés (HMM), qui sont à la base des travaux proposés dans cette thèse. Les HMM sont des méthodes stochastiques dont la théorie de base a été initialement publiée à la fin des années 1960 et au début des années 1970 par Baum et ses collègues (Baum & Petrie, 1966 ; Baum & Eagon, 1967 ; Baum, 1972). Cette théorie, a été mise en œuvre pour des applications de traitement de la parole par Baker (Baker, 1975) à CMU et par Jelinek et ses collègues à IBM (Jelinek, 1969 ; Bahl et al., 1983). Leur utilisation en RAP remonte au début des années 1980, principalement grâce aux travaux de quelques chercheurs tels que Levinson et ses collaborateurs (Levinson et al., 1983 ; Levinson, 1985) et Rabiner et son équipe (Rabiner et al., 1984 ; Rabiner, 1986 ; Rabiner, 1989). Actuellement, les HMM sont largement utilisés non seulement en reconnaissance de la parole, mais aussi dans de nombreux domaines de reconnaissance de formes, tels que la reconnaissance de l'écriture manuscrite (Bougamouza et al., 2018, Bougamouza et al., 2016 ; Samanta, 2018 ; Rabi et al., 2018, Sagar et al., 2020), la reconnaissance de visages (Varma et al., 2020 ; Bobulski, 2016), etc.

Dans ce chapitre, nous allons d'abord présenter la définition d'un HMM et les trois problèmes à résoudre en utilisant ce formalisme. Ensuite, nous abordons le principe de l'apprentissage markovien et la problématique du réglage initial des paramètres de l'apprentissage par la méthode EM (Expectation Maximisation). Nous montrons, après, comment les HMM sont appliqués pour la RAP, et nous présentons quelques travaux de la littérature.

4.2. DEFINITION D'UN HMM

Un HMM est un processus doublement stochastique (Rabiner et Juang 1986). Il a deux propriétés principales. Premièrement, il suppose qu'une séquence d'observations $O = (o_1, o_2, \dots, o_T)$ est produite par une séquence d'états cachés $Q = (q_1, q_2, \dots, q_T)$. Autrement dit, la séquence d'états ayant généré la séquence d'observations est cachée à l'observateur, d'où son nom (caché). Deuxièmement, il vérifie la propriété de Markov qui suppose que, l'état du processus au temps t ne dépend que de son état au temps $t - 1$. En se basant sur cette propriété, on peut déduire que la probabilité $P(q_1, q_2, \dots, q_T)$ que le processus passe par la séquence d'états $Q = (q_1, q_2, \dots, q_T)$ peut être calculée comme suit :

$$P(q_1, q_2, \dots, q_T) = P(q_1) * P(q_2/q_1) * \dots * P(q_T/q_{T-1}) \quad (4.1)$$

Formellement, un HMM à N états et M symboles d'observations discrètes qui peuvent être émis par les états au cours du temps est définit par le triplet $\lambda = (\Pi, A, B)$, tel que :

- Π est un vecteur à N éléments $\pi_i = P(s_0 = s_i)$, $1 \leq i \leq N$, représentants les probabilités que le processus démarre d'un état donné.

- A est la matrice de transition de taille (NxN) contenant les probabilités $a_{ij} = P(s_j/s_i)$ de passer d'un état à un autre.
- B est la matrice d'observation de taille (NxM). Un coefficient $b_i(O_j)$ de B représente la probabilité que le symbole O_j soit émis par l'état s_i.

Les contraintes stochastiques suivantes doivent être vérifiées :

$$\sum_{i=1}^{N} \pi_i = 1, \tag{4.2}$$

$$\sum_{j=1}^{N} a_{ij} = 1, \qquad 1 \leq i \leq N, \tag{4.3}$$

$$\sum_{j=1}^{M} b_i(O_j) = 1, \;\; 1 \leq i \leq N \tag{4.4}$$

Il est à noter que cette définition concerne les HMM à densité discrète. Pour les HMM à densité continue, où on ne dispose pas de symboles d'observation discrets, la matrice d'observation B est remplacée par les paramètres de la loi de probabilité utilisée pour évaluer les probabilités d'observation. En cas de la loi multi-gaussienne (GMM pour Gaussian Mexture Model), utilisée généralement dans les problématiques de reconnaissance de formes, la matrice B est remplacée par les vecteurs moyens et les matrices de covariance des densités gaussiennes. Chaque densité de probabilité associée à un état i est calculée en appliquant la formule :

$$\mathcal{N}(\mu_i, \Sigma_i, O) = \frac{1}{(2\pi)^{P/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(O-\mu_i)^T \Sigma_i^{-1}(O-\mu_i)} \tag{4.5}$$

Où ; P est la dimension (nombre d'éléments) du vecteur O, μ_i le vecteur moyen de la fonction de densité pour l'état i, $|\Sigma_i|$ le déterminant de la matrice de covariance de la fonction de densité associée à l'état i, Σ_i^{-1} est l'inverse de la matrice de covariance de l'état i, et T le nombre moyen d'observations (trames) par séquence.

Les probabilités d'observation $b_i(O_t)$ sont calculées comme une somme pondérée des fonctions de densité gaussienne $\mathcal{N}(\mu_i, \Sigma_i, O_t)$ associées à l'état i.

Dans la littérature, les HMM qui utilisent une distribution multi-gaussiennes des probabilités d'observation sont souvent qualifiés de HMM/GMM.

Un HMM peut être également représenté par un graphe orienté et pondéré, tel que, les sommets représentent les états, les arcs indiquent les transitions entre états, et les poids des arcs correspondent aux probabilités de transition. La matrice de transition permet de définir la topologie du modèle. Un HMM peut avoir une topologie gauche-droite ou

ergodique. Dans le modèle gauche-droite (cf. figure 4.1(a)), seul le bouclage sur le même état ou les transitions vers les états d'indices supérieurs sont autorisées ; les probabilités de transition $a_{ij}, avec\ j < i$ sont donc nulles. Quant à la topologie ergodique, toutes les transitions sont autorisées (cf. figure 4.1(b)).

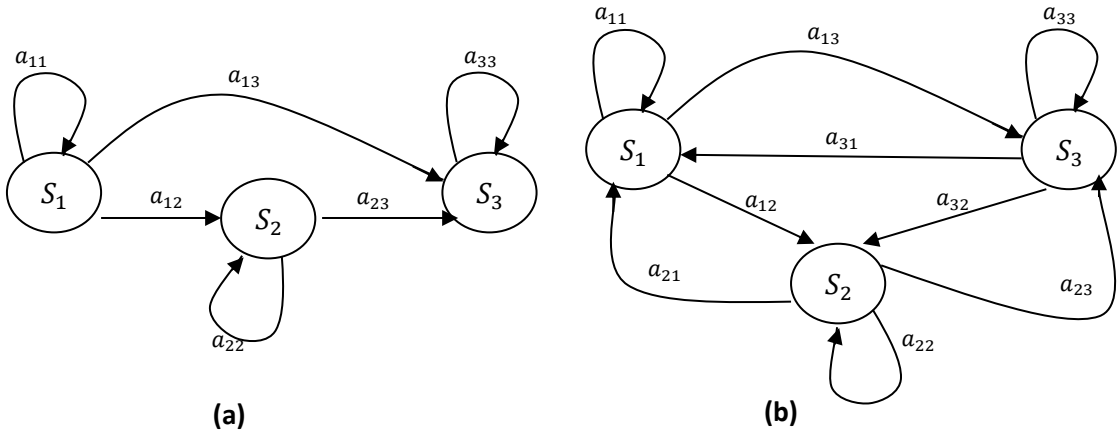


Fig.4.1. Représentation graphique d'un HMM à 3 états
(a) Modèle gauche-droite, (b) modèle ergodique

4.3. LES TROIS PROBLEMES FONDAMENTAUX EN HMM ET LEURS SOLUTIONS

L'utilisation des HMM pour des applications réelles nécessite la résolution de trois problèmes fondamentaux qui sont bien définis par Rabiner (Rabiner, 1988 ; Rabiner, 1989) comme suit :

- Problème d'évaluation (calcul de vraisemblance) : Étant donné la séquence d'observation $O = (o_1, o_2, \dots, o_T)$ et un modèle $\lambda = (\Pi, A, B)$, comment peut-on calculer efficacement $P\ (O/\lambda)$, la probabilité d'observer la séquence O, étant donné le modèle λ ?
- Problème de décodage (reconnaissance) : Étant donné la séquence d'observation $O = (o_1, o_2, \dots, o_T)$ et le modèle λ, comment choisir la séquence d'états $Q = (q_1, q_2, \dots, q_T)$ optimale ayant généré O ?
- Problème de réestimation (apprentissage) : Comment ajuster les paramètres du modèle $\lambda = (\Pi, A, B)$ pour maximiser la vraisemblance $P\ (O/\lambda)$?

Nous allons, dans les sous sections qui suivent, présenter les solutions de ces trois problèmes telles qu'elles sont décrites par Rabiner dans ses travaux (Rabiner, 1988 ; Rabiner, 1989). Ces travaux peuvent être considérés comme références de base pour la reconnaissance de la parole par les HMM.

4.3.1. Solution du problème d'évaluation

On veut calculer la probabilité de la séquence d'observation, $O = (o_1, o_2, \dots, o_T)$, étant donné le modèle λ, c-à-d., la vraisemblance $P\ (O/\lambda)$. La manière la plus simple de le faire,

consiste à énumérer toutes les séquences d'états possibles de longueur T (la longueur de la séquence d'observations). Considérons une telle séquence d'état fixe, $Q = (q_1, q_2, \dots, q_T)$, où q_1 est l'état initial. La probabilité de la séquence d'observation, O, pour la séquence d'états, Q, est calculée en appliquant l'équation 4.6 :

$$P(O/Q, \lambda) = b_{q1}(o_1).\, b_{q2}(o_2) \dots\ b_{qT}(o_T) \tag{4.6}$$

La probabilité d'une telle séquence d'états, Q, peut s'écrire de la manière suivante :

$$P(Q/\lambda) = \pi_{q1}\ a_{q_1 q_2}\ a_{q_2 3}\ \dots\ a_{q_{T-1} q_T} \tag{4.7}$$

La probabilité conjointe de O et Q, qui est la probabilité que O et Q apparaissent simultanément, est simplement le produit des deux termes ci-dessus, c.à.d.,

$$P(O, Q/\lambda) = P(O/Q, \lambda)\, P(Q, \lambda) \tag{4.8}$$

La probabilité de O (étant donné le mode λ) est obtenue en calculant la somme de cette probabilité conjointe sur toutes les séquences d'états possibles j, ce qui donne :

$$P(O/\lambda) = \sum_{tout\ Q} P(O/Q, \lambda)\, P(Q/\lambda) \tag{4.9}$$

$$= \sum_{q_1, q_2, \dots, q_T} \pi_{q_1} b_{q_1}(o_1)\ a_{q_1 q_2}\, b_{q_2}(o_2)\ \dots\ a_{q_{T-1} q_T} b_{q_T}(o_T) \tag{4.10}$$

L'interprétation de l'équation ci-dessus est comme suit : Initialement (au temps $t = 1$), on est dans l'état q_1 avec la probabilité π_{q_1} et on génère le symbole o_1 (dans cet état) avec la probabilité $b_{q_1}(o_1)$. Au temps *t = t + 1 (t = 2)*, on réalise une transition vers l'état q_2 à partir de l'état q_1 avec une probabilité $a_{q_1 q_2}$ et génère le symbole o_2 avec une probabilité $b_{q_2}(o_2)$. Ce processus se poursuit de cette manière jusqu'à la dernière transition (au temps T), de l'état q_{T-1} à l'état q_T avec une probabilité $a_{q_{T-1} q_T}$ et génère le symbole o_T avec une probabilité $b_{q_T}(o_T)$.

Un peu de réflexion devrait convaincre le lecteur que le calcul de $P\,(O/\lambda)$, selon sa définition directe (équation 4.10), nécessite $2T * N^T$ opérations. En effet, à chaque instant $t = 1, 2, \dots, T$, il y a N états possibles qui peuvent être atteints, c-à-d., qu'il existe N^T séquences d'états possibles, et pour chaque séquence d'états, environ $2T$ opérations sont nécessaires pour la somme de l'équation (équation 4.10), plus précisément, on est besoin de $(2T - 1)N^T$ multiplications et $N^T - 1$ additions. Ce calcul est irréalisable, même pour de petites valeurs de N et T; par exemple pour $N = 5$ (états), $T = 100$ (observations), il y a $2 * 100 * 5^{100} \approx 10^{72}$ opérations. Bien évidemment, une procédure plus efficace (moins complexe) est nécessaire pour résoudre le problème d'évaluation. Heureusement, une telle procédure existe et s'appelle l'Algorithme Forward-Backward (Baum, 1972). Le principe de cet algorithme est le suivant :

On considère la variable Forward, $\alpha_t(i)$, définie par :

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = S_i/\lambda) \tag{4.11}$$

Cela correspond à la probabilité d'observer la séquence d'observations partielle $O = (o_1, o_2, \dots, o_t)$ et d'être dans l'état S_i au temps t, étant donné le modèle λ. On peut calculer $\alpha_t(i)$ en appliquant la procédure Forward de manière itérative, comme suit :

1. Initialisation

$$\alpha_1(i) = \pi_i b_i(o_1), \qquad 1 \le i \le N \tag{4.12}$$

2. Itérations

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}), \quad 1 \le t \le T-1, \quad 1 \le j \le N \tag{4.13}$$

3. Terminaison

$$P(O/\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{4.14}$$

La première étape permet d'initialiser la variable $\alpha_1(i)$ en tant que probabilité conjointe de l'état S_i et l'observation initiale o_1. L'étape d'itérations, qui est au cœur du calcul Forward, montre comment l'état S_i peut être atteint au temps $t+1$ à partir des N états possibles S_i $(1 \le i \le N)$, au temps t. Etant donné que $\alpha_t(i)$ est la probabilité que l'événement conjoint que les observations $o_1, o_2, \dots, o_t$ soient observées et que l'état au temps t soit S_i, le produit $\alpha_t(i) * a_{ij}$ est alors la probabilité de l'événement conjoint que $o_1, o_2, \dots, o_t$ soient observés et l'état S_j soit atteint au temps $t+1$ via l'état S_i au temps t. La somme de ce produit sur tous les N états possibles S_i $(1 \le i \le N)$, au temps t a, pour résultat, la probabilité de S_j à l'instant $t+1$ avec toutes les observations partielles précédentes qui l'accompagnent. Une fois que cela est fait et que S_j est connu, il est facile de voir que $\alpha_{t+1}(j)$ est obtenu en prenant en compte l'observation de o_{t+1} dans l'état j, c-à-d., en multipliant la quantité additionnée par la probabilité $b_j(o_{t+1})$. Le calcul de l'équation (équation 4.13) est fait pour tous les états $j, 1 \le j \le N$, pour un t donné. Le calcul est ensuite itéré pour $t = 1, 2, \dots, T-1$. Enfin, l'étape de terminaison indique le calcul de $P(O/\lambda)$ en tant que la somme des variables terminales en Forward, $\alpha_T(i)$. Cela est justifié puisque, par définition nous avons,

$$\alpha_T(i) = P(o_1\ o_2,\ \dots\ o_T, q_T = S_i\ /\lambda) \tag{4.15}$$

Et, par conséquent, P (O/λ) est simplement la somme des $\alpha_T(i)$.

Pour le calcul de $\alpha_t(i)$, $1 \leq t \leq T$, $1 \leq i \leq N$, on voit qu'il nécessite N^2T opérations, plutôt que $2T * N^T$, comme l'exige le calcul direct (équation 4.10). Pour être précis, nous avons besoin de $N(N+1)(T-1)+N$ multiplications et $N\ (N-1)(T-1)$ additions. Pour $N = 5$, et $T = 100$, nous avons besoin d'environ 3000 opérations pour l'algorithme Forward, comparé à 10^{72} pour la méthode directe.

De la même façon, on peut considérer la variable Backward, définie comme suit :

$$\beta_t(i) = P(o_{t+1}\ o_{t+2},\ \ldots\ o_T/q_t = S_i, \lambda) \tag{4.16}$$

Elle correspond à la probabilité de la séquence d'observations partielle de $t+1$ à la fin (T), étant donnés l'état S_i au temps t et le modèle λ. La variable $\beta_t(i)$ peut être calculée de manière itérative, comme suit :

1. Initialisation

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \tag{4.17}$$

2. Itérations

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\ b_j(o_{t+1})\ \beta_{t+1}(j), \quad t = T-1, T-2, \ldots, 1, \quad 1 \leq i \leq N \tag{4.18}$$

L'étape d'initialisation définit arbitrairement $\beta_T(i)$ comme étant 1 pour tout i. L'étape d'itérations montre que, pour être dans l'état S_i au temps t et prendre en compte la séquence d'observation à partir de l'instant $t+1$, il faut considérer tous les états possibles, S_j au temps $t+1$, en tenant compte de la transition de S_i à S_j (le terme a_{ij}), ainsi que l'observation o_{t+1} dans l'état j (le terme $b_j(o_{t+1})$), puis considérer la séquence d'observations partielle restante de l'état j (le terme $\beta_{t+1}(j)$). Le calcul de $\beta_t(i)$, 1≤ t≤T, $1 \leq j \leq N$, nécessite N^2T opérations.

Il est possible de combiner les deux procédures Forward et Backward pour résoudre le problème d'évaluation. Dans ce cas, la vraisemblance $P(O/\lambda)$ est calculée comme suit :

$$P(O/\lambda) = \sum_{i=1}^{N} \alpha_t(i) * \beta_t(i)\,, 1 \leq t \leq T, 1 \leq j \leq N \tag{4.19}$$

On peut également considérer les deux cas particuliers :

- Si $t = 0$:

$$P(O/\lambda) = \sum_{i=1}^{N} \pi_i \beta_1(i)\,, 1 \leq j \leq N \tag{4.20}$$

- Si $t = T$:

$$P(O/\lambda) = \sum_{i=1}^{N} \alpha_T(i)\,, 1 \leq j \leq N \tag{4.21}$$

4.3.2. Solution du problème de décodage

Ce problème est également appelé problème de reconnaissance lorsqu'il est appliqué en RAP. Il existe plusieurs façons de résoudre ce problème, à savoir la recherche de la séquence d'états optimale associée à la séquence d'observations donnée. La difficulté réside dans la définition de la séquence d'état optimale, car il existe plusieurs critères d'optimalité possibles. Par exemple, un critère d'optimalité possible consiste à choisir les états, q_t qui sont individuellement les plus probables. Ce critère maximise le nombre espéré d'états individuels corrects. Pour implémenter cette solution pour le problème 2, nous définissons la variable :

$$\gamma_t(i) = P(q_t = S_i/O, \lambda) \tag{4.22}$$

Qui représente la probabilité d'être dans l'état S_j au temps t, étant donné la séquence d'observations, O, et le modèle λ. L'équation 4.22 peut être exprimée simplement en termes de variables Forward-Backward, comme suit,

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\mathrm{P\ (O\ /\lambda)}} \tag{4.23}$$

Puisque $\alpha_t(i)$ correspond à la sous-séquence d'observations $o_1, o_2, \dots, o_t$ et l'état S_i au temps t, tandis que $\beta_t(i)$ concerne le reste de la séquence d'observations $o_{t+1}, o_{t+2}, \dots, o_T$ étant donné S_i au temps t. Le facteur de normalisation $P\ (O/\lambda)$ rend $\gamma_t(i)$ une mesure de probabilité vérifiant :

$$\sum_{i=1}^{N} \gamma_t(i) = 1 \tag{4.24}$$

En utilisant $\gamma_t(i)$, on peut trouver l'état le plus probable, q_t , au temps t, comme suit :

$$q_t = \underset{1 \le i \le N}{\operatorname{argmax}} \left[\gamma_t(i)\right], \quad 1 \le t \le T \tag{4.25}$$

Bien que l'équation 4.25 maximise le nombre espéré d'états corrects, on pourrait avoir des problèmes avec la séquence d'états résultante. Par exemple, lorsque le HMM a des transitions d'états ayant une probabilité nulle (a_{ij} = 0 pour certains i et j), la séquence d'états optimale peut même ne pas être une séquence d'états valide. Cela est dû au fait que la solution de l'équation 4.25 détermine simplement l'état le plus probable à chaque instant, sans tenir compte de la probabilité d'apparition des séquences d'états.

Une solution possible au problème de décodage consiste à modifier le critère d'optimalité. Par exemple, on pourrait résoudre la séquence d'états qui maximise le nombre espéré de paires d'états correctes ($q_{t,}q_{t+1}$), ou de triples d'états ($q_{t,}q_{t+1}, q_{t+2}$), etc. Bien que ces critères soient raisonnables pour certaines applications, le critère le plus utilisé est de rechercher la meilleure séquence d'états (chemin), c-à-d., de maximiser $P(Q/0, \lambda)$, ce qui est équivalent à maximiser $P(Q, 0\ /\lambda)$. Pour trouver cette séquence d'états optimale, il

existe une technique formelle basée sur des méthodes de programmation dynamique. Cette technique est appelée algorithme de Viterbi (Viterbi, 1967 ; Forney, 1973).

Algorithme de Viterbi : Pour trouver la séquence optimale d'états, $Q = \{q_1, q_2, \dots, q_T\}$, ayant généré la séquence d'observations donnée $O = \{o_1, o_2, \dots, o_T\}$, il faut définir la quantité :

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_T} P[q_1 q_2 \dots q_t = i, o_1, o_2, \dots, o_t/\lambda] \quad (4.26)$$

c.à.d., que $\delta_t(i)$ est la probabilité la plus élevée d'un chemin unique, ayant généré les t premières observations et que l'état au temps t est S_i. Par induction on a :

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) a_{ij}\right] . \ b_j(o_{t+1}) \quad (4.27)$$

Pour récupérer la séquence d'états, on doit garder trace de l'argument qui a maximisé l'équation 4.27, pour chaque t et j. Cela est fait via un tableau $\psi_t(j)$. La procédure complète permettant de trouver la meilleure séquence d'états peut, donc, être définie comme suit :

1. Initialisation

$$\delta_1(i) = \pi_i b_i(o_1), 1 \le i \le N \quad (4.28a)$$

$$\psi_1(i) = 0 \quad (4.28b)$$

2. Itérations

$$\delta_t(j) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(o_t), \ \ 2 \le t \le T, \quad 1 \le j \le N \quad (4.29a)$$

$$\psi_t(j) = \operatorname*{argmax}_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right], \quad 2 \le t \le T, \quad 1 \le j \le N \quad (4.29b)$$

3. Terminaison

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \quad (4.30a)$$

$$q_T^* = \operatorname*{argmax}_{1 \le i \le N} [\delta_T(i)] \quad (4.30b)$$

4. Retour-arrière (Back-tracking)

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \ \ t = T-1, T-2, \dots, 1 \quad (4.31)$$

Il est à noter que l'algorithme de Viterbi est similaire (à l'exception de l'étape de retour-arrière) dans la mise en œuvre à la procédure Forward. La principale différence réside dans la maximisation de l'équation 4.29a par rapport aux états précédents utilisés à la place de la sommation dans l'équation 4.13.

En RAP, l'algorithme de Viterbi peut être allégé en ignorant le calcul des $\psi_t(i)$ et l'étape du Backtracking, car les états de la séquence optimale ne sont liés à aucun phénomène physique, ce qui nous intéresse est la probabilité du chemin optimal et pas le chemin lui-même. Nous allons montrer plus loin dans ce chapitre (section, 4.6) l'application de cet algorithme dans le cadre de la RAP.

4.3.3. Solution du problème de réestimation

Ce problème est également appelé problème d'apprentissage lorsqu'il est appliqué en RAP. C'est le problème le plus difficile à résoudre, car, il n'existe aucun moyen connu de trouver analytiquement le modèle qui maximise la probabilité de la séquence d'observations. En fait, étant donné que toute séquence d'observations finie est un exemple de la base d'apprentissage, il n'existe pas de moyen optimal d'estimer les paramètres du modèle. On peut, cependant, choisir $\lambda = (\Pi, A, B)$ de telle sorte que $P(O/\lambda)$ soit maximisé localement en utilisant des techniques de gradient ou une procédure itérative, telle que l'algorithme de Baum-Welch qui est une implémentation de la méthode Expectation-Maximisation (EM). Cette dernière sera présentée dans la section suivante mais, avant ça, nous allons présenter les formules de réestimation de l'algorithme Baum-Welch.

Soit $\xi_t(i,j)$, la probabilité d'être dans l'état S_i au temps t, et l'état S_j au temps $t+1$, compte tenu du modèle et de la séquence d'observations, c.à.d., :

$$\xi_t(i,j) = P\left[q_t = S_i, q_{t+1} = S_j / O, \lambda\right] \tag{4.32}$$

Il est clair, à partir des définitions des variables Forward et Backward, qu'il est possible de réécrire $\xi_t(i,j)$ sous la forme :

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1})\, \beta_{t+1}(j)}{P(O/\lambda)} \tag{4.33}$$

Nous avons précédemment défini $\gamma_t(i)$ comme étant la probabilité d'être dans l'état S_i au temps t, compte tenu de la séquence d'observation et du modèle. On peut donc relier $\gamma_t(i)$ à $\xi_t(i,j)$ en faisant la somme sur j, ce qui donne :

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{4.34}$$

Si on fait la somme des $\gamma_t(i)$ pour $t = 1$ à $t = T-1$, on obtient une grandeur qui peut être interprétée comme le nombre espéré (dans le temps) de visites de l'état S_i ou bien le nombre de transitions espérées à partir de l'état S_i. De la même façon, la somme de $\xi_t(i,j)$ sur t (de $t = 1$ à $t = T-1$) peut être interprétée comme étant le nombre espéré de transitions entre l'état S_i et l'état S_j.

$$\sum_{t=1}^{T-1} \gamma_t(i) \; = \text{ le nombre espéré de visites de l'état } S_i \qquad (4.35a)$$

$$\sum_{t=1}^{T-1} \xi_{\,t}(i,j) = \text{ le nombre espéré de transitions entre l'état } S_i \text{ et l'état } S_j \qquad (4.35b)$$

En utilisant les formules ci-dessus (et le concept de comptage des occurrences d'événements), on peut donner une méthode de réestimation des paramètres d'un HMM. Un ensemble de formules de réestimation raisonnables pour Π, A et B sont :

$$\bar{\pi} = le\ nombre\ espéré\ de\ visites\ de\ l'état\ S_i au\ temps\ 1$$

$$= \gamma_1(i) \qquad (4.36a)$$

$$\bar{a}_{ij} = \frac{\text{le nombre de transitions espérées entre l'état } S_i \text{ et l'état } S_j}{\text{le nombre de transitions espérées à partir de l'état } S_i}$$

$$= \frac{\sum_{t=1}^{T-1} \xi_{\,t}(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad (4.36b)$$

$$\bar{b_j}(k) = \frac{le\ nombre\ de\ fois\ espéré\ de\ visiter\ l'état\ S_j\ et\ en\ observant\ le\ symbole\ v_k}{le\ nombre\ de\ fois\ espéré\ \ de\ visiter\ l'état\ S_j}$$

$$= \frac{\sum_{t=1}^{T} \gamma_t(j),\ \ o_t = v_k}{\sum_{t=1}^{T} \gamma_t(j)} \qquad (4.36c)$$

En cas des HMM à densités continues, comme c'est montré plus haut, la matrice d'observation est remplacée par les paramètres de la loi multi-gaussienne utilisée. Dans ce cas, les paramètres à estimer sont les vecteurs moyens et les matrices de covariance. Les probabilités d'observation $\bar{b_j}(k)$ sont calculées suivant la formule suivante :

$$\bar{b_j}(k) = \sum_{m=1}^{M} \bar{c}_{jm}\, \mathcal{N}(\bar{\mu}_j, \bar{\Sigma}_j, O_t = v_k), \qquad (4.37)$$

Avec,

$$\bar{c}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)} \qquad (4.38)$$

$$\bar{\mu}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)\, O_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \qquad (4.39)$$

$$\bar{\Sigma}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \,.\, (O_t - \mu_{jk})(O_t - \mu_{jk})'}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{4.40}$$

où (.)' désigne une transposition vectorielle et $\gamma_t(j,k)$ est la probabilité d'être dans l'état j à l'instant t et la $k^{ième}$ composante du mélange représentant O_t, c'est-à-dire,

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{i=1}^{N} \alpha_t(j)\beta_t(j)}\right]\left[\frac{c_{jk}\mathcal{N}(\mu_{jk}, \bar{\Sigma}_{jk}, O_t)}{\sum_{m=1}^{M} c_{jm}\mathcal{N}(\mu_{jm}, \bar{\Sigma}_{jm}, O_t)}\right] \tag{4.41}$$

Ces formules de réestimation seront utilisées, par la suite, dans une procédure itérative appelée algorithme de Baum-Welch, qui n'est qu'une implémentation de la méthode EM pour les modèles de Markov cachés. Cette dernière, va être présentée dans la section suivante.

4.4. APPRENTISSAGE DES HMM PAR EM

Etant donnée une séquence d'observations O, l'apprentissage d'un HMM, comme nous l'avons vu précédemment, consiste à réestimer les paramètres du modèle λ en maximisant la vraisemblance $P(O/\lambda)$. L'estimation du maximum de vraisemblance, (ML : pour Maximum Likelihood en anglais), est la méthode standard pour estimer les paramètres d'un modèle probabiliste. Selon cette méthode, le modèle estimé à partir de O et λ est le modèle λ'_{ML} tel que :

$$\lambda'_{ML} = \underset{\lambda}{\text{argmax}}(\log \mathrm{P}(\mathrm{O}/\lambda)) \tag{4.42}$$

où $log\ P(O/\lambda)$ est le logarithme de vraisemblance de la séquence observée O sachant le modèle λ. Si on suppose que la séquence O est générée par la suite d'états cachés Q, la maximisation de $log\ P(O/\lambda)$ revient alors à maximiser la quantité suivante :

$$\log \frac{\mathrm{P}(\mathrm{O}, \mathrm{Q}/\lambda)}{\mathrm{P}(\mathrm{Q}/\mathrm{O}, \lambda)} = \log \mathrm{P}(\mathrm{O}, \mathrm{Q}/\lambda) - \log\ \mathrm{P}(\mathrm{O}, \mathrm{Q}/\lambda) \tag{4.43}$$

Donc, on peut écrire :

$$\lambda'_{ML} = \underset{\lambda}{\text{argmax}}\ (\log \mathrm{P}(\mathrm{O}, \mathrm{Q}/\lambda) - \log\ \mathrm{P}(\mathrm{O}, \mathrm{Q}/\lambda)) \tag{4.44}$$

Vu que la séquence d'états Q est cachée, il est impossible de résoudre directement le problème du maximum de vraisemblance. Plusieurs solutions approximatives ont été proposées dans la littérature, telles que, la méthode d'Expectation-Maximisation (Dempster et al., 1977), la méthode du gradient (Levinson, 1983), les méthodes variationnelles (Jordan et al., 1999) et autres. Nous nous intéressons ici, à la méthode d'EM qui est celle la plus communément utilisée. Celle-ci considère les variables non observées (les états cachés) comme données manquantes et les remplace par leurs espérances de vraisemblance.

L'algorithme de Baum-Welch (Baum & Petrie, 1966 ; Baum & Eagon, 1967 ; Baum, 1972) est l'implémentation de la méthode d'EM pour les modèles de Markov cachés. Cet algorithme permet de réestimer, petit à petit, les paramètres d'un modèle selon la procédure itérative suivante :

Initialisation :

- Choisir un modèle initial λ_0.

Itérations :

- Étape d'évaluation de l'espérance (E) : on calcule l'espérance de la vraisemblance des données manquantes en tenant compte des dernières variables observées
- Étape de maximisation (M) : on effectue une mise à jour des paramètres en maximisant la vraisemblance trouvée à l'étape E. A la fin de cette étape on obtient un nouveau modèle$\lambda_i$qui sera réutilisé comme point de départ d'une nouvelle phase d'évaluation de l'espérance. La mise à jour des paramètres se fait par les formules de réestimation de Baum-Welch présentées dans la section précédente.
- Répéter E et M jusqu'à convergence ou bien atteindre un nombre maximal d'itérations.

L'algorithme EM converge de manière sûre vers un optimum éventuellement local. Le résultat final dépend de l'initialisation (Christophe, 2001). Pour cette raison, dans l'étape d'initialisation, les paramètres du modèle initial (les probabilités de départ Π, les probabilités de transition A et les fonctions de densité d'observation B), ainsi que la structure du modèle (le nombre d'états et le nombre de gaussiens par états), doivent être soigneusement réglés.

4.5. PROBLEME DU REGLAGE INITIAL DES PARAMETRES DE L'APPRENTISSAGE

Dans cette section, nous allons tout d'abord mettre en évidence, la relation entre les paramètres de la configuration initiale de l'algorithme d'apprentissage et les performances du classifieur markovien, via quelques travaux de la littérature. Ensuite, nous présentons les méthodes d'initialisation les plus répandues.

4.5.1. Influence du réglage initial sur la stabilité du classifieur markovien

Le problème du réglage initial de l'apprentissage markovien a été l'objet d'étude de plusieurs publications dans divers domaines d'application, telles que dans (Yuan et al., 2019 ; Nathan et al., 1996 ; Liu et al., 2004, Moghaddam & Piccardi, 2013 ; Clemente et al., 2012 ; Ge et al., 2016 ; Rabiner et al., 1984 ; Belaïd & Anigbogu, 1994 ; Ferrer et al., 2000), etc. Dans les paragraphes qui suivent, nous présentons quelques-unes de ces publications que nous jugeons les plus intéressantes.

Rabiner et ses collègues dans (Rabiner et al., 1984), ont remarqué que l'évolution du taux de reconnaissance ne suit pas toujours l'augmentation du nombre d'états et qu'il n'y a pas de moyen théorique pour déterminer a priori le nombre d'états optimal. Dans (Belaïd & Anigbogu, 1994), les auteurs ont confirmé les constatations de Rabiner à propos de la relation entre le nombre d'états des HMM et le taux de reconnaissance. Toutefois, ils ont considéré quelques critères pouvant aider à trouver le nombre d'états dans un contexte de reconnaissance de fontes. Ils ont proposé de limiter le nombre d'états par le haut en le fixant inférieur au nombre de symboles ($N < M$). Ceci est logique car, autrement certains états ne seront pas utilisés et seront considérés comme redondants ou superflus. En reconnaissance de la parole, de tels états sont utilisés pour simuler le silence. Ces auteurs ont aussi limité le nombre d'états par le bas (4 dans leur cas), en évitant de le réduire à trop peu d'états, ce qui risquerait de concentrer les observations dans les mêmes états. En plus du nombre d'états, les auteurs ont étudié l'influence du modèle initial et le nombre d'itérations sur les performances dans un cadre de reconnaissance de textes multi-fontes. Les résultats obtenus ont montré que, le choix du modèle initial λ influe grandement sur la vraisemblance du modèle réestimé. En effet, différentes matrices de transition A et matrices d'observation B, peuvent conduire à différentes valeurs de vraisemblance $P(O/\lambda)$. Pour trouver le modèle optimal, on doit chercher les valeurs idéales de A et B conduisant à une valeur de vraisemblance globalement maximum. La figure 4.2 montre la variation de la vraisemblance en fonction du modèle, où λ, $\bar{\lambda}$ et λ^* désignent respectivement, le modèle initial, le modèle réestimé et le modèle optimal. Les pics dans la courbe représentent des points critiques correspondants aux maxima locaux.

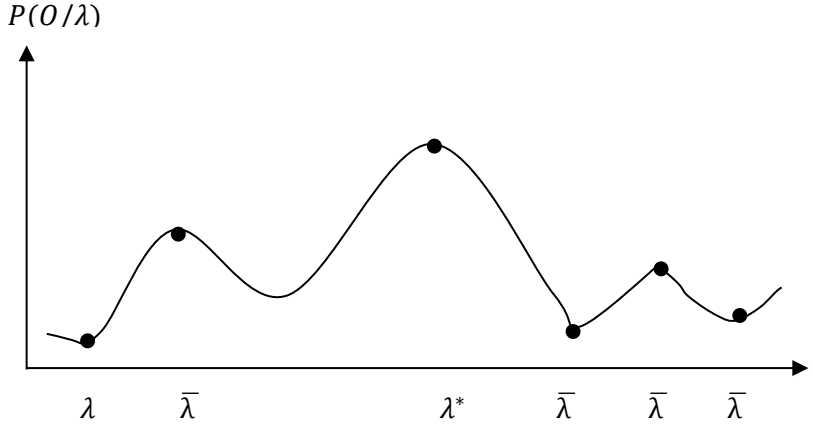


Fig.4.2. Points critiques

Comme l'algorithme d'apprentissage ne garantit pas un maximum global, et comme il n'y a aucun moyen théorique de trouver le pic de $P(O/\lambda)$ globalement maximum, les auteurs dans (Belaïd & Anigbogu, 1994) ont multiplié l'apprentissage des échantillons en l'itérant plusieurs fois. Le résultat obtenu a montré que l'influence de ce nombre d'itérations dépond du nombre d'états choisi. Pour des modèles de 7 à 8 états, la variation est très minime. Par contre, pour des modèles de 4 à 6 états, de fortes variations ont été marquées jusqu'à la quinzième itération, après laquelle le processus semble se stabiliser. Pour stabiliser les modèles, ils ont proposé d'utiliser une fonction de lissage.

4.5.2. Méthodes d'initialisation

Cette sous-section est consacrée à la présentation des méthodes couramment utilisées pour régler les différents paramètres de la configuration initiale de l'algorithme d'apprentissage, à savoir, le nombre d'états, le nombre de densités gaussiennes et le modèle initial de l'algorithme d'EM.

4.5.2.1. Choix du nombre d'états et du nombre de densités gaussiennes

Le réglage initial des paramètres de structure d'un HMM tels que, le nombre d'états et le nombre de gaussiens associées aux états est un problème ouvert, car la théorie (Rabiner, 1988) ne permet pas de guider l'utilisateur dans ses choix, qui doivent être faits en fonction des données modélisées.

Plusieurs idées ont été proposées pour fixer le nombre d'états d'un HMM de chaque mot en RAP. Levinson (Levinson et al., 1983) a suggéré que le nombre d'états devrait correspondre approximativement au nombre d'unités sonores du mot (phonèmes). De son côté, Bakis (Bakis, 1976) a suggéré de choisir le nombre d'états comme étant la moyenne des nombres d'observations des occurrences prononcées du mot à modéliser (bien entendu, ça concerne le cas d'une approche globale). Cependant, comme les durés des élocutions du même mot varient d'une occurrence à une autre, le nombre optimal d'états (Kwong, 2001) ne peut être trouvé qu'en reconfigurant le modèle après chaque entraînement de HMM.

Pour déterminer le nombre de gaussiennes associées aux états d'un HMM, l'auteur dans (Sankar, 1998), a proposé un algorithme appelé GMS (Gaussian Merging-Splitting) ou fractionnement-fusion gaussien. Le fractionnement gaussien itératif et l'algorithme EM sont utilisés pour initialiser le nombre de densités gaussiennes dans chaque état. Partant d'une seule gaussienne, le fractionnement gaussien permet d'augmenter le nombre de gaussiennes à chaque étape de l'apprentissage jusqu'à atteindre le nombre nécessaire de gaussiennes. La fusion gaussienne est effectuée avant chaque opération de fractionnement. Cela garantit, selon l'auteur, qu'à toutes les étapes de l'algorithme, les gaussiennes sont estimées de manière robuste pour tous les états. Dans l'algorithme GMS, le nombre d'états et le nombre maximum de gaussiennes par état doivent être spécifiés par l'utilisateur. Le nombre de gaussiennes est augmenté de manière itérative en utilisant la fusion et le fractionnement jusqu'à ce que le nombre maximal de gaussiennes soit

atteint. Étant donné que la quantité de données d'apprentissage par état varie et que les gaussiennes sont fusionnées jusqu'à ce que toutes les gaussiennes aient un seuil de données, le nombre de gaussiennes par état varie généralement avec les états après un certain nombre d'opérations de fusion et de fractionnement. Les états avec moins de données d'apprentissage ont moins de gaussiennes et vice versa. Le nombre de gaussiennes associées aux états varie donc au fur et à mesure que l'algorithme avance.

La démarche la plus utilisée pour l'initialisation des paramètres, consiste à fixer des valeurs empiriques en exécutant l'algorithme d'apprentissage pour toutes les valeurs raisonnables, puis évaluer le système final et choisir les valeurs donnant le meilleur score. D'autres solutions basées sur les techniques d'optimisation ont été proposées, telles que celle présentée dans (Kwong et al., 2001), où un algorithme génétique est utilisé pour trouver le nombre optimal d'états. Dans (Bhuriyakorn et al., 2008), les auteurs ont proposé une approche d'estimation de la topologie HMM (nombre d'états et transitions entre états) pour une tâche de reconnaissance de phonèmes, dont le processus se déroule en deux étapes : Tout d'abord, un ensemble de topologies appropriées est construit en combinant différentes fonctions objectives et méthodes de génération de topologies. Deuxièmement, un algorithme génétique est utilisé, comme méthode de sélection de topologies. Cet algorithme considère une fonction objective globale et sélectionne, pour chaque phonème, la topologie la mieux adaptée parmi les candidats proposés dans l'étape précédente.

Une autre solution consiste à appliquer les critères de sélection de modèles pour choisir le meilleur modèle final parmi un large ensemble de modèles candidats différents dans leurs structures (nombre d'états ou nombre de densités gaussiennes). Dans (Biem, 2003), un critère de sélection de modèles appelé DIC (Discriminative Information Creterion) a été proposé pour optimiser la topologie des HMM (nombre d'états). Ce critère est basé sur la sélection de modèles les plus discriminants au lieu de modèles basés sur le principe de « rasoir d'Occam », comme pour les critères AIC (Akaike information criterion) et BIC (Bayesian Information Criterion). Nous rappelons que celui-ci (« rasoir d'Occam ») est un principe de parcimonie stipulant dans notre cas, que le modèle doit être suffisamment simple pour un calcul efficace et suffisamment complexe pour que les données soient bien spécifiées.

4.5.2.2. Choix du modèle initial

Les valeurs initiales de la matrice de transition A et le vecteur des probabilités de départ Π, n'ont généralement pas un impact important sur la qualité du modèle réestimé. Il a été montré par (Rabiner, 1988) que, l'initialisation aléatoire, sous réserve des contraintes stochastiques et non nulles (équations 4.2 & 4.3), de ces paramètres, est suffisante pour donner des réestimations utiles dans la quasi-totalité des cas. Il est également courant d'utiliser une initialisation uniforme de ces paramètres. En revanche, pour les paramètres de B, l'expérience a montré que de bonnes estimations initiales sont utiles dans le cas des symboles discrets (HMM à densité discrète), et sont obligatoires dans le cas des HMM à

densités continues. Plusieurs méthodes d'initialisation ont été proposées dans la littérature. Les plus répandues sont les suivantes :

- **La méthode du *k*-means segmentale (segmental *k*-means) :** C'est la méthode d'initialisation standard proposée par (Juang & Rabiner, 1990 ; Rabiner et al., 1986 ; Rabiner, 1988 ; Rabiner, 1989). Elle permet de distribuer les trames de données d'apprentissage sur les états en utilisant l'algorithme de clustering *k*-means et l'algorithme de décodage de Viterbi. Cette technique d'initialisation est exécutée de manière supervisée, nécessitant l'utilisation de nombreux échantillons de données d'apprentissage étiquetées (Clemente et al., 2012). L'inconvénient de cette méthode est que la valeur de *k* doit être fixée à priori.

- **La méthode de démarrage à plat (Flat Start) :** L'idée de cette méthode est de rendre tous les états égaux (Itaya et al., 2005), tous les modèles des classes sont alors initialisés avec des paramètres identiques égaux à la moyenne globale et à la variance des données d'apprentissage (Clemente et al., 2012).

Il est aussi possible, comme il a été montré dans (Rabiner, 1988), d'utiliser comme modèle initial, un modèle aléatoire vérifiant les contraintes stochastiques (equations 4.2, 4.3 et 4.4), ou n'importe quel modèle déjà disponible appris à partir de données appropriées. D'autres méthodes d'initialisation moins populaires ont été proposées dont, nous citons, à titre d'exemples, la méthode DAEM (Deterministic Anealing EM) (Itaya et al., 2005 ; Kurata et al., 2006), la méthode SMEM (Split and Merge EM) (Han & Boves 2006), et la méthode des séquences multiples (Liu et al., 2004). Cette dernière consiste, étant données *K* séquences d'apprentissage, à proposer pour chacune d'elles un modèles initial différent, ce modèle est ensuite utilisé pour apprendre un modèle représentant la séquence, les *K* modèles entrainés sont combinés en calculant la moyenne des valeurs de leurs paramètres, ce qui donne un modèle unique représentant toutes les séquences d'apprentissage.

À notre connaissance, et à l'heure actuelle, il n'existe pas de critères pour choisir entre les méthodes d'initialisation. Toutefois, quelques études comparatives ont été faites, où les résultats reportés ne permettent pas de mettre en évidence la supériorité incontestable d'une méthode sur une autre, et montrent qu'il n'existe pas de méthode universelle qui peut être appliquée avec succès pour tous les domaines d'application ou sur n'importe quelle base de données. Ferrer et ses collègues dans (Ferrer et al., 2000), ont présenté une étude comparative afin de mettre en évidence l'influence de la méthode d'initialisation et du critère d'arrêt de l'algorithme d'apprentissage sur les performances du système. Trois méthodes d'initialisation : *Aléatoire*, *équiprobable* et *occupations égales* (proposée par les hauteurs) et quatre critères d'arrêt : Nombre d'itérations, seuil fixe, validation croisée et moyenne-variance, ont été étudiés. Les auteurs ont montré que les meilleurs résultats ont été trouvés avec la méthode d'initialisation utilisant la méthode d'*occupations égales* et le critère d'arrêt à seuil fixe.

Malgré la littérature abondante sur les méthodes d'initialisation de l'algorithme EM, le problème de réestimation d'un modèle globalement optimal reste toujours posé. Pour traiter de ce problème, deux catégories de solutions sont proposées dans la littérature. La première catégorie regroupe les solutions qui cherchent le réglage initial optimal, en appliquant des algorithmes d'optimisation tel que les algorithmes génétiques (Kwong et al., 2001). La deuxième catégorie, quant à elle, regroupe les solutions qui s'intéressent à la sélection d'un modèle final parmi un ensemble de modèles candidats venant de différents points de départ. Dans ce cas, les critères de sélection de modèles sont souvent utilisés, nous citons à titre d'exemples le travail (Biem, 2003). Malheureusement, comme les fonctions objectives utilisées sont optimisées sur des bases de validation dont les conditions d'enregistrement sont souvent différentes de celles de l'environnement de l'utilisateur final (notamment, en cas des systèmes grand public indépendants du locuteur), les modèles entrainés, dans les deux catégories, sont trop sensibles à la variabilité des données, aux perturbations de l'environnement, et à l'incohérence entre les données d'apprentissage et les données de test. A travers les approches proposées dans le cadre de cette thèse, nous essayons d'alléger ce problème en se basant sur une modélisation markovienne multiple.

4.6. UTILISATION DES HMM EN RAP

Dans ce qui suit, et dans un premier lieu, nous allons expliquer comment les HMM sont appliqués dans les systèmes de RAP, puis présenter une revue bibliographique des travaux utilisant ces modèles.

4.6.1. Principe général

Les HMM sont utilisés en RAP pour modéliser des unités élémentaires de la parole, telles que les mots, les syllabes et les phonèmes. Indépendamment de l'unité de modélisation choisie, la topologie la plus communément utilisée dans la littérature est la topologie gauche-droite, appelée aussi modèle de Bakis, où les transitions de droite à gauches sont interdites. Ce choix est principalement justifié par le fait que la parole est un phénomène évoluant dans le temps. Cette topologie permet de modéliser les variations temporelles (rythme) ; ainsi, la parole lente est réalisée en bouclant plusieurs fois sur le même état. Alors que pour la parole rapide, il y a saut d'états vers la droite. Il y a aussi la variabilité intra et interlocuteur qui peut être prise en compte, partiellement, en utilisant les HMM continus, où les probabilités d'observations sont réalisées à l'aide d'un mélange de lois gaussiennes, associé à chaque état du modèle.

La manière d'utilisation des HMM en RAP varie selon le type d'application visée et selon l'unité de modélisation utilisée. Nous allons montrer, juste après, le principe d'application des HMM en reconnaissance de mots isolés à vocabulaire limité et en reconnaissance de la parole continue à vocabulaire ouvert.

4.6.1.1. La reconnaissance de mots isolés à vocabulaire limité

L'approche de modélisation utilisée dans ce cas est une approche globale (cf. section 2.3.2.2). Chaque mot est donc modélisé par un HMM différent entrainés séparément. Etant donnés un vocabulaire de N mots V= $\{m_1, m_2, \dots, m_N\}$ et une base d'apprentissage contenant plusieurs exemples de chaque mot du vocabulaire (prononcés plusieurs fois par des locuteurs différents pour avoir le maximum de variabilité de prononciation). Après avoir fait l'analyse acoustique des exemples de la base d'apprentissage pour en extraire les vecteurs des caractéristiques, appelés séquences d'observations, ces derniers sont fournis au module d'apprentissage qui consiste à appliquer l'algorithme de Baum-Welch (implémentation de l'algorithme EM pour les HMM) afin de construire un HMM λ_i pour chaque mot du vocabulaire $m_i, 1 \leq i \leq N$. Lorsque on doit reconnaitre un nouvel exemple, le signal acoustique correspondant est d'abord paramétrisé en utilisant la même méthode d'analyse acoustique utilisée lors de l'apprentissage, le résultat est une séquence d'observations O, qui sera ensuite fournie au module de reconnaissance pour calculer la vraisemblance $P(O/\lambda_i)$ par rapport à tous les modèles entrainés en appliquant l'algorithme Forward-Backward, ou la probabilité du meilleur chemin dans chacun des modèles en appliquant l'algorithme de Viterbi. La classe d'appartenance est celle dont le modèle est le plus vraisemblable.

La figure 4.3 illustre l'architecture générale d'un système de reconnaissance de mots isolés à vocabulaire limité, basé sur une modélisation globale par les HMM. Cette approche peut être considérée comme une version améliorée de la programmation dynamique.

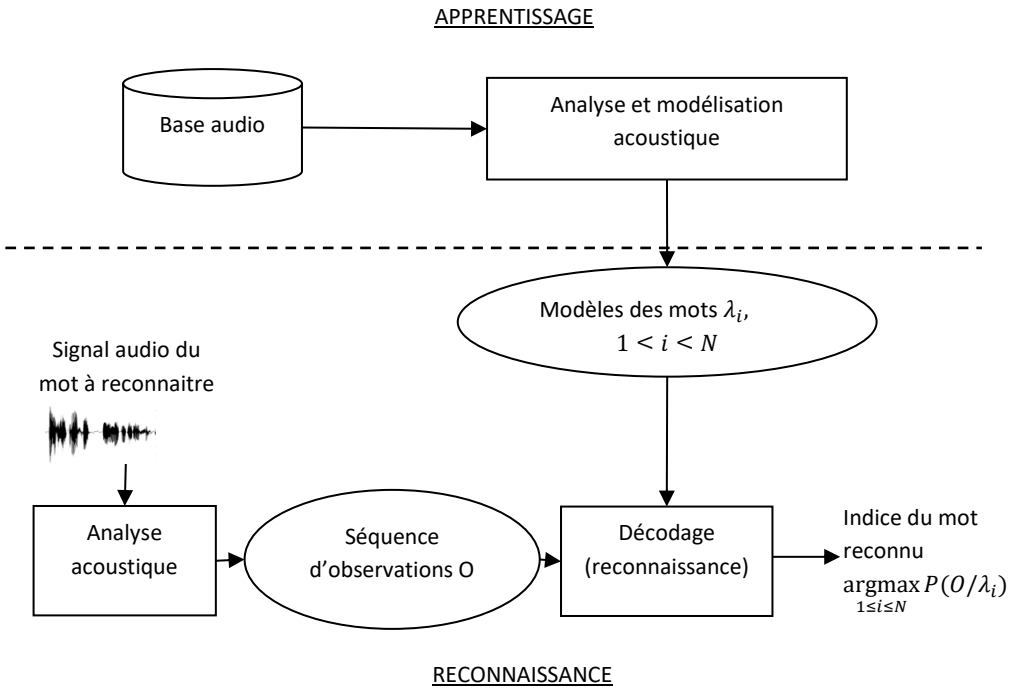


Fig.4.3. Architecture basée HMM d'un système de reconnaissance de mots isolés

4.6.1.2. La reconnaissance de la parole continue à grand vocabulaire

Lorsque la taille du vocabulaire augmente, l'approche globale ne convient plus, car la modélisation de chaque mot par un HMM différent entraine un coût de calcul élevé et un espace de stockage important. Dans ce cas, l'approche analytique est utilisée (cf. section 2.3.2.2), l'unité de modélisation typique est le phonème et les modèles acoustiques appris sont alors des modèles de phonèmes. La construction de ces modèles suit le même principe que celui de l'approche globale. Outre les modèles acoustiques, l'apprentissage consiste à établir, statistiquement, les modèles de langage à partir d'un corpus de textes annotés. Le modèle de langage a pour objectif d'aider le système à déterminer si une suite de mots est plus probable qu'une autre dans la langue modélisée. La probabilité d'apparition d'un mot dans une phrase est calculée en fonction des $n-1$ mots qui le précèdent, et le modèle est appelé modèle *n-gram*.

Dans la phase de reconnaissance, les modèles acoustiques, les modèles de langage et le vocabulaire sont coordonnés dans le module de reconnaissance (décodage) pour transformer les paramètres acoustiques du signal à reconnaitre en une suite de mots qui est la plus vraisemblable. Il s'agit donc, étant donnée une séquence d'observations $O(o_1, o_2, \dots, o_T)$ représentant le signal à reconnaitre, de trouver la suite de mots $W^* = (w_1, w_2, \dots, w_m)$ maximisant la probabilité à postériori $P(W/O)$. Selon la règle de Bayes, on peut écrire :

$$P(W/O) = \frac{P(W) * P(O/W)}{P(O)} \tag{4.45}$$

$$W^* = \underset{W}{\operatorname{argmax}}\, P(W/O) \tag{4.46}$$

$$= \underset{W}{\operatorname{argmax}}\, \frac{P(W) * P(O/W)}{P(O)} \tag{4.47}$$

Et comme $P(O)$ ne dépend pas de la séquence de mots étudiée, la maximisation de $P(W/O)$ revient à maximiser $P(W) * P(O/W)$. L'équation (4.47) peut donc être reformulée comme suit :

$$W^* = \underset{W}{\operatorname{argmax}}\, P(W) * P(O/W) \tag{4.48}$$

Où, $P(W)$ est la probabilité linguistique (probabilité à priori du modèle de langage W) et $P(O/W)$ est la probabilité du modèle acoustique.

Pour un modèle de langage n-gram, la probabilité d'apparition d'un mot w_i dépond de ses $n-1$ prédécesseurs, la probabilité $P(W)$ est décomposée suivant la formule suivante :

$$P(W) = P(w_1, w_2, \dots, w_m) \tag{4.49a}$$

$$= P(w_1) * P(w_2/w_1) * P(w_3/w_1, w_2) * \dots * P(w_m/w_1, w_2, \dots, w_{m-1}) \tag{4.49b}$$

En se limitant seulement à l'historique aux $n-1$ derniers mots, on peut écrire :

$$P(w_m/w_1, w_2, \dots, w_{m-1}) = P(w_m/w_{m-(n-1)}, \dots, w_{m-1}) \quad (4.50)$$

Si $n = 2$ par exemple (modèle bi-gram), $P(W)$ est calculée en appliquant la formule :

$$P(W) = P(w_1) * P(w_2/w_1) * P(w_3/w_2) * \dots * P(w_m/w_{m-1}) \quad (4.51)$$

L'architecture standard d'un système de reconnaissance de la parole continue à grand vocabulaire est schématisée dans la figure 4.4.

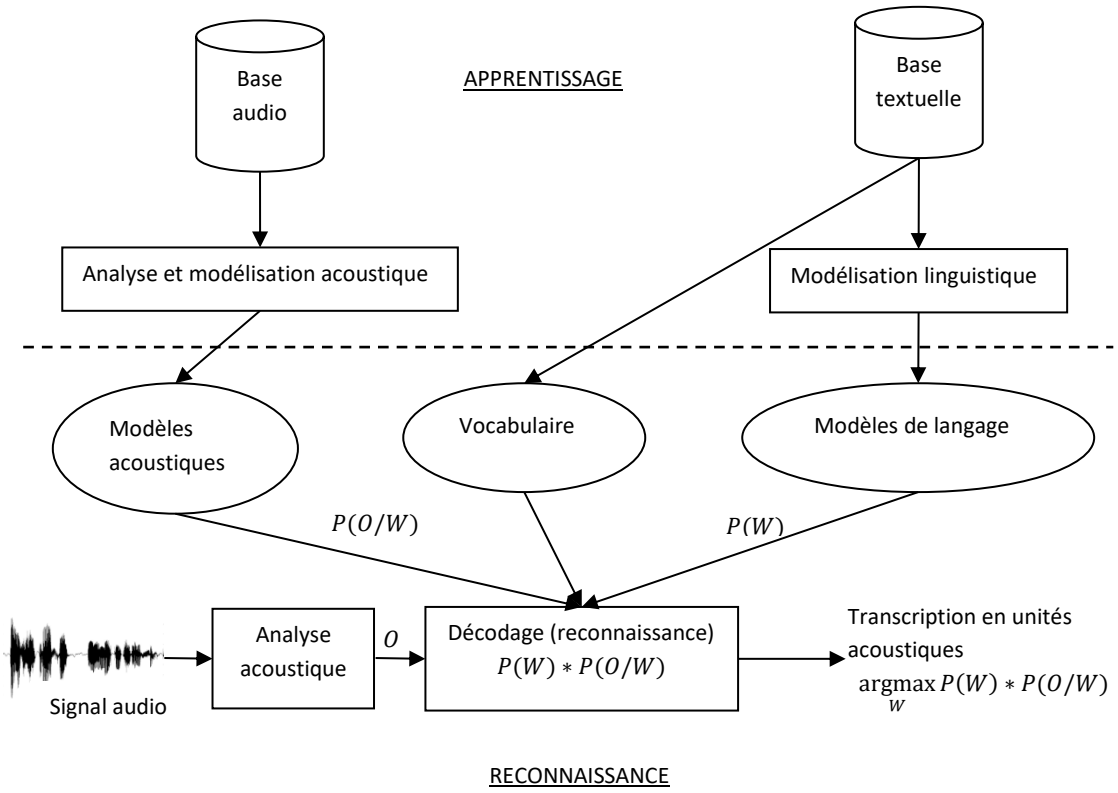


Fig.4.4. Architecture basée HMM d'un système de reconnaissance de la parole continue

4.6.2. Quelques travaux à base des HMM

Bien que leur utilisation ne soit pas nouvelle en RAP, les HMM restent toujours une des méthodes de classification les plus utilisées et continuent à attirer l'attention des chercheurs. En effet, de nombreux travaux récents sont basés sur les HMM. Le tableau 4.1 présente une synthèse non exhaustive de quelques travaux basés HMM pour des langues différentes. Ces travaux s'insèrent dans le cadre des classifieurs individuels vue le fait que

les HMM sont utilisés seuls, comme méthode de classification, dans le système de reconnaissance.

Tableau 4.1. Quelques travaux récents utilisant les HMM pour la RAP

Référence	Ashraf et al., 2010	Ca & Radhab, 2012	Kumar & Aggarw, 2011	Paramonov & Sutula, 2016	Khelifa et al., 2017
Langue	Ourdou	Tamil	Hindi	Russe	Arabe
Type de discours	Mots isolés	Mots isolés	Mots isolés	Mots isolés	/
Vocabulaire	Petit (52 mots)	Petit le (50 mots)	Petit (30 mots)	Petit (100 mots)	59 unités (phonèmes et leurs parties géminées) & 110 allophones
Mode	Multi-locuteur & indépendant du locuteur	Indépendant du locuteur	/	Mono-locuteur	Multi-locuteur
Paramètres des HMM	Topologie : / Nombre d' états : / Nombre de gauss : 8	/	5 à 11 états	/	10 à 16 gaussiennes
Approche	/	/	/	Globale	/
Base de données	Personnelle	Personnelle	Personnelle	Personnelle	Personnelle
Taux de reconnaissance	94.77 % (Multi-loc) & 89.44% (indépendant du loc)	88%	94.63%	97.3%	97% : phonèmes 96% : allophones

Bharali & KalitaK. 2015	Assamais	Mots isolés	Petit (10 chiffres)	Indépendant du locuteur	Nombre d' états : 5	Globale	Personnelle	95%
Al-Maadeed, & Al-Maadeed, 2018	Arabe	Mots isolés	Petit (10 chiffres)	Multi-locuteur	Etats : 3 ou 7 Gauss : 2 ou 4	Globale	Personnelle	80%
Hemakumar et al., 2016	Anglais et langues indiennes	Parole continue	Grand	Indépendant du locuteur	/	Analytique	AN4 de CMU (anglais) & personnelle	85.16%
Kumar et al., 2014	Hindi	Parole continue	Petit	/	/	/	Personnelle	95.08%

Notons que la quasi-totalité des travaux sont évalués en termes de performance (précision) en ignorant, totalement, l'aspect robustesse. Par exemple, dans (Kumar & Aggarw, 2011), les auteurs ont obtenu un taux de reconnaissance moyen égale à 94.63%, où ils ont testé 5 locuteurs différents. L'écart entre le meilleur et le pire taux de reconnaissance est de l'ordre de 10%. Ashraf et ses collaborateurs dans (Ashraf et al., 2010), ont donné un taux de reconnaissance moyen, pour 5 locuteurs, égale à 89.44%, avec un écart allant jusqu'à 6.67%. Ceci montre que les systèmes développés sont peu robustes face à la variabilité interlocuteur.

Malgré qu'ils puissent individuellement donner des résultats satisfaisants, les HMM sont souvent combinés à d'autres méthodes de classification pour améliorer encore les performances des systèmes de RAP. Nous citons à titre d'exemple, HMM/NN (Abdel-Hamid et al., 2012 ; Dubagunta & Doss 2019) et HMM/SVM (Ganapathiraju & Picone, 2000 ; Zarrouk et al., 2018).

4.7. CONCLUSION

Au cours de ce chapitre, nous avons présenté les bases théoriques des HMM, et leur utilisation en RAP avec quelques travaux de la littérature. Nous avons également étudié le problème du réglage initial de l'algorithme EM utilisé souvent pour l'apprentissage

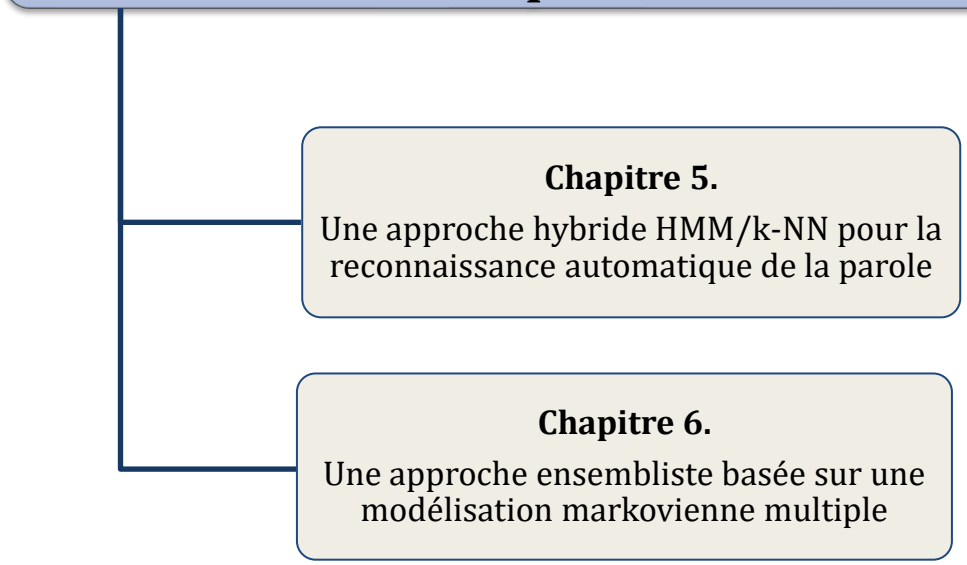
markovien. Nous avons montré que le choix de la configuration initiale joue un rôle important dans la stabilité des performances du système final qui dépendent fortement de la base de validation utilisée. Ainsi, un seul modèle par classe ne peut pas bien modéliser tous les exemples de la classe à cause de la grande variabilité intra-classe, notamment dans les systèmes indépendants du locuteur. Cela affecte négativement la robustesse du système vis-à-vis de la variabilité de données. Pour faire face à ce problème, nous proposons deux différentes approches qui seront présentées en détail dans les deux prochains chapitres.

"New opinions are always suspected, and usually opposed without any other reason, but because they are not already common."

John Locke

Partie II.

Contributions à la Reconnaissance Automatique de la Parle



CHAPITRE 5

UNE APPROCHE HYBRIDE HMM/K-NN POUR LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

SOMMAIRE

5.1. INTRODUCTION

Les résultats publiés récemment montrent une évolution impressionnante dans le domaine de la RAP en général, et de la reconnaissance de mots isolés à vocabulaire limité en particulier. Cependant, d'après notre étude bibliographique, nous avons constaté que la quasi-totalité des travaux sont évalués globalement en termes de performance sans tenir compte de la sensibilité du système à la configuration initiale des paramètres de l'apprentissage. Cette configuration est généralement optimisée expérimentalement sur la base d'apprentissage, par conséquent, les modèles établis sont trop sensibles à la variabilité des données et à l'inadéquation des données d'apprentissage et des données de test. Par ailleurs, tous les travaux publiés traitant de ce problème, notamment ceux basés HMM, utilisent une configuration unique, ce qui donne à la fin un modèle unique par classe. Ce modèle peut ne pas être bien adapté à toutes les occurences de la classe, vu la grande variabilité des données intra-classe, surtout dans les systèmes indépendants du locuteur. Dans ce chapitre, nous proposons d'utiliser une modélisation multiple plutôt qu'un modèle unique, ce qui permet, d'une part, d'éviter le problème de réglage initial des paramètres d'apprentissage des HMM qui n'est pas une tâche triviale, et d'autre part, d'augmenter les performances et la robustesse du système. La modélisation multiple est réalisée à travers une approche hybride intégrant les modèles HMM dans une architecture k-NN.

Avant de décrire et évaluer notre approche, et pour bien comprendre la manière dont nous avons intégré la modélisation markovienne dans des niveaux différents au sein d'une architecture k-NN, nous allons commencer par rappeler le principe général des deux classifieurs de base, HMM et k-NN, dans un cadre d'une approche de modélisation acoustique globale.

5.2. LE CLASSIFIEUR HMM

Grâce à leurs bases mathématiques solides, leur capacité à modéliser des séquences de longueurs variables et la possibilité d'une segmentation implicite de la parole, les HMM restent l'une des méthodes les plus performantes dans le domaine de la reconnaissance vocale. En considérant l'ensemble des séquences d'observations représentant les vecteurs de caractéristiques des exemples de la base d'apprentissage, l'utilisation des HMM pour la RAP consiste à apprendre un modèle acoustique différent pour chaque classe de données. L'apprentissage des modèles est communément réalisé à l'aide de l'algorithme Baum-Welch qui est une implémentation de la méthode itérative EM. L'application de l'algorithme EM pour l'apprentissage des HMM nécessite un réglage initial de certain nombre de paramètres, tels que le nombre d'états, le modèle initial, le nombre d'itérations, et le nombre de densités gaussiennes associées à chaque état dans le cas des modèles HMM/GMM. Durant la phase de reconnaissance, le signal de la parole de l'exemple à reconnaitre est d'abord analysé pour en extraire les vecteurs de caractéristiques, puis l'algorithme de Viterbi est appliqué afin de calculer la vraisemblance que donne chaque modèle à l'exemple à reconnaitre. La classe représentée

par le modèle le plus vraisemblable est choisie comme classe d'appartenance. La figure 5.1 illustre le principe général du classifieur HMM selon une approche globale.

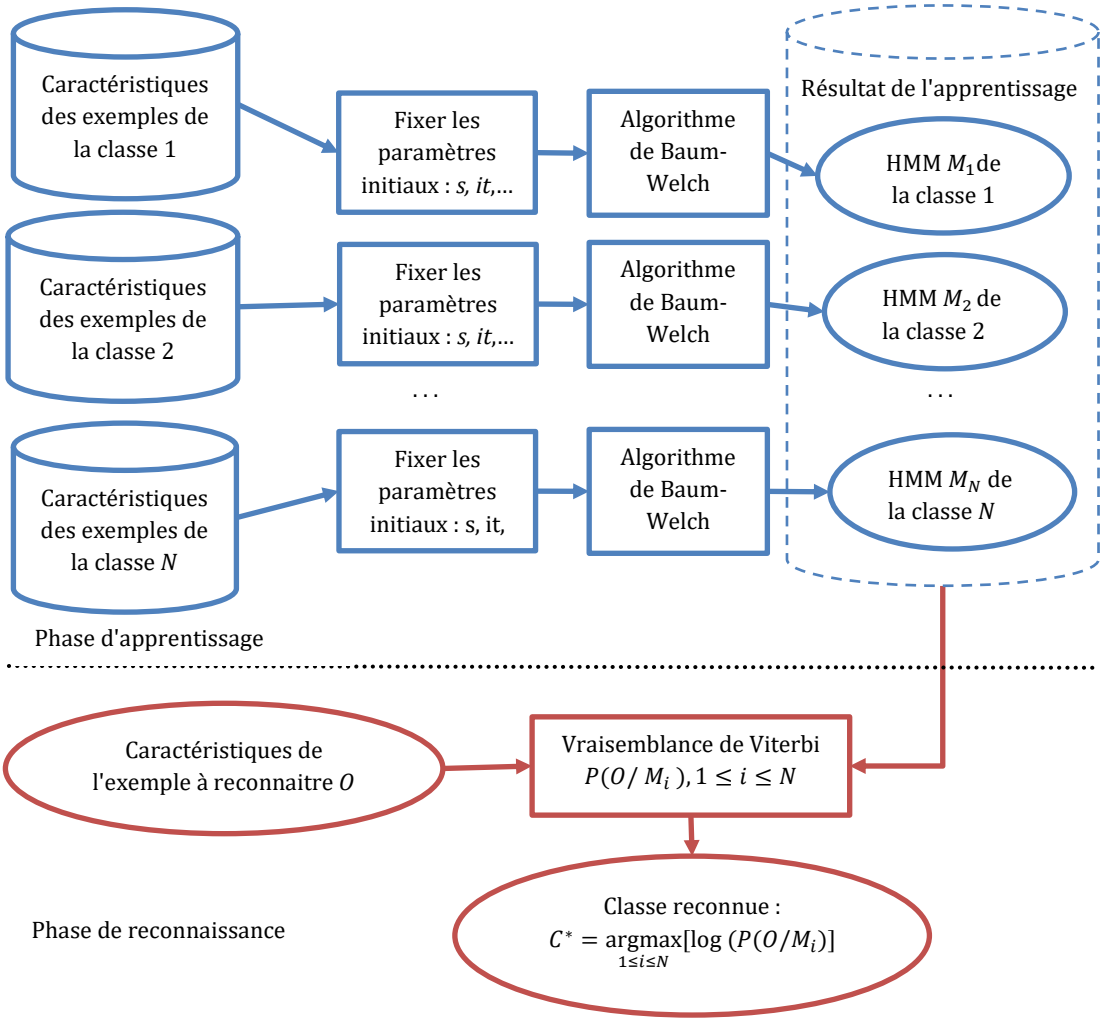


Fig.5.1. Un schéma bloc illustrant le principe du classifieur HMM (approche globale)

Malgré leur grand succès, il est bien connu que l'une des faiblesses du classifieur HMM est le nombre important de paramètres initiaux qui doivent être choisis avec prudence. Ces paramètres sont généralement fixés expérimentalement et ils dépendent fortement des données d'apprentissage et de test utilisées, ce qui affecte considérablement la robustesse et la stabilité du classifieur.

5.3. LE CLASSIFIEUR k-NN

Grâce à sa simplicité d'implémentation et sa robustesse, la méthode k-NN est largement utilisée dans plusieurs domaines de reconnaissance de formes, individuellement (AlKhateeb et al. 2009) ou combinée à d'autres méthodes (Wang et Ju 2008 ; Wang et al. 2015). Cependant, le temps de calcul important et la grande quantité de stockage requise

par la classification k-NN sont les raisons principales qui contraignent son utilisation dans la reconnaissance de la parole.

La figure 5.2 résume le principe de l'algorithme k-NN. Où, x_i et C_i représentent respectivement un vecteur de caractéristiques de référence et son étiquette de classe, NV est le nombre d'exemples de la base d'apprentissage et X est le vecteur de caractéristiques de l'exemple à reconnaitre, k est un entier positif qui doit être choisi soigneusement dans l'intervalle $[1, NV]$. Généralement, il est optimisé empiriquement sur une base de validation.

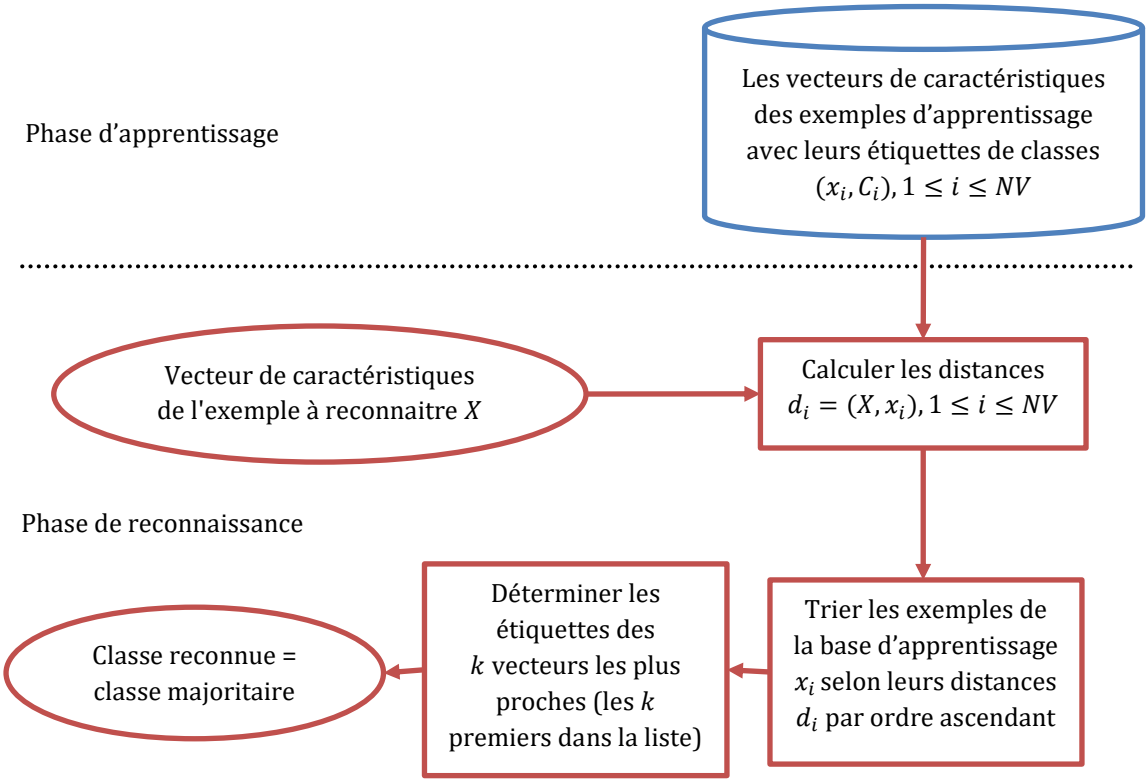


Fig.5.2. Un schéma bloc illustrant le principe de la méthode k-NN

5.4. L'APPROCHE HYBRIDE HMM/k-NN

Les approches hybrides alliant plus d'une méthode de classification, sont de plus en plus utilisées en reconnaissance de formes. Dans le domaine de RAP, l'hybridation la plus communément adoptée est celle utilisant les réseaux de neurones (NN) ou les systèmes à vaste marge (SVM) pour estimer les probabilités d'émission des observations générées par les états des HMM. L'objectif principal de ce type d'hybridation est l'amélioration de la capacité de discrimination et le pouvoir de généralisation des HMM à travers l'intégration des classifieurs plus discriminatifs.

L'objectif de notre approche hybride qui est basée, comme indiqué plus haut, sur une modélisation markovienne multiple et une décision de type k- plus proches voisins, est de concevoir un classifieur héritant à la fois de la robustesse du k-NN et des bonnes

capacités de modélisation acoustique des HMM, tout en évitant le problème de sensibilité des HMM au réglage initial des paramètres de l'apprentissage. Ceci permet d'améliorer les performances du système et sa robustesse à la variabilité de données.

Tout d'abord, nous créons pour chaque classe de données un large ensemble de modèles HMM qui se diffèrent par l'un des paramètres de la configuration initiale de l'algorithme d'apprentissage (le nombre d'états, le modèle initial, le nombre de densités gaussiennes ou le nombre d'itérations). Les meilleurs de ces modèles sont ensuite sélectionnés et stockés avec leurs étiquettes de classe. Dans la phase de reconnaissance, nous proposons d'utiliser la règle de décision k-NN avec, comme mesure de similarité, la vraisemblance de Viterbi. En cas d'ambiguïté, un post-traitement est effectué avant de prendre la décision finale. L'idée d'utiliser plusieurs HMM différents, plutôt qu'un modèle unique, est inspirée du principe de la méthode k-NN pour lequel plusieurs exemples d'apprentissage, décrits par leurs vecteurs de caractéristiques, sont utilisés pour représenter chaque classe. L'utilisation de plusieurs représentants nous permet d'améliorer la robustesse à la variabilité de données, notamment dans le cas des systèmes indépendants du locuteur.

La figure 5.3 représente un schéma général de l'approche proposée. Où N, m et s correspondent respectivement aux : nombre de classes, nombre de modèles générés par classe, et nombre de modèles sélectionnés par classe.

Dans ce qui suit, nous expliquerons en détail chaque étape de l'approche proposée. Pour des fins de simplification, nous supposons que le paramètre modifié de la configuration initiale est le nombre d'états des HMM. Le même principe s'applique aux autres cas, à savoir le modèle initial, le nombre de densités gaussiennes par état, et le nombre d'itérations de l'algorithme EM.

5.4.1. L'analyse acoustique (extraction des caractéristiques)

L'analyse acoustique est la première phase de tout système de reconnaissance vocale. Elle vise à réduire la quantité de données que nous devons traiter et à extraire les caractéristiques pertinentes et discriminantes du signal de parole. Selon notre approche, nous proposons d'utiliser la méthode MFCC (cf. section 2.3.2.1). Ce choix se justifie par deux raisons principales. D'une part, elle est l'une des méthodes les plus répandues dans le domaine de RAP et de nombreuses publications prouvent sa supériorité sur d'autres méthodes. D'autre part, la base de données utilisée ne permet pas d'appliquer d'autres méthodes d'analyse, car elle ne comprend que des coefficients MFCC extraits des signaux vocaux, et les fichiers audio ne sont pas disponibles. Cela ne posera pas de problème, car notre contribution ne concerne pas la phase d'analyse acoustique, mais plutôt celle de classification, et tous les systèmes étudiés dans cette thèse sont basés sur les mêmes caractéristiques MFCC.

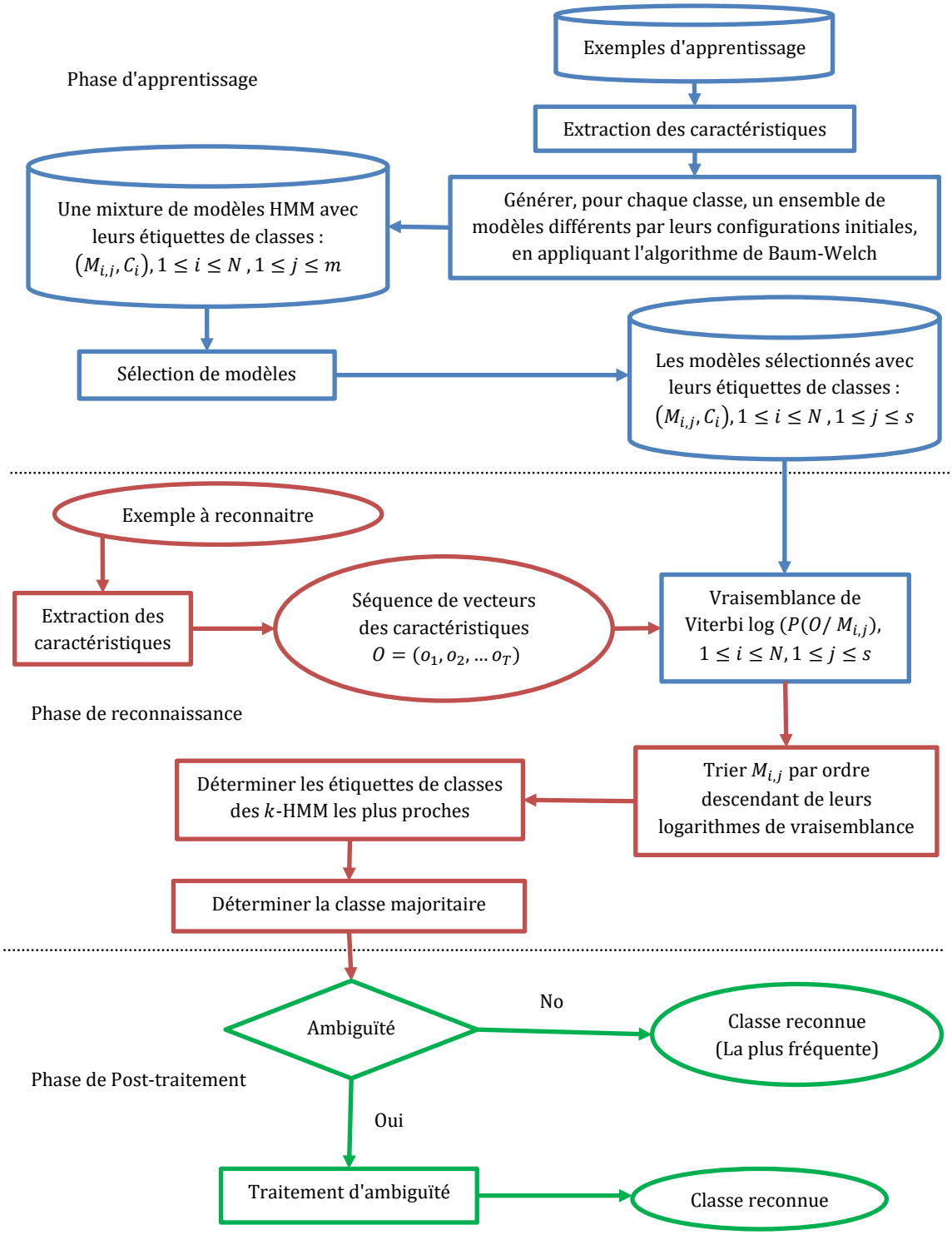


Fig.5.3. Un schéma bloc de l'architecture hybride HMM/k-NN

5.4.2. L'apprentissage (modélisation multiple)

Comme nous l'avons déjà évoqué précédemment, l'apprentissage du système hybride HMM/k-NN, basé sur une modélisation markovienne multiple, se déroule en deux étapes principales : La génération des modèles, et leur sélection. La figure 5.4 représente un

schéma fonctionnel décrivant les différentes étapes de la phase d'apprentissage. Notons que le processus de la figure 5.4 (a) se répète pour chaque classe C_i.

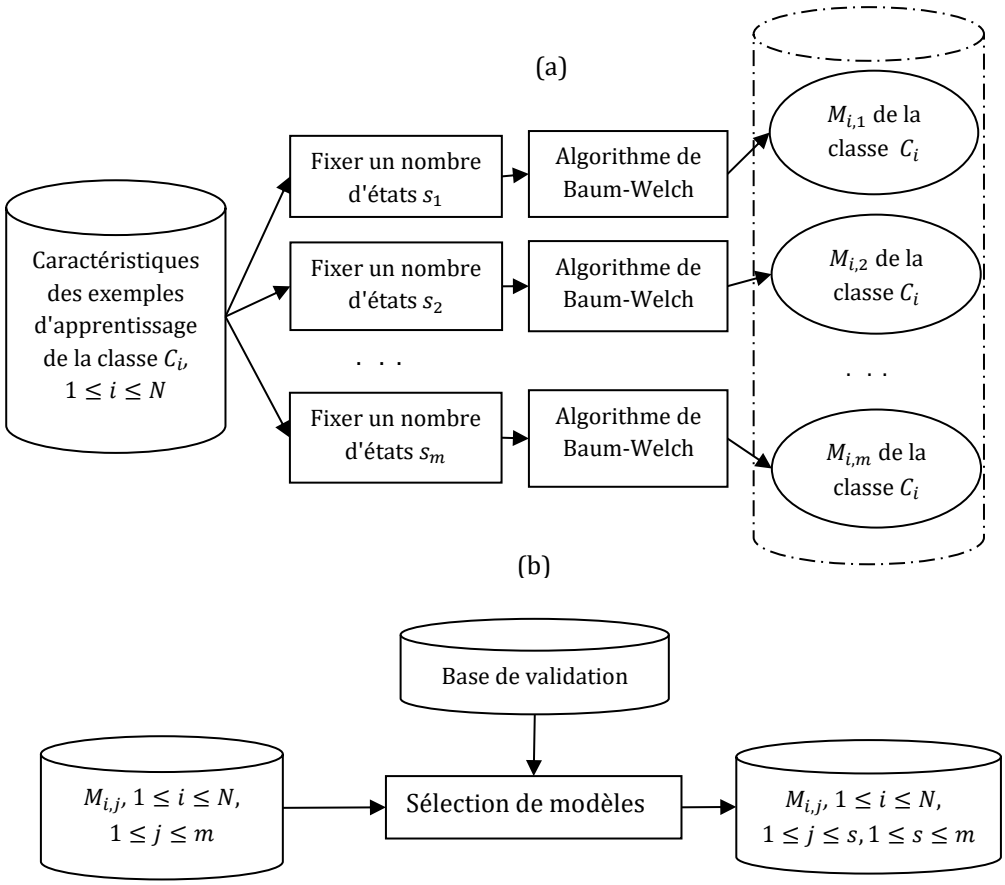


Fig.5.4. Principe général de la phase d'apprentissage dans l'architecture hybride HMM/k-NN, (a) étape 1 : génération de plusieurs modèles pour chaque classe, (b) étape 2 : sélection des s meilleurs modèles

5.4.2.1. Génération des modèles

Rappelons que l'apprentissage dans un classifieur HMM de base permet de représenter chaque classe de données par un modèle unique, et celui du classifieur k-NN, consiste seulement à stocker plusieurs exemples sous forme de vecteurs de caractéristiques pour représenter chaque classe. L'apprentissage proposé (Fig.5.4(a)) peut se présenter comme un mélange des deux. En considérant les vecteurs de caractéristiques de la base d'apprentissage, il consiste, à générer pour chaque classe C_i , $(1 \leq i \leq N)$ un ensemble de m HMM différents $(M_{i,1}, M_{i,2}, \dots, M_{i,m})$, où m est le nombre de configurations initiales. La différence entre les modèles se fait simplement en faisant varier l'un des paramètres de la configuration initiale (le nombre d'états en l'occurrence). Afin de réduire le nombre de HMM par classe, nous proposons d'utiliser uniquement des nombres d'états appartenant

à l'intervalle des valeurs plausibles $[s_1, s_m]$, où s_1 et s_m sont choisis de manière empirique. L'algorithme ci-dessous résume l'étape de génération des modèles.

Algorithme Génération des modèles

Données :

- Vecteurs de caractéristiques de l'ensemble d'apprentissage
- $C = \{C_1, C_2, \dots, C_N\}$: l'ensemble des N classes
- S: l'intervalle des nombres d'états possibles

Sorties :

Un ensemble de modèles différents pour chaque classe

Début

Pour tout $c \in \{C_1, C_2, \dots, C_N\}$ **faire**

Pour tout $s \in S$ **faire**

Appliquer l'algorithme d'EM sur l'ensemble d'apprentissage de la classe c pour générer un HMM avec s états.

Fin

Fin

Fin

Retourner l'ensemble des modèles avec leurs étiquettes de classes

5.4.2.2. Sélection des modèles

Afin de réduire l'espace de stockage et le temps de calcul pendant la phase de reconnaissance, sans affecter les performances du système, une sélection statique des s modèles les plus performants parmi les m modèles générés est effectuée (voir la figure 5.4(b)). Les modèles HMM sélectionnés seront ensuite fournis au module de reconnaissance qui se base sur une décision k-NN. Notons que la sélection est faite à l'aide d'une base de validation autre que celle d'apprentissage et celle de test.

À l'issue de la phase d'apprentissage, nous obtenons $(N * m)$ HMM avant la sélection du modèle et $(N * s)$ HMM avec leurs étiquettes de classe après la sélection du modèle, où $(1 \leq s \leq m)$.

5.4.3. La reconnaissance

Comme illustré sur la figure 5.3, la reconnaissance d'un nouvel exemple par le système hybride se pratique selon le même principe que la reconnaissance k-NN. Une fois l'extraction des caractéristiques de l'exemple à reconnaitre effectuée, l'algorithme de Viterbi et la règle k-NN sont combinés en procédant comme suit :

1. Calculer le logarithme de vraisemblance $(log\ P\ (O\ /\ M_{i,j})), 1 \leq i \leq N, 1 \leq j \leq s)$ que chaque HMM stocké $(M_{i,j})$ fait correspondre à l'exemple à reconnaitre O à l'aide de l'algorithme de Viterbi,

2. Trier tous les HMM par ordre décroissant en fonction de leurs logarithmes de vraisemblances,

3. Déterminer les k premiers HMM et leurs étiquettes de classe,

4. Sélectionner la classe majoritaire dans la liste des k voisins les plus proches (si elle existe) et affecter-la à l'exemple à reconnaitre.

5. S'il n'y a pas de classe majoritaire, un post-traitement est effectué avant de prendre la décision finale.

Pour montrer comment les HMM sont intégrés au sein de l'architecture k-NN, nous résumons dans les deux algorithmes ci-dessous le principe de reconnaissance pour le classifieur k-NN de base et pour le classifieur hybride HMM/k-NN.

Algorithme Reconnaissance *k*-NN

- Soit $A = \{(x', c) \setminus x' \in R^d, c \in C\}$ l'ensemble des échantillons d'apprentissage (plusieurs échantillons par classe).
- Soit x l'exemple dont on souhaite déterminer la classe

Pour chaque $(exemple\ (x', c) \in A)$ **faire**

Calculer la distance $D(x, x')$

Fin

Pour chaque $\{x' \in k\text{-}ppv(x)\}$ **faire**

Compter le nombre d'occurrences (échantillons) de chaque classe

Fin

Déterminer la classe majoritaire

Si (non ambiguïté) **alors**

Attribuer à x la classe majoritaire

Si non

Post-traitement

Fin

Fin

Algorithme Reconnaissance HMM/*k*-NN

- Soit $A = \{(m', c) \setminus m'$ est un HMM, $c \in C\}$ l'ensemble des modèles HMM créés à partir des échantillons d'apprentissage (plusieurs HMM par classe)
- Soit x l'exemple dont on souhaite déterminer la classe

Pour chaque (HMM$(m', c) \in A$) faire

Calculer la vraisemblance $P(x, m')$ par Viterbi

Fin

Pour chaque $\{m' \in k\text{-}ppv(x)\}$ faire

Compter le nombre de modèles HMM de chaque classe

Fin

Déterminer la classe majoritaire

Si (non ambiguïté) alors

Attribuer à x la classe majoritaire

Si non

Post-traitement

Fin

Fin

5.4.4. Le post-traitement

L'objectif de la phase de post-traitement est de permettre la prise de décision en cas d'ambiguïté, c'est-à-dire en cas où il n'y a pas de classe majoritaire. Pour cela, nous proposons deux méthodes différentes, l'une basée sur la maximisation de la vraisemblance moyenne des modèles, et l'autre basée sur la minimisation de la somme des positions (rangs) des modèles.

5.4.4.1. Traitement d'ambiguïté en maximisant la vraisemblance moyenne des modèles

Cette méthode que nous appelons ATL (l'abréviation anglaise de **A**mbiguity **T**reatment based models' **L**ikelihoods) est basée sur la maximisation des vraisemblances moyennes. Elle consiste à calculer, pour chaque classe candidate, la moyenne des logarithmes de vraisemblances de ses HMM appartenants à la liste des k voisins les plus proches en utilisant l'équation (5.1), puis à sélectionner la classe correspondante à la valeur la plus élevée comme le montre l'équation (5.2).

$$avg\_likelihood(C_i) = \frac{1}{n} \sum_{j=1}^{n} \log\left(P(O/M_{i,j})\right);\ 1 \leq i \leq nm. \qquad (5.1)$$

Où, nm est le nombre de classes concurrentes (ayant le plus grand nombre de modèles dans le k voisinage), n le nombre de modèles HMM que possède chaque classe candidate dans la liste des k plus proches voisins, et O l'exemple à reconnaitre.

La classe C^*choisie est celle qui maximise la moyenne de vraisemblance $avg\_likelihood$ (équation 5.2).

$$C^* = \underset{1 \le i \le nm}{\operatorname{argmax}}(avg\_likelihood(C_i)) \tag{5.2}$$

5.4.4.2. Traitement d'ambiguïté en minimisant la somme des positions des modèles

Dans cette méthode que nous appelons ATP (l'abréviation de **A**mbiguity **T**reatment based models' **P**ositions), nous proposons d'associer à chaque HMM dans la liste des k voisins les plus proches sa position (rang) en fonction des logarithmes de vraisemblance par rapport à l'exemple à reconnaitre. Parmi les nm classes concurrentes, la classe reconnue C* est celle minimisant la somme des positions de ses HMM (cf. équations 5.3 et 5.4).

$$sp(C_i) = \sum_{j=1}^{n} Pos(M_{i,j}, O);\ 1 \le i \le nm. \tag{5.3}$$

Où $sp(C_i)$ est la somme des positions des modèles de la classe C_i et $Pos\ (M_{i,j}, O)$ représente la position du modèle $M_{i,j}$ par rapport à l'exemple à reconnaitre O.

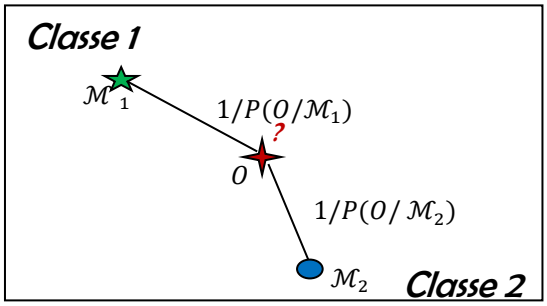
$$C^* = \underset{1 \le i \le nm}{\operatorname{argmin}}(sp(C_i)) \tag{5.4}$$

5.5. COMPARAISON THEORIQUE

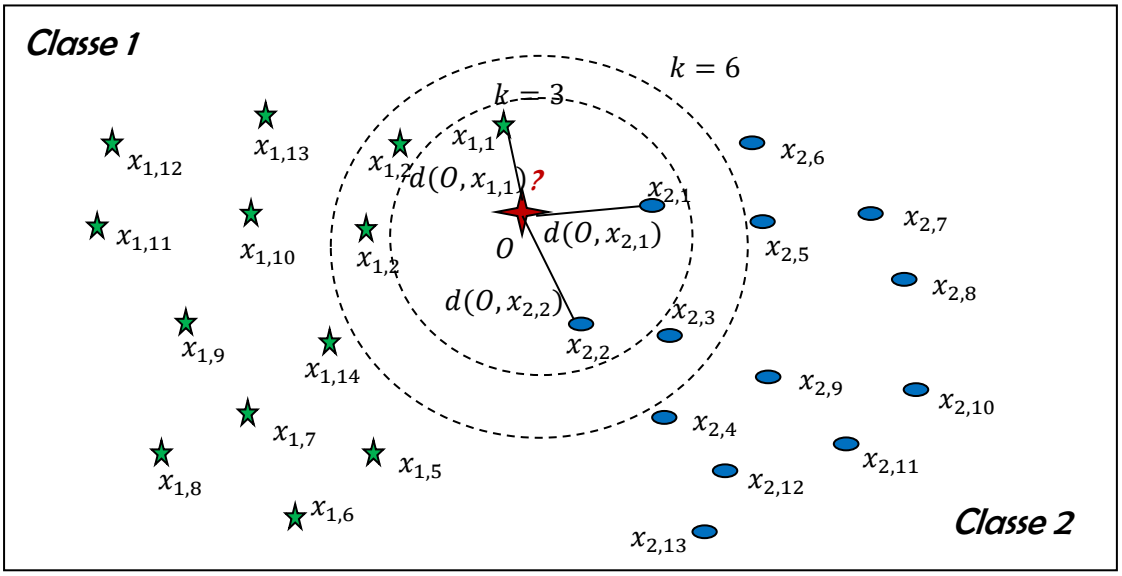
Dans cette section, nous comparons théoriquement le système hybride HMM/k-NN avec les deux systèmes de base : HMM et k-NN. La principale différence entre le système HMM et le système k-NN est que, dans le premier (Fig.5.5 (a)), l'apprentissage consiste à construire un modèle unique M_i pour représenter chaque classe de données i, et ce, en appliquant l'algorithme EM sur les séquences des vecteurs des caractéristiques de tous les exemples d'apprentissage $x_{i,j}$ de la classe i. Au niveau de décision, nous sélectionnons la classe maximisant la vraisemblance $P(O/M_i)$ que le modèle M_i donne à l'exemple à reconnaitre O. Alors que, dans le système k-NN (Fig.5.5 (b)), l'apprentissage consiste, simplement, à stocker tous les vecteurs des caractéristiques des exemples d'apprentissage $x_{i,j}$ avec leurs étiquettes de classe, et la décision consiste à sélectionner la classe majoritaire dans l'ensemble des k vecteurs les plus proches. Quant au système hybride proposé (Fig.5.5 (c)), il se caractérise par une architecture k-NN, une modélisation HMM, et une décision k-NN basée sur la vraisemblance de Viterbi. L'apprentissage consiste à stocker plusieurs modèles HMM $M_{i,j}$ pour chaque classe i (au lieu de plusieurs vecteurs de caractéristiques ou d'un modèle unique). Au niveau décision, la classe majoritaire dans l'ensemble des k modèles les plus proches, en termes de vraisemblance de Viterbi $P\ (O\ /\ M_{i,j})$, est sélectionnée.

La figure 5.5 résume le principe des trois classifieurs via un exemple illustratif dans un contexte de classification binaire (bi-classes).

(a)



(b)



(c)

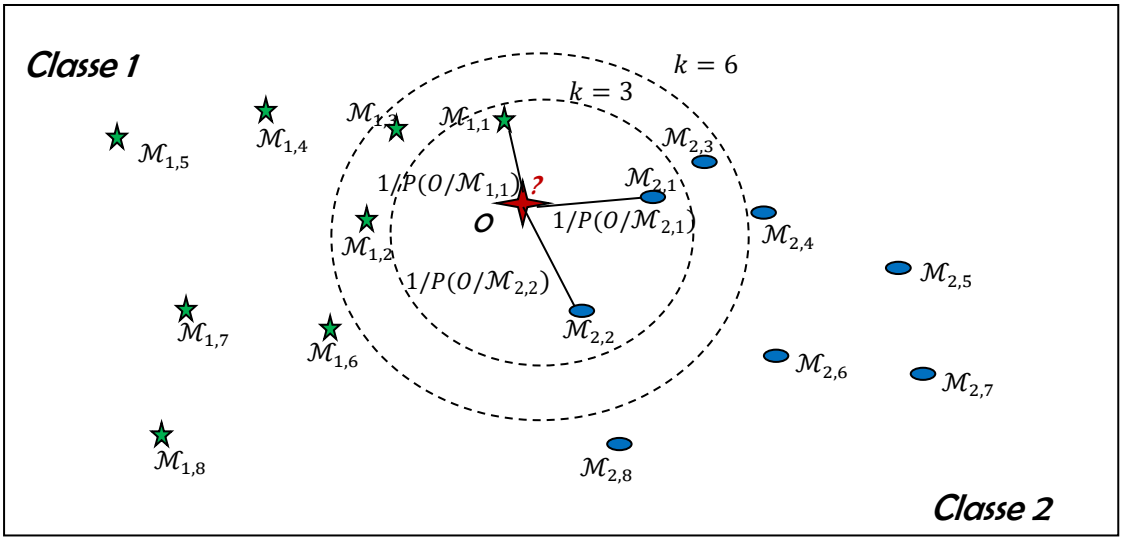


Fig.5.5. Exemple illustratif du principe du classifieur (a) HMM, (b) k-NN et (c) HMM/k-NN

Sur la figure 5.5 (a), l'étoile correspond à l'unique HMM $\mathcal{M}_1$ représentant la classe 1, et le cercle correspond à l'unique HMM $\mathcal{M}_2$ représentant la classe 2. L'exemple à reconnaitre O est affecté à la classe 2, celle qui maximise le logarithme de la vraisemblance $P(O/\mathcal{M}_i)$, car, comme le montre la figure, on a $1/P(O/\mathcal{M}_2) < 1/P(O/\mathcal{M}_1)$.

Sur la figure 5.5 (b), les étoiles représentent les vecteurs de caractéristiques de l'apprentissage $x_{1,j}$ de la classe 1, et les cercles représentent les vecteurs de caractéristiques des exemples de l'apprentissage $x_{2,j}$ de la classe 2. Si k=3, l'exemple à reconnaitre O est affecté à la classe 2, car parmi les 3 plus proches exemples, on a 2 exemples appartenant à la classe 2 et un seul exemple appartenant à la classe 1. Pour k=6, on n'a pas de classe majoritaire (3 exemples pour chaque classe), on est, donc, face à un problème d'ambiguïté. Pour résoudre ce problème, en cas d'une application bi-classes, il suffit de choisir une valeur impaire pour k. En revanche, pour une application multi-classes, un postraitement permettant d'enlever cette ambiguïté est indispensable.

Sur la figure 5.5 (c), chaque classe est représentée par plusieurs HMM $M_{i,j}$. Comme pour le cas du classifieur k-NN, l'exemple à reconnaitre O est affecté à la classe majoritaire. Ainsi pour k=3, la classe majoritaire (ayant plus de modèles dans la liste des k-HMM les plus proches) est la classe 2, alors que pour k=6, on n'a pas de classe majoritaire. Ce problème est résolu grâce aux deux méthodes de postraitement proposées ATL et ATP (cf. section 5.4.4 & section 5.6.3.1).

Dans les tableaux 5.1, 5.2, 5.3 et 5.4, nous donnons une comparaison théorique des trois systèmes k-NN, HMM et HMM/k-NN. Le tableau 5.1 explique le principe d'apprentissage de chaque système, le tableau 5.2 montre l'entrée et le principe de reconnaissance, et les tableaux 5.3 et 5.4 résument respectivement les avantages et les inconvénients de chaque système.

Tableau 5.1. L'apprentissage des trois classifieurs k-NN, HMM et HMM/k-NN

	k-NN	HMM	HMM/k-NN
Apprentissage	Stocker les vecteurs de caractéristiques des exemples d'apprentissage et leurs étiquettes de classe : Chaque classe est représentée par **plusieurs exemples**.	Créer un HMM unique pour chaque classe, et le stocker avec son étiquette de classe : Chaque classe est représentée par **un seul HMM.**	Créer plusieurs HMM pour chaque classe, et les stocker avec leurs étiquettes de classe : Chaque classe est représentée par **plusieurs HMM.**

Tableau 5.2. La reconnaissance dans les trois classifieurs k-NN, HMM et HMM/k-NN

	k-NN	**HMM**	**HMM/k-NN**
Entrées du module de reconnaissance	- Le vecteur des caractéristiques de l'exemple testé, - **Mélange de vecteurs de caractéristiques** avec leurs étiquettes de classes de tous les exemples d'apprentissage.	- La séquence d'observations représentant l'exemple testé, - **Un seul HMM** par classe avec son étiquette de classe.	- La séquence d'observations représentant l'exemple testé, - **Mélange de modèles HMM** (par classe) avec leurs étiquettes de classes.
Principe de la reconnaissance	- Comparaison de type **exemple/exemple**, - Calcul des **distances** entre le vecteur de caractéristiques de l'exemple à reconnaître et tous les vecteurs de caractéristiques stockés, - Choisir la classe **la plus fréquente dans la liste des k vecteurs les plus proches** en termes de distance.	- Comparaison de type **exemple/modèle,** - Calcul du logarithme de la **vraisemblance** de la séquence à reconnaître pour chaque HMM stocké à l'aide de l'algorithme de **Viterbi**, - Choisir l'étiquette de classe correspondant à **la valeur la plus élevée** du logarithme de la **vraisemblance**.	- Comparaison de type **exemple/modèle**, - Calcul du logarithme de la **vraisemblance** de la séquence à reconnaître pour chaque HMM stocké à l'aide de l'algorithme de **Viterbi**, - Choisir la classe **la plus fréquente dans la liste des k HMM les plus proches** en termes de logarithme de la **vraisemblance**.

Tableau 5.3. Avantages des trois classifieurs k-NN, HMM et HMM/k-NN

k-NN	HMM	HMM/k-NN
- Précision élevée en cas d'utilisation de grandes quantités de données d'apprentissage. - Simplicité d'implémentation et robustesse. - La possibilité d'ajouter de nouveux exemples de données sans refaire l'apprentissage.	- Moins de ressources, comparé au classifieur k-NN (quelques modèles stockés). - Fondement mathématique et algorithmes efficaces pour l'apprentissage et la reconnaissance des séquences de longueurs variables (cas du signal de parole). - Aucune normalisation de données n'est nécessaire.	- Bon compromis entre temps de calcul et précision. - Prendre en compte la sensibilité des HMM à la variabilité des données et aux paramètres d'apprentissage. - Basé sur une modélisation acoustique solide plutôt que des vecteurs de caractéristiques et sur la vraisemblance comme mesure de similarité plutôt que la distance. - Aucune normalisation de données n'est nécessaire.

Tableau 5.4. Inconvénients des trois classifieur k-NN, HMM et HMM/k-NN

k-NN	HMM	HMM/k-NN
- La nécessité d'une grande quantité de données étiquetées pour avoir des performances acceptables. - Un grand espace de stockage. - Un temps de calcul important : besoin de calculer les distances entre le vecteur de caractéristiques à reconnaître et les vecteurs de caractéristiques de tous les exemples d'apprentissage. - L'utilisation de la distance peut ne pas refléter la similitude dans le cas de grandes dimensions - La nécissité d'une normalisation de la taille des vecteurs des caractéristiques, ce qui provoque une perte d'information conséquente.	- La qualité (précision) des modèles est très sensible à la variabilité des données et aux paramètres d'apprentissage. - La nécessité d'une grande quantité de données étiquetées pour l'apprentissage.	- Le temps de calcul pendant la phase de reconnaissance est plus important que celui du classifieur HMM de base (on utilise plusieurs modèles au lieu d'un seul).

5.6. EXPERIMENTATIONS, RESULTATS ET DISCUSSION

L'objectif principal de cette section est de valider notre approche proposée, en mesurant sa qualité en termes de performances et de robustesse, et en la comparant, d'une part, avec les classifieurs de base HMM et *k*-NN et, d'autre part, avec les travaux de la littérature ayant utilisé la même base de données.

Pour l'implémentation, nous avons utilisé Matlab 2013 et HTK (HMM toolkit) qui est une boîte à outils logicielle propriétaire pour la gestion des HMM. Il est principalement destiné à la reconnaissance de la parole.

5.6.1. La base de données utilisée

Pour bien apprécier les performances des classifieurs présentés dans ce chapitre, nous avons utilisé la base de données standard UCI Spoken Arabic Digit SAD (Lichman 2013). Il s'agit de la base la plus citée en ce qui concerne les systèmes de reconnaissance de la parole arabe dans la dernière décennie. Elle contient 8800 échantillons prononcés par 88 locuteurs arabes (44 hommes et 44 femmes). Chaque chiffre est répété 10 fois par le même locuteur. Chaque échantillon est représenté par une série de 13 coefficients MFCC. La base d'apprentissage contient 6600 échantillons prononcés par 66 locuteurs, et la base de test contient 2200 échantillons prononcés par 22 locuteurs qui n'ont pas participé à la base d'apprentissage. Les systèmes conçus sont, alors, des systèmes indépendants du locuteur. Nous avons divisé la base d'apprentissage en deux parties. La première partie, contenant 5280 échantillons, réservée à l'apprentissage des modèles, et la deuxième, contenant 1320 échantillons, sert à la validation. Cette dernière est utilisée pour évaluer la qualité des modèles durant l'étape de sélection dans la phase d'apprentissage. Alors que, la base de test est utilisée pour évaluer le système final.

Nous avons choisi d'utiliser la base SAD pour deux raisons principales : Premièrement, elle est librement disponible sur le Net, ce qui nous permet de faire une comparaison directe avec les travaux précédents. Deuxièmement, un nombre relativement élevé de locuteurs ont participé à la réalisation de la base. Ceci permet d'étudier efficacement le problème de la variabilité interlocuteurs. Toutefois, sa limite principale est la non disponibilité des fichiers audio ; seules les caractéristiques MFCC sont distribuées, ce qui nous a empêché de tester d'autres techniques d'analyse acoustique telles que LPC, LPCC, PLP, etc.

La reconnaissance des chiffres parlés est nécessaire dans de nombreuses applications basées sur les chiffres, comme le composeur téléphonique vocal, les réservations de compagnies aériennes, les systèmes bancaires, l'automatisation de formulaires, et divers autres domaines.

La reconnaissance des chiffres prononcés est l'une des tâches les plus ardues dans le domaine de la RAP (Saleh & Wazir, 2018), notamment pour les langues peu dotées telles que l'Arabe. En effet, plusieurs travaux récents ont été appliqués sur des bases de chiffres

arabes, nous citons, à titre d'exemples, (Saleh & Wazir, 2018), (Wazir & Chuah, 2019), (Guerid et al., 2018), (Touazi & Debyeche, 2017), (Guerid & Houacine, 2019).

Avant d'évaluer notre approche hybride, nous allons d'abord, montrer expérimentalement l'effet du réglage initial des paramètres de l'apprentissage sur la stabilité du classifieur HMM de base.

5.6.2. Effet du réglage initial des paramètres de l'apprentissage sur la stabilité du classifieur HMM de base

Pour montrer la non stabilité du classifieur markovien vis-à-vis le réglage initial des paramètres de l'apprentissage, nous avons effectué plusieurs expérimentations. Dans chacune d'elles, nous examinons le comportement du classifieur en fonction du paramètre étudié en fixant les valeurs des autres paramètres. L'approche de modélisation utilisée est une approche globale, où un HMM de type gauche-droite à densité continue est généré pour chaque classe.

Le protocole expérimental que nous avons suivi pour étudier le comportement du système en fonction du paramètre étudié à chaque expérimentation, est résumé dans l'algorithme suivant.

Début

Initialisation :

Fixer les valeurs des paramètres non concernés

Itérations :

Pour chaque valeur possible du paramètre étudié faire

Construire un classifieur (un HMM par classe) par la méthode EM

Evaluer les performances du classifieur en termes de taux de reconnaissance

Fin pour

Examiner la variation du taux de reconnaissance en fonction du paramètre étudié

Fin

5.6.2.1. Effet du nombre d'états sur les performances

Durant notre expérience dans la modélisation acoustique de la parole par approche globale, nous avons constaté qu'il est rarement utile de choisir un nombre d'états inférieur à 5 ou supérieur à 20. Par conséquent, nous avons limité la plage de valeurs possibles à l'intervalle [5,20].

La figure 5.6 montre la relation entre le nombre d'états des HMM utilisés et les performances du classifieur en termes de taux de reconnaissance.

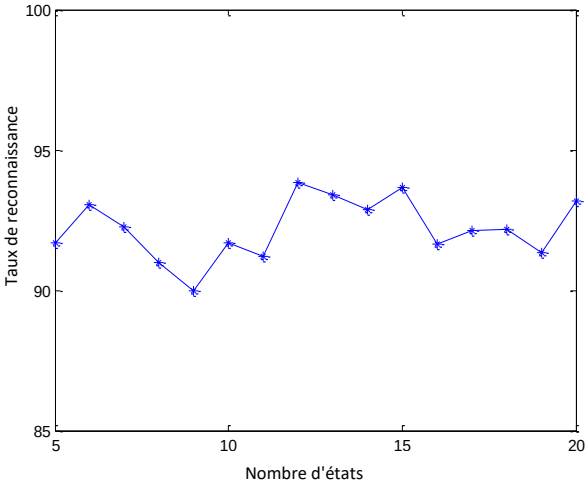


Fig.5.6. Effet du nombre d'états sur les performances du classifieur markovien

En examinant la courbe dans la figure 5.6, deux remarques principales peuvent être avancées:

- Les performances du classifieur HMM sont sensibles au nombre d'états (ici, dans la figure 5.6, le nombre optimal d'états est 12 et le meilleur taux de reconnaissance est 93,81%). Cette sensibilité peut diminuer la robustesse du système face à la variabilité des données.

- Il n'existe pas de règle permettant de déterminer le nombre d'états optimal (on ne peut pas imaginer comment va être le comportement du système si le nombre d'états est augmenté ou diminué), car le nombre d'états approprié à un exemple donné peut ne pas l'être à un autre exemple de la même classe, et ce, à cause de la grande variabilité intra-classe.

5.6.2.2. Effet du modèle initial sur les performances

Comme nous l'avons présenté dans le chapitre précédent, la 1ière étape de la méthode EM consiste à proposer un modèle initial. L'inconvénient majeur de cette méthode est que l'estimation des paramètres des modèles HMM n'est pas optimale puisqu'elle dépend fortement des valeurs des paramètres du modèle initial.

Dans cette expérimentation, nous avons fixé le nombre d'états à 12, le nombre de densités gaussiennes associées à chaque état à 1 et le nombre d'itérations de l'algorithme EM à 25, et avons varié les valeurs du modèle initial $\lambda_0 = (\pi_0, A_0, \mu_0, \sigma_0)$ afin d'examiner son impact sur les performances du système.

Les probabilités de départ (le vecteur π_0) et les probabilités de transition (la matrice A_0) sont choisies à chaque fois de façon aléatoire. Tandis que les valeurs du vecteur moyen μ_0 et de la matrice de covariance , σ_0 sont calculées en changeant à chaque fois les séquences d'observations (exemples) utilisées pour l'initialisation qui est fondée, essentiellement, sur l'algorithme de clustering « segmental k-means ». La figure 5.7 illustre la variation du

taux de reconnaissance du système en fonction de la variation du modèle initial. Elle montre clairement la sensibilité du classifieur HMM aux valeurs des paramètres du modèle initial.

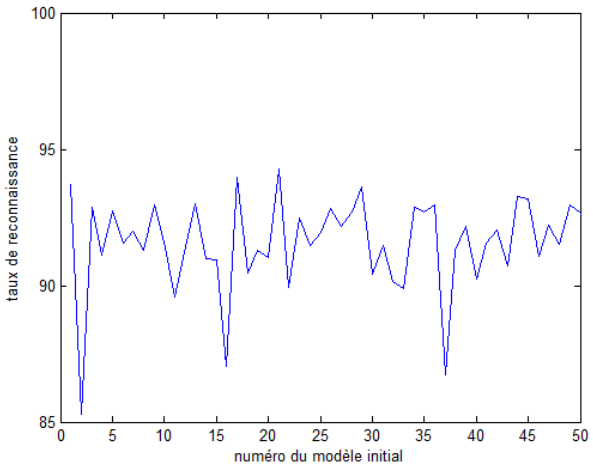


Fig.5.7. Effet du modèle initial sur les performances du classifieur markovien

5.6.2.3. Effet du nombre de densités gaussiennes sur les performances

Dans cette expérimentation, nous avons fixé le nombre d'états à 12, le nombre d'itération à 15 et un modèle initial fixe (choisi aléatoirement). Le nombre de densités gaussiennes par état est varier dans l'intervalle [1,15]. La figure 5.8 montre le comportement du classifieur HMM de base en fonction du nombre de densités gaussiennes.

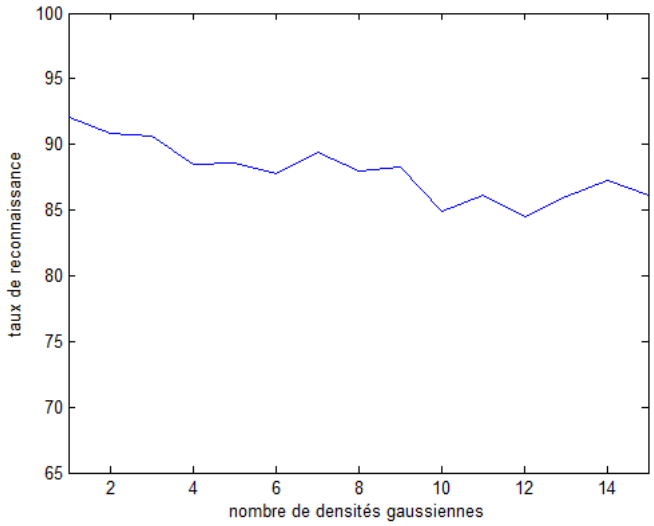


Fig.5.8. Effet du nombre de densités gaussiennes sur les performances du classifieur markovien

A partir de la figure 5.8, deux remarques principales peuvent être avancées : Premièrement, le classifieur HMM de base est très sensible à la variation du nombre de densités gaussiennes. Deuxièmement, la variation du taux de reconnaissance du système est inversement proportionnellement à celle du nombre de densités gaussiennes. Ce problème est lié, à notre avis, principalement à la base de données utilisée dont laquelle les exemples sont très courts (c.-à.-d., le nombre de vecteurs par exemple est insuffisant pour être répartis entre tous les états du modèle), ce qui conduit à une dégradation de performances avec l'augmentation du nombre de densités gaussiennes.

5.6.2.4. Effet du nombre d'itérations d'EM sur les performances

Durant cette expérimentation, nous avons fixé le nombre d'états, le modèle initial et le nombre de densités gaussiennes par état, et avons varié le nombre d'itérations de l'algorithme d'apprentissage dans l'intervalle [15,110] avec un pas de 5 itérations. Les résultats illustrés sur la figure 5.9 montrent la non stabilité du classifieur HMM de base vis à vis du nombre d'itérations.

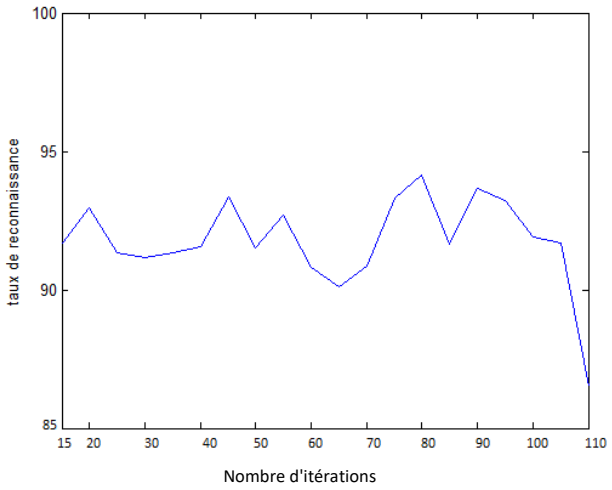


Fig.5.9. Effet du nombre d'itérations sur les performances du classifieur markovien

5.6.3. Evaluation de l'approche proposée

Cette section présente les séries d'expérimentations que nous avons réalisées afin d'évaluer l'efficacité de l'approche hybride.

5.6.3.1. L'effet de la valeur de *k* et l'apport du post-traitement proposé sur l'ambiguïté

Dans cette section, nous visons à étudier le comportement du système hybride sans et avec post-traitement proposé, en se focalisant essentiellement sur la modélisation multiple à différents nombres d'états. Le nombre de modèles générés, dans ce cas, est 160 (16 modèles * 10 classes), car l'intervalle des nombres d'états considérés est [5,20], ce qui donne 16 valeurs possibles, et donc 16 modèles différents par classe.

La figure 5.10 montre la variation du taux de reconnaissance du système hybride en fonction de la valeur de k, et ce, avant l'application du post-traitement proposé (Fig.5.10(a)) et après son application (Fig.5.10 (b)). Il est à noter que dans cette expérimentation, nous avons utilisé tous les 160 modèles générés sans sélection, k va donc prendre des valeurs entières allant de 1 à 160.

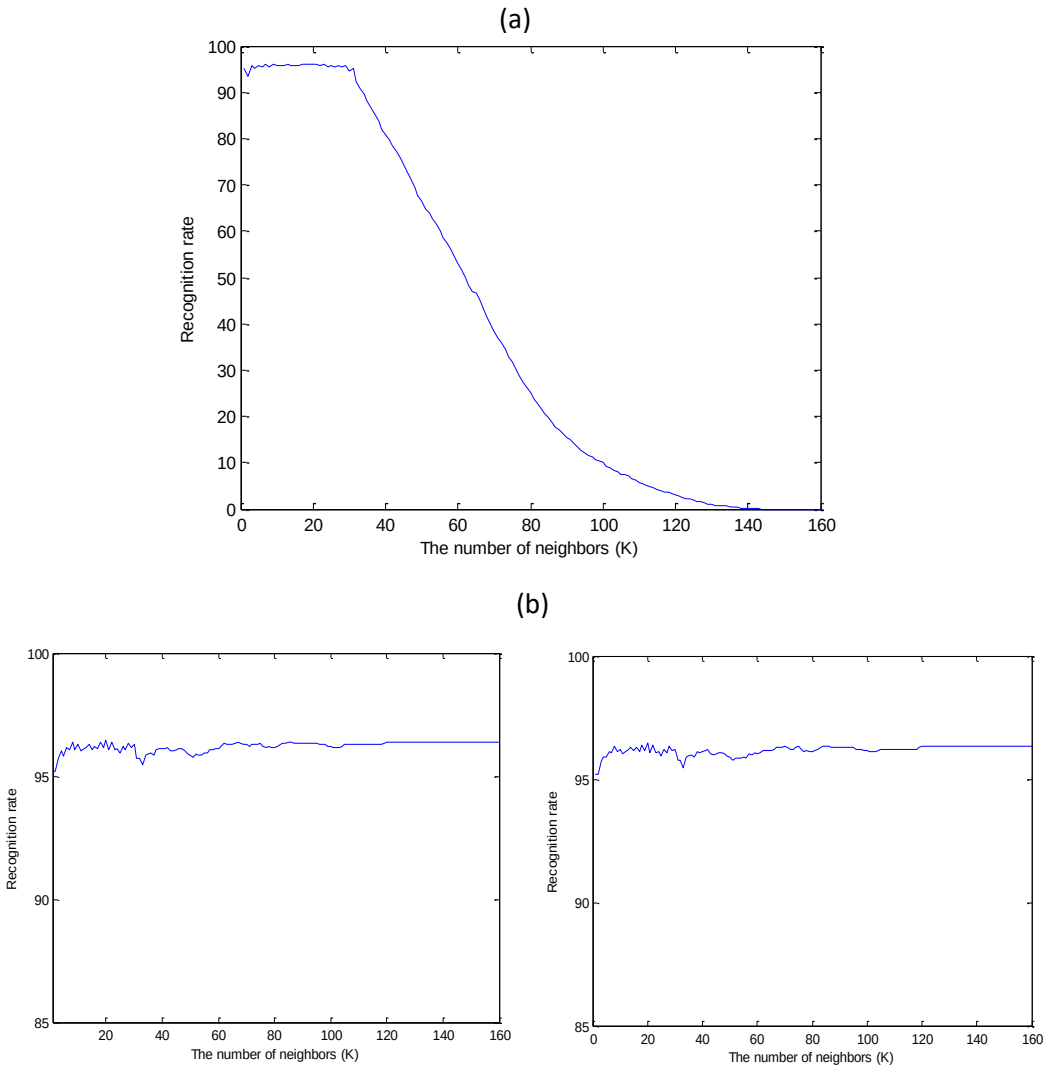


Fig.5.10. Taux de reconnaissance en fonction du nombre de voisins les plus proches (k) dans le système HMM/k-NN:

(a) avant le post-traitement,

(b) après le post-traitement : (gauche) traitement d'ambiguïté en maximisant la vraisemblance moyenne (ATL), (droite) traitement d'ambiguïté en minimisant la somme des positions dans la liste des k voisins (ATP) (ici, le meilleur taux de reconnaissance = 96,45% est atteint avec k = 20 pour les deux méthodes ATL et ATP)

Comme le montre la figure 5.10, les performances du classifieur hybride HMM/k-NN en cas d'une modélisation multiple à différents nombres d'états, dépondent fortement du choix du nombre de voisins les plus proches, k. La figure 5.10 (a) montre que l'augmentation de la valeur de k conduit à une dégradation dramatique des performances. Cela peut être justifié par l'augmentation des cas d'ambiguïté, c-à-d., les cas où on n'a pas de classe majoritaire. Si k=160 par exemple, nous remarquons que le taux de reconnaissance devient nul, car le taux d'ambiguïté dans ce cas est 100%. En effet, dans les 160 modèles les plus proches, chaque classe possède exactement 16 modèles, il n'y a donc pas de classe majoritaire. Ce problème est résolu en utilisant les méthodes de post-traitement proposées ATL (Fig.5.10 (b), gauche) et ATP (Fig.5.10 (b), droite). Les résultats présentés sur la figure 5.10 montrent l'apport du post-traitement. Nous remarquons également que, dans le cas de l'utilisation de l'ensemble entier des modèles générés (160 HMM), aucune supériorité n'a été marquée entre les deux méthodes proposées.

Des résultats pareils sont obtenus avec les trois autres paramètres d'apprentissage, à savoir le modèle initial, le nombre de densités gaussiennes et le nombre d'itérations. Cependant, nous avons marqué une légère supériorité de la méthode ATP sur la méthode ATL en cas des modèles crées à partir de différents modèles initiaux, et en cas des modèles ayant différents nombres de densités gaussiennes.

5.6.3.2. L'impact de la sélection des modèles sur les performances du classifieur hybride HMM/k-NN

Comme nous l'avons mentionné précédemment, les résultats montrés sur la figure 5.10 ont été obtenus lorsque tous les modèles générés sont utilisés dans la phase de reconnaissance, c'est-à-dire sans sélection de modèles. Afin de réduire la complexité en temps de calcul et en espace de stockage, sans affecter la performance du système, nous proposons de sélectionner et stocker uniquement les modèles les plus performants en termes de taux de reconnaissance sur la base de validation. Nous avons initialement créé pour chaque classe un ensemble de 16 HMM différents par leurs nombres d'états, nous les avons triés par ordre décroissant en fonction de leurs performances, puis nous avons stocké seulement les s premiers modèles. Dans le tableau 5.5, nous rapportons les résultats des expérimentations menées sur les données de test, qui n'ont pas été utilisées dans le processus de développement des modèles (apprentissage et validation).

Dans le tableau 5.5, les cases en surbrillance du gris (lorsque $s = 6$ avec la méthode ATL et $s = 10$ avec la méthode ATP) correspondent à nos meilleurs résultats. Les valeurs soulignées (pour $s = 1$) aux meilleurs taux de reconnaissance du système HMM de base et les valeurs en gras (pour $s = 16$) sont les meilleurs taux de reconnaissance en utilisant l'ensemble de tous les HMM générés (c-à-d., sans sélection).

Tableau 5.5 Taux de reconnaissance et meilleure valeur de k en fonction du nombre de modèles sélectionnés dans le cas de différents nombres d'états

Le nombre de modèles sélectionnés par classe (s)	Traitement d'ambiguïté par la méthode ATL		Traitement d'ambiguïté par la méthode ATP	
	Meilleure valeur de k	Taux de reconnaissance (%)	Meilleure valeur de k	Taux de reconnaissance (%)
1	1	93.81	1	93.81
2	5	95.81	4	95.54
3	11	96.00	4	95.86
4	17	96.40	6	96.18
5	23	96.50	8	96.45
6	29	96.86	29	96.68
7	35	96.81	35	96.59
8	14	96.68	12	96.68
9	44	96.63	14	96.59
10	50	96.77	50	96.72
11	42	96.68	55	96.54
12	19	96.63	19	96.68
13	16	96.59	16	96.59
14	16	96.45	16	96.40
15	10	96.45	10	96.45
16	20	**96.45**	20	**96.45**

La première remarque intéressante qui peut être faite, à partir du tableau 5.5, est qu'il n'existe pas de corrélation directe entre le nombre de modèles sélectionnés et les performances du système. En effet, le taux de reconnaissance, dans le cas de la méthode ATL, augmente avec l'augmentation du nombre de modèles jusqu'à $s = 6$ où il atteint sa meilleure valeur, puis il commence à changer de façon irrégulière. Ce comportement irrégulier du système en fonction du nombre de modèles sélectionnés se manifeste encore plus clairement dans le cas de la méthode ATP.

Nous remarquons, également, que notre meilleur résultat (96,86%) est atteint avec la méthode ATL lorsque ($s = 6\ et\ k = 29$). L'utilisation des 6 modèles les plus performants seulement conduit à augmenter le taux de reconnaissance de 96,45% à 96,86% et à réduire le nombre de HMM stockés de 160 (16 modèles * 10 classes) à 60 (6 modèles * 10 classes). Ainsi, une réduction de 62,5% d'espace de stockage et du temps de calcul en phase de reconnaissance est possible. Nous remarquons également que dans le cas de l'ATL, l'utilisation de 5 HMM par classe seulement génère un taux de reconnaissance de 96,50% qui dépasse celui obtenu lors de l'utilisation de 16 HMM par classe. Cela pourrait être un bon compromis entre le taux de reconnaissance et le temps de calcul durant le test.

Contrairement à l'expérimentation précédente (Fig.5.10), où nous n'avons marqué aucune supériorité entre ATL et ATP, cette expérimentation montre que, la méthode ATL donne des résultats légèrement meilleurs que ceux de la méthode ATP.

Ces résultats concernent seulement le cas de différents nombres d'états. Pour les autres cas, nous avons suivi la même démarche expérimentale, mais pour des fins de lisibilité et pour éviter de se perdre dans les chiffres, nous nous limitons aux résultats finaux qui sont récapitulés dans le tableau 5.6. Les meilleurs taux de reconnaissance (TR), dans chaque cas, sont marqués en gras et le meilleur de ces TR est souligné.

Tableau 5.6. Récapitulatif des résultats de sélection de modèles dans le système hybride HMM/k-NN

		Nombre d'états		Modèle initial		Nombre de gaussiennes		Nombre d'itérations	
		Sans sélection	Avec sélection	Sans sélection	Avec sélection	Sans sélection	Avec sélection	Sans sélection	Avec sélection
ATL	TR	96.45	**96.86**	96.45	**97.18**	92.45	**94.36**	95.86	**96.27**
	Valeur de k	20	29	117	12	4	13	98	12
	Nombre de modèles (s)	16	6	66	8	15	3	20	8
ATP	TR	96.45	**96.72**	96.56	**97.13**	92.77	**93.90**	95.81	**96.18**
	Valeur de k	20	50	117	16	4	15	24	12
	Nombre de modèles (s)	16	10	66	13	15	3	20	8

Comme prévu, en analysant les résultats présentés dans le tableau 5.6, nous pouvons remarquer que, dans tous les cas, l'utilisation d'un sous ensemble de modèles sélectionnés permet d'offrir des performances meilleures que celles de l'ensemble entier des modèles générés. Le meilleur TR (97.18%) est enregistré dans le cas de différents modèles initiaux avec la méthode de post-traitement ATL.

Nous pouvons également remarquer que la méthode ATL donne des résultats légèrement meilleurs que ceux de la méthode ATP. Par conséquent, celle-ci sera dorénavant utilisée comme méthode de post-traitement dans toutes les expérimentations à venir.

5.6.3.3. Matrices de confusion

Afin d'étudier statistiquement les erreurs de classification du meilleur système pour chaque cas, nous avons généré les matrices de confusion présentées dans les tableaux 5.7, 5.8, 5.9 et 5.10.

Un élément a_{ij} d'une matrice de confusion représente le nombre d'exemples de la classe i qui ont été affectés à la classe j. La diagonale indique, donc, le nombre d'exemples correctement reconnus.

Tableau 5.7. Matrice de confusion du système hybride en cas de différents nombres d'états

	Zéro Sifr	Un wahid	Deux ithnan	Trois thalatha	Quatre arbaa	Cinq khamsa	Six Setta	Sept sebaa	Huit thamania	Neuf tissaa	TR (%)
Zéro (sifr)	215	0	0	1	0	0	0	4	0	0	97.72
Un (wahid)	0	220	0	0	0	0	0	0	0	0	100
Deux (ithnan)	0	0	206	1	0	1	1	7	2	2	93.63
Trois (thalatha)	0	0	0	220	0	0	0	0	0	0	100
Quatre (arbaa)	0	0	0	0	204	0	0	14	0	2	92.72
Cinq (khamsa)	0	0	0	0	0	220	0	0	0	0	100
Six (setta)	6	0	0	0	0	0	213	1	0	0	96.81
Sept (sebaa)	5	0	0	0	7	0	0	208	0	0	94.54
Huit (thamania)	0	0	3	4	0	0	0	5	208	0	94.54
Neuf (tissaa)	0	0	0	0	0	0	1	2	0	217	98.63

Tableau 5.8. Matrice de confusion du système hybride en cas de différents modèles initiaux

	Zéro sifr	Un wahid	Deux ithnan	Trois thalatha	Quatre arbaa	Cinq khamsa	Six Setta	Sept sebaa	Huit thamania	Neuf tissaa	TR (%)
Zéro (sifr)	211	0	0	2	0	0	0	4	0	3	95.90
Un (wahid)	0	219	1	0	0	0	0	0	0	0	99.54
Deux (ithnan)	0	0	211	3	0	1	0	3	1	1	95.90
Trois (thalatha)	1	0	0	206	5	0	0	7	1	0	93.63
Quatre (arbaa)	0	2	0	0	217	1	0	0	0	0	98.63
Cinq (khamsa)	0	0	0	0	4	216	0	0	0	0	98.18
Six (setta)	3	0	0	0	0	0	217	0	0	0	98.63
Sept (sebaa)	10	0	0	0	2	0	0	208	0	0	94.54
Huit (thamania)	0	0	0	2	1	0	0	1	216	0	98.18
Neuf (tissaa)	0	0	0	0	0	0	2	1	0	217	98.63

Tableau 5.9. Matrice de confusion du système hybride en cas de différents nombres de densités gaussiennes

	Zéro sifr	Un wahid	Deux ithnan	Trois thalatha	Quatre arbaa	Cinq khamsa	Six setta	Sept sebaa	Huit thamania	Neuf tissaa	TR (%)
Zéro (sifr)	214	0	0	0	0	0	1	3	0	2	97.27
Un (wahid)	0	220	0	0	0	0	0	0	0	0	100
Deux (ithnan)	0	1	199	1	0	3	3	5	2	6	90.45
Trois (thalatha)	9	1	0	180	3	1	0	25	1	0	81.81
Quatre (arbaa)	0	1	0	0	212	3	0	4	0	0	96.36
Cinq (khamsa)	0	0	0	0	0	220	0	0	0	0	100
Six (setta)	8	0	0	0	0	0	211	1	0	0	95.90
Sept (sebaa)	5	0	0	0	1	1	3	209	0	1	95.00
Huit (thamania)	3	4	1	4	3	0	0	6	199	0	90.45
Neuf (tissaa)	0	0	0	0	0	1	5	2	0	212	96.36

Tableau 5.10. Matrice de confusion du système hybride en cas de différents nombres d'itérations

	Zéro Sifr	Un wahid	Deux ithnan	Trois thalatha	Quatre arbaa	Cinq khamsa	Six Sitta	Sept sebaa	Huit thamania	Neuf tissaa	TR (%)
Zéro (sifr)	211	0	0	3	0	0	0	4	0	2	95.90
Un (wahid)	0	220	0	0	0	0	0	0	0	0	100
Deux (ithnan)	0	0	201	0	0	2	5	6	3	3	91.36
Trois (thalatha)	3	0	0	204	6	0	0	6	1	0	92.72
Quatre (arbaa)	0	0	0	0	210	2	0	8	0	0	95.45
Cinq (khamsa)	0	0	0	0	4	216	0	0	0	0	98.18
Six (sitta)	0	0	0	0	0	0	219	1	0	0	99.54
Sept (sebaa)	8	0	0	1	2	0	0	209	0	0	95.00
Huit (thamania)	1	0	1	4	1	0	0	3	210	0	95.45
Neuf (tissaa)	0	0	0	0	0	0	2	0	0	218	99.09

En analysant les différentes matrices de confusion, nous observons que la majorité des confusions concernent des classes qui sont phonétiquement proches (par exemple, 7 (sebaa) avec 4 (arbaa), 0 (sifr) avec 6 (sitta) et 0 (sifr) avec 7 (sebaa)), ou qui partage le même type de voyelles dans le même ordre (par exemple, la confusion de 5 (khamsa) avec 4 (arbaa), de 9 (tissaa) avec 6 (sitta) et de 8 (thamania) avec 3 (thalatha)). D'autres confusions inattendues ont également surgi telles que la confusion de 2 (ithnan) avec 6 (sitta), avec 7 (sebaa) et avec 9 (tissaa) et de 3 (thalatha) avec 4 (arbaa) et 7 (sebaa). Cependant, il est intéressant de signaler que ces dernières erreurs de confusion sont enregistrées pour le même locuteur, cela nous laisse penser qu'il s'agit d'un problème de mal prononciation. Malheureusement, rien ne permet de confirmer ce constat, car, comme nous l'avons évoqué précédemment, les fichiers audio de la base de données utilisée ne sont pas disponibles.

Une solution au problème de confusion consiste, à notre avis, à considérer d'autres caractéristiques discriminantes telles que la combinaison de plus d'une méthode d'analyse acoustique, à savoir la méthode MFCC avec LPC ou l'ajout des caractéristiques dynamiques des coefficients MFCC (dérivées premières et secondes) qui ont montré des améliorations notables des résultats dans de nombreuses publications. En effet, il a été montré dans une étude précédente sur la base de données SAD (Hammami et al. 2012) que l'addition des caractéristiques dynamiques des coefficients MFCC conduit à améliorer considérablement le taux de reconnaissance du classifieur markovien (*"It has been shown that, the second-order derivatives of MFCC parameters compared to the MFCC yield improved rates of 4.60% for Continuous HMM"* (Hammami et al. 2012)). Malheureusement,

seuls les coefficients MFCC de la base de données SAD sont disponibles sur le Net, ce qui nous a empêché de considérer d'autres caractéristiques pour améliorer encore nos résultats.

Une autre solution éventuelle consiste à appliquer la technique de l'apprentissage discriminatif (*discriminative training*) ou à utiliser les critères discriminatifs de sélection de modèles, notamment le critère DIC (*Discriminative Information Criterion*) permettant de construire des modèles plus discriminatifs.

5.6.3.4. Comparaison avec les classifieurs de base

Le système hybride proposé est comparé en termes de performance, de robustesse et de temps de calcul avec les deux systèmes de base HMM et k-NN. Les résultats sont calculés sur la base de données SAD décrites précédemment.

- **Comparaison en termes de performances**

Le graphique dans la figure 5.11 montre les résultats d'évaluation des performances des trois systèmes en termes de taux de reconnaissance moyen (TR). Les systèmes sont triés de gauche à droite du moins performant au plus performant.

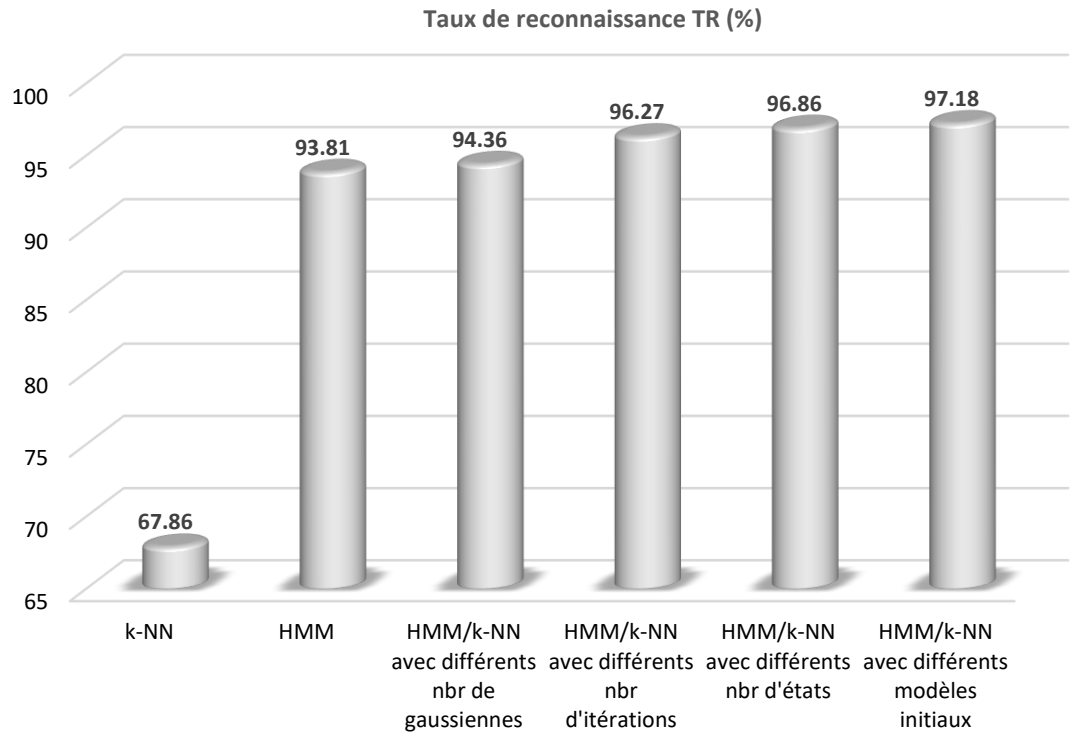


Fig.5.11. Comparaison des performances des systèmes k-NN, HMM et HMM/k-NN

La figure 5.11 montre que le système hybride donne toujours des résultats meilleurs que ceux des classifieurs de base HMM et k-NN. Il est également intéressant de remarquer la supériorité du système utilisant différents modèles initiaux sur tous les autres cas avec un TR égale à 97.18%. Cela pourrait, à notre avis, être justifié par le fait que les modèles

générés dans ce cas sont plus diversifiés et, donc, capables de prendre en compte plus de variabilité de données.

- **Comparaison en termes de robustesse**

Le système développé est indépendant du locuteur, car les 22 locuteurs de test n'ont pas participé à l'enregistrement de la base d'apprentissage, ce qui implique une variabilité intra-classe importante due à la grande variabilité interlocuteurs. Un système robuste ne doit pas être trop sensible à cette variabilité.

Pour étudier la robustesse de notre système hybride face à la variabilité interlocuteurs, nous proposons d'utiliser deux paramètres de dispersion tirés de la théorie de probabilité et des statistiques, à savoir l'écart-type et le coefficient de variation. Nous avons calculé l'écart type des taux de reconnaissance des locuteurs. Plus cette écart-type est faible plus le système est robuste et, donc, peu sensible à la variation du locuteur.

L'écart type interlocuteurs est calculé comme suit :

$$\sigma = \sqrt{\frac{1}{n}\sum_{loc=1}^{n}(Tr_{loc} - \mu)^2} \tag{5.5}$$

Où, n est le nombre de locuteurs (ici, n =22), Tr_{loc} est le taux de reconnaissance sur l'ensemble d'exemples du locuteur loc et μ est le taux de reconnaissance moyen des locuteurs (correspond au taux de reconnaissance global du système).

Nous avons, également, calculé le coefficient de variation (CV), dit écart-type relatif RSD (Relative Standard Deviation). C'est une mesure relative de la dispersion des données autour de la moyenne. Ce coefficient est défini comme le rapport entre l'écart-type σ et la moyenne μ, et s'exprime, souvent, en pourcentage (voir équation 5.6). Son avantage est qu'il permet de comparer le degré de variation, même si les moyennes sont différentes.

Plus la valeur du coefficient de variation est faible, plus la dispersion autour du taux de reconnaissance moyen est faible, et donc, plus la robustesse du système face à la variabilité interlocuteurs est grande.

$$CV = \frac{\sigma}{\mu} * 100 \tag{5.6}$$

Le tableau 5.11 présente les résultats obtenus en termes de robustesse mesurée par l'écart-type et le coefficient de variation interlocuteurs.

Les résultats illustrés dans le tableau 5.11 montrent clairement que, dans tous les cas, notre système hybride est plus robuste que le classifieur HMM de base, pour lequel nous avons enregistré l'écart-type et le coefficient de variation les plus importants (11.2512 et 11.99). Ainsi, les systèmes hybrides peuvent être triés par ordre décroissant de leur robustesse comme suit : Le cas de différents modèles initiaux avec un écart-type de 4.4859 et un coefficient de variation de 4.62, le cas de différents nombres d'itérations avec

un écart-type de 6.1812 et un coefficient de variation de 6.42, le cas de différents nombres d'états avec un écart-type de 7.8334 et un coefficient de variation de 8.09. En fin, le cas de différents nombres de densités gaussiennes avec un écart-type de 8.8670 et un coefficient de variation de 9.40.

Tableau 5.11. Evaluation de l'approche hybride en termes de robustesse

	HMM de base	HMM/k-NN Nombre d'états	HMM/k-NN Modèle initial	HMM/k-NN Nombre de gaussiennes	HMM/k-NN Nombre d'itérations
TR Moyen μ	93.81	**96.86**	**97.18**	**94.36**	**96.27**
Nombre de modèles	1	6	8	3	8
Ecart-type σ	11.2512	**7.8334**	**4.4859**	**8.8670**	**6.1812**
Coefficient de variation CV	11.99	**8.09**	**4.62**	**9.40**	**6.42**

- **Comparaison en termes de temps de calcul**

Comparé au classifieur HMM de base, il est évident que la modélisation markovienne multiple est plus coûteuse en temps de calcul et en espace de stockage. En fait, cela n'est pas un problème si important pour les raisons suivantes :

- Dans le cas du classifieur HMM de base, le TR obtenu peut se diminuer considérablement en changeant la base de test, car le classifieur HMM est très sensible à la variabilité de données. Cependant, notre système est relativement stable et plus robuste. Nous pensons que l'amélioration, des performances et de la robustesse, obtenue pourrait compenser le temps de calcul durant le test qui reste acceptable. En effet, le temps de reconnaissance estimé d'un exemple de test, en utilisant 6 modèles par classe, est d'environ 0,1 secondes seulement. Par un simple calcul, nous pouvons déduire que notre système peut reconnaître environ 10 mots par seconde, ce qui est plus rapide que le rythme de la parole normal.

- Concernant la phase d'apprentissage, le temps de calcul n'est pas un problème aussi important parce que, contrairement au test, l'apprentissage se fait préalablement en mode hors ligne. Ainsi, pour le système HMM de base, l'apprentissage se fait en plusieurs essais (plusieurs cycles apprentissage-test) dans le but est de fixer les meilleurs paramètres et/ou choisir les meilleurs modèles, donc, pratiquement le temps d'apprentissage du système hybride est comparable à celui du système HMM de base, il peut même être inférieur en évitant de tester, à chaque fois, les modèles appris, car ces derniers ne seront pas mis en compétition mais ils seront tous gardés.

- Durant la reconnaissance, nous devons calculer la vraisemblance pour chaque HMM indépendamment des autres, et durant l'apprentissage, les HMM sont générés indépendamment les uns des autres. Le temps de calcul de notre système peut donc être considérablement réduit en utilisant le parallélisme grâce aux processeurs multi-cœurs très puissants.

5.6.3.5. Comparaison avec des travaux de la littérature

Afin de se situer par rapport aux approches de classification existantes dans la littérature, nous comparons, dans le tableau 5.12, nos résultats en termes de taux de reconnaissance TR avec certains travaux récents sur la base SAD.

D'après le tableau, il semble clairement que notre système fournit un taux de reconnaissance meilleur que ceux des travaux précédents. Notons par ailleurs que, malgré son importance, la robustesse, dans tous les travaux précédents de RAP, n'a pas été évaluée (quantifiée), ce qui nous a empêché de la considérer durant notre comparaison. Théoriquement, nous pensons que notre système est plus robuste, car il permet de prendre en compte la variabilité de données grâce à la modélisation multiple proposée.

Il est, également, interressant de remarquer que notre approche donne de meilleurs résultats, comparée même aux approches les plus récentes, à savoir celles basées sur l'apprentissage profond (Deep Learning), telles que le réseau de neurones convolutif CNN et le réseau de neurones récurrent de longue mémoire à court terme LSTM.

Tableau 5.12. Comparaison des performances avec l'état de l'art

Référence	Méthode de classification	TR (%)
Hu et al., 2011	Réseaux de neurones et ondelettes où l'ondelette de *Morlet* est introduite dans la couche cachée.	96.72
Zhang & Zhang, 2011	Intégration du réseau de neurones et *k*-means clustering	90.68
Hammami et al., 2012	Modèle d'approximation des distributions d'arbres	93.16
	DHMM (HMM à densité discrète)	90.97
Ettaouil et al., 2013	Un modèle hybridant les réseaux de neurones artificiels et les HMM (NN/HMM)	86
Li et al., 2013	Réseau de neurones probabiliste	92.41
Li et al., 2015	Ensembles de réseaux neuronaux probabilistes	93.95
Shen et al., 2017	Combinaison de plus proche voisin à large marge (LMNN) et la déformation temporelle dynamique (DTW)	92.7

Ramli & Chien, 2017	Règles de décision pondérées dans une représentation collaborative des classifieurs	96.85
	Systèmes à Vaste Marge SVM	94.98
Lawal, 2017	Réseaux abductifs multiples	96.89
Iwana & Uchida, 2017	Réseaux de neurones convolutifs (CNN) avec alignement dynamique des poids	96.95
	Réseaux de neurones convolutifs (CNN)	94.77
	Réseau de neurones récurrent de longue mémoire à court terme (Long Short-Term Memory (LSTM) networks)	96
Iwana et al., 2019	Un réseau de neurones utilisant l'alignement dynamique entre les entrées et les poids.	96.10
Notre approche (Hazmoune et al., 2018)	HMM/k-NN (cas de différents nombres d'états)	96.86
	HMM/k-NN (cas de différents modèles initiaux)	**97.18**
	HMM/k-NN (cas de différents nombres de gaussiennes)	94.36
	HMM/k-NN (cas de différents nombres d'itérations)	96.27

5.7. CONCLUSION

Dans ce chapitre, nous avons présenté une nouvelle approche hybride couplant les HMM et la règle k-NN. Chaque classe est représentée par plusieurs HMM qui se diffèrent par leurs configurations initiales. Dans la phase de reconnaissance, la décision est prise en combinant l'algorithme de Viterbi avec la règle k-NN où, la classe la plus fréquente dans la liste des k-HMM les plus proches, en termes de logarithme de vraisemblance, est affectée à l'exemple à reconnaitre. En cas d'ambiguïté, un post-traitement est effectué avant de prendre la décision finale. Deux méthodes de post-traitement ont été proposées. La première consiste à sélectionner la classe la plus fréquente qui maximise les vraisemblances moyennes de ses HMM appartenant à la liste des k voisins les plus proches. La deuxième méthode est basée sur la sélection de la classe minimisant la somme des positions de ses HMM par rapport à l'exemple à reconnaitre. Les résultats obtenus ont montré l'apport des deux méthodes dans le traitement d'ambiguïté avec une légère supériorité de la première.

L'intérêt de notre approche est qu'elle permet de faire rejoindre les avantages des deux classifieurs de base HMM et k-NN, et de réduire leurs inconvénients respectifs. En effet, elle permet, d'une part, de tirer profit de la robustesse du k-NN lors de l'utilisation de plusieurs représentants pour la même classe (modélisation multiple) au lieu d'un représentant unique (cas du classifieur HMM de base), ce qui conduit à prendre en compte la variabilité des données et, par conséquent, d'améliorer la robustesse et le pouvoir de

généralisation du classifieur. D'autre part, elle permet de tirer profit de la solide modélisation acoustique des HMM en représentant les classes par des modèles acoustiques, au lieu des vecteurs de caractéristiques (cas du classifieur k-NN). Cela permet, ainsi, de réduire l'espace de stockage parce qu'on n'aura pas besoin de stocker les vecteurs de caractéristiques de tous les exemples de la base d'apprentissage qui, en pratique, peut contenir des millions d'exemples, mais seul un nombre restreint de HMM doit être gardé.

Comme prévu, les résultats obtenus en termes de performances et de robustesse, sur la base de données UCI Spoken Arabic Digit, sont encourageants et confirment la supériorité de l'approche proposée sur les travaux précédents et sur un k-NN et un HMM classique. De plus, notre approche permet d'alléger le problème du réglage initial des paramètres d'apprentissage des HMM grâce à la modélisation multiple.

Dans le prochain chapitre nous allons présenter notre deuxième approche ensembliste qui partage avec l'approche HMM/k-NN une première étape basée sur une modélisation markovienne multiple.

CHAPITRE 6

UNE APPROCHE ENSEMBLISTE BASEE SUR UNE MODELISATION MARKOVIENNE MULTIPLE

SOMMAIRE

6.1. INTRODUCTION

Comme nous l'avons montré dans les deux chapitres précédents, le réglage initial des paramètres de l'apprentissage des HMM joue un rôle crucial dans la stabilité du classifieur et sa robustesse à la variabilité de données. Pour éviter ce problème, nous proposons, dans ce chapitre, une méthode ensembliste homogène basée sur une modélisation markovienne multiple, dans laquelle nous exploitons la sensibilité des classifieurs HMM au choix des différents paramètres d'apprentissage pour créer la diversité nécessaire entre les classifieurs de base. En outre, nous réalisons une série d'expérimentations dans le but est d'examiner la relation éventuelle entre chaque paramètre d'apprentissage et dix différentes mesures de diversité, utilisées souvent dans la littérature, et ce, pour finalement choisir, parmi ces mesures, celles qui sont les mieux adaptées pour évaluer la qualité de l'ensemble proposé. L'objectif, à terme de cette approche est d'améliorer les performances et la robustesse du système de reconnaissance en combinant les prédictions de plusieurs classifieurs différents.

6.2. DESCRIPTION DE L'APPROCHE PROPOSEE

Le but des systèmes de reconnaissance de la parole est d'atteindre le meilleur taux de reconnaissance possible. En général, un certain nombre de classifieurs sont testés dans ces systèmes, et le plus approprié est choisi pour le problème en question. Malheureusement, ce test ne peut se faire que sur des bases de données limitées qui ne peuvent pas prendre en considération toutes les variations possibles de données. Ceci rend les classifieurs générés trop sensibles et peu robustes à la variabilité de données. Cela peut justifier le fait que les classifieurs font généralement des erreurs différentes sur différents échantillons de données, ce qui signifie qu'en combinant des classifieurs différents, nous pouvons créer un ensemble qui prend des décisions plus précises et plus robustes. Afin d'avoir des classifieurs avec différentes erreurs, il est recommandé de créer divers classifieurs et de les regrouper en ce qu'on appelle ensemble de classifieurs. La diversité entre les classifieurs peut être créée en choisissant des classifieurs hétérogènes basés sur des algorithmes différents, ou des classifieurs homogènes, mais qui se diffèrent dans leurs prédictions.

Pour avoir un ensemble de classifieurs homogènes avec des prédictions différentes, trois méthodes principales sont couramment utilisées. La première méthode consiste à apprendre les classifieurs sur des sous-ensembles différents de caractéristiques, comme la méthode des sous-espaces aléatoires RSM (Random Subspace Method). La deuxième méthode consiste à utiliser des sous-ensembles différents de données, telle que dans le Bagging et le Boosting. La troisième méthode consiste à utiliser des architectures différentes pour le même classifieur.

L'approche proposée s'insère dans le cadre des méthodes ensemblistes à classifieurs homogènes. Elle consiste à fusionner les décisions de plusieurs classifieurs markoviens obtenus à partir de différentes configurations initiales de l'algorithme d'apprentissage. L'utilisation de différentes configurations initiales permet, d'une part, d'avoir des

classifieurs avec des prédictions différentes et, d'autre part, de réduire l'effet des paramètres initiaux de l'apprentissage sur la stabilité du classifieur.

En plus des avantages des méthodes ensemblistes homogènes par rapport aux méthodes hétérogènes, à savoir la réduction de l'espace de stockage en utilisant les mêmes algorithmes pour tous les classifieurs de base, et le fait qu'aucune normalisation n'est nécessaire pour homogénéiser les sorties des classifieurs, notre approche possède d'autres avantages par rapport aux autres méthodes homogènes. Ces avantages peuvent être résumés dans les 2 points suivants :

- Contrairement au Boosting et au Bagging, aucun fractionnement de la base d'apprentissage n'est nécessaire, car tous les classifieurs de base sont appris sur la base entière. Ceci convient fortement aux classifieurs markoviens qui exigent une grande quantité de données pour atteindre des performances acceptables,

- Contrairement à la méthode des sous espaces aléatoires (RSM), le même ensemble de caractéristiques est utilisé pour tous les classifieurs de base. Ceci permet de réduire, considérablement, le temps de calcul en évitant de calculer à chaque fois de nouvelles caractéristiques, tant au niveau apprentissage qu'au niveau reconnaissance.

La figure 6.1 illustre un schéma bloc de l'architecture générale de l'approche proposée. Dans ce qui suit, nous présenterons en détail chaque étape de notre approche en mettant un accent particulier sur la manière par laquelle est créée la diversité entre les classifieurs de base. Notons que dans cette architecture, nous choisissons une modélisation globale où l'unité de base utilisée est le mot. Cette approche est la mieux adaptée aux systèmes de reconnaissance de mots isolés à vocabulaire limité. Cependant, l'architecture proposée est standard et peut s'appliquer à n'importe quel autre type d'unité (phonèmes, syllabes, etc.) et ce, en remplaçant les modèles de mots par les modèles de l'unité de modélisation choisie.

6.2.1. Extraction des caractéristiques (analyse acoustique)

L'extraction des caractéristiques est une étape clé dans tout système de RAP. Elle consiste à extraire un ensemble de caractéristiques pertinentes du signal de parole afin de réduire la dimensionnalité des données et de faciliter la tâche de classification. Pour obtenir des performances élevées, les caractéristiques extraites doivent être discriminantes, pertinentes et robustes. Plusieurs techniques d'extraction de caractéristiques ont été largement utilisées dans la littérature, telles que les coefficients MFCC, LPC, PLP, RASTA-PLP (RelAtive SpecTrAl Transform - Perceptual Linear Prediction), etc.

Dans notre thèse, nous optons pour la méthode MFCC (voir section 2.3.2.1.) pour les deux raisons suivantes : Tout d'abord, c'est la plus populaire de la littérature et elle a prouvé son efficacité. Deuxièmement, seuls les coefficients MFCC de la base de données que nous avons utilisée sont disponibles sur le Net (pas de fichiers audio pour calculer d'autres paramètres).

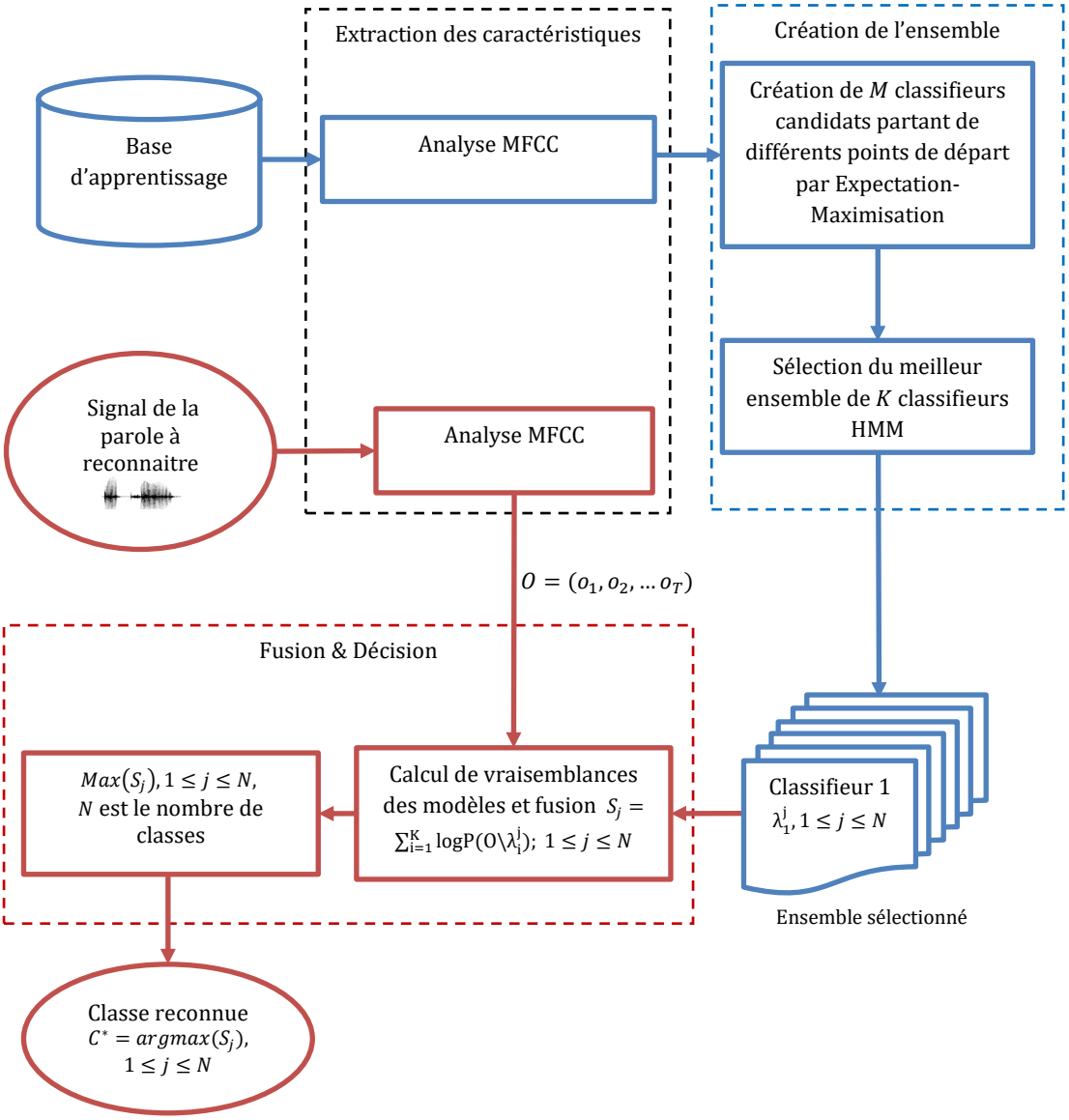


Fig.6.1. Schéma général de l'approche proposée

6.2.2. Création de l'ensemble

La création d'ensemble de classifieurs nécessite la construction de classifieurs individuels différents les uns des autres. Ces derniers ne doivent pas forcément être les plus performants possible mais doivent, lorsque combinés, fournir des performances meilleures que le plus performant des classifieurs de l'ensemble. Dans cette section, nous présentons la méthode de création de l'ensemble proposée. Elle se déroule en deux étapes : Premièrement, un large ensemble de classifieurs candidats, qui se diffèrent dans l'un des paramètres de la configuration initiale de l'algorithme d'apprentissage, est créé en utilisant la même base d'apprentissage et le même ensemble de caractéristiques. A la seconde étape, le meilleur sous-ensemble de classifieurs candidats en termes de performances et de diversité est sélectionné.

6.2.2.1. Génération des classifieurs candidats

Une fois l'analyse acoustique est faite afin d'extraire les vecteurs de caractéristiques de tous les échantillons de l'apprentissage d'une classe, l'algorithme EM (cf. chapitre 4) est utilisé pour générer M modèles pour chaque classe avec des configurations initiales différentes. L'approche de modélisation utilisée, comme indiqué précédemment dans ce chapitre, est une approche globale. Les modèles HMM entrainés sont donc des modèles de mots. Tous ces modèles ont une topologie gauche-droite, c'est-à-dire que seul le bouclage sur le même état ou les transitions vers les états suivants sont autorisées. À chaque état, nous associons m densités gaussiennes. Les modèles qui ont la même configuration initiale sont regroupés pour former un classifieur HMM individuel.

Voici ci-dessous, un algorithme résumant la démarche suivie pour générer les classifieurs candidats. Dans cet algorithme on se base sur le nombre d'états pour créer la diversité entre les classifieurs. Le même principe s'applique pour les autres méthodes de création de l'ensemble, à savoir différents modèles initiaux, différents nombres de densités gaussiennes et différents nombres d'itérations de l'algorithme d'apprentissage.

Pour chaque $s \in S \setminus S$: l'ensemble des nombres d'états possibles
  Pour chaque classe $c \in \{c_1, c_2, \dots, c_N\}$ \ l'ensemble des N classes
    Appliquer l'algorithme d'EM pour générer un HMM avec s états.
  Fin
  Regrouper les HMM établis dans un classifieur à s états
Fin
  Retourner l'ensemble des M classifieurs candidats

A l'issue de cette étape, nous obtenons un ensemble de M classifieurs candidats (M est la cardinale de l'ensemble S représentant le nombre de configurations initiales différentes par leurs nombres d'états). Dans cet ensemble, K ($1 \leq K \leq M$) classifieurs seulement seront sélectionnés pour être combinés.

6.2.2.2. Sélection du meilleur ensemble de classifieurs

L'un des problèmes les plus importants concernant la création d'ensembles de classifieurs est la sélection du meilleur sous-ensemble. Le mécanisme pour ce faire est conçu pour sélectionner des classifieurs adéquats à partir d'un large ensemble de classifieurs différents, afin que l'ensemble sélectionné puisse atteindre des performances optimales. La particularité de cette sélection est qu'elle ne s'intéresse pas nécessairement aux qualités individuelles des membres, mais plutôt aux qualités globales de l'ensemble.

La sélection peut être statique au niveau d'apprentissage où l'ensemble sélectionné est utilisé pour tous les exemples de test, ou dynamique à la reconnaissance où un nouvel ensemble est sélectionné pour chaque nouvel exemple de test. Dans ce travail, nous utilisons une sélection statique avec, comme critère de sélection, les performances

globales et la diversité de l'ensemble évaluées sur une base de validation, autre que celles d'apprentissage et de test.

Notons qu'en raison de la plage limitée de valeurs utiles de chaque paramètre de la configuration initiale de l'apprentissage, le nombre possible de classifieurs candidats est relativement restreint. Par conséquent, nous n'utilisons pas ici une technique d'optimisation pour la sélection des ensembles, mais nous proposons simplement de regrouper aléatoirement les classifieurs en un nombre limité de groupes différents puis de sélectionner le meilleur groupe de classifieurs parmi tous les groupes créés.

Le résultat final de l'étape d'apprentissage est un ensemble de K classifieurs, où chacun possède N modèles, avec N est le nombre de classes.

Bien que l'approche ensembliste et l'approche hybride HMM/k-NN, présentée dans le chapitre 5, partagent cette première étape de modélisation multiple, elles se diffèrent par la manière dont les modèles générés sont organisés dans l'espace de représentation et exploités durant la phase de reconnaissance. Dans l'approche hybride, les modèles et leurs étiquettes de classes sont stockés de façon aléatoire (l'espace de représentation peut être vu comme un mélange de modèles étiquetés), alors que dans l'approche ensembliste, les modèles générés sont organisés de telle sorte que chaque classifieur de l'ensemble ait un modèle par classe et que les modèles du même classifieur aient les mêmes paramètres de la configuration initiale de l'algorithme d'apprentissage. Par exemple, le premier classifieur possède N HMM à 5 états, chacun représente une classe, et le deuxième classifieur possède N HMM à 6 états, chacun représente une classe, etc. La figure 6.2 donne un exemple démonstratif de l'organisation des modèles dans les deux approches proposées. Pour des fins de simplification, nous supposons que le nombre de classes est limité à N=2 et le nombre de modèles par classe à M=6.

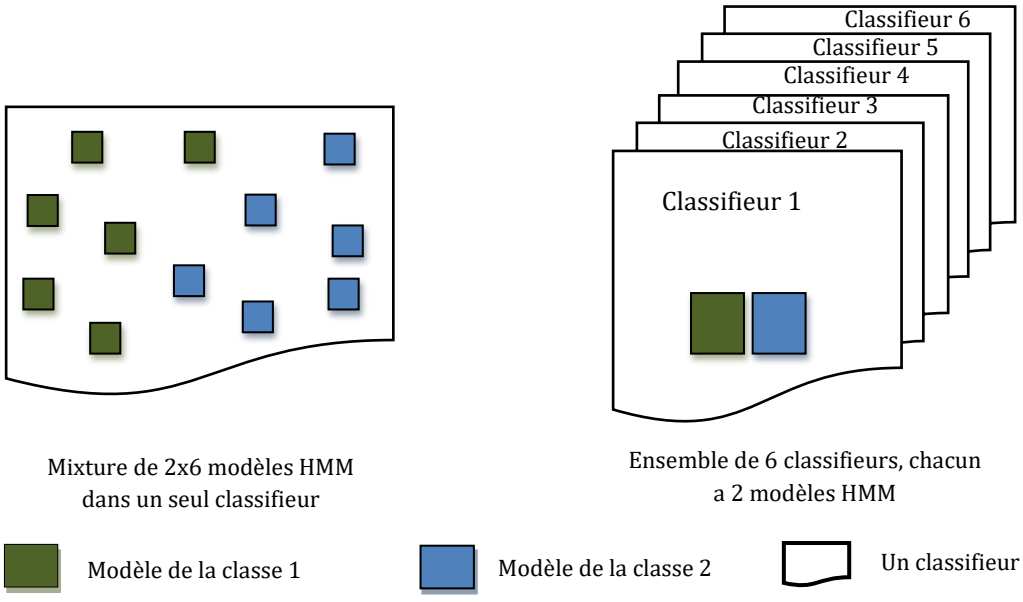


Fig.6.2. Organisation des modèles générés dans l'espace de représentation : l'approche hybride (à gauche), et l'approche ensembliste (à droite)

6.2.3. Fusion et Reconnaissance

Etant donné un signal acoustique de l'exemple à reconnaitre e, les N classes possibles, et l'ensemble des K classifieurs créés et sélectionnés lors de la phase d'apprentissage.

Pour classifier le nouvel exemple e, il est d'abord paramétrisé et représenté sous forme d'une suite de vecteurs de caractéristiques appelée séquence d'observations O. Cette opération est réalisée en appliquant la même méthode d'analyse acoustique utilisée pour paramétriser les exemples de la base d'apprentissage, en l'occurrence la méthode MFCC. Puis, il est passé à l'ensemble des classifieurs pour calculer son vraisemblance $P(O/\lambda_i^j)$ par rapport à tous les modèles des classes de chaque classifieur, avec λ_i^j est le $i^{ième}$ HMM de la $j^{ième}$ classe.

En fin, pour fusionner les réponses des classifieurs de base, nous proposons d'utiliser la somme des logarithmes de vraisemblances. L'idée est d'affecter à l'exemple à reconnaitre O la classe C^* qui maximise la somme des logarithmes de vraisemblances données par les classifieurs de base pour chaque modèle de classe, selon les équations suivantes :

$$S_j = \sum_{i=1}^{K} \log P(O \backslash \lambda_i^j);\ 1 \leq j \leq N \tag{6.1}$$

$$C^* = argmax(S_j), 1 \leq j \leq N \tag{6.2}$$

Dans la section expérimentations, nous allons également expérimenter la règle du vote majoritaire.

6.3. EXPERIMENTATIONS, RESULTATS ET DISCUSSION

Objectifs

Dans cette section, nous visons à étudier les points suivants :

- L'impact des différents paramètres de la configuration initiale de l'algorithme d'apprentissage sur la création d'ensembles basés HMM, en termes de performance et de diversité. Pour cela, quatre paramètres vont être étudiés : Le nombre d'états des HMM, le modèle initial de l'algorithme d'EM, le nombre de densités gaussiennes par état, et le nombre d'itérations de l'algorithme d'apprentissage.
- La robustesse à la variabilité intra-classe (variabilité interlocuteurs en particulier).
- L'impact de chaque méthode de création d'ensemble proposée sur 10 mesures de diversité prises de la littérature, ainsi que la relation entre ces mesures.
- L'impact de la taille de l'ensemble sur les performances.
- La relation entre la diversité et les performances de l'ensemble.

- L'impact de la diversité sur le gain de combinaison, qui est la différence entre le taux de reconnaissance de l'ensemble et le meilleur taux de reconnaissance des classifieurs individuels.

Configuration initiale

Les plages de valeurs et la configuration initiale des paramètres de l'algorithme d'apprentissage, pour chaque méthode de création de l'ensemble, sont présentées dans le tableau 6.1. La première colonne du tableau représente les méthodes de création d'ensemble proposées. Dans la deuxième colonne, nous indiquons le nombre de classifieurs candidats générés. La troisième colonne représente la plage de valeurs du paramètre modifié que nous limitons à celles qui sont raisonnables. Les colonnes 4, 5, 6 et 7 représentent la valeur choisie respectivement pour le nombre d'états, le modèle initial, le nombre de densités gaussiennes par état et le nombre d'itérations de l'algorithme d'apprentissage.

Tableau 6.1. Plage de valeurs et configuration initiale des paramètres de l'algorithme d'apprentissage pour chaque méthode de création de l'ensemble

	Le nombre de classifieurs générés	Plage de valeurs raisonnables du paramètre modifié	Le nombre d'états	Le modèle initial	Le nombre de densités gaussiennes	Le nombre d'itérations
Différents nombres d'états	26	De 5 à 30 états	/	Un seul modèle aléatoire	1	25
Différents modèles initiaux	66	Modèles aléatoires	10	/	1	25
Différents nombres de densités gaussiennes	12	De 1 à 12 gaussiennes	10	Un seul modèle aléatoire	/	25
Différents nombres d'itérations	20	De 15 à 110 avec un pas de 5 itérations	10	Un seul modèle aléatoire	1	/

Mesures de diversité utilisées

Les mesures de diversité permettent de quantifier la complémentarité des classifieurs individuels. Elles sont utilisables pour étudier la relation entre la diversité et les performances d'un ensemble de classifieur ou comme critère de sélection de l'ensemble.

Pour mesurer la diversité de l'ensemble, nous avons utilisé 4 mesures *par paire (pairwise)* et 6 mesures *globales (non-pairwise)*. Où une mesure *par paire* est calculée en faisant la moyenne des valeurs mesurées pour toutes les paires de classifieurs de base, tandis

qu'une mesure globale est prise sur l'ensemble des sorties des classifieurs. Le tableau 6.2 résume les mesures utilisées et leurs caractéristiques.

Notons que les mesures de diversité de l'ensemble, comme l'indique le tableau 6.2, peuvent être classées en deux catégories : La première catégorie comprend les mesures qui reflètent la similarité, c-à-d., plus la valeur est faible (↓), plus la diversité est grande. La deuxième catégorie regroupe les mesures qui reflètent la diversité, c-à-d., plus la valeur est élevée (↑) plus la diversité est grande. $Q, \rho, DF, k\ et\ \theta$ appartiennent à la première catégorie, alors que les autres mesures appartiennent à la seconde. On trouvera une bonne présentation de ces dix mesures dans (Kuncheva et Whitaker, 2001 ; Kuncheva et Whitaker, 2003 ; Shipp et Kuncheva, 2002).

Tableau 6.2. Les mesures de diversité utilisées

Mesure de diversité	Référence	Type	Similarité (↓) ou Diversité (↑)
Q-statistic (Q)	(Yule 1900)	Pairwise	↓
Correlation (ρ)	(Sneath & Sokal 1973)	Pairwise	↓
Disagreement (D)	(Ho 1998; Skalak 1996)	Pairwise	↑
Double Fault (DF)	(Giacinto & Roli 2001)	Pairwise	↓
Entropy of the votes (Ent)	(Cunningham & Carney 2000)	*Non-pairwise*	↑
Difficulty Index (θ)	(Hansen Salamon 1990)	*Non-pairwise*	↓
Kohavi-Wolpert variance (kw)	(Kohavi & Wolpert 1996)	*Non-pairwise*	↑
Interrater Agreement (k)	(Dietterich 2000; Fleiss et al., 2013)	*Non-pairwise*	↓
Generalized Diversity (GD)	(Partridge & Krzanowski 1997)	*Non-pairwise*	↑
Coincident Failure Diversity (CFD)	(Partridge & Krzanowski 1997).	*Non-pairwise*	↑

Pour l'implémentation des différentes mesures de diversité, nous avons utilisé un toolbox Matlab disponible sur ce lien : https://lucykuncheva.co.uk/ensemble_diversity.html.

Base de données utilisée

Afin d'analyser et évaluer les performances de l'approche proposée, plusieurs expérimentations ont été réalisées sur la base de données Spoken Arabic Digits (SAD) décrite dans le chapitre précédent. Nous avons divisé la base d'apprentissage en deux parties. La première, contenant 5280 exemples, est réservée à l'apprentissage des modèles, et la seconde partie, contient 1320 exemples pour la validation. Cette dernière est utilisée pour évaluer la qualité de l'ensemble en termes de performances et de diversité durant l'étape de sélection de la phase d'apprentissage. Alors que, la base de test est utilisée pour évaluer le système final.

6.3.1. Expérimentation 1 : Evaluation des performances de l'approche proposée

Cette expérimentation est réalisée dans le but de comparer les performances de l'approche proposée et celles des classifieurs de base en termes de taux de reconnaissance. Elle permet de valider et montrer le rôle de la complémentarité des classifieurs générés à partir de configurations initiales différentes dans le cadre d'un ensemble homogène de classifieurs markoviens. Les résultats présentés dans le tableau 6.3 correspondent au sous-ensemble le plus performant parmi un grand nombre de classifieurs candidats pour chaque méthode de création de l'ensemble. La première colonne représente les différentes méthodes de création de l'ensemble proposées. La deuxième colonne donne les performances du meilleur classifieur individuel. La troisième colonne représente les performances de l'ensemble en appliquant la règle de vote majoritaire comme stratégie de fusion, et la dernière colonne indique les performances de l'ensemble avec la règle de la somme des logarithmes de la vraisemblance.

Tableau 6.3. Taux de reconnaissance (%) du meilleur classifieur individuel et de l'ensemble pour les quatre méthodes de création de l'ensemble

	Meilleur taux individuel	Taux de l'ensemble avec la règle de vote majoritaire	Taux de l'ensemble avec la règle de la somme de vraisemblances
Différents nombres d'états	93.409	94.500	96.54
Différents modèles initiaux	**92.854**	**95.693**	**97.000**
Différents nombres de densités gaussiennes	92.590	94.000	94.818
Différents nombres d'itérations	92.757	95.700	96.31

Le tableau 6.3 révèle que la fusion des classifieurs donne toujours de meilleurs résultats, et ce, par comparaison à tous les classifieurs individuels. Cela montre que l'exploitation de la sensibilité des HMM à la configuration initiale de l'apprentissage a un impact important sur la création d'ensembles basées HMM, principalement lors de l'utilisation

de différents modèles initiaux avec la règle de la somme de vraisemblances, où nous obtenons le meilleur taux de reconnaissance (97%).

Une deuxième remarque peut être faite à partir du tableau 6.3. C'est que la règle de la somme de vraisemblances dépasse la règle du vote majoritaire dans tous les cas. Par conséquent, celle-ci sera utilisée, sans le mentionner, comme méthode de fusion dans toutes les expérimentations à venir.

D'autres résultats concernant le cas de différents modèles initiaux, sur une autre base de données personnelle et en utilisant un algorithme génétique pour la sélection d'ensembles, sont publiés dans (Hazmoune et al., 2013a ; Hazmoune et al., 2013b) et présentés en annexe.

6.3.2. Expérimentation 2 : Evaluation de la robustesse à la variabilité des données

Nous nous intéressons ici à l'étude du gain en termes de robustesse et de stabilité de notre approche face à la variabilité interlocuteurs. Pour ce faire, nous avons calculé l'écart type et le coefficient de variation interlocuteurs, présentés dans le chapitre 5. Plus les valeurs de ces paramètres sont faibles plus le système est robuste, et inversement, plus elles sont élevées, plus il y aura de chances que la performance du système se dégrade fortement du fait de la variabilité interlocuteurs. Le tableau 6.4 résume les résultats obtenus en termes de robustesse mesurée par l'écart-type et le coefficient de variation interlocuteurs.

Tableau 6.4. Evaluation de la robustesse à la variabilité interlocuteurs

	HMM de base	Ensemble avec différents nombres d'états	Ensemble avec différents modèles initiaux	Ensemble avec différents nombres de gaussiennes	Ensemble avec différents nombres d'itérations
TR Moyen (%)	93.81	**96.54**	**97**	**94.45**	**96.31**
Taille de l'ensemble	1	4	9	3	4
Ecart type	11.2512	**7.8177**	**5.1824**	**8.6008**	**6.6288**
Coefficient de variation	11.99	**8.10**	**5.34**	**9.11**	**6.88**

A partir du tableau 6.4, nous pouvons clairement remarquer la stabilité de notre approche et sa robustesse à la variabilité intra-classe, notamment dans le cas de modélisation

multiple à partir de différents modèles initiaux, où nous avons obtenu un écart type de 5.1824 et un coefficient de variation de 5.34, qui sont beaucoup plus faibles que ceux du classifieur HMM de base, pour lequel un écart-type de 11.2512 et un coefficient de variation de 11.99 ont été marqués.

6.3.3. Expérimentation 3 : L'impact de la taille de l'ensemble sur les performances

Le but de cette expérimentation est d'examiner la relation entre la taille de l'ensemble et les performances pour chaque méthode de création de l'ensemble. Le protocole expérimental (figure 6.3) suivi pour atteindre cet objectif est le suivant : Nous commençons par utiliser un seul classifieur, puis nous ajoutons progressivement un autre, et nous évaluons la précision de l'ensemble, et ainsi de suite, jusqu'à ce que tous les classifieurs générés soient utilisés. Les courbes dans la figure 6.4 montrent les résultats obtenus.

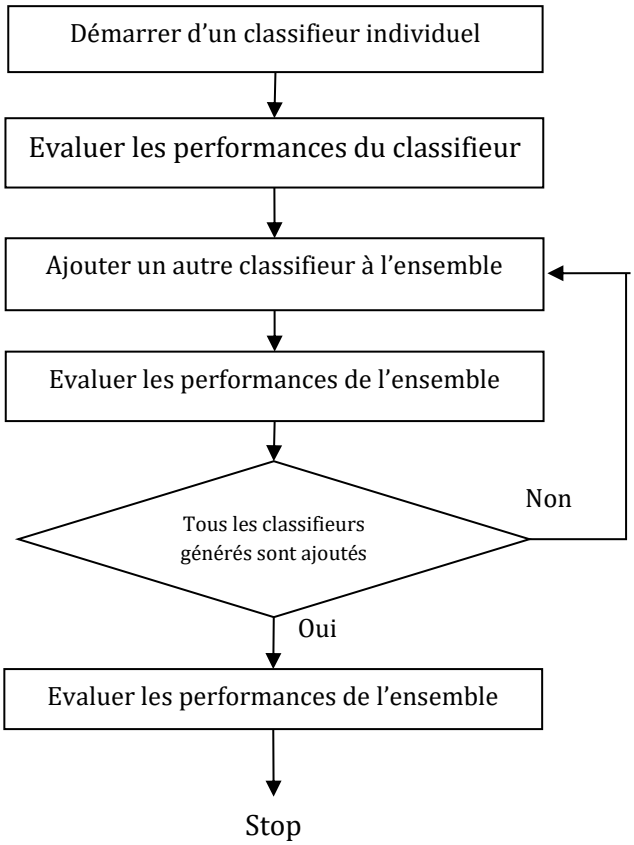


Fig.6.3. Protocole expérimental pour étudier l'impact de la taille de l'ensemble sur les performances

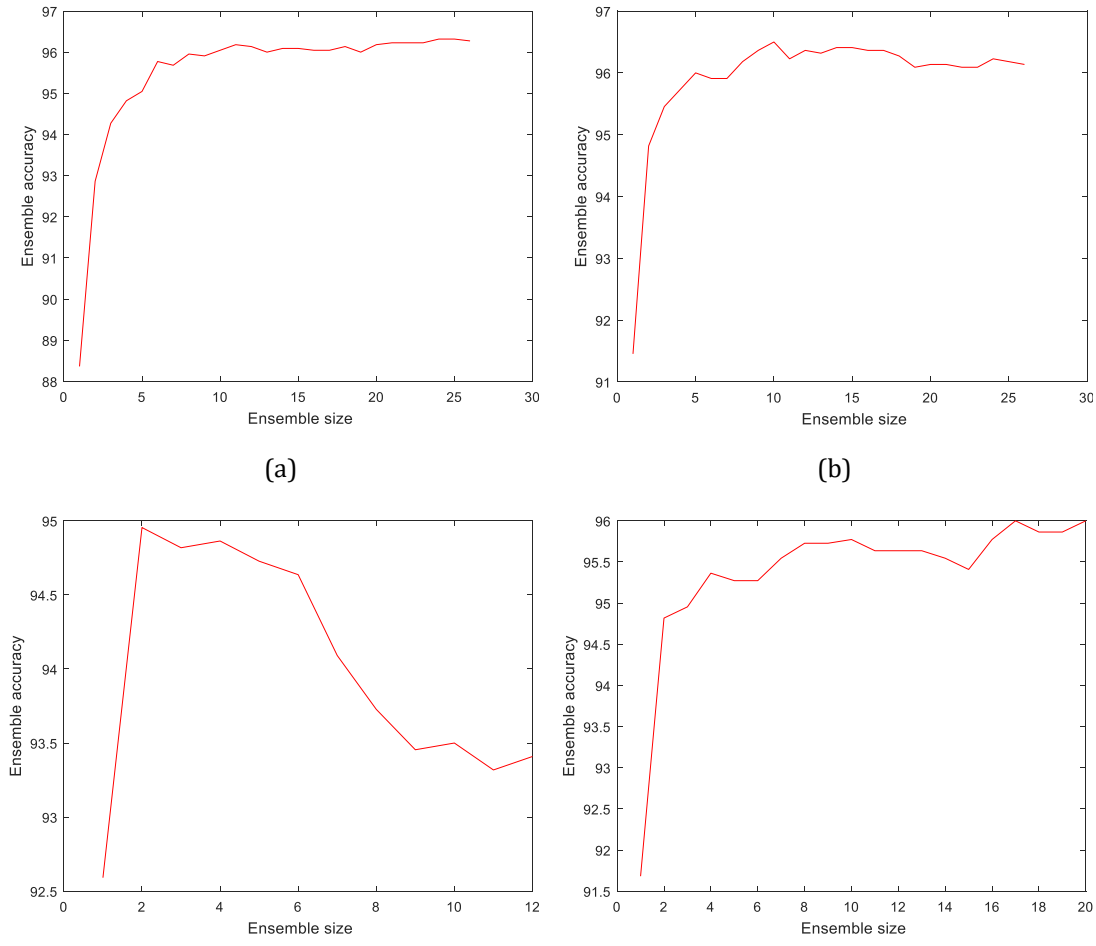


Fig.6.4. L'impact de la taille de l'ensemble sur les performances dans le cas de : (a) différents nombres d'états, (b) différents modèles initiaux, (c) différents nombres de densités gaussiennes, et (d) différents nombres d'itérations

Une analyse des résultats présentés dans la figure 6.4 conduit aux observations suivantes :

- L'ensemble donne toujours de meilleurs résultats que n'importe quel classifieur individuel. Ceci confirme les résultats obtenus dans l'expérimentation précédente.
- La taille de l'ensemble a un léger impact sur les performances dans les trois cas : Différents nombres d'états (Fig.6.4 (a)), différents modèles initiaux (Fig.6.4 (b)), et différents nombres d'itérations (Fig.6.4 (d)).
- L'augmentation du nombre de classifieurs de base n'améliore pas systématiquement les performances de l'ensemble, notamment dans le cas de différents nombres de densités gaussiennes (Fig.6.4 (c)). Pour ce cas précis, nous

avons marqué un impact inverse, qui peut s'expliquer par le fait que, le classifieur ajouté à chaque fois est plus faible que les classifieurs précédents. En effet, comme il a été montré dans le chapitre 5, les performances des classifieurs générés dans ce cas baissent lorsque le nombre de densités gaussiennes augmente. Par exemple, les performances des trois premiers classifieurs sont respectivement 92.59%, 92.50 et 90,59%, par conséquent, l'ajout du troisième classifieur à l'ensemble des deux premiers classifieurs affectera négativement et significativement les performances globales de l'ensemble.

À partir de ces observations, nous pouvons conclure que l'enrichissement de l'ensemble par l'addition d'un nouveau membre n'est pas forcément utile. Cela dépend de ses performances par rapport aux autres membres de l'ensemble et de son effet sur la diversité. Néanmoins, la taille de l'ensemble pourrait avoir un impact important sur les performances de l'ensemble si les classifieurs de base sont à la fois performants et divers.

6.3.4. Expérimentation 4 : Comparaison des 4 méthodes de création de l'ensemble en termes de performances et de diversité

Dans cette expérimentation, nous visons à étudier les quatre méthodes proposées de création de l'ensemble en termes de performances et de diversité.

Dans le tableau 6.5, les valeurs sont calculées pour 10 ensembles différents choisis aléatoirement, chacun contient 5 classifieurs de base. La première ligne dans ce tableau représente les méthodes de création de l'ensemble. La deuxième ligne représente le taux de reconnaissance moyen des 10 ensembles sur la base de validation. La ligne 3 reporte le taux de reconnaissance moyen des 10 ensembles sur la base de test. Les autres lignes indiquent les valeurs moyennes des différentes mesures de diversité ($Q, \rho, D, DF, kw, k, Ent, \theta, GD\ et\ CFD$) calculées sur la base de validation. Les meilleures valeurs de performances et de diversité sont marquées en gras et soulignées.

Les conclusions suivantes peuvent être tirées à partir du tableau 6.5 :

- Que ce soit pour la base de validation (ligne 2) ou pour la base de test (ligne 3), les quatre méthodes de création de l'ensemble peuvent être triées, en termes de performances, allant du meilleur au pire comme suit : Différents modèles initiaux (93.84% & 95.66%), différents nombres d'états (93.63% & 95.50%), différents nombres d'itérations (93.12% & 95.30%) et différents nombres de densités gaussiennes (90.87% & 92.97%).

- La majorité des mesures de diversité ($Q, \rho, DF, k, \theta, GD\ et\ CFD$) montrent que les classifieurs créés partant de modèles initiaux différents sont les plus diversifiés. Donc, on peut déduire qu'il existe un accord entre les performances et ces mesures de diversité.

- Nous pouvons clairement observer que $D, kw\ et\ Ent$ ne sont pas appropriés pour prédire laquelle des méthodes de création de l'ensemble proposées est meilleure. En effet, les meilleures valeurs de ces trois mesures sont obtenues en cas d'utiliser

différents nombres de densités gaussiennes. Cependant, cette méthode est la pire en termes de performances globales (92,977% vs 95,509%, 95,660% et 95,306%). Par ailleurs, en excluant ce cas de notre comparaison, nous obtenons un consensus total entre les dix mesures de diversité.

Tableau 6.5. Comparaison des 4 méthodes de création de l'ensemble en termes de performances et de diversité

		Différents nombres d'états	**Différents modèles initiaux**	Différents nombres de densités gaussiennes	Différents nombres d'itérations
Taux de reconnaissance moyen des 10 ensembles sur la base de validation		93.636	**93.846**	90.871	93.125
Taux de reconnaissance moyen des 10 ensembles sur la base de test		95.509	**95.660**	92.977	95.306
Mesures *Pairwise* calculées sur la base de validation	$Q \downarrow$	0.885	**0.856**	0.879	0.870
	$\rho \downarrow$	0.448	**0.409**	0.468	0.430
	$D \uparrow$	0.103	0.114	**0.123**	0.110
	$DF \downarrow$	0.052	**0.049**	0.073	0.052
Mesures *Non-Pairwise* calculées sur la base de validation	$kw \uparrow$	0.041	0.045	**0.049**	0.044
	$k \downarrow$	0.446	**0.403**	0.465	0.425
	$Ent \uparrow$	0.150	0.165	**0.180**	0.161
	$\Theta \downarrow$	0.052	**0.049**	0.066	0.052
	$GD \uparrow$	0.495	**0.533**	0.462	0.512
	$CFD \uparrow$	0.703	**0.729**	0.677	0.711

6.3.5. Expérimentation 5 : Sélection de l'ensemble

Il n'existe pas actuellement de consensus dans la littérature concernant le choix d'une mesure de diversité ou d'une autre mesure pour la sélection d'ensemble, car cela dépond essentiellement du problème étudié. Cette expérimentation vise à étudier les relations éventuelles entre les différentes mesures de diversité et les performances de l'ensemble dans le but est de choisir, parmi ces mesures, celle les mieux adaptées à notre approche, et de montrer si la diversité au sein des classifieurs de base est suffisante pour la sélection de l'ensemble. Pour atteindre ce but, nous avons considéré cinq ensembles numérotés de 1 à 5 pour chaque méthode de création de l'ensemble. Les dix mesures de diversité sont calculées sur la base de validation, alors que les performances des ensembles sont évaluées sur la base de test. Le tableau 6.6 présente les résultats obtenus dans le cas d'utilisation des classifieurs qui se diffèrent dans leurs nombres d'états. Le tableau 6.7 présente les résultats de l'utilisation de classifieurs avec différents modèles initiaux. Le

tableau 6.8 présente les résultats obtenus en utilisant des classifieurs avec différents nombres de densités gaussiennes. Les résultats, dans le cas de l'utilisation des classifieurs avec différents nombres d'itérations de l'apprentissage, sont rapportés dans le tableau 6.9. Dans chacun de ces tableaux, la première colonne représente le numéro de l'ensemble, la seconde colonne représente le taux de reconnaissance (TR) de l'ensemble et les autres colonnes représentent les mesures de diversité. Pour plus de lisibilité, les meilleures valeurs sont soulignées.

Tableau 6.6. Relation entre les mesures de diversité et les performances de l'ensemble dans le cas de l'utilisation de nombres d'états différents

Numéro de l'ensemble	TR (%) de l'ensemble sur la base de test	Mesures Pairwise sur la base de validation				Mesures Non-Pairwise sur la base de validation					
		Q ↓	ρ ↓	D ↑	DF ↓	kw ↑	K ↓	Ent ↑	θ ↓	GD ↑	CFD ↑
1	95.045	0.864	0.401	0.104	0.044	0.042	0.398	0.150	0.045	0.543	0.753
2	95.636	0.902	0.424	0.081	0.035	0.032	0.421	0.115	0.037	0.534	0.757
3	95.727	0.879	0.394	0.087	0.034	0.035	0.393	0.127	0.037	0.559	0.755
4	95.227	0.918	0.460	0.074	0.037	0.029	0.458	0.108	0.039	0.501	0.708
5	95.909	0.904	0.428	0.079	0.035	0.031	0.426	0.115	0.037	0.530	0.726

Tableau 6.7. Relation entre les mesures de diversité et les performances de l'ensemble dans le cas de l'utilisation de différents modèles initiaux

Numéro de l'ensemble	TR (%) de l'ensemble sur la base de test	Mesures Pairwise sur la base de validation				Mesures Non-Pairwise sur la base de validation					
		Q ↓	ρ ↓	D ↑	DF ↓	kw ↑	k ↓	Ent ↑	θ ↓	GD ↑	CFD ↑
1	96.000	0.811	0.335	0.114	0.034	0.045	0.313	0.160	0.037	0.623	0.811
2	96.181	0.859	0.372	0.091	0.033	0.036	0.374	0.132	0.036	0.576	0.763
3	95.545	0.864	0.371	0.090	0.032	0.036	0.369	0.130	0.035	0.580	0.770
4	96.636	0.799	0.302	0.107	0.029	0.043	0.296	0.155	0.033	0.645	0.810
5	94.772	0.888	0.425	0.090	0.040	0.036	0.423	0.130	0.042	0.526	0.731

Tableau 6.8. Relation entre les mesures de diversité et les performances de l'ensemble dans le cas de l'utilisation des nombres différents de densités gaussiennes

Numéro de l'ensemble	TR (%) de l'ensemble sur la base de test	Mesures Pairwise sur la base de validation				Mesures Non-Pairwise sur la base de validation					
		Q ↓	ρ ↓	D ↑	DF ↓	kw ↑	k ↓	Ent ↑	θ ↓	GD ↑	CFD ↑
1	94.727	0.887	0.426	0.094	0.043	0.037	0.425	0.136	0.044	0.522	0.731
2	91.227	0.907	0.508	0.106	0.070	0.042	0.506	0.157	0.065	0.432	0.646
3	94.818	0.883	0.401	0.089	0.036	0.029	0.400	0.134	0.044	0.551	0.733
4	92.454	0.916	0.503	0.093	0.058	0.031	0.502	0.140	0.063	0.445	0.647
5	91.090	0.902	0.498	0.107	0.068	0.036	0.497	0.161	0.071	0.441	0.628

Tableau 6.9. Relation entre les mesures de diversité et les performances de l'ensemble dans le cas de l'utilisation de différents nombres d'itérations

Numéro de l'ensemble	TR (%) de l'ensemble sur la base de test	Mesures Pairwise sur la base de validation				Mesures Non-Pairwise sur la base de validation					
		Q ↓	ρ ↓	D ↑	DF ↓	kw ↑	k ↓	Ent ↑	θ ↓	GD ↑	CFD ↑
1	95.727	0.893	0.419	0.090	0.038	0.036	0.412	0.131	0.040	0.538	0.743
2	94.590	0.890	0.421	0.093	0.040	0.037	0.411	0.133	0.041	0.537	0.747
3	96.000	0.886	0.391	0.080	0.030	0.032	0.390	0.115	0.033	0.566	0.760
4	95.727	0.887	0.401	0.085	0.033	0.034	0.394	0.121	0.036	0.558	0.771
5	95.045	0.908	0.445	0.082	0.038	0.032	0.440	0.120	0.040	0.514	0.712

Comme le montrent les résultats du tableau 6.6, il n'existe pas de relation bien établie entre les performances de l'ensemble et les mesures de diversité dans le cas de nombres d'états différents. En effet, l'ensemble le plus diversifié (numéro 3) n'est pas le plus

performant (numéro 5). Par conséquent, il n'est pas utile d'utiliser ces mesures de diversité comme critère de sélection de l'ensemble.

Contrairement au tableau 6.6, les résultats des tableaux 6.7, 6.8 et 6.9 indiquent clairement que :

- En cas d'utilisation de différents modèles initiaux pour créer l'ensemble (tableau 6.7), si l'on exclut les mesures $D, kw\ et\ Ent$, l'ensemble le plus diversifié (numéro 4) en termes de toutes les autres mesures de diversité est bien le plus performant. De même, pour le cas où l'on utilise différents nombres de densités gaussiennes (tableau 6.8), l'ensemble le plus diversifié (numéro 3) étant également le plus performant.

- En cas d'utilisation de nombres d'itérations différents comme méthode de création de l'ensemble (tableau 6.9), à l'exception de $D\ et\ Ent$, l'ensemble le plus diversifié en termes de toutes les mesures de diversité est également le plus performant (numéro 3).

A partir des 4 derniers tableaux, la relation entre les différentes mesures de diversité peuvent être récapitulées dans le tableau 6.10. Les croix indiquent les mesures de diversité ayant réussi à sélectionner l'ensemble le plus performant en termes de taux de reconnaissance.

Tableau 6.10. Tableau récapitulatif de la relation entre les différentes mesures de diversité et leur impact sur la sélection de l'ensemble

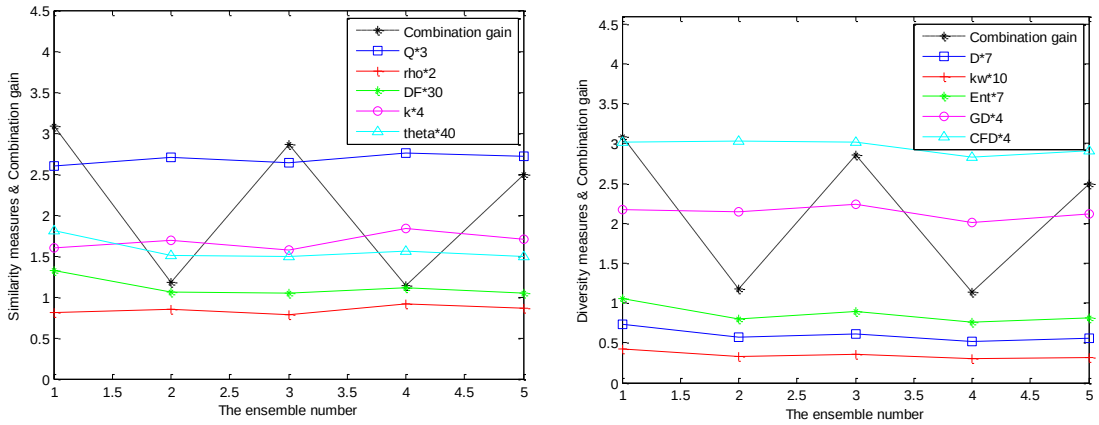
	Q	ρ	D	DF	kw	k	Ent	θ	GD	CFD
Différents nombres d'états										
Différents modèles initiaux	X	X		X		X		X	X	X
Différents nombres de densités gaussiennes	X	X		X		X		X	X	X
Différents nombres d'itérations	X	X		X	X	X		X	X	X

En conclusion de cette expérimentation, comme le montre le tableau 6.10, nous pouvons nous assurer que toutes les mesures de diversité, à l'exception de $D, kw\ et\ Ent$, peuvent jouer un rôle important dans la sélection de l'ensemble, sauf dans le cas d'utilisation de nombres d'états différents, où l'ensemble le plus diversifié n'est pas forcément le plus performant. Donc, pour sélectionner le meilleur ensemble, il suffit d'utiliser l'une des 7 mesures $Q, \rho, DF, k, \theta, GD, et\ CFD$ car un accord total est marqué entre toutes ces mesures. S'il n'y avait pas de consensus total, Il serait possible, de les faire voter pour améliorer encore le résultat.

6.3.6. Expérimentation 6 : L'impact de la diversité sur le gain de combinaison

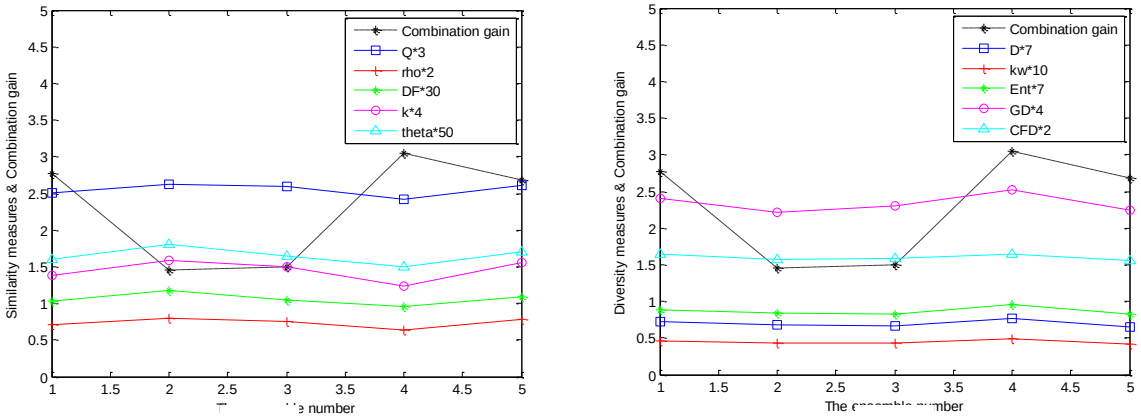
Nous visons à travers cette expérimentation à examiner la relation entre les différentes mesures de diversité et le gain de combinaison. Ce dernier correspond à la différence entre le taux de reconnaissance de l'ensemble et celui du meilleur classifieur de base. A cet effet, nous avons construit plusieurs ensembles par chacune des 4 méthodes proposées de création de l'ensemble. Pour chaque ensemble, nous avons calculé le gain de combinaison et la diversité de ses classifieurs de base en utilisant les dix mesures de diversité. Les résultats obtenus sont illustrés dans les figures 6.5, 6.6, 6.7 et 6.8, où l'axe des abscisses représente les numéros des ensembles générés et l'axe des ordonnées représente le gain de combinaison et les mesures de diversité.

Nous avons divisé les mesures de diversité en deux groupes : Celles qui reflètent la similarité entre les classifieurs de base (Q, ρ (repéré par *rho* dans les figures), $DF, k\ et\ \theta$ (repéré par *theta* dans les figures), elles sont illustrées sur les figures de gauche 6.5 (a), 6.6(a), 6.7(a) et 6.8(a), et celles reflétant la diversité des classifieurs de base ($D, kw, Ent, GD\ et\ CFD$) qui sont illustrées sur les figures de droite 6.5(b), 6.6(b), 6.7(b) et 6.8(b). Comme indiqué plus haut dans cette section, dans le cas des mesures de similarité, plus la valeur est petite, plus la diversité est grande, et dans le cas des mesures de diversité, plus la valeur est grande, plus la diversité est grande. Afin de rendre possible la représentation de plusieurs mesures différentes dans le même graphique, nous avons multiplié chaque mesure de diversité par une valeur positive, cela n'aura aucun effet sur la relation entre les courbes car toutes les mesures de la même courbe sont multipliées par la même valeur.



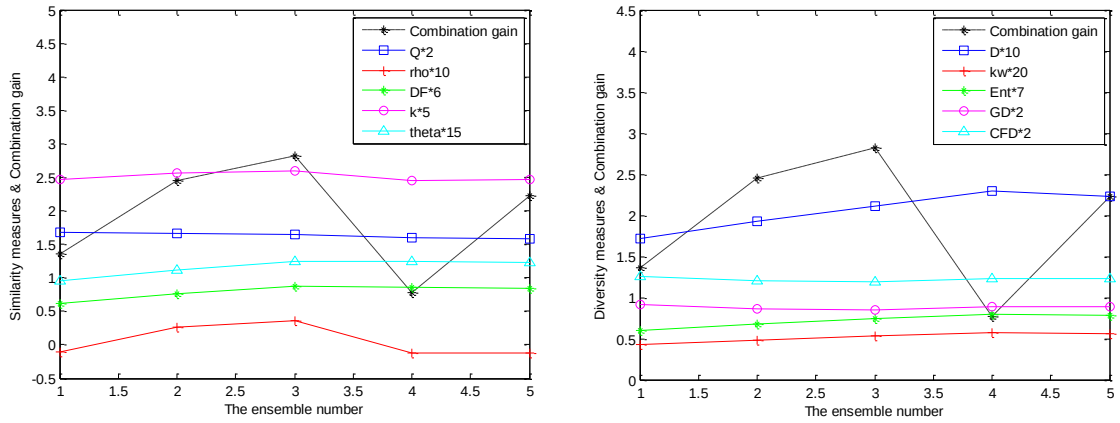
(a) (b)

Fig.6.5. Relation entre la diversité et le gain de combinaison dans le cas de différents nombres d'états : (a) Mesures de similarité (↓), et (b) Mesures de diversité (↑)



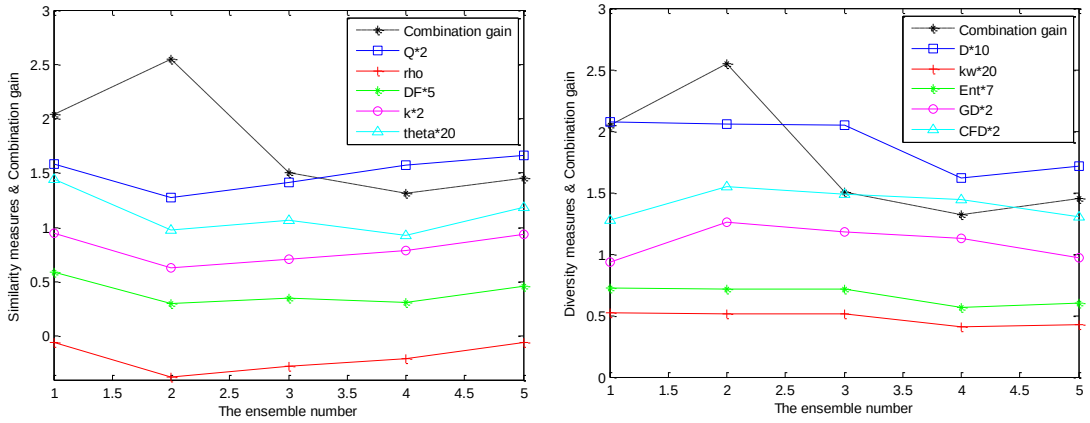
(a) (b)

Fig.6.6. Relation entre la diversité et le gain de combinaison dans le cas de différents modèles initiaux : (a) Mesures de similarité (↓), et (b) Mesures de diversité (↑)



(a) (b)

Fig.6.7. Relation entre la diversité et le gain de combinaison dans le cas de différents nombres de densités gaussiennes : (a) Mesures de similarité (↓), et (b) Mesures de diversité (↑)



(a) (b)

Fig.6.8. Relation entre la diversité et le gain de combinaison dans le cas de différents nombres d'itérations : (a) Mesures de similarité (↓), et (b) Mesures de diversité (↑)

A partir de la figure 6.5 où les classifieurs se diffèrent dans leurs nombres d'états, nous remarquons que toutes les mesures de diversité sauf DF et θ de l'ensemble numéro 1 (Fig.6.5 (a)), s'accordent très bien avec le gain de combinaison et avec les courbes de la Fig.6.5 (b)). Par exemple, le gain de combinaison le plus important est atteint dans l'ensemble numéro 1 et la plus petite valeur de chaque mesure de similarité (Fig.6.5 (a)) et la plus grande valeur de chaque mesure de diversité (Fig.6.5 (b)) sont atteintes dans le même ensemble. De même, le plus faible gain de combinaison est obtenu dans l'ensemble le moins diversifié (numéro 4) au sens de toutes les mesures de diversité.

Dans le cas d'utilisation de différents modèles initiaux pour créer la diversité entre les classifieurs de base (Fig.6.6), le gain de combinaison est totalement corrélé à toutes les mesures de diversité présentées dans les deux figures 6.6 (a) et 6.6 (b). Le gain de combinaison le plus important est obtenu dans l'ensemble le plus diversifié (numéro 4), et le gain le moins important correspond à l'ensemble le moins diversifié (numéro 2).

Dans le cas de l'utilisation d'ensembles de classifieurs avec différents nombres de densités gaussiennes, la figure 6.7 montre l'existence d'une certaine corrélation entre les mesures de diversité, mais il n'y a pas de relation pertinente entre le gain de combinaison et les différentes mesures de diversité.

A partir de la figure 6.8 (le cas d'utilisation d'ensembles de classifieurs créés avec différents nombres d'itérations de l'algorithme d'apprentissage), nous remarquons qu'à l'exception du cas de l'ensemble 5, et le cas de $DF\ et\ \theta$ dans l'ensemble 4, le gain de combinaison s'améliore lorsque la diversité augmente pour toutes les mesures de diversité. De plus, il y a un accord total entre $D, Ent\ et\ kw$, et il y a également une relation pertinente entre ces trois mesures et le gain de combinaison.

De cette expérimentation, on peut conclure que, dans la quasi-totalité des cas, il y a une corrélation directe entre la diversité et le gain de combinaison. Cependant, l'utilisation unique des mesures de diversité pour évaluer la qualité d'un ensemble peut ne pas être utile, car cela permet de favoriser les ensembles les plus diversifiés qui génèrent un gain de combinaison important en négligeant les performances individuelles des classifieurs. Ceci pourrait conduire, si les classifieurs individuels sont faibles, à des performances globales faibles, malgré un gain de combinaison important.

Prenons l'exemple de 4 ensembles : E1 composé de 2 classifieurs faibles (taux de 70% et 75%) mais très diversifiés, l'ensemble E2 composé de 2 classifieurs forts (taux de 80%, et 85%) mais peu diversifiés, l'ensemble E3 composé de 2 classifieurs faibles (taux de 70% et 75%) et peu diversifiés, et l'ensemble E4 composé de 2 classifieurs forts (taux de 80%, et 85%) et très diversifiés. En se basant sur la corrélation directe entre la diversité de l'ensemble est le gain de combinaison, nous supposons que le gain de combinaison de E1 est 7%, celui de E2 est 2%, celui de E3 est 2% et celui de E4 est 6%. Les performances globales qui peuvent être calculées en sommant le taux de reconnaissance du meilleur classifieur individuel et le gain de combinaison sont résumées dans le tableau 6.11.

Tableau 6.11. Exemple montrant la relation de complémentarité entre la diversité de l'ensemble et les performances individuelles

Ensemble	Diversité	Meilleur taux individuel	Gain de combinaison	Taux de l'ensemble
E1	Très forte	75% (faible)	7%	75+7=82%
E2	Faible	85% (fort)	2%	85+2=87%
E3	Faible	75% (faible)	2%	75+2=77%
E4	Forte	85% (fort)	6%	85+6=91%

Nous remarquons que E4 donne des résultats meilleurs que E1 malgré que ce dernier soit plus diversifié et présente un gain de combinaison plus important. Cela est justifié par le fait que les classifieurs de E1 sont beaucoup plus faibles. Pour les deux ensembles E2 et E3 qui ont le même degré de diversité, on remarque que E2 est plus performant car ses classifieurs individuels sont plus performants. En comparant les ensembles E2 et E4 qui ont les mêmes performances individuelles, nous constatons que E4 est plus performant, car ses classifieurs individuels sont plus diversifiés et, donc, présentent un gain de combinaison plus important.

A travers cet exemple, nous pouvons déduire qu'il y a une relation de complémentarité entre la diversité de l'ensemble et les performances individuels. Pour augmenter les performances de l'ensemble, nous devons donc sélectionner, dans la mesure du possible, les classifieurs de base qui sont à la fois les plus performants (pour obtenir un taux de

reconnaissance moyen élevé) et les plus diversifiés (pour obtenir un gain de combinaison élevé).

6.3.7. Comparaison des résultats

Dans ce qui suit, les résultats de l'approche ensembliste seront comparés, d'une part, avec l'approche hybride proposée et, d'autre part, avec des travaux précédents sur la même base de données.

6.3.7.1. Comparaison avec l'approche hybride HMM/k-NN

Le tableau 6.12 présente une comparaison des deux approches proposées en termes de performances et de robustesse, où les résultats obtenus sont comparables. Les meilleurs résultats sont enregistrés dans le cas de différents modèles initiaux avec un taux de reconnaissance, un écart-type et un coefficient de variation, respectivement, de 97.18%, 4.4859 et 4.62 pour l'approche hybride et de 97%, 5.1824 et 5.34 pour l'approche ensembliste.

Tableau 6.12. Comparaison de performance et de robustesse des deux approches proposées

		Différents nombres d'états	Différents modèles initiaux	Différents nombres de gaussiens	Différents nombres d'itérations
L'approche ensembliste	TR Moyen (%)	96.54	97	94.45	96.31
	Ecart-type	7.8177	5.1824	8.6008	6.6288
	Coefficient de Variation CV	8.10	5.34	9.11	6.88
L'approche hybride	TR Moyen (%)	96.86	**97.18**	94.36	96.27
	Ecart-type	7.8334	**4.4859**	8.8670	6.1812
	Coefficient de variation CV	8.09	**4.62**	9.40	6.42

6.3.7.2. Comparaison avec les travaux de la littérature

Pour bien apprécier notre approche ensembliste, elle est mise en comparaison, dans le tableau 6.13, avec les travaux récents sur la base de données SAD.

Tableau 6.13. Comparaison des performances de l'approche ensembliste avec la littérature

Référence	Méthode de classification	TR (%)
Hu et al., 2011	Réseaux de neurones et ondelettes où l'ondelette de *Morlet* est introduite dans la couche cachée.	96.72
Zhang & Zhang, 2011	Intégration du réseau de neurones et k-means clustering	90.68
Hammami et al., 2012	Modèle d'approximation des distributions d'arbres	93.16
	DHMM (HMM à densité discrète)	90.97
Ettaouil et al., 2013	Un modèle hybridant les réseaux de neurones artificiels et les HMM (NN/HMM)	86
Li et al., 2013	Réseau de neurones probabiliste	92.41
Li et al., 2015	Ensembles de réseaux neuronaux probabilistes	93.95
Shen et al., 2017	Combinaison de plus proche voisin à large marge (LMNN) et la déformation temporelle dynamique (DTW)	92.7
Ramli & Chien, 2017	Règles de décision pondérées dans une représentation collaborative des classifieurs	96.85
	Systèmes à Vaste Marge SVM	94.98
Lawal, 2017	Réseaux abductifs multiples	96.89
Iwana & Uchida, 2017	**Réseaux de neurones convolutifs (CNN) avec alignement dynamique des poids**	**96.95**
	Réseaux de neurones convolutifs (CNN)	**94.77**
	Réseau de neurones récurrent de longue mémoire à court terme (Long Short-Term Memory (LSTM) networks)	**96**
Iwana et al., 2019	Un réseau de neurones utilisant l'alignement dynamique entre les entrées et les poids	96.10
Notre approche ensembliste	Cas de différents nombres d'états	96.54
	Cas de différents modèles initiaux	**97**
	Cas de différents nombres de gaussiennes	94.45
	Cas de différents nombres d'itérations	96.31

Nous remarquons, à partir du tableau 6.13, que le meilleur taux de reconnaissance 97% (souligné dans le tableau) est enregistré pour notre approche dans le cas d'une modélisation multiple à partir de différents modèles initiaux. Pour les autres cas de notre approche, les performances sont comparables avec celles des meilleurs travaux de la littérature.

Il est également interressant de remarquer que, en plus de sa superiorité sur les approches de classification classiques (HMM de base, k-NN, SVM et NN), notre approche donne de meilleurs résultats, comparée aux approches les plus récentes basées sur le Deep learning (marquées en gras dans le tableau), telles que le réseau de neurones convolutif CNN (94.77%), le réseau de neurones convolutif avec alignement dynamique des poids (96.95%), et le réseau de neurones récurrent de longue mémoire à court terme LSTM (96%).

6.4. CONCLUSION

Dans ce chapitre, nous avons présenté une méthode d'ensemble, basée HMM pour la reconnaissance de la parole, qui consiste à mettre en coopération plusieurs classifieurs HMM partant de différentes configurations initiales. Nous avons étudié quatre manières pour faire la différence entre ces configurations : Différents nombres d'états, différents modèles initiaux, différents nombres de densités gaussiennes par état, et différents nombres d'itérations de l'algorithme d'apprentissage. Les résultats comparatifs avec le classifieur de base HMM et d'autres classifieurs utilisés dans des travaux précédents montrent l'efficacité de notre approche en termes de performance et de robustesse, notamment dans le cas où différents modèles initiaux ont été utilisés comme méthode de création de l'ensemble.

Une analyse expérimentale de l'impact de ces paramètres sur la création et la sélection de l'ensemble, en termes de diversité et de performances, a été réalisée afin de déterminer le paramètre ayant le plus grand effet sur la qualité de l'ensemble. Nous avons également étudié l'impact de la taille de l'ensemble sur les performances, ainsi que la relation entre dix mesures de diversité et le gain de combinaison. Cette étude expérimentale nous a permet de tirer les conclusions suivantes :

- La taille de l'ensemble a un léger impact sur les performances. De plus, l'ajout d'un nouveau classifieur à l'ensemble peut avoir un impact inverse sur la qualité de l'ensemble (le cas où différents nombres de densités gaussiennes sont utilisés, par exemple). En fait, il est intuitif d'avancer que combiner moins de classifieurs à la fois performants et diversifiés est préférable, en termes de performance et de complexité, que de combiner un grand nombre de classifieurs faibles.

- Il existe une corrélation forte, d'une part entre les mesures $Q, \rho, DF, k, \theta, GD\ et\ CFD$, et d'autre part, entre $D, kw\ et\ Ent$. Ces trois dernières mesures ne sont pas utiles pour évaluer la qualité de l'ensemble dans les quatre cas de création de diversité. Cependant, un consensus total a été marqué entre toutes les autres mesures $Q, \rho, DF, k, \theta, GD\ et\ CFD$, où le gain de combinaison le

plus important est enregistré dans l'ensemble le plus diversifié. Nous pouvons, dès lors, nous assurer que l'utilisation de l'une de ces dernières mesures, combinée avec les performances des classifieurs individuels, comme critère de sélection, aura un impact important sur la qualité de l'ensemble.

CONCLUSION GENERALE

SOMMAIRE

BILAN

Le travail de recherche présenté dans cette thèse, est axé sur deux contributions principales, qui s'inscrivent dans le cadre général de la RAP. L'objectif était de proposer des approches efficaces permettant de concevoir des SRAP à la fois performants et robustes face à la variabilité de données, aspect fondamental et problématique des données acoustiques.

Après avoir présenté un état de l'art des connaissances dans le domaine de la RAP en général et relatives aux approches de classification en particulier, nous avons pu constater que, depuis leur introduction en RAP aux années 80, les HMM continuent à attirer l'attention des chercheurs, et ils sont devenus largement acceptés comme technique standard de la RAP (Mustafa et al., 2019). Néanmoins, comme la majorité des méthodes de classification, les HMM souffrent du problème de sensibilité à la configuration initiale des paramètres de l'apprentissage. Cette sensibilité rend les SRAP conçus peu robustes à la variabilité de données et à l'inadéquation des données d'apprentissage et des données de test. L'approche markovienne classique consiste à générer, pour chaque classe de données, un HMM unique partant d'une seule configuration initiale. Ce dernier ne peut pas modéliser efficacement toutes les occurrences de la même classe, car ces configurations/modèles sont optimisés sur une base de données limitée, qui ne peut pas prendre en compte toute la variabilité intra-classe, notamment dans les systèmes indépendants du locuteur. Contrairement à cette approche, nos contributions sont basées sur une modélisation markovienne multiple à partir de différentes configurations initiales. Cette modélisation a été mise en œuvre à travers deux approches différentes.

La première approche est une approche hybride intégrant les HMM dans une architecture k-NN. Au lieu de représenter chaque classe par un ensemble de vecteurs de caractéristiques, extraits directement des signaux de la parole, comme le cas de la méthode k-NN, l'approche hybride consiste à générer, pour la même classe, un ensemble de modèles HMM qui se diffèrent par l'un des paramètres de leurs configurations initiales. Durant la phase de reconnaissance, la classe majoritaire dans les k-HMM les plus proches en termes de vraisemblance est sélectionnée. En cas d'ambiguïté, un post-traitement est effectué selon deux méthodes proposées. La première consiste à sélectionner la classe la plus fréquente maximisant les vraisemblances moyennes de ses HMM qui appartiennent à la liste des k voisins les plus proches. La deuxième méthode est basée sur la sélection de la classe minimisant la somme des positions de ses HMM par rapport à l'exemple à reconnaitre. L'intérêt de cette approche est qu'elle permet de faire rejoindre les avantages du classifieur HMM et ceux de la méthode k-NN, et de réduire leurs inconvénients, comme la sensibilité du classifieur HMM à la variabilité de données, et la gourmandise en ressources et en temps de calcul de la méthode k-NN.

La deuxième approche est ensembliste. Elle partage avec l'approche hybride HMM/k-NN une première étape de modélisation multiple. Les modèles HMM générés sont, d'abord, regroupés dans un ensemble initial de classifieurs qui se diffèrent par leurs configurations initiales. Ensuite, une sélection statique du meilleur sous-ensemble est effectuée en se basant sur les performances et la diversité comme critère de sélection. Durant la phase

de reconnaissance, les classifieurs sélectionnés sont mis en coopération, et leurs sorties sont fusionnées selon une règle de combinaison avant de prendre la décision finale.

L'avantage principal des deux approches est que, grâce à la modélisation multiple proposée, elles permettent d'améliorer les performances et la robustesse à la variabilité de données, et de contourner le délicat problème d'optimisation de la configuration initiale de l'algorithme d'apprentissage markovien. Cependant, comme toute méthode ensembliste et hybride, l'inconvénient des approches proposées est qu'elles sont plus couteuses en ressources qu'un classifieur HMM de base. Une manière de remédier à ce problème est bien évidemment le parallélisme. En effet, la répartition de calcul sur plusieurs processeurs serait très avantageuse.

A partir des tests expérimentaux effectués sur la base de données SAD, nous avons pu montrer la supériorité, en termes de performance et de robustesse face à la variabilité de données, de l'approche de modélisation multiple, dans ses deux versions hybride et ensembliste, sur les classifieurs individuels, ainsi que sur des travaux précédents, notamment ceux basés sur l'apprentissage profond (Deep Learning), tels que les réseaux de neurones convolutifs et les réseaux de neurones récurrent de longue mémoire à court terme LSTM. Nous avons également constaté que l'utilisation de la modélisation multiple à partir de différents modèles initiaux est plus robuste et plus performante, comparée aux autres cas de la modélisation multiple, à savoir le cas de différents nombres d'états, le cas de différents nombres de densités gaussiennes, et le cas de différents nombres d'itérations. Aussi, il est intéressant de noter une légère supériorité de l'approche hybride sur l'approche ensembliste dans la majorité des cas de la modélisation multiple.

Outre l'originalité des approches proposées, nous avons réalisé une étude expérimentale approfondie afin de mettre en évidence le rôle de la diversité au sein des classifieurs de base de l'approche ensembliste, ainsi que la relation entre les différentes mesures de diversité couramment utilisées dans la littérature. Cette étude nous a permis de déterminer, parmi ces mesures, celles les plus adaptées à notre approche. Les résultats obtenus, ont montré une corrélation directe entre un sous-ensemble de mesures, à savoir $Q, \rho, DF, k, \theta, GD\ et\ CFD$ qui ont réussi à sélectionner le meilleur ensemble en termes de performance et de gain de combinaison. Quand, au reste des mesures, à savoir $D, kw\ et\ Ent$, notre étude a montré une certaine corrélation entre elles, mais qu'elles ne sont pas utiles pour la sélection des ensembles de classifieurs considérés dans notre approche.

PERSPECTIVES

Ce travail nous a ouvert de nombreuses perspectives de recherche et d'application que nous résumons dans les points suivants :

- Pour l'approche hybride, une perspective est ouverte sur l'amélioration des méthodes de post-traitement proposées, sachant que les erreurs enregistrées pour le système hybride sont des erreurs d'ambiguïté, et que la classe correcte est toujours parmi les classes majoritaires. Une méthode de post-traitement bien appropriée pourrait donc conduire à un taux de reconnaissance de 100%.

- Afin d'améliorer encore les résultats de l'approche ensembliste, il serait intéressant d'étudier d'autres règles de fusion, à savoir le vote pondéré et la somme pondérée des vraisemblances.

- Pour développer un SRAP plus complet, il est important d'introduire une stratégie de rejet permettant de traiter le cas d'exemples qui n'appartiennent pas au vocabulaire considéré. Ceci pourrait être réalisé en fixant un seuil empirique de vraisemblance.

- Les approches proposées sont appliquées sur une base de chiffres arabes isolés. Une perspective majeure de ce travail, est leur adaptation à la parole continue à grand vocabulaire. Pour ce faire, il suffit d'utiliser une approche analytique basée sur les phonèmes contextuels à la place de l'approche globale. Les modèles appris, dans ce cas, seront donc des modèles de phonèmes, à partir desquels, les modèles de mots seront construits. De plus, les modèles de langage doivent être intégrés dans l'architecture du système afin de modéliser la langue. Quant à la segmentation de la parole en unités de base, elle pourrait être réalisée implicitement grâce aux HMM.

- Un travail futur à court terme consiste à étudier la capacité de nos approches de s'adapter à d'autres sources de variabilité, autres que la variabilité interlocuteurs, à savoir le bruit ambiant qui pourrait affecter la stabilité et la robustesse des SRAP.

- Une perspective reste également ouverte, et qui consiste en la recherche d'une méthode de sélection efficace des modèles des classes de l'approche hybride et des ensembles de classifieurs de l'approche ensembliste. Dans ce sens, nous proposons d'utiliser les critères de sélection de modèles tels que BIC (Bayesien Information Creterion), AIC (Akaike Information Criterion) et DIC (Discriminative Information Creterion), et les techniques d'optimisation méta-heuristiques récentes telle que l'algorithme des Loups Gris (Grey Wolf Optimizer GWO) proposé par (Mirjalili et al., 2014).

- En fin, un travail futur, également important, serrait la validation des approches proposées sur d'autres bases de données, et ce, pour pouvoir généraliser nos conclusions à d'autres domaines d'application de la RAP en particulier, et de la reconnaissance de formes en général.

BIBLIOGRAPHIE

Abdel-Hamid, O., Deng, L., & Yu, D. (2013). Exploring convolutional neural network structures and optimization techniques for speech recognition. In Interspeech (Vol. 11, pp. 73-5).

Abdel-Hamid, O., Mohamed, A. R., Jiang, H., & Penn, G. (2012). Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In 2012 IEEE international conference on Acoustics, speech and signal processing (ICASSP) (pp. 4277-4280). IEEE.

Akamine, M., & Ajmera, J. (2012). Decision tree-based acoustic models for speech recognition. EURASIP Journal on Audio, Speech, and Music Processing, 2012(1), 10.

Alalshekmubarak, A., & Smith, L. S. (2013). A novel approach combining recurrent neural network and support vector machines for time series classification. In 9th International Conference on Innovations in Information Technology (IIT), (pp. 42-47). IEEE.

Alessandro, N., Doval, B., Beux, S. L., Woodruff, P., & Fabre, Y. (2006). Ramcess: Realtime and accurate musical control of expression in singing synthesis. In eINTERFACE'06-SIMILAR NoE Summer Workshop on Multimodal Interfaces.

Al-Hajj, R., Mokbel, C., & Likforman-Sulem, L. (2007). Combination of HMM-based classifiers for the recognition of Arabic handwritten words. In Ninth International Conference on Document Analysis and Recognition (Vol. 2, pp. 959-963). IEEE.

AlKhateeb, J. H., Khelifi, F., Jiang, J., & Ipson, S. S. (2009). A new approach for off-line handwritten Arabic word recognition using KNN classifier. In IEEE International Conference on Signal and Image Processing Applications (ICSIPA), (pp. 191-194). Malaysia IEEE.

Al-Maadeed, N., & Al-Maadeed, S. (2018, November). Person-Dependent and Person-Independent Arabic Speech Recognition System. In Recent Trends in Computer Applications: Best Studies from the 2017 International Conference on Computer and Applications, Dubai, UAE (p. 267). Springer.

Al-Qatab, B. A., & Ainon, R. N. (2010). Arabic speech recognition using hidden Markov model toolkit (HTK). In International Symposium in Information Technology (ITSim), (Vol. 2, pp. 557-562). IEEE.

Anthony, G., Gregg, H., & Tshilidzi, M. (2007). Image classification using SVMs: one-against-one vs one-against-all. arXiv preprint arXiv:0711.2914.

Asafuddoula, M., Verma, B., & Zhang, M. (2017). An incremental ensemble classifier learning by means of a rule-based accuracy and diversity comparison. In International Joint Conference on Neural Networks (pp. 1924-1931). IEEE.

Ashraf, J., Iqbal, N., Khattak, N. S., & Zaidi, A. M. (2010, March). Speaker independent Urdu speech recognition using HMM. In 2010 The 7th International Conference on Informatics and Systems (INFOS) (pp. 1-5). IEEE.

Aulia, M. N., Mubarok, M. S., Novia, W. U., & Nhita, F. (2017, May). A comparative study of MFCC-KNN and LPC-KNN for hijaiyyah letters pronounciation classification system. In 2017 5th International Conference on Information and Communication Technology (ICoIC7) (pp. 1-5). IEEE.

Azizi, N., Tlili-Guiassa, Y., & Zemmal, N. (2013). A computer-aided diagnosis system for breast cancer combining features complementarily and new scheme of SVM classifiers fusion. *International Journal of Multimedia and Ubiquitous Engineering*, *8*(4), 45-58.

Bahl, L. R., deSouza, P. V., Gopalakrishnan, P. S., Nahamoo, D., & Picheny, M. A. (1991, April). Decision trees for phonological rules in continuous speech. In [Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing (pp. 185-188). IEEE.

Baker, J. (1975). The DRAGON system--An overview. IEEE Transactions on Acoustics, Speech, and Signal Processing, 23(1), 24-29.

Barrault, L. (2008). Diagnostic pour la combinaison de systèmes de reconnaissance automatique de la parole, thèse de doctorat en sciences de l'Université d'Avignon et des Pays de Vaucluse).

Batista, L., Granger, E., & Sabourin, R. (2012). Dynamic selection of generative–discriminative ensembles for off-line signature verification. *Pattern Recognition*, *45*(4), 1326-1340.

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine learning, 36(1-2), 105-139.

Baum, L. (1972). An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. Inequalities, 3, 1-8.

Baum, L. E., & Eagon, J. A. (1967). An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. Bulletin of the American Mathematical Society, 73(3), 360-363.

Baum, L. E., & Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. The annals of mathematical statistics, 37(6), 1554-1563],

Belaïd, A., & Anigbogu, J. C. (1994). Mise à contribution de plusieurs classifieurs pour la reconnaissance de textes multifontes. Traitement du signal, 11(1), 57-76.

Bellman, R. E. (1957). Dynamic Programming. Princeton Univ. Press, Princeton, NJ.

Besbes, S., & Lachiri, Z. (2016, March). Multi-class SVM for stressed speech recognition. In 2016 2nd international conference on advanced technologies for signal and image processing (ATSIP) (pp. 782-787). IEEE.

Bharali, S. S., & Kalita, S. K. (2015). A comparative study of different features for isolated spoken word recognition using HMM with reference to Assamese language. International Journal of Speech Technology, 18(4), 673-684.

Bhuriyakorn, P., Punyabukkana, P., & Suchato, A. (2008, August). A genetic algorithm-aided hidden markov model topology estimation for phoneme recognition of thai continuous speech. In 2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (pp. 475-480). IEEE.

Biem, A. (2003). A model selection criterion for classification: Application to hmm topology optimization. In Seventh International Conference on Document Analysis and Recognition. (pp. 104-108). IEEE.

Bobulski, J. (2016). 2DHMM-based face recognition method. In Image Processing and Communications Challenges 7 (pp. 11-18). Springer, Cham.

Bougamouza, F., Hazmoune, S. and Benmohammed, M. (2018). Normalisation of handwriting speed for online Arabic characters recognition. Int. J. Computational Vision and Robotics, Vol. 8, No. 6, pp.591–605.

Bougamouza, F., Hazmoune, S., & Benmohammed, M. (2016). Using Mel Frequency Cepstral Coefficient method for online Arabic characters handwriting recognition. In 2016 5th

International Conference on Multimedia Computing and Systems (ICMCS), (pp. 87-92). IEEE.

Box, G. E. (1979). Robustness in the strategy of scientific model building. In Robustness in statistics (pp. 201-236). Academic Press.

Britto Jr, A. S., Sabourin, R., & Oliveira, L. E. (2014). Dynamic selection of classifiers—a comprehensive review. *Pattern Recognition*, 47(11), 3665-3680.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, *2*(2), 121-167.

Ca, M. V., & Radhab, V. (2012). Speaker Independent Isolated Speech Recognition System for Tamil Language using HMM. Procedia Engineering, 30, 1097-1102.

Cavalin, P. R., Sabourin, R., & Suen, C. Y. (2012). LoGID: An adaptive framework combining local and global incremental learning for dynamic selection of ensembles of HMM. *Pattern Recognition*, 45(9), 3544-3556.

Chebotar, Y., & Waters, A. (2016). Distilling Knowledge from Ensembles of Neural Networks for Speech Recognition. In Interspeech (pp. 3439-3443).

Cho, S. B., & Kim, J. H. (1995). Combining multiple neural networks by fuzzy integral for robust classification. *IEEE Transactions on Systems, Man, and Cybernetics*, *25*(2), 380-384.

Chow, C. K. (1965). Statistical independence and threshold functions. IEEE Transactions on Electronic Computers, (1), 66-68.

Christophe Saint-Jean (2001). Classification paramétrique robuste partiellement supervisée en reconnaissance des formes. Modélisation et simulation. Thèse de doctorat de l'université de La Rochelle, Français. tel-00145895

Clairet, S. (2004). Compensation articulatoire dans la production des occlusives du français, Thèse de doctorat. Université Aix Marseille I-Université de Provence, Aix-en-Provence.

Clarkson, P., & Moreno, P. J. (1999). On the use of support vector machines for phonetic classification. In IEEE International Conference on Acoustics, Speech, and Signal Processing. (Vol. 2, pp. 585-588). IEEE.

Clemente, I. A., Heckmann, M., & Wrede, B. (2012). Incremental word learning: Efficient hmm initialization and large margin discriminative adaptation. Speech Communication, 54(9), 1029-1048.

Cohen, I., Sebe, N., Garg, A., Chen, L. S., & Huang, T. S. (2003). Facial expression recognition from video sequences: temporal and static modeling. *Computer Vision and image understanding*, 91(1), 160-187.

Cruz, R. M., Sabourin, R., & Cavalcanti, G. D. (2018). Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41, 195-216.

Cunningham, P., & Carney, J. (2000). Diversity versus quality in classification ensembles based on feature selection. In *European Conference on Machine Learning* (pp. 109-116). Springer, Berlin, Heidelberg.

Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE transactions on acoustics, speech, and signal processing, 28(4), 357-366.

Debnath, S., & Roy, P. (2019). Isolated Word Recognition Based on Different Statistical Analysis and Feature Selection Technique. In Cognitive Informatics and Soft Computing (pp. 463-473). Springer, Singapore.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1), 1-2.

Deselaers, T., Heigold, G., & Ney, H. (2007). Speech recognition with state-based nearest neighbour classifiers. In Interspeech-2007, 2093-2096.

Dhanashri, D., & Dhonde, S. B. (2017). Isolated Word Speech Recognition System Using Deep Neural Networks. In Proceedings of the International Conference on Data Engineering and Communication Technology (pp. 9-17). Springer Singapore.

Dhingra, S. D., Nijhawan, G., & Pandit, P. (2013). Isolated speech recognition using MFCC and DTW. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2(8), 4085-4092.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, *40*(2), 139-157.

Ding, J., & Chang, C. W. (2016). An adaptive hidden Markov model-based gesture recognition approach using Kinect to simplify large-scale video data processing for humanoid robot imitation. *Multimedia Tools and Applications*, 75(23), 15537-15551.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern classification. 2nd edition. John Wiley & Sons, New York.

Durbin, J. (1960). "The fitting of time series models." Rev. Inst. Int. Stat., v. 28, pp. 233–243.

Ekbal, A., & Saha, S. (2010). Classifier ensemble selection using genetic algorithm for named entity recognition. *Research on Language and Computation*, 8(1), 73-99.

Ellis, D. P. (2000, April). Stream combination before and/or after the acoustic model. In Proc. Int. Conf. on Acoustics, Speech and Signal Processing (Vol. 3, pp. 1635-1638).

En-Naimani, Z., Lazaar, M., & Ettaouil, M. (2014). Hybrid system of optimal self organizing maps and hidden Markov model for Arabic digits recognition. *WSEAS Transactions on Systems*, 13(60), 606-616.

Ettaouil, M., Lazaar, M., & En-Naimani, Z. (2013, May). A hybrid ANN/HMM models for arabic speech recognition using optimal codebook. In 2013 8th International Conference on Intelligent Systems: Theories and Applications (SITA) (pp. 1-5). IEEE.

Fabrice LAURI, (2004). Adaptation au locuteur de modèles acoustiques markoviens pour la reconnaissance automatique de la parole, thèse de doctorat, université Nancy 2

Feng, W., Guan, N., Li, Y., Zhang, X., & Luo, Z. (2017, May). Audio visual speech recognition with multimodal recurrent neural networks. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 681-688). IEEE.

Fern, X. Z., & Lin, W. (2008). Cluster ensemble selection. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *1*(3), 128-141.

Ferrer, M. A., Alonso, I. G., & Travieso, C. M. (2000). Influence of initialisation and stop criteria on HMM based recognisers. Electronics Letters, 36(13), 1165-1166.

Fix, E., & Hodges, J.L. (1951). Discriminatory analysis, nonparametric discrimination: Consistency properties, Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas.

Fleiss, J. L., Levin, B., & Paik, M. C. (2013). *Statistical methods for rates and proportions*. John Wiley & Sons.

Forney, G. D. (1973). The viterbi algorithm. Proceedings of the IEEE, 61(3), 268-278.

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and computation*, *121*(2), 256-285.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In icml (Vol. 96, pp. 148-156).

Ganapathiraju, A., Hamaker, J. E., & Picone, J. (2004). Applications of support vector machines to speech recognition. *IEEE Transactions on Signal Processing*, 52(8), 2348-2355.

Ganapathiraju, A., Hamaker, J., & Picone, J. (2000). Hybrid SVM/HMM architectures for speech recognition. In Sixth International Conference on Spoken Language Processing. (vol.4, pp. 504-507).

Ge, Y., Zhang, X., Chen, Q., & Jiang, M. (2016). Initialization of the HMM-based delay model in networked control systems. Information Sciences, 364, 1-15.

Geiger, J., Schenk, J., Wallhoff, F., & Rigoll, G. (2010). Optimizing the number of states for HMM-based on-line handwritten whiteboard recognition. In International Conference on Frontiers in Handwriting Recognition (ICFHR), (pp. 107-112). IEEE.

Giacinto, G., & Roli, F. (2001). Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, *19*(9-10), 699-707.

Gilpin, S. A., & Dunlavy, D. M. (2009, September). Relationships between accuracy and diversity in heterogeneous ensemble classifiers. In SAND2009, 6940C. Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Giuliani, D., Gerosa, M., & Brugnara, F. (2006). Improved automatic speech recognition through speaker normalization. Computer Speech & Language, 20(1), 107-123.

Guerid, A., & Houacine, A. (2019). Recognition of isolated digits using DNN–HMM and harmonic noise model. IET Signal Processing, 13(2), 207-214.

Guerid, A., Saboune, H., & Houacine, A. (2018, April). Recognition of isolated digits using HMM and harmonic noise model. In 2018 1st International Conference on Computer Applications & Information Security (ICCAIS) (pp. 1-5). IEEE.

Gunter, S., & Bunke, H. (2002). Creation of classifier ensembles for handwritten word recognition using feature selection algorithms. In *Eighth International Workshop on Frontiers in Handwriting Recognition, 2002. Proceedings* (pp. 183-188). IEEE.

Gunter, S., & Bunke, H. (2003). Optimizing the number of states, training iterations and Gaussians in an HMM-based handwritten word recognizer. In Seventh International Conference on Document Analysis and Recognition (pp. 472-476). IEEE.

Hai, N. T., Van Thuyen, N., Mai, T. T., & Van Toi, V. (2015). MFCC-DTW Algorithm for Speech Recognition in an Intelligent Wheelchair. In 5th International Conference on Biomedical Engineering in Vietnam (pp. 417-421). Springer, Cham.

Hamdi, A., & Frigui, H. (2015). Ensemble hidden Markov models with application to landmine detection. *EURASIP Journal on Advances in Signal Processing*, 2015 (1), 75.

Hammami, N., & Bedda, M. (2010). Improved tree model for Arabic speech recognition. In 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), (Vol. 5, pp. 521-526).IEEE.

Hammami, N., Bedda, M., & Farah, N. (2011). HMM parameters estimation based on cross-validation for Spoken Arabic Digits recognition. In International Conference on Communications, Computing and Control Applications (CCCA), (pp. 1-4). IEEE.

Hammami, N., Bedda, M., & Farah, N. (2012). Tree distributions approximation model for robust discrete speech recognition. International Journal of Speech Technology, 15(4), 455-462.

Hammami, N., Bedda, M., & Nadir, F. (2012). The second-order derivatives of MFCC for improving spoken arabic digits recognition using tree distributions approximation model and HMM. In International Conference on Communications and Information Technology (ICCIT), (pp. 1-5). IEEE.

Hammami, N., Bedda, M., Farah, N., & Lakehal-Ayat, R. O. (2013). Spoken Arabic Digits recognition based on (GMM) for e-Quran voice browsing: Application for blind category. In Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences (32519), (pp. 123-127). IEEE.

Han, Y., & Boves, L. (2006). EM Algorithm with Split and Merge in Trajectory Clustering for Automatic Speech Recognition. Department of Language and Speech, Radboud University Nijmegen.

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, *12*(10), 993-1001.

Hasan, M., & Boris, F. (2006). SVM: Machines à vecteurs de support ou séparateurs à vastes marges. Rapport technique, Versailles St Quentin, France. Cité, 64.

Hawarah, L. (2008). Une approche probabiliste pour le classement d'objets incomplètement connus dans un arbre de décision, thèse de doctorat, Université Joseph-Fourier - Grenoble I, 2008. Français. tel-00335313v2.

Hazmoune, S., Bougamouza, F., Mazouzi, S., & Benmohammed, M. (2018). A new hybrid framework based on hidden Markov models and K-nearest neighbors for speech recognition. International Journal of Speech Technology, 21(3), 689-704. (Springer), DOI: 10.1007/s10772-018-9535-4

Hazmoune, S., Bougamouza, F., Mazouzi, S., & Benmohammed, M. (2013a). A novel speech recognition approach based on multiple modeling by hidden Markov models. In International Conference on Computer Applications Technology (ICCAT), 2013 (pp. 1-IEEE, DOI: 10.1109/ICCAT.2013.6522028

Hazmoune, S., Bougamouza, F., Mazouzi, S., & Benmohammed, M. (2013b). Contributions to HMM-based Speech Recognition Systems, International Journal of Computational Linguistics Research (IJCLR), ISSN 0976-4178, Vol.4 No.1, (pp.38-47).

Hemakumar, G., Punithavalli, M., & Thippeswamy, K. (2016). Large Vocabulary in Continuous Speech Recognition Using HMM and Normal Fit. International Journal of Computer Trends and technology (IJCTT), 42, 2231-2803.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. The Journal of the Acoustical Society of America, 87(4), 1738-1752.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, *20*(8), 832-844.

Hu, X., Zhan, L., Xue, Y., Zhou, W., & Zhang, L. (2011, October). Spoken arabic digits recognition based on wavelet neural networks. In 2011 IEEE International Conference on Systems, Man, and Cybernetics (pp. 1481-1485). IEEE.

Huo, Q., & Lee, C. H. (2000). A Bayesian predictive classification approach to robust speech recognition. IEEE transactions on speech and audio processing, 8(2), 200-204.

Imtiaz, M. A., & Raja, G. (2016, November). Isolated Word Automatic Speech Recognition (ASR) System using MFCC, DTW & KNN. In 2016 Asia Pacific Conference on Multimedia and Broadcasting (APMediaCast) (pp. 106-110). IEEE.

Itaya, Y., Zen, H., Nankaku, Y., Miyajima, C., Tokuda, K., & Kitamura, T. (2005). Deterministic annealing EM algorithm in acoustic modeling for speaker and speech recognition. IEICE transactions on information and systems, 88(3), 425-431

Iwana, B. K., & Uchida, S. (2017). Dynamic Weight Alignment for Convolutional Neural Networks. CoRR.

Iwana, B. K., Frinken, V., & Uchida, S. (2019). DTW-NN: A novel neural network for time series recognition using dynamic alignment between inputs and weights. Knowledge-Based Systems, 104971.

Jacob, B. (1995). Un outil informatique de gestion de Modèles de Markov Cachés : expérimentations en Reconnaissance Automatique de la Parole. Thèse de doctorat, Université de Paul Sabatier de Toulouse III.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. IEEE Transactions on pattern analysis and machine intelligence, 22(1), 4-37.

Jelinek, F. (1969). Fast sequential decoding algorithm using a stack. IBM journal of research and development, 13(6), 675-685.

Jiang, Z., Ding, X., Peng, L., & Liu, C. (2012). Analyzing the information entropy of states to optimize the number of states in an HMM-based off-line handwritten Arabic word recognizer. In 21st International Conference on Pattern Recognition (ICPR), (pp. 697-700). IEEE.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. Machine learning, 37(2), 183-233.

Juang, B. H., Rabiner, L., & Wilpon, J. G. (1986). On the use of bandpass liftering in speech recognition. In ICASSP'86. IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 11, pp. 765-768). IEEE.

Juang, BH, Rabiner, LR, (1990). The segmental K- means Algorithm for Estimating Parameters of Hidden Markov Models. Acoustics, Speech and Signal Processing, IEEE Transactions, 38(9), 1639 – 1641.

Junqua, J. C., & Haton, J. P. (2012). Robustness in automatic speech recognition: fundamentals and applications (Vol. 341). Springer Science & Business Media.

Kacur, J., & Urbancikova, L. (2018, June). Recognition of Isolated Words Using Feedforward Neural Networks. In 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP) (pp. 1-4). IEEE.

Kamper, H., Jansen, A., & Goldwater, S. (2015). Fully unsupervised small-vocabulary speech recognition using a segmental bayesian model. In Sixteenth Annual Conference of the International Speech Communication Association.

Kamper, H., Jansen, A., & Goldwater, S. (2017). A segmental framework for fully-unsupervised large-vocabulary speech recognition. Computer Speech & Language, 46, 154-174.

Kanisha, B., Lokesh, S., Kumar, P. M., Parthasarathy, P., & Chandra Babu, G. (2018). Speech recognition with improved support vector machine using dual classifiers and cross fitness validation. Personal and ubiquitous computing, 22(5-6), 1083-1091.

Karthikeyan, K. S., Priya, K. J., & Gupta, D. (2019). Analysis of Digit Recognition in Kannada Using Kaldi Toolkit. In Emerging Research in Electronics, Computer Science and Technology (pp. 813-821). Springer, Singapore.

Këpuska, V. Z., & Elharati, H. A. (2015). Robust speech recognition system using conventional and hybrid features of MFCC, LPCC, PLP, RASTA-PLP and hidden markov model classifier in noisy conditions. Journal of Computer and Communications, 3(06), 1.

Khalid Hallouli, (2004) Reconnaissance de caractères par méthodes markoviennes et réseaux bayésiens, thèse de doctorat, Télé-com ParisTech, HAL Id: pastel-00000740, https://pastel.archives-ouvertes.fr/pastel-00000740

Khelifa, M. O., Elhadj, Y. M., Abdellah, Y., & Belkasmi, M. (2017). Constructing accurate and robust HMM/GMM models for an Arabic speech recognition system. International Journal of Speech Technology, 20(4), 937-949.

Kim, J. H., Kim, K. K., & Suen, C. Y. (2000). Hybrid schemes of homogeneous and heterogeneous classifiers for cursive word recognition. In Proc 7th International Workshop on Frontiers in Handwriting Recognition, Amsterdam. Netherlands (pp. 433-442).

Kim, Y. W., & Oh, I. S. (2008). Classifier ensemble selection using hybrid genetic algorithms. *Pattern Recognition Letters*, 29(6), 796-802.

Ko, A. H., Sabourin, R., & Britto Jr, A. S. (2008). From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, 41(5), 1718-1731.

Koerich, A. L., & Poitevin, C. (2005). Combination of homogeneous classifiers for musical genre classification. In IEEE International Conference on Systems, Man and Cybernetics (Vol. 1, pp. 554-559).

Kohavi, R., & Wolpert, D. H. (1996). Bias plus variance decomposition for zero-one loss functions. In *ICML* (Vol. 96, pp. 275-83).

Kumar, A., Dua, M., & Choudhary, T. (2014). Continuous hindi speech recognition using monophone based acoustic modeling. International Journal of Computer Applications, 24.

Kumar, K., & Aggarwal, R. K. (2011). Hindi speech recognition system using HTK. International Journal of Computing and Business Research, 2(2), 2229-6166.

Kuncheva, L. I. (2014). Combining pattern classifiers: methods and algorithms. John Wiley & Sons.

Kuncheva, L. I., & Whitaker, C. J. (2001). Ten measures of diversity in classifier ensembles: limits for two classifiers. In *A DERA/IEE Workshop on Intelligent Sensor Processing (Ref. No. 2001/050),* (pp. 10-1). IET.

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, *51*(2), 181-207.

Kurata, D., Nankaku, Y., Tokuda, K., Kitamura, T., & Ghahramani, Z. (2006). Face recognition based on separable lattice HMM. In 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings (Vol. 5, pp. V-V). IEEE.

Kwong, S., Chau, C. W., Man, K. F., & Tang, K. S. (2001). Optimisation of HMM topology and its model parameters by genetic algorithms. Pattern recognition, 34(2), 509-522.

Kwong, S., Chau, C. W., Man, K. F., & Tang, K. S. (2001). Optimisation of HMM topology and its model parameters by genetic algorithms. *Pattern recognition*, 34(2), 509-522.

L. Breiman (1996), Bagging predictors, *Machine Learning* , 2, 123–140.

Lawal, I. A. (2017). Spoken character classification using abductive network. International Journal of Speech Technology, 20(4), 881-890.

Lazli, L. (2007). Système Neuro-Markovien basé sur la fusion de données floues et génétiques : Application pour la Reconnaissance automatique de la parole. Thèse de doctorat en sciences, université Badji Mokhtar – Annaba, Algérie.

Lee, H. K., & Kim, J. H. (1999). An HMM-based threshold model approach for gesture recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 21(10), 961-973.

Lee, K. F., Hon, H. W., & Reddy, R. (1990). An overview of the SPHINX speech recognition system. IEEE Transactions on Acoustics, Speech, and Signal Processing, 38(1), 35-45.

Levinson, N. (1947). "The Wiener RMS error criterion in filter design and prediction." J. Math. Phys., v. 25, pp. 261–278.

Levinson, S. E. (1985). Structural methods in automatic speech recognition. Proceedings of the IEEE, 73(11), 1625-1650.

Levinson, S. E., Rabiner, L. R., & Sondhi, M. M. (1983). An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. Bell System Technical Journal, 62(4), 1035-1074.

Li, X. G., Yao, M. F., Jian, L. R., & Li, Z. J. (2013). The Application of Probabilistic Neural Network in Speech Recognition Based on Partition Clustering. In Applied Mechanics and Materials (Vol. 263, pp. 2173-2178). Trans Tech Publications.

Li, X., Zhang, S., Li, S., & Chen, J. (2015, August). An improved method of speech recognition based on probabilistic neural network ensembles. In 2015 11th International Conference on Natural Computation (ICNC) (pp. 650-654). IEEE.

Lichman, M. (2013). UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, URL http://archive.ics.uci.edu/ml.

Lior, R. (2014). *Data mining with decision trees: theory and applications* (Vol. 81). World scientific.

Liu, N., Davis, R. I., Lovell, B. C., & Kootsookos, P. J. (2004, April). Effect of initial HMM choices in multiple sequence training for gesture recognition. In International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. (Vol. 1, pp. 608-613). IEEE.

Looney, C. G. (1997). *Pattern recognition using neural networks: theory and algorithms for engineers and scientists*. Oxford University Press, Inc..

Luana Batista n, Eric Granger, Robert Sabourin(2012), Dynamic selection of generative–discriminative ensembles for off-line signature verification, *Pattern Recognition*, 45, 1326–1340.

Luo, X. (2011). Chinese speech recognition based on a hybrid SVM and HMM architecture. In International Symposium on Neural Networks (pp. 629-635). Springer, Berlin, Heidelberg.

Ma, C., Randolph, M. A., & Drish, J. (2001). A support vector machines-based rejection technique for speech recognition. In IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (ICASSP'01). (Vol. 1, pp. 381-384). IEEE.

Manikandan, J., & Venkataramani, B. (2011). Design of a real time automatic speech recognition system using Modified One Against All SVM classifier. Microprocessors and Microsystems, 35(6), 568-578.

Mansour, A. H., Salh, G. Z. A., & Mohammed, K. A. (2015). Voice recognition using dynamic time warping and mel-frequency cepstral coefficients algorithms. International Journal of Computer Applications, 116(2).

Mao, S., Jiao, L. C., Xiong, L., & Gou, S. (2011). Greedy optimization classifiers ensemble based on diversity. *Pattern Recognition*, 44(6), 1245-1261.

Marković, B. G., Stevanović, G., Jovičić, S. T., Mijić, M., & Galić, J. (2017). Recognition of Normal and Whispered Speech Based on RASTA Filtering and DTW Algorithm. In Proceedings of the Int. Conf. IcETRAN-2017 (pp. 8-2).

Masmoudi, S., Frikha, M., Chtourou, M., & Hamida, A. B. (2011). Efficient MLP constructive training algorithm using a neuron recruiting approach for isolated word recognition system. *International Journal of Speech Technology*, 14(1), 1-10.

Matsui, T., Kanno, T., & Furui, S. (1996). Speaker recognition using HMM composition in noisy environments. *Computer Speech & Language*, 10(2), 107-116.

Miao, Y., Gowayyed, M., & Metze, F. (2015, December). EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU) (pp. 167-174). IEEE.

Minker, W., & Néel, F. (2002). Développement des technologies vocales. *Le travail humain*, 65(3), 261-287.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software,* 69, 46-61.

Misra, H., Bourlard, H., & Tyagi, V. (2003, April). New entropy based combination rules in HMM/ANN multi-stream ASR. In 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). (Vol. 2, pp. II-741). IEEE.

Moghaddam, Z., & Piccardi, M. (2013). Training initialization of hidden Markov models in human action recognition. IEEE Transactions on Automation Science and Engineering, 11(2), 394-408.

Mohamad, R. A. H., Likforman-Sulem, L., & Mokbel, C. (2009). Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(7), 1165-1177.

Moore, G. E. (1965). Cramming more components onto integrated circuits.

Muda, L., Begam, M., & Elamvazuthi, I. (2010). Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *Journal of Computing*, 2(3), 138-143.

Mustafa, M. K., Allen, T., & Appiah, K. (2019). A comparative review of dynamic neural networks and hidden Markov model methods for mobile on-device speech recognition. Neural Computing and Applications, 31(2), 891-899.

Nathan, K., Senior, A., Subrahmonia, J., 1996. Initialization of hidden markov models for unconstrained on-line handwriting recognition. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3502–3505.

Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H. G., & Ogata, T. (2015). Audio-visual speech recognition using deep learning. Applied Intelligence, 42(4), 722-737.

Norris, D., & McQueen, J. M. (2008). Shortlist B: a Bayesian model of continuous speech recognition. Psychological review, 115(2), 357.

Orhan, U., Hekim, M., & Ozer, M. (2011). EEG signals classification using the K-means clustering and a multilayer perceptron neural network model. Expert Systems with Applications, 38(10), 13475-13481.

Palaz, D., Doss, M. M., & Collobert, R. (2015, April). Convolutional neural networks-based continuous speech recognition using raw speech signal. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4295-4299). IEEE.

Paramonov, P., & Sutula, N. (2016). Simplified scoring methods for HMM-based speech recognition. Soft Computing, 20(9), 3455-3460.

Partalas, I., Tsoumakas, G., & Vlahavas, I. P. (2008). Focused Ensemble Selection: A Diversity-Based Method for Greedy Ensemble Selection. In ECAI (pp. 117-121).

Partridge, D., & Krzanowski, W. (1997). Software diversity: practical statistics for its measurement and exploitation. *Information and software technology*, *39*(10), 707-717.

Pépiot, E. (2013). Voix de femmes, voix d'hommes : différences acoustiques, identification du genre par la voix et implications psycholinguistiques chez les locuteurs anglophones et francophones. Thèse de doctorat, Université Paris VIII Vincennes-Saint Denis, Français. tel-00821462.

Pérez-Cruz, F., & Bousquet, O. (2004). Kernel methods and their potential use in signal processing. IEEE Signal Processing Magazine, 21(3), 57-65.

Quinlan, J. R. (1987, January). Decision trees as probabilistic classifiers. In Proceedings of the Fourth International Workshop on Machine Learning (pp. 31-37). Morgan Kaufmann.

Rabi, M., Amrouch, M., & Mahani, Z. (2018). Cursive Arabic Handwriting Recognition System Without Explicit Segmentation Based on Hidden Markov Models.

Rabiner, L. R. (1988). Mathematical foundations of hidden Markov models. In Recent advances in speech understanding and dialog systems (pp. 183-205). Springer, Berlin, Heidelberg.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257-286.

Rabiner, L. R., & Juang, B. H. (1992). Hidden Markov models for speech recognition—strengths and limitations. *In Speech Recognition and Understanding* (pp. 3-29). Springer, Berlin, Heidelberg.

Rabiner, L. R., Levinson, S. E., & Sondhi, M. M. (1984). On the Use of Hidden Markov Models for Speaker-Independent Recognition of Isolated Words From a Medium-Size Vocabulary. AT&T Bell Laboratories Technical Journal, 63(4), 627-642.

Rabiner, L. R., Wilpon, J. G., and Juang, B. H. (1986). A Segmental k-Means Training Procedure for Connected Word Recognition. AT&T Technical Journal, 65(3), 21-31.

Rabiner, L.R., & Juang, B. (1986). An introduction to hidden Markov models. *IEEE assp magazine*, 3(1), 4-16.

Ramli, D. A., & Chien, T. W. (2017). Extreme Learning Machine based weighting for decision rule in Collaborative Representation Classifier. Procedia Computer Science, 112, 504-513.

RamÍrez, M., Sotaquirá, M., De La Cruz, A., Maria, E., Avellaneda, G., & Ochoa, A. (2016). An automatic speech recognition system for helping visually impaired children to learn Braille. In XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA), (pp. 1-4). IEEE.

Rao, K. S., Reddy, V. R., & Maity, S. (2015). *Language Identification Using Spectral and Prosodic Features*. Springer.

Ravanelli, M. (2017). Deep learning for Distant Speech Recognition. Thèse de doctorat, Unitn.

Reichl, W., & Chou, W. (2000). Robust decision tree state tying for continuous speech recognition. IEEE Transactions on Speech and Audio Processing, 8(5), 555-566.

Rokach, L., & Maimon, O. (2005). Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *35*(4), 476-487.

Sagar, S., Dixit, S., & Mahesh, B. V. (2020). Offline Cursive Handwritten Word Using Hidden Markov Model Technique. In Smart Intelligent Computing and Applications (pp. 525-535). Springer, Singapore.

Sak, H., Senior, A., Rao, K., & Beaufays, F. (2015). Fast and accurate recurrent neural network acoustic models for speech recognition. arXiv preprint arXiv:1507.06947.

SALEH, Abdul Aziz et WAZIR, Mahfoudh Ba, (2018). Spoken arabic digits recognition using deep learning. Thèse de doctorat. University of Malaya.

Samanta, O., Roy, A., Parui, S. K., & Bhattacharya, U. (2018). An HMM framework based on spherical-linear features for online cursive handwriting recognition. Information Sciences, 441, 133-151.

Samarasinghe, S. (2016). *Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition*. CRC Press.

Sankar, A. (1998, February). Experiments with a Gaussian merging-splitting algorithm for HMM training for speech recognition. In Proceedings of DARPA Speech Recognition Workshop (pp. 99-104).

Schmidt, M., Schels, M., & Schwenker, F. (2010). A hidden markov model based approach for facial expression recognition in image sequences. In IAPR Workshop on Artificial Neural Networks in Pattern Recognition (pp. 149-160). Springer, Berlin, Heidelberg.

Seltzer, M. L., Raj, B., & Stern, R. M. (2004). A Bayesian classifier for spectrographic mask estimation for missing feature speech recognition. Speech Communication, 43(4), 379-393.

Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press, available at http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning

Sharma, P., & Kaur, M. (2013). Classification in pattern recognition: A review. International Journal of Advanced Research in Computer Science and Software Engineering, 3(4).

Shen, J., Huang, W., Zhu, D., & Liang, J. (2017). A novel similarity measure model for multivariate time series based on LMNN and DTW. Neural Processing Letters, 45(3), 925-937.

Shipp, C. A., & Kuncheva, L. I. (2002). Relationships between combination methods and measures of diversity in combining classifiers. *Information fusion*, *3*(2), 135-148.

Skalak, D. B. (1996, August). The sources of increased accuracy for two proposed boosting algorithms. In Proc. American Association for Artificial Intelligence, AAAI-96, *Integrating Multiple Learned Models Workshop* (Vol. 1129, p. 1133).

Sneath, P. H., & Sokal, R. R. (1973). *Numerical taxonomy. The principles and practice of numerical classification*.

Solera-Urena, R., Garcia-Moral, A. I., Peláez-Moreno, C., Martinez-Ramon, M., & Diaz-de-Maria, F. (2011). Real-time robust automatic speech recognition using compact support vector

machines. IEEE Transactions on Audio, Speech, and Language Processing, 20(4), 1347-1361.

Song, I., Chung, J., Kim, T., & Bengio, Y. (2018, April). Dynamic Frame Skipping for Fast Speech Recognition in Recurrent Neural Network Based Acoustic Models. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4984-4988). IEEE.

Spalanzani, A. (1999). Algorithmes évolutionnaires pour l'étude de la robustesse des systèmes de reconnaissance de la parole. Thèse de doctorat, Université Joseph-Fourier-Grenoble I.

Sumon, S. A., Chowdhury, J., Debnath, S., Mohammed, N., & Momen, S. (2018, September). Bangla short speech commands recognition using convolutional neural networks. In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP) (pp. 1-6). IEEE.

Sun, J., Sun, J., Abida, K., & Karray, F. (2012, June). A novel template matching approach to speaker-independent arabic spoken digit recognition. In International Conference on Autonomous and Intelligent Systems (pp. 192-199). Springer, Berlin, Heidelberg.

T. Ho (1998), The random subspace method for constructing decision forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (8), 832–844.

Tebelskis, J. (1995). Speech Recognition using Neural Networks. Thèse de doctorat, université de Carnegie Mellon, Pittsburgh, Pennsylvania.

Thubthong, N., & Kijsirikul, B. (2001). Support vector machines for Thai phoneme recognition. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(06), 803-813.

Tohkura, Y. (1987). A weighted cepstral distance measure for speech recognition. IEEE Transactions on acoustics, speech, and signal processing, 35(10), 1414-1422

Touazi, A., & Debyeche, M. (2017). An experimental framework for Arabic digits speech recognition in noisy environments. International Journal of Speech Technology, 20(2), 205-224.

Vapnik, V. N. (1995). The nature of statistical learning. Theory. N-Y, Springer-Verlag, 1995.

Varma, S., Shinde, M., & Chavan, S. S. (2020). Analysis of PCA and LDA Features for Facial Expression Recognition Using SVM and HMM Classifiers. In Techno-Societal 2018 (pp. 109-119). Springer, Cham.

Vintsyuk, T. K. (1968). Speech discrimination by dynamic progamming. Kibenetika, 4 :81–88.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE transactions on Information Theory, 13(2), 260-269.

Wang, Q., & Ju, S. (2008, October). A mixed classifier based on combination of HMM and KNN. In Fourth International Conference on Natural Computation. ICNC'08. (Vol. 4, pp. 38-42). IEEE.

Wang, X. H., Liu, A., & Zhang, S. Q. (2015). New facial expression recognition based on FSVM and KNN. *Optik-International Journal for Light and Electron Optics*, 126(21), 3132-3134.

WAZIR, A. S. M. B., & CHUAH, J. H. (2019). Spoken Arabic Digits Recognition Using Deep Learning. In 2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS) (pp. 339-344). IEEE.

Weston, J., & Watkins, C. (1999). Support vector machines for multi-class pattern recognition. In *Esann* (Vol. 99, pp. 219-224).

Woźniak, M., Graña, M., & Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. Information Fusion, 16, 3-17.

Wu, C. (2018). Structured Deep Neural Networks for Speech Recognition. Thèse de doctorat, Université de Cambridge

Xu, C. (2014). Model constrution in Speech recognition on time and space sampling point of view. In IEEE 9th Conference on Industrial Electronics and Applications (ICIEA), (pp. 1095-1097). IEEE.

Xu, Y., Siohan, O., Simcha, D., Kumar, S., & Liao, H. (2015). Exemplar-based large vocabulary speech recognition using k-nearest neighbors. In International Conference on Acoustics, Speech and Signal Processing (ICASSP), (pp. 5167-5171). IEEE.

Xue, S., Jiang, H., Dai, L., & Liu, Q. (2016). Speaker adaptation of hybrid NN/HMM model for speech recognition based on singular value decomposition. Journal of Signal Processing Systems, 82(2), 175-185.

Ye, R., & Dai, Q. (2018). A Novel Greedy Randomized Dynamic Ensemble Selection Algorithm. *Neural Processing Letters*, 47(2), 565-599.

Yfantis, E. A., & Elison, J. D. (1970). Vector interpolation for time alignment in speech recognition. WIT Transactions on Modelling and Simulation, 23.

Yfantis, E. A., Lazarakis, T., Angelopoulos, A., Elison, J. D., & Zhang, Y. (1999, March). On time alignment and metric algorithms for speech recognition. In Proceedings 1999 International Conference on Information Intelligence and Systems (Cat. No. PR00446) (pp. 423-428). IEEE.

Yuan DONG (2013), Modélisation probabiliste de classifieurs d'ensemble pour des problèmes à deux classes, thèse de doctorat Université de Technologie de Troyes

Yuan, S., Zhang, J., Chen, J., Qiu, L., & Yang, W. (2019). A uniform initialization Gaussian mixture model–based guided wave–hidden Markov model with stable damage evaluation performance. Structural Health Monitoring, 18(3), 853-868.

Yule, G. U. (1900). VII. On the association of attributes in statistics: with illustrations from the material of the childhood society, &c. *Phil. Trans. R. Soc. Lond. A*, *194*(252-261), 257-319.

Zarrouk, E., & Benayed, Y. (2016). Hybrid SVM/HMM Model for the Arab Phonemes Recognition. International Arab Journal of Information Technology (IAJIT), 13(5).

Zarrouk, E., Ayed, Y. B., & Gargouri, F. (2014). Hybrid continuous speech recognition systems by HMM, MLP and SVM: a comparative study. *International Journal of Speech Technology*, 17(3), 223-233.

Zarrouk, E., Ayed, Y. B., & Gargouri, F. (2018). Arabic Continuous Speech Recognition Based on Hybrid SVM/HMM Model. Communication and Signal Processing: Extended Papers, 8, 145.

Zeinali, H., Sameti, H., & Burget, L. (2017). HMM-based phrase-independent i-vector extractor for text-dependent speaker verification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(7), 1421-1435.

Zelaia, A., Alegria, I., Arregi, O., & Sierra, B. (2011). A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. *Applied Soft Computing*, *11*(8), 4981-4990.

Zelinka, P., Sigmund, M., & Schimmel, J. (2012). Impact of vocal effort variability on automatic speech recognition. Speech Communication, 54(6), 732-742.

Zhang, J., & Zhang, M. (2011, April). A speech recognition method based clustering neural network integration. In 2011 International Conference on Electric Information and Control Engineering (pp. 1120-1122). IEEE.

Zhang, S. X., & Gales, M. J. (2012). Structured SVMs for automatic speech recognition. IEEE Transactions on Audio, Speech, and Language Processing, 21(3), 544-555.

Zhang, X., Povey, D., & Khudanpur, S. (2015). A diversity-penalizing ensemble training method for deep learning. In INTERSPEECH, 3590-3594.

Zhang, X., Sun, J., & Luo, Z. (2014). One-against-all weighted dynamic time warping for language-independent and speaker-dependent speech recognition in adverse conditions. *PloS ONE*, 9(2), e85458. https://doi.org/10.1371/journal.pone.0085458.

Zighed, D. A. (2008). Arbres de décision en situation d'asymétrie, thèse de doctorat, Université Lyon II.

Zouari, H. K. (2004). Contribution à l'évaluation des méthodes de combinaison parallèle de classifieurs par simulation. Thèse de doctorat, Université de Rouen.

PUBLICATIONS DE L'AUTEUR

Bougamouza, F., Hazmoune, S., & Benmohammed, M. (2016). Using Mel Frequency Cepstral Coefficient method for online Arabic characters handwriting recognition. In 2016 5th International Conference on Multimedia Computing and Systems (ICMCS), (pp. 87-92). IEEE.

Bougamouza, F., Hazmoune, S. and Benmohammed, M. (2018). Normalisation of handwriting speed for online Arabic characters' recognition. Int. J. Computational Vision and Robotics, Vol. 8, No. 6, pp.591–605.

Hazmoune, S., Bougamouza, F., & Benmohammed, M. (2010). La reconnaissance automatique de la parole, par combinaison de classifieurs markoviens Sélectionnés Par Algorithmes Génétiques. Premier séminaire national sur la modélisation et la simulation informatiques des systèmes industriels, Skikda.

Hazmoune, S., Bougamouza, F., Mazouzi, S., & Benmohammed, M. (2018). A new hybrid framework based on hidden Markov models and K-nearest neighbors for speech recognition. International Journal of Speech Technology, 21(3), 689-704. (Springer), DOI: 10.1007/s10772-018-9535-4

Hazmoune, S., Bougamouza, F., Mazouzi, S., & Benmohammed, M. (2013a). A novel speech recognition approach based on multiple modeling by hidden Markov models. In International Conference on Computer Applications Technology (ICCAT), 2013 (pp. 1-IEEE, DOI: 10.1109/ICCAT.2013.6522028

Hazmoune, S., Bougamouza, F., Mazouzi, S., & Benmohammed, M. (2013b). Contributions to HMM-based Speech Recognition Systems, International Journal of Computational Linguistics Research (IJCLR), ISSN 0976-4178, Vol.4 No.1, (pp.38-47).

Hazmoune, S., Bougamouza, F., Mazouzi, S., & Benmohammed, M. (2013c). Arabic Speech Recognition based on Multiple HMM Modeling and Genetic Selection, the 3rd International Conference on Information Systems and Technologies (ICIST'2013), Tangier, Morocco.

ANNEXE : SELECTION GENETIQUE DE CLASSIFIEURS DANS UNE APPROCHE MARKOVIENNE ENSEMBLISTE

1. INTRODUCTION

Cette annexe est consacrée à la présentation des résultats supplémentaires de l'approche ensembliste sur une base de données personnelle. Ces résultats concernent, seulement, le cas d'utilisation de différents modèles initiaux comme méthode de création de l'ensemble. La sélection de classifieurs est réalisée en utilisant un algorithme génétique. Et la fusion des classifieurs est faite selon la règle de moyenne de vraisemblances.

Notons que, durant la reconnaissance, un seuil de rejet relatif à chaque classe est utilisé afin de traiter le cas d'exemples qui n'appartiennent pas au vocabulaire. Ce seuil est calculé sur la base d'apprentissage et celle de validation, en appliquant la formule empirique suivante :

$$seuil(w) = \frac{1}{3} Min\big(Pr(w_i)\big),$$

Avec, w_i est la $i^{ème}$ occurrence de la classe w et $\mathrm{Pr}(\mathrm{w_i})$ est la moyenne de vraisemblances données par les classifieurs de base pour l'occurrence i de la classe w.

2. UN ALGORITHME GENETIQUE POUR LA SELECTION D'ENSEMBLES DE CLASSIFIEURS

La sélection de classifieurs à combiner est l'un des aspects les plus importants de notre approche ensembliste. Elle peut être réalisée en utilisant des techniques d'optimisation combinatoire comme les algorithmes génétiques, les colonies de fourmis et l'essaim de particules. Nous avons proposé dans (Hazmoune et al., 2013a) et (Hazmoune et al., 2013b) une sélection génétique qui se déroule selon les étapes suivantes :

a. Population initiale, codage d'individus et évaluation de la fonction objective (fitness)

La population initiale est un ensemble aléatoire de 20 chromosomes (pop_size = 20), chaque chromosome contient 100 gènes codés en binaire (chaque gène correspond à un classifieur candidat), où la présence de 1 signifie que le classifieur correspondant à ce gène est sélectionné, et la présence de 0 signifie qu'il ne l'est pas. Afin de réduire le temps de réponse et l'espace de stockage, le nombre de classifieurs sélectionnés dans chaque chromosome est limité entre 2 et 5. La population initiale est créée aléatoirement, car la position de l'optimum dans l'espace de solutions est totalement inconnue.

Une fois qu'une génération est créée, la valeur de fitness de chaque chromosome de la population est calculée. Le fitness choisi pour notre problème, est le taux de reconnaissance de la base de validation.

b. La reproduction

Dans cette étape, de nouveaux individus (fils) sont créés en effectuant les opérations suivantes :

- **La sélection :** La méthode de sélection utilisée est la méthode élitiste. Elle consiste à sélectionner les meilleurs chromosomes en fonction de leurs fitness.
- **Le croisement :** Une fois la sélection terminée, un croisement à un point de coupure est appliqué, ce point est choisi aléatoirement entre 2 et 99, la combinaison se fait entre le chromosome i ($1 \leq i \leq pop\_size\ /2$) et le chromosome j ($j = i + pop\_size/2$). Après le croisement, si nous obtenons des fils identiques à leurs parents ou des fils qui ne vérifient pas la contrainte ($2 \leq n \leq 5$), le point de coupure sera changé, et le croisement doit être recommencé jusqu'à aboutir à de nouveaux chromosomes. La probabilité de croisement est fixée à $\rho\_c = 0{,}8$.
- **La mutation :** Contrairement au croisement, la mutation se fait au sein du même chromosome. Elle est effectuée avec une faible probabilité ($\rho\_m = 0{,}02$ dans notre cas). Tous les chromosomes provenant du croisement sont soumis à une mutation, où deux positions dans le chromosome sont choisies au hasard et les gènes correspondants sont inversés. Si ces gènes sont identiques, les positions de mutation doivent être modifiées. Cela, permet d'assurer l'évolution des populations et de bien exploiter l'espace de recherche.
- **Évaluation de la nouvelle population :** L'évaluation se fait en calculant le taux de reconnaissance pour chaque chromosome de la nouvelle population.
- **Le remplacement (élitisme) :** les parents et les fils sont triés par ordre décroissant de leur fitness et les (pop_size) premiers chromosomes sont choisis comme individus de la nouvelle population.

c. La convergence

Le critère d'arrêt que nous avons choisi est le nombre de générations déterminé par expérimentation. La convergence vers une solution satisfaisante est trouvée après 20 générations. L'algorithme génétique s'arrête, donc, lorsqu'il atteint cette valeur, et l'ensemble de n classifieurs maximisant le fitness est sélectionné. S'il y a deux ensembles ayant la même valeur de fitness, nous choisissons celui qui contient le moins de classifieurs afin de réduire l'espace de stockage et le temps de réponse du système durant la phase de reconnaissance. Le résultat de cette étape est un ensemble de n classifieurs.

Il est à noter que les paramètres ($\rho\_c, \rho\_m, pop\_size, critères\ d'arrêt$) et les méthodes de reproduction sont tous choisis empiriquement.

3. BASE DE DONNEES UTILISEE

L'approche proposée est évaluée sur une base de données des chiffres arabes (de 0 à 9). Elle contient des occurrences des 10 chiffres prononcées par 30 locuteurs (15 femmes et

15 hommes). Chaque chiffre est répété 10 fois par le même locuteur, nous obtenons donc 300 occurrences pour chaque chiffre, ce qui donne au total 3000 fichiers audio. La base de données est divisée en 3 parties : 1500 exemples pour l'apprentissage, 500 exemples pour la validation et 1000 exemples pour le test. Les exemples de la base de validation et celle de test sont prononcés par des locuteurs qui n'ont pas participé à l'apprentissage. Le système est, donc, indépendant du locuteur.

Dans toutes les expérimentations effectuées, l'analyse acoustique est réalisée selon la méthode MFCC. Le signal de la parole est segmenté en trames de 256 échantillons, et la fenêtre d'analyse est décalée de 128 échantillons. Chaque trame est représentée par un vecteur composé de 13 coefficients cepstraux, le logarithme d'énergie et leurs dérivées premières et secondes. A l'issue de l'analyse acoustique d'un signal parole, nous obtenons, donc, une suite de vecteurs. Chacun d'eux contient 42 éléments.

4. RESULTATS

Afin de montrer l'efficacité de la sélection génétique proposée, nous avons testé trois méthodes : Sélection des meilleurs classifieurs individuels en termes de taux de reconnaissance, sélection aléatoire des classifieurs, et sélection génétique de classifieurs en maximisant le taux de reconnaissance de l'ensemble et en minimisant le nombre de classifieurs sélectionnés. Les résultats d'évaluation, en plusieurs essais, sont résumés dans les tableaux 1, 2 et 3. Le tableau 1, correspond à $n = 2$ classifieurs sélectionnés. Le tableau 2, correspond à $n = 3$ et le tableau 3 correspond à $n = 4$. Dans tous les tableaux, nous comparons les taux de reconnaissance (TR) des trois méthodes de sélection avec ceux des différents classifieurs individuels. Les meilleurs taux de reconnaissance sont marqués en gras.

À partir des tableaux, nous pouvons constater que :

- Le classifieur HMM est sensible au choix du modèle initial : Chaque fois que le modèle initial est changé, un nouveau modèle final est obtenu (différence de taux de reconnaissance d'un classifieur individuel à un autre).
- Le taux de reconnaissance de l'approche ensembliste est toujours supérieur à ceux des classifieurs individuels.
- Les meilleurs classifieurs individuels ne sont pas nécessairement les meilleurs pour la combinaison (essai $A1$, essai $A2$ et essai $A3$ vs essai $B1$, essai $B2$ et essai $B3$).
- Les meilleurs résultats sont obtenus en utilisant la sélection génétique (essai C).
- La combinaison de trois classifieurs pourrait être un bon compromis entre le taux de reconnaissance et le temps de réponse du système.

Nous pouvons résumer que l'approche ensembliste proposée est bien meilleure que l'approche classique. Le taux de reconnaissance a atteint la valeur de 95% en utilisant 2 classifieurs sélectionnés par algorithme génétique et 96,3333% en utilisant 3 classifieurs contre (70 à 89%) pour l'approche classique.

Tableau 1. Comparaison des performances de l'approche ensembliste et les classifieurs individuels ($n = 2$ classifieurs)

		TR du 1ier classifieur (%)	TR du 2ième classifieur (%)	TR de l'ensemble des 2 classifieurs (%)
Meilleurs classifieurs individuels	Essai A1	89.0000	87.3333	**91.7778**
	Essai A2	87.3333	87.8889	**89.5556**
	Essai A3	86.2222	86.6667	**88.4444**
Sélection aléatoire	Essai B1	86.7778	83.4444	**91.2222**
	Essai B2	86.7778	87.8889	**91.7778**
	Essai B3	80.6667	83.4444	**90.1111**
Sélection génétique	Essai C	80.6667	86.6667	**95**

Tableau 2. Comparaison des performances de l'approche ensembliste et les classifieurs individuels ($n = 3$ classifieurs)

		TR du 1ier classifieur (%)	TR du 2ième classifieur (%)	TR du 3ième classifieur (%)	TR de l'ensemble des 3 classifieurs (%)
Meilleurs classifieurs individuels	Essai A1	85.1111	89.0000	87.3333	**94**
	Essai A2	87.8889	87.3333	87.8889	**90.1111**
	Essai A3	86.2222	85.6667	83.4444	**91.2222**
Sélection aléatoire	Essai B1	83.4444	82.8889	89.0000	**94.5556**
	Essai B2	79.5556	85.1111	89.0000	**92.8889**
	Essai B3	80.6667	81.7778	81.7778	**95.1111**
Sélection génétique	Essai C	83.4444	83.4444	89.0000	**96.3333**

Tableau 3. Comparaison des performances de l'approche ensembliste et les classifieurs individuels ($n = 4$ classifieurs)

		TR du 1ier classifieur (%)	TR du 2ième classifieur (%)	TR du 3ième classifieur (%)	TR du 4ième classifieur (%)	TR de l'ensemble des 4 classifieurs (%)
Meilleurs classifieurs individuels	Essai A1	85.1111	89.0000	87.3333	85.6667	**95.1111**
	Essai A2	87.3333	83.4444	84.0000	89.0000	**92.8889**
	Essai A3	85.1111	86.7778	85.6667	85.1111	**93.4444**
Sélection aléatoire	Essai B1	86.7778	81.2222	85.1111	87.8889	**95.6667**
	Essai B2	81.2222	87.8889	85.6667	81.2222	**94**
	Essai B3	79.0000	85.6667	82.8889	89.0000	**94.5556**
Sélection génétique	Essai C	87.8889	87.3333	79.0000	85.6667	**96.8889**