République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Badji Mokhtar-Annaba
Faculté des Sciences de l'Ingénieur
Département Informatique

Mémoire

Présenté en vue de l'obtention du diplôme de **Magistère en Informatique**
Option : Reconnaissance de formes et intelligence artificielle

Titre:

Modèle probabiliste indexé par des arbres pour la reconnaissance de formes : Application à la reconnaissance automatique de la parole

Par : Mr.Nacereddine HAMMAMI
Encadré par : Mr. Mokhtar Sellami, Professeur U. ANNABA

Jury

Dr. BAHI Halima	Université de Annaba	Présidente
Pr. SELLAMI Mokhtar	Université de Annaba	Promoteur
Dr. SOUICI Labiba	Université de Annaba	Examinateur
Dr. KHADIR Tarek	Université de Annaba	Examinateur
Dr. SERIDI Hamid	Université de Guelma	Examinateur

2008 - 2009

Remerciements

Je rends grâce à Allah qui m’a donné l’aide, la patience et le courage pour accomplir ce modeste travail scientifique.

Je voudrai remercie tout particulièrement mon encadreur Mr. Sellami Mokhtar, professeur au département d’informatique de l’université de Badji Mokhtar Annaba, pour sa confiance en moi. Son support moral et scientifique a été indispensable pour accomplir ce travail. Je dois avouer que j’ai eu la chance de travailler avec un homme dextrement compétant mais surtout un homme formidable.

Mes remerciements s'adressent à mes rapporteurs, qui ont bien voulu accepter d'évaluer le présent travail et ce malgré toutes les responsabilités qu'ils assument. Je les remercie pour le temps qu'ils consacreront à la lecture de ce mémoire et je souhaite qu'ils y trouvent entière satisfaction.

Abstract

The probabilistic inference has become a core technology in AI, largely due to development in graph theory for the representation and manipulation of complex probability distributions.

We have proposed in this work to apply a probabilistic model based on tree structures to the problems of speech recognition. The latter is an active area of research since the early 50s.

We developed in this paper a system of recognition of isolated speech using the tree model. We used a novel method consists in a first step in using models in which interactions are described by spanning trees carried by G, and thereafter to reduce the complexity of algorithms and calculations we have imposed a properly structure of tree to the problem of recognition of words. The results are good.

We have detailed the results at word vocabularies for different languages. To our knowledge the first time that the tree models are used to perform speech recognition.

These new methods have been tested and compared with those of most models used in speech recognition including hidden Markov models (HMMs). Benchmark data bases for the recognition task of isolated words, shows competitive recognition rates.

Résumé

L'inférence probabiliste est devenue une technologie de base dans l'IA, en grande partie due au développement dans la théorie des graphes pour la représentation et la manipulation des distributions probabilistes complexes.

Nous avons proposé dans ce travail d'appliquer un modèle probabiliste basé sur les structures d'arbre aux problèmes de la reconnaissance automatique de la parole. Ce dernier est un domaine d'études actif depuis le début des années 50.

Nous avons développé dans ce mémoire un système de reconnaissance de la parole isolée en utilisant le modèle d'arbre. Nous avons utilisé une méthode originale consiste dans une première étape à utiliser des modèles pour lesquels les interactions sont décrites par des arbres couvrants portés par G, par la suite et afin de réduire la complexité des algorithmes et des calculs nous avons imposé une structure d'arbre adéquate au problème de la reconnaissance des paroles. Les résultats obtenus sont bons.

Nous avons détaillé les résultats obtenus au niveau du mot pour des vocabulaires de différentes langues. C'est à notre connaissance la première fois que les modèles d'arbre sont utilisés pour effectuer une reconnaissance de la parole isolée.

Ces nouvelles méthodes, qui ont été testées et comparées à celles des modèles les plus utilisés en reconnaissance de la parole notamment les modèles de Markov caches (HMMs). Des bases *benchmark* pour la tâche de reconnaissance des mots isolés, montre des taux de reconnaissance compétitifs.

Liste des tableaux

Liste des figures

Table des Matières

Chapitre I

Introduction générale

De nos jours, il n'y a aucun doute que les machines et les ordinateurs sont largement répandus presque par chaque personne pour faciliter la gestion et le stockage de l'information. Pour cette raison, les outils comme les ordinateurs, et la microélectronique ont connu une évolution considérable dans ces dernières années.

Cette évolution a permis de faciliter la communication entre l'homme et la machine par l'usage de la parole où l'information transmise à la machine est un signal vocal. Après traitement la machine répond par un autre signal vocal adéquat.

Il est clair que cette opération nécessite des traitements sur le signal vocal telles que reconnaissance - synthèse. Où en remarque la présence de tous les champs d'un système TLH (technologie de langue humaine).

La reconnaissance automatique de la parole (RAP) traite le fait d'identifie le discours humain; processus de langage naturel (PLN) avec extraction d'information des phrases ou la gestion du dialogues, connexes aussi avec l'intelligence artificielle (IA); et les techniques de conversions texte parole (CTP).

Il n'est pas difficile d'imaginer une situation où TLH peut être appliqué avec succès, et par conséquent RAP, mais les applications les plus importantes peut être la dictée (transcriptions médicales, légales ou d'affaires), applications de téléphone (téléphone opérations bancaires, audio messagerie) et applications conçue pour rendre quelques services accessible aux personnes handicapées (distributeur automatique de billets pour personnes aveugles, une conversation téléphonique assistée par ordinateur pour sourd ou sourd-muet ou paramètres commandés par voix pour des personnes avec des problèmes musculaires).

La reconnaissance automatique de la parole (RAP) est un exemple typique d'un problème de classification automatique des modèles (formes), le but d'un RAP est de déterminé la séquence

des mots la plus probable pour un flux d'information acoustique. Un système RAP se réalise en deux phases apprentissage et classification.

Mais, avant que les modèles soient appris ou classés. Le signal parole a besoin d'être codé dans un vecteur de paramètres acoustiques représentatifs.

I.1 Historique

La reconnaissance de la parole est une discipline récente. Vers 1950 apparut le premier système de reconnaissance de chiffres, appareil entièrement câblé et très imparfait. Vers 1960, l'introduction des méthodes numériques et l'utilisation des ordinateurs changent la dimension des recherches. Néanmoins, les résultats demeurent modestes car la difficulté du problème avait été largement sous-estimée, en particulier en ce qui concerne la parole continue. Vers 1970, la nécessité de faire appel à des contraintes linguistiques dans le décodage automatique de la parole avait été jusque-là considérée comme un problème d'ingénierie. La fin de la décennie 70 voit se terminer la première génération des systèmes commercialisés de reconnaissance de mots. Les générations suivantes, mettant à profit les possibilités sans cesse croissantes de la microinformatique, posséderont des performances supérieures (systèmes multi locuteurs, parole continue).

On peut résumer en quelques dates les grandes étapes de la reconnaissance de la parole :

• 1952 : Reconnaissance des 10 chiffres, pour un mono locuteur, par un dispositif électronique câblé

• 1960 : Utilisation des méthodes numériques

• 1965 : Reconnaissance de phonèmes en parole continue

• 1968 : Reconnaissance de mots isolés par des systèmes implantés sur gros ordinateurs (jusqu'à 500 mots)

• 1969 : Utilisation d'informations linguistiques

• 1971 : Lancement du projet ARPA aux USA (15 millions de dollars) pour tester la faisabilité de la compréhension automatique de la parole continue avec des contraintes raisonnables

• 1972 : Premier appareil commercialisé de reconnaissance de mots

• 1976 : Fin du projet ARPA ; les systèmes opérationnels sont HARPY, HEARSAY I et II et HWIM

• 1978 : Commercialisation d'un système de reconnaissance à microprocesseurs sur une carte de circuits imprimés

• 1981 : Utilisation de circuits intégrés VLSI (Very Large Scale Integration) spécifiques du traitement de la parole
• 1981 : Système de reconnaissance de mots sur un circuit VLSI
• 1983 : Première mondiale de commande vocale à bord d'un avion de chasse en France
• 1985 : Commercialisation des premiers systèmes de reconnaissance de plusieurs milliers de mots
1986 : lancement du projet japonais ATR de téléphone avec traduction automatique en temps réel
• 1988 : Apparition des premières machines à dicter par mots isolés
• 1989 : Recrudescence des modèles connexionnistes neuromimétiques
• 1990 : Premières véritables applications de dialogue oral homme-machine
• 1994 : IBM lance son premier système de reconnaissance vocale sur PC
• 1997 : Lancement de la dictée vocale en continu par IBM

I.2. Problématique

La problématique de la reconnaissance de parole est particulièrement riche et complexe. Il existe en effet un grands nombre de problèmes différents à traiter dans lesquels les formes (mot ou phrase..) à reconnaître sont nombreuses, soumises à une variabilité importante et donc source de confusion.

Les Systèmes de Reconnaissance Automatique de la Parole (SRAP), avec la diversité des techniques qui les sous-tendent, réagissent inégalement par rapport à la multitude de situations aux quelles ils sont confrontés en milieu réel. Il est intéressant d'observer que dans toute cette panoplie de techniques, il n'existe pas de système adapté à toutes les situations de l'élocution.

La modélisation acoustique par les méthodes les plus performantes de l'état de l'art reste insuffisante; cette faiblesse est un facteur limitant des systèmes de RAP.

Nous cherchons à améliorer la qualité de la modélisation acoustique, en appliquant des modèles probabilistes autres que les systèmes à base de modèles de Markov cachés.

I.3. Contribution

Les techniques de l'état de l'art proposent de compenser le manque de discrimination par une réestimation des paramètres des modèles à l'aide d'un critère discriminant.

Nous développons un système de reconnaissance des mots isolés utilisant un modèle statistique probabiliste [24] [25]. Nous avons utilisé une méthode originale basée sur l'utilisation d'un modèle d'arbre avec une structure prédéfini de l'arbre probabiliste.

Les résultats obtenus sur deux bases de données sont très bons. C'est à notre connaissance la première fois que les modèles d'arbre utilisé pour effectuer une reconnaissance de la parole notamment dans la langue arabe.

I.4 Structure du mémoire

Le mémoire est organisé en quatre chapitres :

Dans le second chapitre, nous décrirons l'architecture générale d'un système de reconnaissance de la parole et les différents éléments qui le composent. Après avoir mis en évidence les principales caractéristiques du signal de parole et les difficultés rencontrées lors de sa modélisation, nous résumerons l'analyse acoustique préliminaire effectuée sur la parole avant le processus de reconnaissance.

Dans le troisième chapitre nous décrirons les Méthode les plus utilisées en reconnaissance de la parole notamment les modèles de Markov cachés, (HMMs), dans le chapitre 4 nous présentons le modèle d'arbre développés ainsi que les résultats obtenus.

Nous donnerons enfin dans de ce mémoire, nos conclusions sur les travaux que nous avons effectués, ainsi que les perspectives que nous envisageons pour la poursuite de notre travail.

Chapitre II

La reconnaissance automatique de la parole

Avant d'aborder le sujet de ce mémoire, à savoir le modèle d'arbre pour la reconnaissance automatique de la parole, il est indispensable de comprendre comment un système de reconnaissance automatique de la parole opère pour fournir l'énoncé de la phrase qui a été prononcée par un locuteur. L'objectif était de proposer une vue synthétique des éléments à la base des systèmes actuels.

II.1 Les problèmes de variabilité de la parole [2][3][23]

La complexité du signal acoustique de parole résulte de l'interaction de nombreux facteurs de variabilité (figure 2.1).

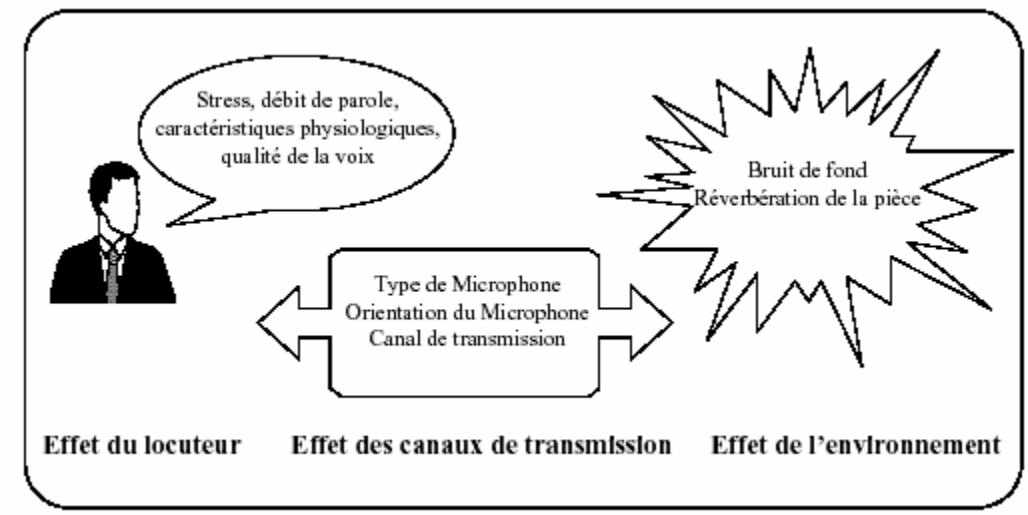


Figure 2.1 Sources de variabilité du signal de parole

Certains sont inhérents au signal de parole, comme la redondance des informations qui y sont contenues ou les effets de la coarticulation.

Les autres facteurs correspondent aux sources de variabilités, qui peuvent rendre la représentation de deux signaux acoustiques correspondant au même message très différentes. Ces sources de variabilités au niveau du signal acoustique sont dues au locuteur lui-même, à l'environnement ou au canal de transmission du signal (microphone).

II.1.1 Redondance des informations contenues dans le signal

La représentation dans le domaine temporel du signal acoustique numérisé est caractérisée par une redondance d'informations qui ne sont pas fondamentalement nécessaires pour reconnaître correctement le message qui a été prononcé.

Outre le message proprement dit, la communication parlée véhicule effectivement de nombreuses autres informations paralinguistiques, comme le sexe du locuteur, son identité, son état de santé, son état émotionnel, etc. Pour un SRAP, ce flux d'informations représente une quantité colossale de données à exploiter. Par exemple, un signal échantillonné à 16 kHz sur 16 bits représente un débit de 256 KBits/s, ce qui implique que le SRAP doit traiter 32000 octets de données par seconde [2].

Pour des raisons de rapidité d'exécution, SRAP cherchera donc à minimiser ce flux important de données en ayant recours à une étape de prétraitement du signal, afin de le débarrasser des informations superflues et inutiles pour la reconnaissance d'un message.

II.1.2 Phénomènes de coarticulation

Tout message peut être décomposé en une suite de mots, qui peuvent à leur tour être décrits comme une suite d'unités acoustiques. Cela laisse supposer que la parole est un processus séquentiel, au cours duquel des unités élémentaires et indépendantes se succèdent.

Toutefois, les phonéticiens eux-mêmes éprouvent parfois des difficultés à identifier individuellement ces sons caractéristiques du langage dans un signal de parole, même si quelques événements acoustiques particuliers peuvent être détectés. La parole est en réalité un continuum sonore, où il n'existe pas de pause perceptible entre les mots qui pourrait faciliter leur localisation automatique par un SRAP En outre, lors de la production d'un message, l'inertie de l'appareil phonatoire et l'anticipation du geste articulatoire influencent la production de chaque son, si bien que la réalisation acoustique d'un son est fortement perturbée par les sons qui le précèdent mais également par ceux qui le suivent. Ces effets s'étendent sur la durée d'une syllabe, voire même au-delà, et sont amplifiés par un rythme d'élocution soutenu. Le choix de l'unité acoustique directement identifiable par un SRAP est alors primordial.

On distingue habituellement trois classes d'unités acoustiques les phonèmes, les unités courtes infra- phonémiques (ou phones) et les unités longues supra- phonémiques »phones, triphones, semi-syllabes, syllabes, mots).

Une unité courte peut être en général mieux identifiée, mais ne possédant pas de statut linguistique particulier, leur concaténation pour former des unités plus longues est problématique. L'utilisation de phonèmes souffre d'une mauvaise modélisation des effets de coarticulation et d'une difficulté pour les localiser. Toutefois leur nombre assez faible facilite la mise en oeuvre du SRAP [2].

En ce qui concerne les unités longues enfin, leur utilisation permet une meilleure modélisation des effets de la coarticulation interne, mais la mise en œuvre du SRAP n'est pas aisée en raison de leur nombre important.

II.1.3 Variabilités inter-locuteurs et intra-locuteur

La variabilité inter-locuteurs, qui est généralement considérée comme étant a priori la plus importante, suggère que la prononciation d'un même énoncé par deux personnes est différente.

Les différences physiologiques entre locuteurs de l'appareil phonatoire, comme la longueur du conduit vocal, la forme et le volume des cavités résonnantes, ou la forme des lèvres, influencent la réalisation acoustique d'un même message. Pour s'en convaincre, il suffit de considérer par exemple les voix d'enfants et d'adultes, qui sont les plus reconnaissables car les caractéristiques de leurs appareils phonatoires sont les plus différenciées. A ces différences physiologiques s'ajoutent les habitudes acquises au sein du milieu social et géographique, comme la vitesse d'élocution, ou les accents régionaux. Dans la figure 2.2, deux locuteurs ont prononcé le même message, le premier avec un débit de parole normale, le second avec un débit de parole rapide.

Ces différences au niveau de la réalisation d'un même message sont clairement observables sur les signaux de parole et sur les spectrogrammes représentés dans la figure.

a) Variabilité intra-locuteur

La variabilité intra-locuteur identifie les différences dans le signal produit par une même personne. Cette variation peut résulter de l’état physique ou moral du locuteur. Une maladie des voies respiratoires peut ainsi dégrader la qualité du signal de parole de manière à ce que celui-ci devienne totalement incompréhensible, même pour un être humain. L’humeur ou l'émotion du locuteur peut également influencer son rythme d’élocution, son intonation ou sa phraséologie [2].

Il existe un autre type de variabilité intra-locuteur lié à la phase de production de parole ou de préparation à la production de parole. Cette variation est due aux phénomènes de coarticulation. Il est possible de voir la phase de production de la parole comme un compromis entre une minimisation de l’énergie consommée pour produire des sons et une maximisation des scores d’atteinte des cibles que sont les phonèmes tels qu’ils sont théoriquement définis par la phonétique.

Un locuteur adoptera donc un compromis qui est généralement partagé par une vaste majorité de la communauté de langage à laquelle il appartient bien que ce compromis lui soit propre du fait de sa physionomie particulière. Ce compromis peut d'ailleurs être retrouvé à un plus haut niveau

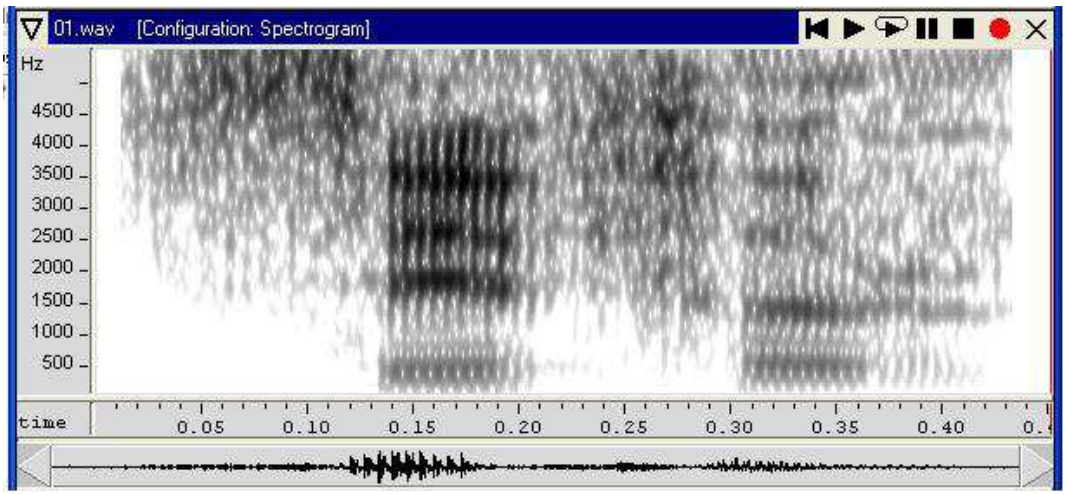
avec la notion d'idiolecte. Ce locuteur essaiera, lors d'une phase de production de parole, d'atteindre les buts qui lui sont fixés par les différents éléments de sa phrase tout en conservant un rythme naturel de production de la parole. Les cibles peuvent alors être modifiées du fait d'un certain contexte phonétique. Ce contexte peut être antérieur, lorsque le phonème provoquant une modification se trouve avant le phonème considéré, ou postérieur lorsque le phonème perturbateur se trouve après.

La coarticulation peut enfin se produire à l'échelle d'un ou de plusieurs phonèmes adjacents, ce dernier cas étant cependant très rare. La variabilité intra-locuteur est cependant beaucoup plus limitée que la variabilité inter-locuteur que nous allons étudier maintenant. Il est en effet possible, malgré les problèmes énoncés ci-avant, de mettre en oeuvre des systèmes automatiques d'identification du locuteur, à la manière d'une personne reconnaissant une voix familière. Cette capacité est la preuve qu'une certaine constance existe dans la phase de production de la parole par un même individu.

b) Variabilité inter-locuteur

La variabilité inter-locuteur est un phénomène majeur en reconnaissance de la parole. Comme nous venons de le rappeler, un locuteur reste identifiable par le timbre de sa voix malgré une variabilité qui peut parfois être importante.

La contrepartie de cette possibilité d'identification à la voix d'un individu est l'obligation de donner aux différents sons de la parole une définition assez souple pour établir une classification phonétique commune à plusieurs personnes.



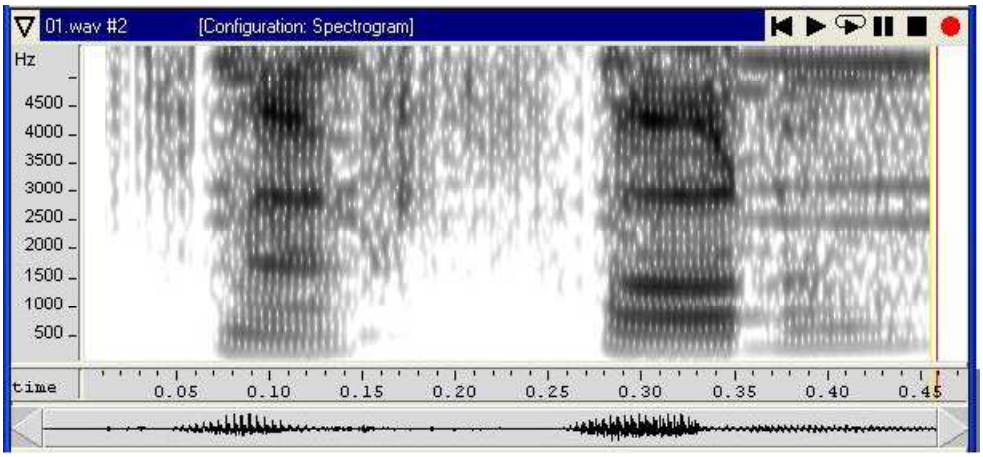


Figure 2.2 Variabilité inter-locuteurs

La cause principale des différences inter-locuteurs est de nature physiologique. La parole est principalement produite grâce aux cordes vocales qui génèrent un son à une fréquence de base, le fondamental. Cette fréquence de base sera différente d'un individu à l'autre et plus généralement d'un genre à l'autre, une voix d'homme étant plus grave qu'une voix de femme, la fréquence du fondamental étant plus faible. Ce son est ensuite transformé par l'intermédiaire du conduit vocal, délimité à ses extrémités par le larynx et les lèvres. Cette transformation, par convolution, permet de générer des sons différents qui sont regroupés selon les classes que nous avons énoncées précédemment. Or le conduit vocal est de forme et de longueur variables selon les individus et, plus généralement, selon le genre et l'âge. Ainsi, le conduit vocal féminin adulte est, en moyenne, d'une longueur inférieure de 15% à celui d'un conduit vocal masculin adulte [3].

Le conduit vocal d'un enfant en bas âge est bien sûr inférieur en longueur à celui d'un adulte. Les convolutions possibles seront donc différentes et, le fondamental n'étant pas constant, un même phonème pourra avoir des réalisations acoustiques très différentes. La variabilité inter-locuteur trouve également son origine dans les différences de prononciation qui existent au sein d'une même langue et qui constituent les accents régionaux.

Ces différences s'observeront d'autant plus facilement qu'une communauté de langue occupera un espace géographique très vaste, sans même tenir compte de l'éventuel rayonnement international de cette communauté et donc de la probabilité qu'a la langue d'être utilisée comme seconde ou, pire, troisième langue par un individu de langue maternelle étrangère.

Là aussi, la définition phonétique tout autant qu'une définition stricte d'un vocabulaire ou d'une grammaire peuvent être mises à mal. La variabilité inter-locuteur telle qu'elle vient d'être présentée permet de comprendre aisément pourquoi les méthodes de reconnaissance des formes

fondées sur la quantification de concordances entre une forme à analyser et un ensemble de définitions strictes plus ou moins formelles ne peuvent être appliquées, avec un succès limité, qu'à des applications où le nombre de définitions est restreint, limitant ainsi le nombre des possibles.

D'une manière générale, la définition assez floue des différents phonèmes ou des différents mots d'une langue est la cause de nombreuses erreurs de classification dans les systèmes de décodage acoustico-phonétique, DAP. Mais la variabilité inter-locuteur, malgré son importance évidente, n'est pas encore la variabilité la plus importante car les différences au sein des classes phonétiques sont en nombre restreint.

L'environnement du locuteur est porteur d'une variabilité beaucoup plus importante, comme nous allons le voir brièvement dans le paragraphe suivant.

II.1.4 Variabilités dues à l'environnement et au canal de transmission

L'absence de bruit de fond est dans la pratique impossible. A moins d'être dans une chambre isolée, n'importe lequel des appareils que nous utilisons émet un bourdonnement qui est la plupart du temps audible et qui génère des parasites dans le signal acoustique. Dans certains cas, ce bruit de fond peut être si élevé qu'il influe directement sur la prononciation du locuteur, le poussant à ralentir son élocution et à augmenter l'intensité sonore de son discours (effet Lombard)[3].

Par ailleurs, le microphone utilisé par le locuteur pour transmettre son message au système possède des caractéristiques spécifiques 4 et peut alors avoir des qualités d'acquisition plus ou moins bonnes de certaines fréquences. L'acquisition de certaines fréquences peut également être rendue imparfaite selon l'angle et la distance du microphone lors de son utilisation .

Enfin, le canal de transmission (fil, ondes radio, etc.) peut introduire des parasites dans le signal.

II.2 Problématique de reconnaissance de la parole

Pour bien appréhender le problème de la reconnaissance automatique de la parole, il est bon d'en comprendre les différents niveaux de complexité et les différents facteurs qui en font un problème difficile.

- Le système doit-il être optimisé pour un unique locuteur ou est-il destiné à devoir se confronter à plusieurs utilisateurs ?

On peut aisément comprendre que les systèmes dépendants d'un seul locuteur sont plus faciles à développer et sont caractérisés par de meilleurs taux de`reconnaissance que les systèmes indépendants du locuteur étant donné que la variabilité du signal de parole est plus limitée. Cette dépendance au locuteur est cependant acquise au prix d'un entraînement spécifique à chaque utilisateur. Ceci n'est néanmoins pas toujours possible. Par exemple, dans le cas d'applications téléphoniques, on comprend bien que les systèmes puissent être utilisés par n'importe qui et donc être indépendants du locuteur. Bien que la méthodologie de case reste la même, cette indépendance au locuteur est obtenue par l'acquisition de nombreux locuteurs (couvrant si possible les différents dialectes) qui sont utilisés simultanément pour l'entraînement de modèles susceptibles d'en extraire toutes les caractéristiques majeures. Une solution intermédiaire parfois utilisée consiste à développer des systèmes capables de s'adapter rapidement (de façon supervisée ou non) au nouveau locuteur.

- Le système reconnaît-il des mots isolés ou de la parole en continue ?

Evidemment, il est plus simple de reconnaître des mots isolés bien séparés par des périodes de silence que la séquence de mots constituant une phrase. En effet, dans ce dernier cas, non seulement la frontière entre les mots n'est plus connue mais les mots deviennent fortement articulés (c'est-à-dire que la prononciation de chaque mot est affectée par le mot qui précède ainsi que par celui qui suit.

Dans le cas de la parole continue, le niveau de complexité varie également selon qu'il s'agisse de texte lu, de texte parlé ou, beaucoup plus difficile, de langage naturel0avec ses hésitations, phrases grammaticalement incorrectes, faux départs, etc.. .

Un autre problème, qui commence à être bien maîtrisé, concerne la reconnaissance de mots clés en parole libre. Dans ce dernier cas, le vocabulaire à reconnaître est relativement petit et bien défini mais le locuteur n'est pas contraint de parler en mots isolés. Par exemple, si un utilisateur est invité à répondre par «oui» ou «non», il peut répondre «oui, s'il vous plaît». Dans ce contexte, un problème qui reste particulièrement difficile est le rejet de phrases ne contenant aucun mot clé.

La taille du vocabulaire et son degré de confusion sont également des facteurs importants. Les petits vocabulaires sont plus faciles à reconnaître que les grands vocabulaires, étant donné que dans ce dernier cas, les possibilités de confusion augmentent. Certains petits vocabulaires peuvent cependant s'avérer particulièrement difficiles à traiter ; ceci est le cas, par exemple, pour

l'ensemble des lettres de l'alphabet, contenant surtout des mots très courts et proches au niveau acoustique.

- le système est-il robuste ?

Autrement dit, le système est-il capable de fonctionner proprement dans des conditions difficiles ? En effet, de nombreuses variables pouvant affecter significativement les performances des systèmes de reconnaissance ont été identifiées :

- bruits d'environnement (dans une rue, un bistrot etc.…)
- Déformation de la voix par l'environnement (réverbérations, échos, etc.…)
- Qualité du matériel utilisé (micro, carte son etc.…)
- Bande passante fréquentielle limitée (fréquence limitée d'une ligne téléphonique)
- Elocution inhabituelle ou altérée (stress, émotions, fatigue, etc.…)
- Certains systèmes peuvent être plus robustes que d'autres à l'une ou l'autre de ces perturbations, mais en règle générale, les systèmes de reconnaissance de la parole sont encore sensibles à ces perturbations.

II.3 Etapes intervenant dans le processus de reconnaissance

La reconnaissance automatique de la parole peut être interprétée comme une tâche de particulière de reconnaissance de formes [3]. Le principe général de la reconnaissance automatique peut être résumé par la figure 2.3.

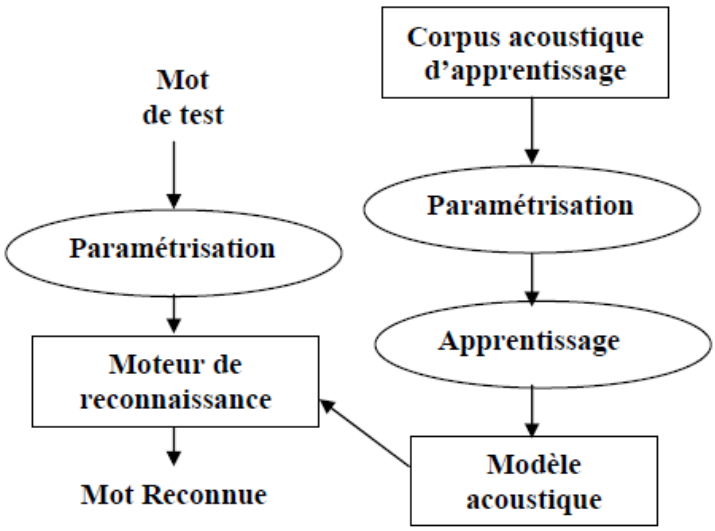


Figure 2.3 Schéma général de la reconnaissance automatique

Un système de reconnaissance est composé principalement de trois modules:

• un module de traitement acoustique

• un module d'apprentissage

• un moteur de reconnaissance

Tout d'abord, le message vocal, capté par un microphone, est converti en signal numérique. Il est ensuite analysé dans un étage d'analyse acoustique. A l'issue de cette étape, le signal est représenté par des vecteurs de coefficients pertinents pour la modélisation des mots de vocabulaire. Dans l'étape d'apprentissage, on crée un modèle de mot.

A la reconnaissance, un module de classification va mesurer la similarité entre les paramètres acoustiques du signal prononcé et les modèles des mots présents dans la base. En dernier lieu, un module de décision, basé sur une stratégie de décision donnée, fournit la réponse du système. On peut également introduire un module d'adaptation pour augmenter les performances du système de reconnaissance. En ce qui concerne le module de reconnaissance acoustique, nous présenterons les techniques de reconnaissance la plus employée à l'heure actuelle :

• La programmation dynamique

• La modélisation par modèles de Markov.

• Les approches fondées sur les réseaux de neurones

• Des approches hybrides mélangeant modèles de Markov et réseaux de neurones.

II.3.1 Un Module de Traitement Acoustique

Un système de paramétrisation du signal, appelé aussi pré-traitement acoustique, se décompose en deux blocs (figure 2.4), le premier de mise en forme (numérisation, Pré-Accentuation, Décomposition en trames et fenêtrage figure 2.5) et l’autre de calcul de coefficients.

Le signal analogique est fourni en entrée et une suite discrète de vecteurs, appelée trame acoustique est obtenue en sortie.

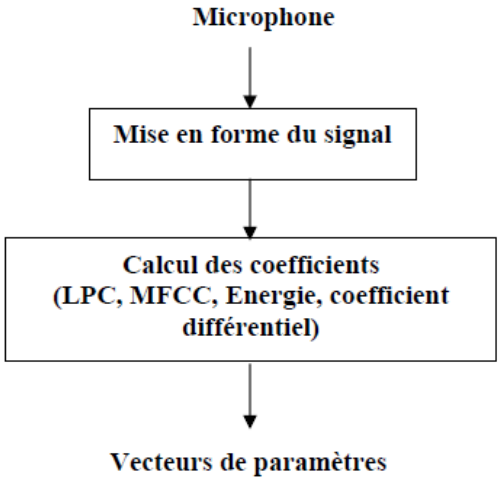


Figure 2.4 Différents étapes de traitement acoustique

II.3.1.1 Etape de mise en forme

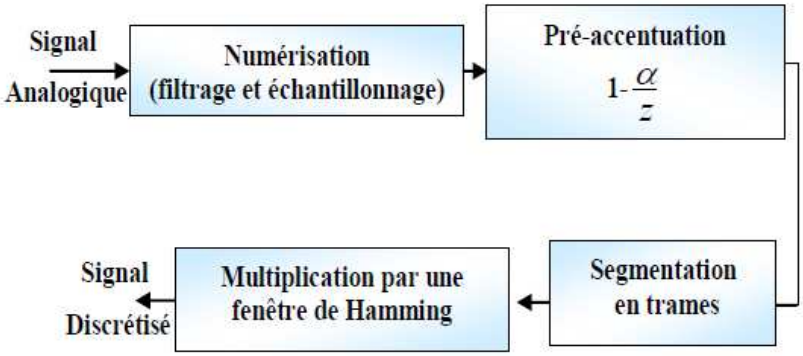


Figure 2.5 Mise en forme du signal

a) Numérisation

Pour être utilisable par un ordinateur, un signal doit tout d'abord être numérisé. Cette opération tend à transformer un phénomène temporel analogique, le signal sonore dans notre cas, en une suite d'éléments discrets, les échantillons.

Ceux-ci sont obtenus avec une carte spécialisée courante de nos jours dans les ordinateurs depuis l'avènement du multimédia. La numérisation sonore repose sur deux paramètres : la quantification et la fréquence d'échantillonnage. La quantification définit le nombre de bits sur lesquels on veut réaliser la numérisation. Elle permet de mesurer l'amplitude de l'onde sonore à chaque pas de l'échantillonnage. De plus, cette quantification peut suivre une échelle linéaire ou

logarithmique (comme l'échelle μ-law), cette dernière privilégiant la résolution de la quantification pour les niveaux faibles au détriment des niveaux forts.

Le choix de la fréquence d'échantillonnage est aussi déterminant pour la définition de la bande passante représentée dans le signal numérisé. Le théorème de Shannon nous indique que la fréquence maximale f max présente dans un signal échantillonné à une fréquence est égale à la moitié de fe. Un signal échantillonné à 16000 Hertz contient donc une bande de fréquences allant de 0 à 8000 Hertz. D'après ce principe, il est donc inutile de numériser un signal téléphonique à plus de 6800 Hertz, car le résultat ne contiendrait pas plus d'informations fréquentielles. Pourtant, comme la majorité des cartes ne proposent que certaines fréquences d'acquisition, le signal téléphonique est généralement échantillonné à une fréquence de 8000 Hz, ce qui, de plus, facilite la définition de filtres fréquentiels.

b) Détection les frontières des mots (début et fin de mot)

Comme nous sommes dans le cas de mots isolés, les frontières des mots (début et fin de mot) sont généralement déterminées en repérant les intersections de la courbe d'énergie du signal avec un ou plusieurs seuils évalués expérimentalement [3].

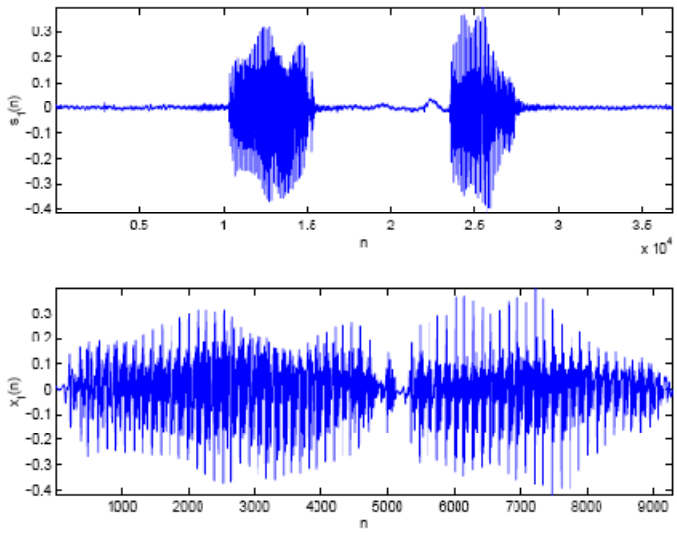


Figure 2.6 La détection du début et fin du chiffre wahid

c) Pré-Accentuation

L'étape de pré- accentuation (ou pré-emphase) consiste à accentuer les hautes fréquences. On fait généralement appel à un filtre de la forme :

$$H(Z) = 1 - z^{-1} \text{ Avec } 0.9 < a < 1.0 \qquad (2.1)$$

où a est généralement égal à 0.95 (figure 2.6).

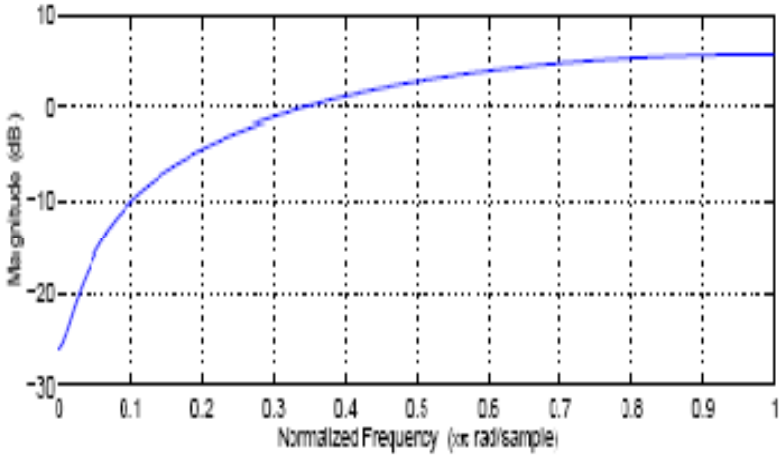


Figure 2.7 Le spectre d'un filtre de Pré-Accentuation

L'intérêt de cette pré-emphase est d'applatir le spectre du signal de parole et de filtrer la composante continue de façon à se placer dans des conditions « optimales » vis-à-vis des traitements ultérieurs, notamment le calcul d'un modèle autorégressif [3][4][6].

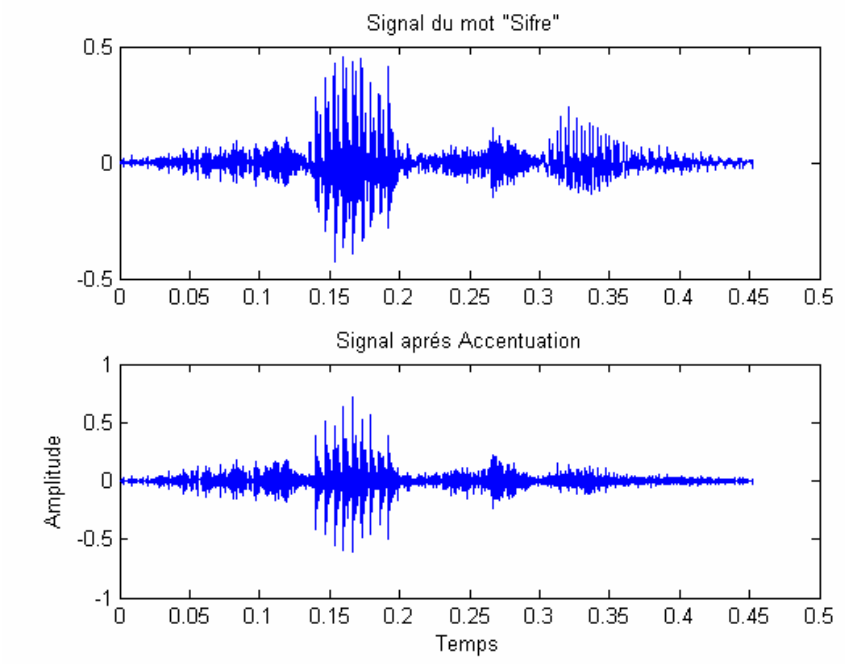


Figure 2.8 Accentuation de mot « SIFRE »

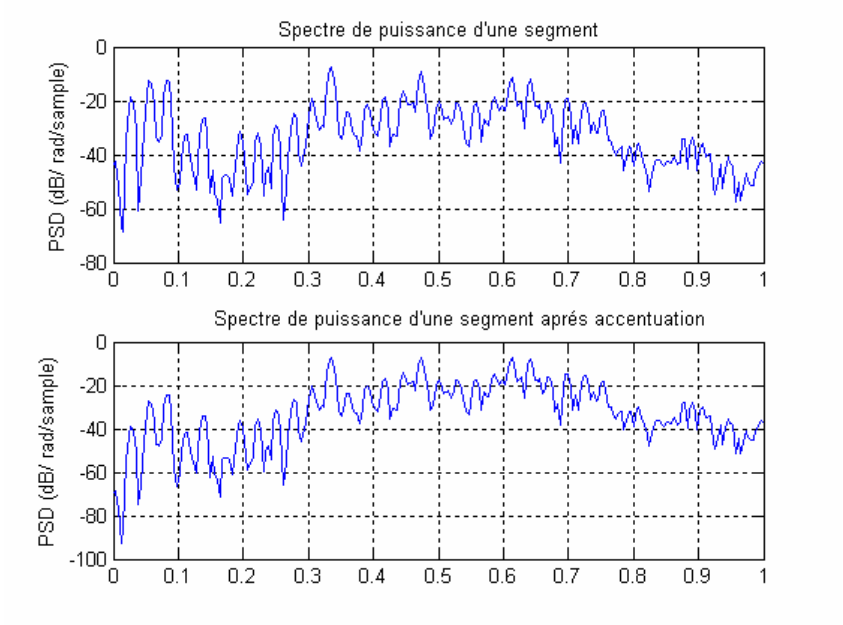


Figure 2.9 Densité spectrale de puissance d'un segment avant et après accentuation

d) Décomposition en trames et fenêtrage

Le signal de parole est ensuite décomposé en trames dont la durée est proche de 30 ms. Chaque trame correspond à une portion sur laquelle le signal de parole peut être considéré comme stationnaire.

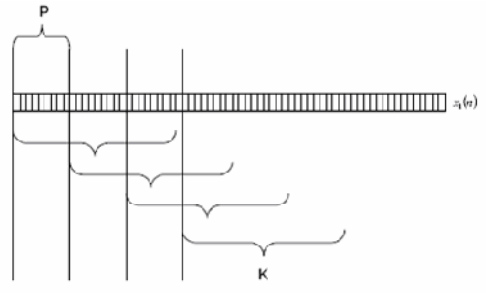


Figure 2.10 Décomposition en trames d'une séquence $x_1(n)$

Ensuite, on applique une fenêtre qui a pour fonction d'atténuer le signal au début et à la fin de chaque trame.

Le choix se porte généralement sur les fenêtres de Hanning ou de Hamming:

$$Hanning(n) = O.5 + 0.4\, cos\left(2\pi\, \frac{n}{N-1}\right) \quad (2.2)$$

$$Hanning - généralisée(n) = \alpha + (1-\alpha) cos\left(2\pi \frac{n}{N-1}\right) \quad (2.3)$$

$$Hamming(n) = O.54 + 0.46\, cos\left(2\pi\, \frac{n}{N-1}\right) \quad (2.4)$$

N étant la largeur de la fenêtre et α un paramètre. Dans le domaine spectral, ce fenêtrage permet d'atténuer les lobes secondaires associés aux différentes composantes fréquentielles du signal.

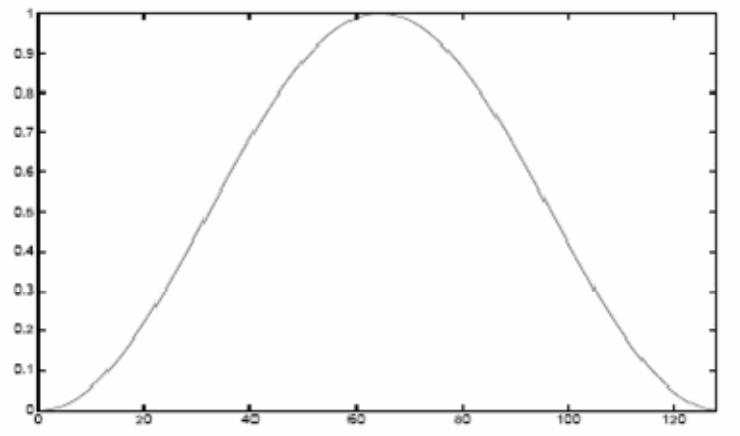


Figure 2.11 Fenêtre de Hanning sur 128 points

II.3.1.2 Etape de paramétrisation

Le problème de la reconnaissance de la parole est notamment axé sur une classification des divers sons intervenant dans la construction des mots et des phrases.

Depuis de nombreuses années, les recherches ont montré l'importance de l'enveloppe spectrale pour la classification de ces sons. Cette enveloppe spectrale fait apparaître certaines "bosses" appelées formants résultant des résonances imposées par la configuration du conduit vocal à l'instant considéré. Ces constatations ont guidé l'utilisation de représentations paramétriques du signal dans les systèmes de reconnaissance automatique de la parole.

A partir des échantillons d'une portion de signal considérée comme stationnaire, un module de traitement de signal extrait un nombre réduit de paramètres représentatifs, qui peuvent généralement être assimilés à une représentation compacte de l'enveloppe spectrale de la portion considérée. Parmi les méthodes les plus courantes, il convient de citer ici celles basées sur l'utilisation d'un banc de filtres, ainsi que celles utilisant une modélisation autorégressive du signal de parole. Ces deux types de méthodes sont parfois combinés [4].

Différents auteurs proposent également d'utiliser certains aspects du fonctionnement de l'oreille, par exemple pour définir les spécifications du banc de filtres. Il est également possible d'aller plus loin encore dans l'utilisation des propriétés physiologiques et psychoacoustiques en effectuant un traitement non linéaire à la sortie des différents filtres de façon à obtenir des paramètres représentant les impulsions transmises au cerveau par les nerfs auditifs.

Les sections suivantes dressent un aperçu sommaire des méthodes le plus utilisées. Un système de paramétrisation du signal a pour rôle de fournir et d'extraire des informations caractéristiques et pertinentes du signal pour produire une représentation moins redondante de la parole. Le signal analogique est fourni en entrée et une suite discrète de vecteurs, appelée trame acoustique est obtenue en sortie. En reconnaissance de la parole, les paramètres extraits doivent être :

 Pertinents: Extraits de mesures suffisamment fines, ils doivent être précis mais leur nombre doit rester raisonnable afin de ne pas avoir de coût de calcul trop important dans le module de décodage.

 Discriminants: Ils doivent donner une représentation caractéristique des sons de base et les rendre facilement séparables.

 Robustes: Ils ne doivent pas être trop sensibles à des variations de niveau sonore ou à un bruit de fond. La conversion du signal acoustique en séquence de vecteurs d'observation repose sur un modèle régit par un ensemble de paramètres numériques. La paramétrisation du signal de parole consiste à estimer les valeurs des paramètres du modèle permettant l'observation du signal de parole. Il existe de nombreux modèles de parole. On distingue :

 Les modèles articulatoires : Ils permettent de réaliser une simulation numérique du mécanisme de phonation. Les paramètres codent dans ce cas la position de la langue, l'ouverture des lèvres,…La paramétrisation fait intervenir des équations de mécanique des fluides.

 Les modèles de production: Ils permettent de réaliser une simulation de l'équivalent électrique de l'appareil phonatoire. Cet équivalent est en fait un modèle linéaire simplifié du modèle articulatoire. Dans ce cas, on considère le signal de parole comme étant produit par un ensemble de générateurs et de filtres numériques. Les paramètres calculés sont ceux qui contrôlent ces éléments. On trouvera dans cette catégorie, les codages LPC (*Linear Prediction Coding*) et AR (*AutoRegressive coding*).

 Les modèles phénoménologiques: Ils cherchent à modéliser le signal indépendamment de la façon dont il a été produit. Les algorithmes associés à la paramétrisation sont issus du traitement du signal. Les modèles basés sur l'analyse de Fourier en sont un exemple. Les coefficients les plus utilisés en reconnaissance de la parole sont certainement les cepstres. Ils peuvent être extraits de deux façons soit par l'analyse paramétrique, à partir d'un modèle de production de type LPC, soit par l'analyse spectrale (modèle phénoménologique).

Dans le premier cas, on parlera de LPCC (Lincar Prediction Cepstral Coefficient) et dans le deuxième de MFCC (Mel Frequency Cepstral Coefficients).

II.3.1.3 Modèle autorégressif - Analyse LPC [4][5]

Le principe du modèle autorégressif du signal de parole est de modéliser le processus phonatoire par un système de synthèse élémentaire comprenant un module d'excitation à gain variable G, suivi par un filtre tout-pôles d'ordre p (approche LPC: *''Linear Predictive Coding''*). Les coefficients du filtre sont considérés constants (hypothèse de quasi-stationnarité) pendant des intervalles de temps réduits de l'ordre de 30 ms.

L'excitation u est soit périodique (train d'impulsions, ou plus généralement signal périodique dont le spectre d'amplitude est un train d'impulsions, ce qui permet de modéliser les déphasages entre les différentes harmoniques), soit stochastique (bruit blanc), et éventuellement mixte, de façon à pouvoir modéliser les sons voisés ainsi que les sons non-voisés. Remarquons que pour le cas des sons purement voisés, l'excitation du système représentera l'action opérée par la vibration des cordes vocales, alors que le filtre représentera l'action du conduit vocal.

Pour le cas de sons partiellement non voisés par contre, le signal acoustique est le résultat d'un processus plus complexe faisant intervenir la frication, c'est à dire les perturbations créées par le passage de l'air au travers des constrictions du conduit vocal ou des lèvres. L'interprétation du modèle n'est donc plus aussi simple. Ce modèle reste cependant très utilisé en pratique car, quel que soit la nature périodique ou apériodique du signal, la fonction de transfert du filtre sera un bon

modèle de l'enveloppe spectrale du signal, caractéristique essentielle pour la distinction des sons linguistiques. Un échantillon s(n) est calculé de la sorte:

$$s(n) = \sum_{i=1}^{P} a_i \, s(n-i) + Gu(n) \qquad (2.5)$$

En effectuant la transformation en z, on obtient

$$S(z) = \sum_{i=1}^{P} a_i z^{-1} \, S(z) + GU(z) \qquad (2.6)$$

La fonction de transfert du filtre est bien évidemment exprimée par:

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{1-\sum_{i=1}^{P} a_i \, z^{-I}} \qquad (2.7)$$

et devra idéalement avoir un ordre suffisamment élevé pour modéliser avec précision la structure en formants du spectre du signal. L'ordre ne sera cependant pas trop élevé, et ce pour éviter la modélisation de détails spectraux au contenu linguistique négligeable.

On estime en général avoir besoin d'une paire de pôles par kHz de bande passante, plus 3 ou 4 pôles pour l'excitation glottique et la radiation des lèvres. Pour une fréquence d'échantillonnage de 8 kHz, on choisira donc un ordre de Il ou 12. Les expériences de reconnaissance vocale montrent que ces valeurs sont raisonnables.

Les paramètres de ce modèle, à savoir le gain, l'excitation et les coefficients ai peuvent être estimées par des méthodes d'analyse. Une interprétation de ces méthodes d'analyse est de séparer la source et la structure, et donc d'obtenir des paramètres de structure ai relativement "propres" car débarrassés de données moins importantes comme la fréquence fondamentale du son, les déphasages entre les harmoniques et les petites variations dans l'enveloppe spectrale. Ces données

sont généralement considérées comme du bruit pour la reconnaissance automatique de la parole. A partir du modèle qui vient d'être décrit, une estimation de l'échantillon s(n) peut-être calculée de la sorte:

$$\hat{s}(n) = \sum_{i=1}^{P} a_i s(n-i) \qquad (2.8)$$

L'erreur de prédiction $\hat{s}(n) - s(n)$ vaut donc :

$$s(n) - \sum_{i=1}^{P} a_i s(n-i) \qquad (2.9)$$

Une estimation des paramètres ai peut être obtenue par minimisation de la somme des carrés des erreurs de prédiction sur une trame de parole provenant des étapes de traitement précédentes, ce qui conduit à un système linéaire de p équations à p inconnues faisant intervenir la fonction de covariance du signal s. En limitant l'ordre de la somme des erreurs de prédiction par définition d'une fenêtre de signal de durée limitée, on peut montrer que les éléments intervenant dans le systèmes d'équation sont les $p + 1$ premiers éléments de la fonction d'autocorrélation du signal. De plus, la matrice du système est une matrice de Toeplitz (les éléments de toutes les diagonales sont égaux) symétrique. Cette particularité permet l'utilisation d'une méthode de résolution particulièrement efficace appelée récursion de Durbin.

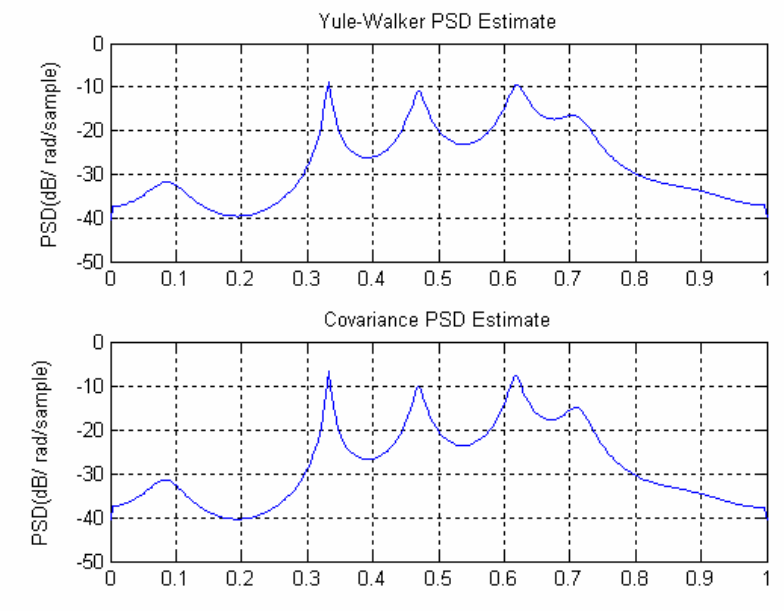


Figure 2.12 La densité Spectrale d'une trame estimé par Deux méthodes de prédiction

II.3.1.4 Analyse par banc de filtres

Sur base des trames d'analyse, il s'agit ici de calculer les énergies dans un ensemble de bandes de fréquence couvrant l'ensemble du spectre utile. Ce calcul peut être effectué dans le domaine temporel sur base de filtres définissant les différentes bandes de fréquence choisies.

Il peut également être effectué dans le domaine fréquentiel, par exemple à partir de la transformée de Fourier discrète de la trame de signal. Le nombre de filtres sera suffisamment important pour représenter avec précision l'enveloppe spectrale du signal, mais suffisamment réduit pour éviter de représenter des détails spectraux n'ayant que peu d'intérêt pour l'identification des sons linguistiques.

En pratique, le nombre de filtres est généralement inférieur à 32.

a) Divers jeux de paramètres

Sur base de la représentation issue du banc de filtres, il est possible d'effectuer une analyse par prédiction linéaire et d'en déduire divers jeux de paramètres. Il suffit en effet d'effectuer une transformée de Fourier inverse pour obtenir une représentation temporelle, et ensuite utiliser les méthodes citées à la Section 2.7.3. Le calcul de cepstres par cette méthode (Pt(4), figure 2.17) est à la base de l'analyse PLP qui permet de combiner l'intérêt d'un banc de filtres suivant une échelle non-linéaire avec le lissage opéré par le modèle autorégressif. Il est également possible de calculer directement les cepstres par transformée de Fourier inverse du logarithme de la représentation en banc de filtres. Cette méthode (Pt(6). figure 2.17) est à la base de l'approche MFCC.

La représentation issue du banc de filtres (Pt(5) figure 2.17) peut également être utilisée directement. Dans ce travail, elle a été utilisée dans le cadre de l'approche de reconnaissance multi-bande pour calculer les paramètres représentatifs des différentes bandes de fréquence. Il est également possible de combiner les avantages du banc de filtres non-linéaire avec l'analyse LPC pour obtenir une représentation de type banc de filtres (Pt(3).figure 2.17) [3].

b) Analyse MFCC (''Mel Frequency cepstral coefficients '')

Dans le cadre d'une application de reconnaissance de la parole, seule l'estimation de l'enveloppe spectrale est nécessaire [2][4].

L'extraction de coefficients MFCC est basée sur l'analyse par banc de filtres qui consiste à filtrer le signal par un ensemble de filtres passe-bande. L'énergie en sortie de chaque filtre est attribuée à sa fréquence centrale.

Pour simuler le fonctionnement du système auditif humain, les fréquences centrales sont réparties uniformément sur une échelle perceptive. Plus la fréquence centrale d'un filtre n'est élevée, plus sa bande passante est large. Cela permet d'augmenter la résolution dans les basses fréquences, zone qui contient le plus d'information utile dans le signal de parole.

Les échelles perceptives les plus utilisées sont l'échelle Mel ou l'échelle Bark Du point de vue performance des systèmes de reconnaissance de la parole, ces deux échelles sont quasiment identiques.

Dans nos expériences, nous avons fait le choix d'utiliser l'échelle Mel.

$$Mel(f) = \frac{1000}{\log(2)}\left(1 + \frac{f}{1000}\right),\ f \text{ représente la fréquence} \quad (2.11)$$

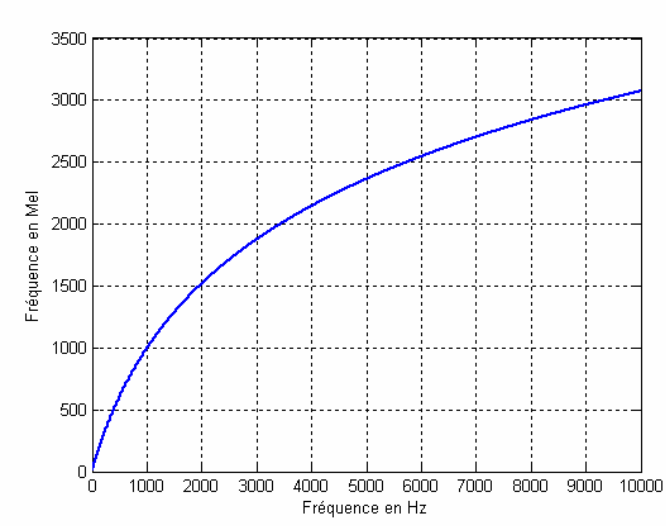


Figure 2.13 Transformation Hz en Mel

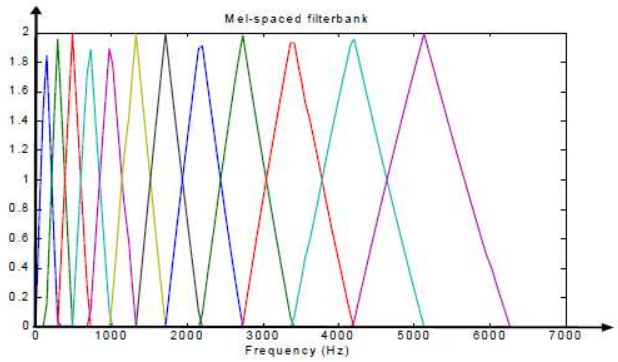


Figure 2.14 Bank de filtres triangulaires de Mel

Le nombre de filtres utilisés dans une telle analyse est choisi de manière empirique Zwicker propose 24 filtres [2].

De la même manière, on choisit empiriquement le type des filtres optimaux pour la reconnaissance de la parole. Avant tout calcul, il est nécessaire d'effectuer quelques opérations pour mettre en forme le signal de parole.

La figure 2.15 illustre l'ensemble de ces opérations. Après cette mise en forme du signal (commune à la plupart des méthodes d'analyse de la parole), une transformée de Fourier discrète (DFT Discret Fourier Transform), en particulier FFT (Transformée de Fourier Rapide Fast Fourier Transform), est appliquée pour passer dans le domaine fréquentiel et pour extraire le spectre du signal. Ensuite le filtrage est effectué en multipliant le spectre obtenu par les gabarits des filtres. Ces filtres sont en général, soit triangulaires soit sinusoïdaux.

Dans nos expériences, nous avons choisi d'utiliser des filtres triangulaires répartis sur une échelle Mel.

$$C_n = \sqrt{\frac{2}{k}} \sum_{k=i}^{N} \left(logS_k cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{k}\right]\right) \qquad (2.12)$$

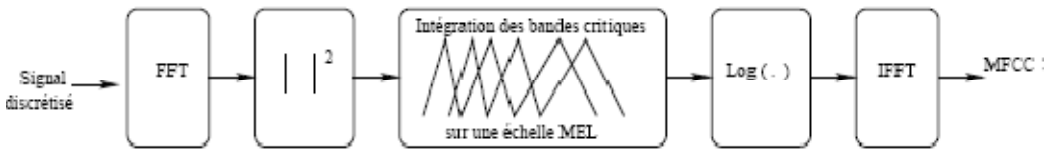


Figure 2.15 Calcul des coefficients MFCC (Mel-Scale Frequency Cepstral coefficients)

Le traitement décrit dans le paragraphe précédent permet d'obtenir une estimation de l'enveloppe spectrale (densité spectrale lissée). Il est possible d'utiliser les sorties du banc de filtres comme entrée pour le système de reconnaissance. Cependant, d'autres coefficients dérivés des sorties d'un banc de filtres, sont plus discriminants, plus robustes au bruit ambiant et moins corrélés entre eux. Il s'agit des coefficients cepstraux dérivés des sorties du banc de filtres répartis linéairement sur l'échelle Mel, ce sont les coefficient "MFCC".

Le cepstre est défini comme la transformée de Fourier inverse du logarithme de la densité spectrale.

Ceci a une interprétation du point de vue de la déconvolution homomorphique alors que le filtrage linéaire permet de séparer des composantes combinées linéairement, dans le cas de composantes combinées de façon non linéaire (multiplication ou convolution), les méthodes homomorphiques permettent de se ramener au cas linéaire. Pour le signal de parole, la source d'excitation glottique est convoluée avec la réponse impulsionnelle du conduit vocal considéré comme un filtre linéaire :

$$s(t) = e(t) * h(t) \qquad (2.13)$$

où s(t) est le signal de parole, effl est la source d'excitation glottique et h(t) est la réponse impulsionnelle du conduit vocal. L'application à l'équation précédente du logarithme du module de la transformée de Fourier donne :

$$Log\,|S(f)| = Log\,|E(f)| + Log\,|H(f)| \qquad (2.14)$$

Par une transformée de Fourier inverse on obtient :

$$s(cef) = e(cef) + h(cef) \qquad (2.15)$$

La dimension du nouveau domaine est compatible avec le temps et s'appelle la quéfrence (cef), le nouveau domaine s'appelle le domaine quéfrentiel. Un filtrage dans ce domaine s'appelle liftrage. Ce domaine est intéressant pour faire la séparation du conduit vocal et de la source d'excitation. En effet, si les contributions relevant du conduit vocal et les contributions de la source d'excitation évoluent avec des rapidités différentes dans le temps, alors il est possible de les séparer par application d'une simple fenêtre dans le domaine quéfrentiel (liftrage passe-bas pour le conduit vocal). Le conduit vocal possède une contribution fréquentielle assez lisse qui abouti à un cepstre basse-quéfrence. Réciproquement, la source possède une contribution qui varie très rapidement dans le domaine fréquentiel, son cepstre sera donc dans les hautes quéfrences.

Le domaine quéfrentiel est le domaine idéal pour séparer les deux composantes, car non seulement leur contributions sont séparées dans ce domaine, mais aussi elles sont additives [3]. Les étapes d'une analyse MFCC sont présentées dans la figure 2.12.

c) Analyse PLP (''Perceptual. Linear Prediction'')

Les fréquences centrales du banc de filtres suivent une échelle perceptuelle dont l'unité est le Bark. La fréquence en Bark B peut être obtenue par l'expression :

$$B = 6\ln\left(\frac{f}{600} + \sqrt{\left(\frac{f}{600}\right)^2 + 1}\right) \qquad (2.16)$$

où f est la fréquence en Hertz. La figure 2.13 montre que cette loi est quasi-logarithmique pour les fréquences supérieures à 1000 Hz et également que 15 bandes de fréquence de 1 Bark permettent de couvrir la plage de fréquence de 0 à 4000 Hz. Un filtrage équi-énergie est ensuite appliqué aux sorties des filtres.

Il consiste grossièrement à amplifier les sorties des filtres à haute fréquence centrale. Il s'agit d'une implémentation fréquentielle de la pré-accentuation, typiquement réalisée dans le domaine temporel (voir figure 2.17). Finalement, les valeurs obtenues sont compressées par une fonction racine cubique. Ce traitement est basé sur les conclusions d'études psychoacoustiques relatives à la perception auditive et aux caractéristiques fonctionnelles de l'oreille moyenne. Cette analyse conduit donc à une représentation en banc de filtres. Celle-ci peut être utilisée comme paramètres représentatifs ou peut servir de point de départ à une analyse plus adaptée au problème de la reconnaissance de la parole. Une transformée de Fourier discrète inverse 10 peut être appliquée aux bandes critiques, de façon à obtenir des coefficients d'auto corrélation qui seront alors utilisés de façon classique pour effectuer une analyse LPC et finalement extraire des cepstres.

L'algorithme de transformée de Fourier rapide n'est pas utilisé car le nombre de points n'est pas forcément une puissance de 2 et qu'il est de toute façon très faible. Rappelons ici l'expression de la transformée de Fourier discrète (DFT) d'une fenêtre de signal comprenant N échantillons :

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\left(-j2\pi\frac{kn}{N}\right)},\ k = 0, \ldots, N \quad (2.17)$$

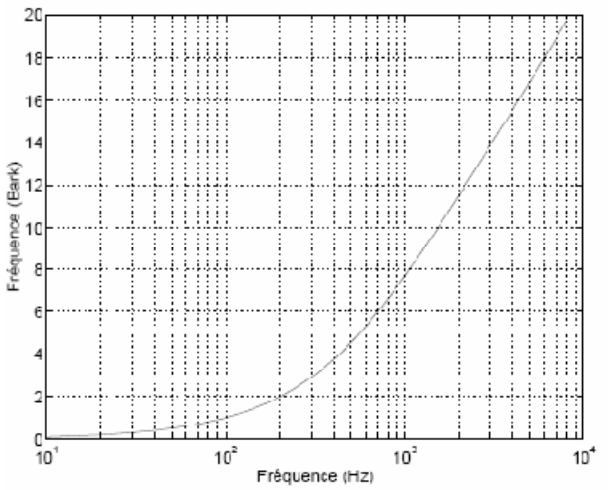


Figure 2.16 Loi Bark en fonction de la fréquence en Hz

et de la transformée de Fourier discrète inverse:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\left(-j2\pi\frac{kn}{N}\right)}, \; k = 0, \dots, N \quad (2.18)$$

$$x(n) = \frac{1}{N}\sum_{n=0}^{N-1} X(n) e^{\left(j2\pi\frac{kn}{N}\right)}, \; t = 0, \dots, N \quad (2.19)$$

On peut également appliquer la transformée de Fourier discrète inverse (DCT en pratique) aux logarithmes des racines carrées des énergies des bandes critiques il On obtient alors des paramètres qu'on peut qualifier de cepstres perceptuels. La différence avec les paramètres des paragraphes précédents est qu'aucun lissage par modélisation autorégressive n'est appliqué. On peut cependant supposer que ces paramètres sont également de bonne qualité vu qu'un lissage fréquentiel est déjà obtenu grâce au filtrage en bandes critiques.

d) Rasta PLP

La méthode PLP [3][6], dont l'algorithme repose sur des spectres à court terme de la parole, résiste difficilement aux contraintes qui peuvent lui être imposées par la réponse fréquentielle d'un canal de communication.

II.3.1.5. Paramètres dynamiques- Contexte

Le vecteur de paramètres issus des méthodes précédentes peut être complété par un vecteur correspondant aux dérivées temporelles premières et secondes de ces paramètres. Ces dérivées sont estimées sur base de plusieurs trames adjacentes [. L'approche permet d'introduire une information concernant le contexte temporel de la trame courante. Une approche plus directe consiste à utiliser plusieurs trames successives en entrée du système de reconnaissance. Cette approche est courante lorsque le système de classification est un réseau de neurones artificiels. Des expériences ont montré un optimum autour de 9 à 15 trames (décalées de 10 ms) pour plusieurs tâches différentes.

II.3.1.6. Schéma complet d'analyse du signal de parole

La figure 2.17 donne un schéma représentant les méthodes d'analyses classiques. Il fait appel aux modules décrits aux sections précédentes, auxquelles on se référera pour plus de détails et de liens vers d'autres publications. Toutes ces méthodes sont fondamentalement similaires.

Elles visent à extraire des paramètres de structure représentant l'enveloppe spectrale de courtes trames de signal.

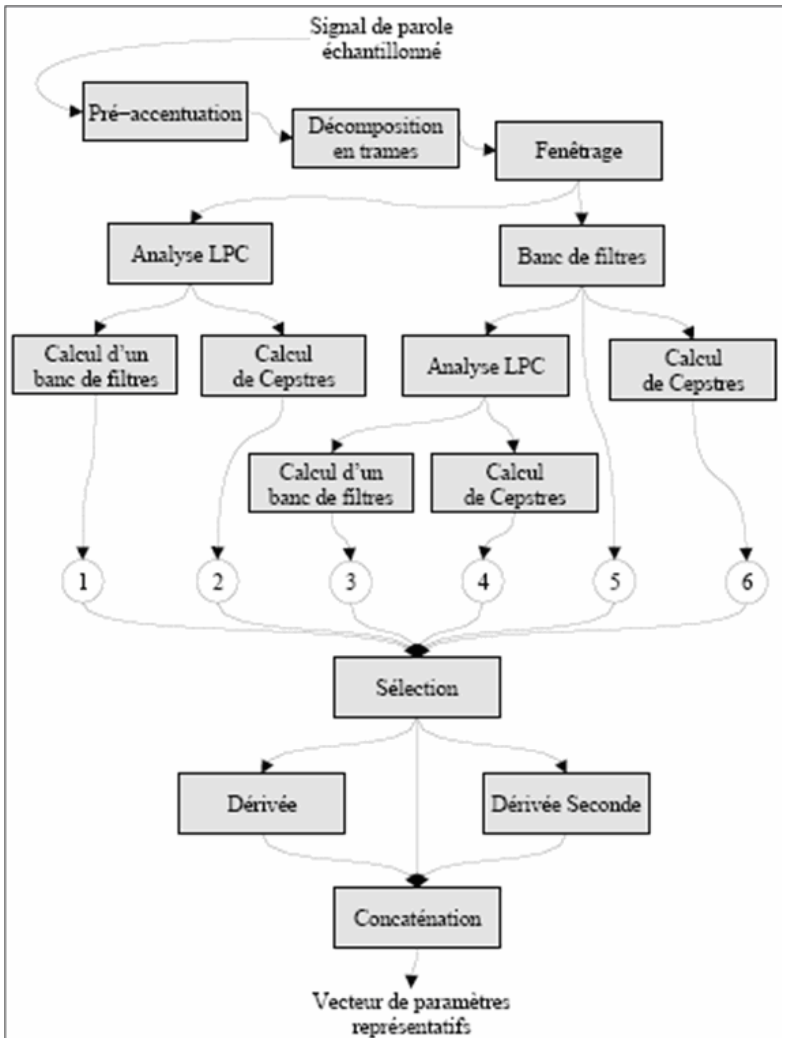


Figure 2.17 Schéma général d'analyse du signal de parole

II.3.2 Phase d'apprentissage

Pendant laquelle un ou plusieurs locuteurs prononcent une ou plusieurs fois chacun des mots de l'application prévue. Ces prononciations sont toutes prétraitées puis conservées telles quelles ou bien moyennées dans un dictionnaire de références en tant que " images acoustiques " ou sous forme d'un modèle mathématique.

II.3.3 Moteur de reconnaissance

Où le signal à reconnaître subit le même prétraitement que la phase précédente. Il est ensuite comparé aux références contenues dans le dictionnaire (image acoustique ou modèle). Le calcul d'une « distance » ou « probabilité" et sa comparaison à un seuil permet ou non de retenir la ou les références les plus proches. Mais les différences de prononciations et les variations de débit

d'élocution, parfois importantes et non linéaires imposent l'utilisation d'algorithmes de comparaison tels que la comparaison dynamique ou les chaînes de Markov. C'est une méthode bien adaptée aux applications monolocuteur, à faible vocabulaire et plutôt à mots isolés.

II.5 Conclusion

Nous avons présenté dans ce chapitre le principe de fonctionnement d’un système de reconnaissance automatique de la parole et la variabilité du signal vocale ainsi que description des applications principales de la reconnaissance vocale et de ses domaines connexes.

Chapitre III

Les Méthodes de la reconnaissance de la parole

Dans ce chapitre, nous nous intéresserons aux bases de la reconnaissance Automatique de la parole (RAP) et nous verrons quels sont les fondements théoriques des différents algorithmes utilisés. Pour ce faire, nous détaillerons la façon dont l'ordinateur traite le signal de parole par le biais de sa paramétrisation. Nous verrons quelles sont les méthodes les plus employées actuellement pour la reconnaissance acoustique du signal notamment les méthodes probabiliste(HMMs). Ce chapitre s'est attaché à présenter un bref état de l'art des différentes méthodes intervenant en reconnaissance automatique de la parole ainsi qu'une description des applications principales de la reconnaissance vocale et de ses domaines connexes.

III.1 Les système de reconnaissance de la parole

Nous trouvons deux approches de reconnaissance de parole [3]

III.1.1 La Méthode Globale

Cette méthode considère le plus souvent le mot ou le phonème comme unité de reconnaissance minimale, c'est-à-dire indécomposable. Dans cette méthode nous comparons globalement le message d'entrée (mot, phrase) aux différentes références stockées dans un dictionnaire en utilisant des algorithmes de programmation dynamique ou des modèles de Markov cachés (HMM Hidden Markov Model).

L'avantage de cette méthode est d'éviter l'explicitation des connaissances relatives aux transitions qui apparaissent entre les phonèmes. La généralisation de la méthode à des unités enchaînées présente un certain intérêt.

En effet les unités phonétiques sont représentées par des modèles et les connaissances phonétiques, lexicales et syntaxiques sont compilées dans un seul réseau, ce qui rend le système de reconnaissance très homogène, des niveaux acoustiques jusqu'aux niveaux linguistiques.

La reconnaissance consiste alors à trouver le meilleur chemin dans le réseau global pour reconnaître une phrase prononcée. Ce type de méthode est utilisé dans les systèmes suivants

- Reconnaissance de mots isolés.
- Reconnaissance d'unités enchaînées
- Reconnaissance de parole dictée avec pauses entre les mots.

III.1.2 La Méthode Analytique

Cette méthode fait intervenir un modèle phonétique de langage. Il y a plusieurs unités minimales pour la reconnaissance qui peuvent être choisies (syllabe, demi-syllabe, diphone, phonème, phone homogène, etc.).

Le choix parmi ces unités dépend des performances des méthodes de segmentation utilisées. La reconnaissance par cette méthode, passe par la segmentation du signal de la parole en unités de décision puis par l'identification de ces unités en utilisant des méthodes de reconnaissance des formes (classification statistique, réseaux de neurones, etc.) ou des méthodes d'intelligence artificielle (systèmes experts par exemple). Cette méthode est beaucoup mieux adaptée aux systèmes à grand vocabulaire et pour la parole continue.

Les problèmes qui peuvent apparaître dans ce type de système sont dus en particulier aux erreurs de segmentation (délétions, insertions, substitutions, recouvrements) et d'étiquetage phonétique. C'est pourquoi le DAP (Décodage Acoustico- Phonétique) est fondamental dans une telle approche.

III.2 Techniques statistique probabiliste pour la reconnaissance de la parole

Un calcul probabiliste à l'aide de modèles stochastiques; les plus utilisés actuellement dérivent des modèles de Markov cachés (Hidden Markov Models HMM).

Les modèles de Markov cachés (HMMs") [5][6][8] sont imposés comme la technologie prédominante en reconnaissance de la parole ces dernières années. Nous allons revoir les bases nécessaires à l'utilisation de ce type de modèle pour la reconnaissance automatique de la parole.

Ces modèles se sont avérés les mieux adaptés aux problèmes de la reconnaissance de la parole. La quasi-totalité des outils de reconnaissance de la parole disponibles actuellement sur lemarché sont basés sur cette technologie.

Un modèle de Markov caché est un automate stochastique particulier capable, après avoir été entraîné, d'estimer la probabilité qu'une séquence d'observations ait été générée par ce modèle. Idéalement, il faudrait pouvoir associer à chaque phrase possible un modèle. Il va de soi que ceci est irréalisable en pratique car le nombre de modèles serait beaucoup trop élevé.

Des sous-unités lexicales comme le mot, la syllabe, ou le phonème sont utilisées afin de réduire le nombre de paramètres à entraîner. A chacune de ces unités est associé un modèle de Markov caché constitué d'un nombre fini d'états prédéterminés.

III.2.1 Qu'est ce qu'un HMM?

Un HMM est un automate probabiliste d'états finis (figure 3.1). Il est constitué d'états (les noeuds), reliés entre eux par des transitions (les arcs). Une transition entre un état s_i et un état s_j rend possible le passage entre ces deux états.

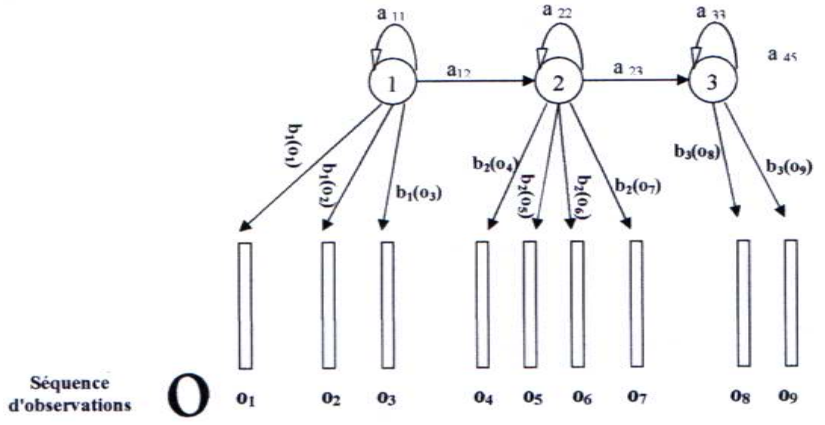


Figure 3.1 Exemple d'un HMM à trois états état associé à une observation O

A chaque instant *t*, un HMM se trouve dans un état $q_t \in S$ avec $S = \{s_1, s_2, s_3, \ldots\ldots, s_n\}$ où N est le nombre total d'états du modèle, et il émet l'observation $o_t \in v$où v est un ensemble éventuellement continu d'observations que le HMM peut potentiellement générer.

A chaque état s_i est associée une distribution de probabilité d'observations $b_j(o_t)$ et à chaque transition d'un état s_i vers un autre état est associée une probabilité de transition $p(q_t = s_i/q_1 q_2 q_3 \ldots q_{t-1})$ la distribution de probabilité de générer l'observation o_t à l'état s_i au temps t; la probabilité de transition représente la probabilité de passer d'un état vers un autre état sachant la suite d'états précédents. Cette probabilité constitue une sorte d'historique des états par lesquels le HMM est passé.

La longueur n de cet historique $q_1 q_2 q_3 \ldots\ldots q_n$ détermine l'ordre du HMM. Un HMM d'ordre 1 ne conservera ainsi que l'état précédemment visité à l'instant t-1. Dans la figure 1.3, cette probabilité de transition se réduit à $a_{j1} = p(q_t = s_i/q_{t-1} = s_j)$ où seul l'état visité au temps t-1 estmémorisé.

A l'issue de ce processus, ce modèle permet ainsi de générer une séquence de T observation $O = (o_1, o_2, o_3, \ldots\ldots o_T)$. Seule la séquence d'observations *O* est connue, la séquence d'états $Q = (q_1, q_2, q_3, \ldots\ldots q_T)$ ayant permis de la générer restant inconnue, ce qui explique le substantif "caché" des modèles de Markov.

III.2.2 Eléments d'un modèle de Markov caché

Un modèle de Markov caché noté $\lambda = (S, v, A, B, \pi)$ est défini par :

- Ses états, en nombre n, qui composent l'ensemble S = $\{s_1, s_2, \ldots s_n\}$. L'état où se trouve le HMM à l'instant t est noté q_t ($q_t \in S$);
- ν est l'ensemble discret ou continu des observations qu'un HMM peut générer. o_t désigne un élément observé à l'instant t ;
- Une matrice A de probabilités de transition entre les états de la chaîne :

$$a_{ij} = A(i,j) = P(q_{t+1}|q_t = s_i) \quad (3.1)$$

est la matrice des probabilités de transition sur l'ensemble des états du modèle. La probabilité de transition est la probabilité de choisir la transition $a_{Bij}B$ pour accéder à l'état q_j étant donné un processus à l'état q_i . Pour un HMM d'ordre un, cette probabilité ne dépend que de l'état précédent:

$$\forall t, k: P(q_t = s_j | q_{t-1} = s_i, q_{t-2} = s_k) = P(q_t = s_j | q_{t-1} = s_i) \quad (3.2)$$

Elle dépend des deux précédents dans le cas d'un HMM d'ordre deux :

$$\forall t, k: P(q_t = s_j | q_{t-1} = s_i, q_{t-2} = s_k, \ldots.) = P(q_t = s_j | q_{t-1} = s_i, q_{t-2} = s_k) \quad (3.3)$$

En d'autres termes, l'évolution du système entre deux instants t - 1 et t ne dépend que de l'État de ce système au temps t-1 (ordre 1) ou des deux instants précédents t-1 et t-2 (ordre deux).

- Une matrice B de probabilités d'observation : $b_j(o_t)$ est la probabilité d'observer le o_t quand le modèle se trouve dans l'état j, soit :

$$b_j(o_t) = P(O_t | q_t = s_j) \quad 1 \leq j \leq N \quad (3.4)$$

La probabilité d'observation $b_j(o_t)$ a_{ij} a les propriétés suivantes :

$$b_j(o_t) \geq 0 \quad \forall t, k \quad (3.5)$$

$$\sum_{k=1}^{M} b_t(o_t) = 1$$

La forme que prend cette distribution détermine le type du HMM. C'est ainsi qu'on parle de HMMs discrets, semi-continus, continus, etc.

- Un vecteur π de densités de probabilité initiale :

$\pi = \{\pi_i\}_{i=1,2,\ldots,n}, \pi_i$ Un ensemble de densités de probabilités initiales, représente la probabilité que l'état de départ de modèle soit l'état s_i, soit :

$$\pi_i = P(q_1 = s_i) \quad 1 \leq i \leq n \tag{3.6}$$

Avec : $$\pi_i \geq 0 \ \ \forall i \tag{3.7}$$

$$\sum_{i=1}^{n} \pi_i = 1$$

III.2.3 Propriétés des HMMs utilisées en RAP

Pour réduire le temps de calcul, des hypothèses simplificatrices sont communément émises dans le cadre de la RAP en ce qui concerne les propriétés des modèles de Markov cachés. Ainsi

- Un modèle de Markov est stationnaire, si bien que :

$$\forall t, k: \ P\left(q_t = s_i \middle| q_{t-1} = s_j\right) = P\left(q_{t+k} = s_i \middle| q_{t+k-1} = s_j\right) \tag{3.8}$$

- Les observations sont considérées comme indépendantes, c'est-à-dire que :

$$P(o_t / q_1 q_2 \ldots . q_t \, , o_1 o_2 \ldots . o_{t-1}) = P(o_t / q_1 q_2 \ldots . q_t) \tag{3.9}$$

- La probabilité d'émission d'une observation ne dépend que de l'état courant. Ainsi

$$P(o_t / q_1 q_2 q_3 \ldots . q_t) = p(o_t / q_t) \tag{3.10}$$

On cherchera à construire des HMM pour lesquels les suites observables sont les exemples que l'on cherche à modéliser. Les séquences observées sont définies comme des phrases, mot … etc.

En outre, dans la plupart des systèmes de reconnaissance automatique de la parole actuels, les modèles acoustiques sont représentés par des HMMs d'ordre 1, c'est-à-dire que la probabilité d'être dans un état donné si à l'instant t, en sachant que t - 1 états ont été visités, est égal à la probabilité d'être dans l'état s_i en ne considérant que l'état s_j précédemment visité. En d'autre termes :

$$P(q_t = s_i / q_1 q_2 \ldots . q_{t-1} = s_j) = p(q_t = s_i / q_{t-1} = s_j) \tag{3.11}$$

Dans ce cas, on peut noter $a_{ij} = \mathrm{P}(\mathrm{q_t} = \mathrm{s_i} | \mathrm{q_{t-1}} = \mathrm{s_j})$ la probabilité de passer de l'état s_j à l'état s_i, avec ∀i, j = 1,2,....N . L'ensemble des probabilités de L'ensemble des probabilités de transition du HMM est alors caractérisé par la matrice N x N suivante :

$$\begin{bmatrix} a_{11}, a_{12}, \dots\dots\dots\dots\dots & a_{1N} \\ a_{21}, a_{22}, \dots\dots\dots\dots\dots & a_{2N} \\ a_{31}, a_{32}, \dots\dots\dots\dots\dots & a_{3N} \\ \vdots & \\ a_{N1}, a_{N2}, \dots\dots\dots\dots\dots & a_{NN} \end{bmatrix} \tag{3.12}$$

Avec

$$a_{ti} \geq 0 \ \ \forall i, j \tag{3.13}$$

$$\sum_{j=1}^{n} a_{ij} = 1 \ \ \forall j$$

Enfin, afin de mieux modéliser la variabilité du signal, la fonction de densité de probabilités d'observation associé à un état s_i est traditionnellement représenté par :

- un mélange de **M** Gaussiennes
- une densité de probabilité discrète au moyen de la quantification vectorielle

III.2.4 Densité d'observation discrète par quantification vectorielle

Le principe de l'estimateur de fonction de probabilité discret par quantification vectorielle est de partitionner l'espace de représentation en associant à chaque zone un représentant. Celui-ci est soit le centre de gravité de la partition (*centroïde* (la moyenne des éléments de la partition), soit le vecteur le plus proche des autres (prototype). L'habitude veut que l'on parle dans les deux cas de prototypes.

L'ensemble des prototypes constitue le dictionnaire (*codebook*) de l'espace de représentation discrétisé. Divers algorithmes appartenant à la classification automatique non supervisée ont été proposés pour la construction du dictionnaire des prototypes.

III.2.5 Densité d'observation continue

Nous avons voir le cas où les observations prennent des valeurs dans un alphabet fini discret et nous pouvions donc utiliser une loi de probabilité discrète dans chaque état du modèle. Une telle approche n'est pas compatible avec des observations qui sont des signaux continus.

Bien sûr, quantifier le signal pourrait permettre de résoudre le problème, mais cela ne pourrait entraîner que des dégradations. Il est donc préférable d'utiliser des modèles de Markov cachés avec des densités d'observation continues.

La représentation la plus générale de la fonction des densités de probabilités d'observation associé à un état s_i est traditionnellement représenté par un mélange de M fonction élémentaire(mixture), c'est-à-dire que:

$$b_i(o_t) = \sum_{K=1}^{M} c_{i,k} b_{jk}(o_t), i = 1,2, \ldots . N \qquad (3.14)$$

Avec M est le nombre de mixture et la contrainte imposé sur les coefficients de pondération $c_{i,k}$ est :

$$c_{i,k} \geq 0 \qquad i = 1,2, \ldots N \qquad (3.15)$$

$$\sum_{k=1}^{M} c_{i,k} = 1 \qquad i = 1,2, \ldots M$$

Ou $b_{ik}(o_t)$ est une densité de probabilité de dimension D comme une fonction elliptique de vecteur moyenne μ_{ik} et matrice de covariance $\sum ik$

$$b_{ik}(o_t) = \emptyset(o_t, \mu_{ik}, \sum ik) \qquad (3.16)$$

Dans la plus part des cas la densité de probabilité élémentaire est une fonction gaussienne :

$$b_{ik}(o_t) = \emptyset(o_t, \mu_{ik}, \sum ik) = \frac{1}{(2\pi)^{D/2} |\sum ik|^{1/2}} e^{\frac{1}{2}(o_t - \mu_{ik})^T \sum_{IK}^{-1} (o_t - \mu_{ik})} \qquad (3.17)$$

L'hypothèse d'une indépendance entre les D dimensions de l'espace de représentation autorise l'utilisation de matrices de covariance diagonales ; ce qui limite le nombre de paramètres à estimer et simplifie les calculs.

$$\emptyset(o_t, \mu_{ik}, \sum ik) = \frac{1}{(2\pi)^{D/2} \left| \prod_t^D \sigma_{ikl} \right|^{1/2}} e^{-\sum_l^D \frac{(o_t - \mu_{ikl})}{2\sigma_{ikl}^2}} \qquad (3.18)$$

Le principal inconvénient de l'estimateur continu est qu'il repose sur l'hypothèse que la fonction élémentaire utilisée est cohérente avec la loi réelle des données. Or, les vecteurs d'analyse n'ont pas, en général, de distribution gaussienne [Barras, 1996; Montacié, Caraty et al., 1996]. L'existence d'une somme particulière de fonctions qui converge vers la loi réelle est montrée

théoriquement mais pas la manière de l'obtenir. Ainsi, le choix du nombre de fonctions élémentaires dans la somme pondérée, tout comme l'apprentissage leurs paramètres, est guidé par des heuristiques et ne permet pas, en général, d'assurer une convergence vers la loi réelle.

III.2.6 Les trois problèmes des HMM

Il y a trois problèmes à résoudre pour que la théorie des HMM puisse donner naissance à des algorithmes :

- **Problème 1**

L'évaluation de la probabilité de l'observation d'une séquence. Etant donné la suite d'observations O et un HMM λ , comment évaluer la probabilité d'observation $P(O|\lambda)$?

- **Problème 2**

La recherche du chemin le plus probable (décodage), ou estimation de la partie cachée, ou encore décision. Soit la suite d'observations O et un modèle λ , comment trouver une suite d'états $Q = q_1, q_2, \dots q_T$ qui soit optimale selon un certain critère ?

- **Problème 3**

L'apprentissage. Comment ajuster les paramètres du modèle λ pour maximiser $P(O|\lambda)$, à partir de séquences d'apprentissage dont on sait qu'elles ont été émises par ce modèle ?

III.2.6.1 Solution au problème 1 « évaluation de probabilité»

Il s'agit de calculer La probabilité de la suite d'observations O, étant donné le modèle λ, elle est égale à la somme sur tous les chemins d'états possibles Q des probabilités conjointes de O et de Q:

$$P(O|\lambda) = \sum_Q P(O, Q|\lambda) = \sum P(O|Q, \lambda)\, P(Q|\lambda) \qquad (3.19)$$

Or, on a les relations :

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots . a_{q_{T-1} q_T}$$

$$P(O|Q, \lambda) = b_{q_1}(O_1) b_{q_2}(O_2) \dots b_{q_T}(O_T) \qquad (3.20)$$

D'où :

$$P(O|\lambda) = \sum_{q_1,q_2,\dots,q_T} \pi_{q_1}\, b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \dots a_{q_{T-1} q_T} b_{q_T}(O_T) \qquad (3.21)$$

Cette formule directe nécessite $n^T - 1$ additions et $(2T-1)n^T$ multiplications (n^T étant le nombre de chemins possibles de longueur T), soit $2Tn^T$ opérations.

a) Evaluation Par Les Fonctions Forward-Backward.

On considère que l'observation peut se faire en deux temps : d'abord, émission du début de l'observation O(1: t) en aboutissant à l'état i q au temps t, puis, émission de la fin de l'observation O(t +1:T) sachant que l'on part de q_i au temps t.

Dans ce cas, l'évaluation de l'observation est égale à :

$$P(O|\lambda) = \sum_{q_i} \alpha(t, q_i)\, \beta(t, q_i) \qquad (3.22)$$

Où α (*t, q_i*) est la probabilité d'émettre le début O (1: *t*) et d'aboutir à q_i à l'instant t (α pour t croissant), et β (*t, q_i*) est la probabilité d'émettre la fin O (*t* +1:*T*) sachant que l'on part de q_i à l'instant t (β pour t décroissant).

a.1) Procédure récursive directe « Forward »

On notera désormais α (*t, q_i*) par α_t (*i*):

$$\alpha_t(i) = P(O_1 O_2 \dots O_t, q_t = s_i|\lambda) \qquad (3.23)$$

α_t (*i*) est calculée de manière récursive comme suit:

- Initialisation

$$\alpha_t(i) = \pi_i b_i(O_1) \quad 1 \le i \le n; \qquad (3.24)$$

- Récurrence

$$\alpha_{t-1}(j) = \left[\sum_{i=1}^{n} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}) \quad 1 \le t \le T-1,\ 1 \le j \le n; \qquad (3.25)$$

- Terminaison

$$P(O|\lambda) = \sum_{i=1}^{n} \alpha_T(i) \qquad (3.26)$$

Ce calcul est basé sur le fait que pour émettre le début de l'observation O(1: t +1) et aboutir dans l'état si au temps t + 1, on doit nécessairement être dans l'un des états si à l'instant t.

Ce calcul nécessite n + n(n +1) (T −1) multiplications et (n −1)n(T −1) additions, soit une complexité enO (n^2 T).

a.2) Procédure récursive inverse « Backward »

β (t, q_i) sera noté β_t (i) dans la suite :

$$\beta_t(i) = P(O_{t+1}O_{t+2} \dots O_t | q_T = s_i, \lambda) \qquad (3.27)$$

On déduit β_t de β_{1+t} par

- Initialisation

$$\beta_T(i) = 1 \quad 1 \le i \le n; \qquad (3.28)$$

- Récurrence

$$\beta_t(i) = \sum_{j=1}^{n} a_{ij}\, b_j(O_{t+1})\beta_{t+1}(j) \quad T-1 \ge t \ge 1, \ 1 \le i \le n; \qquad (3.29)$$

Le calcul de β est aussi enO(n^2 T).

La probabilité d'observation est obtenue en prenant les valeurs de α et de β à un instant t quelconque

$$P(\mathrm{O}|\lambda) = \sum_{q_t} \alpha_t(i)\, \beta_t(i) \qquad (3.30)$$

Cependant, on utilise le plus souvent les valeurs obtenues pour deux cas particuliers (t = 0) ou (t = T), ce qui donne :

$$P(\mathrm{O}|\lambda) = \sum_{q_t} \alpha_t(i) = \sum_{q_t} \pi_i\ \beta_0(i) \qquad (3.31)$$

Ou bien par la relation

$$P(\mathrm{O}|\lambda) = \sum_{\mathrm{i}=1}^{\mathrm{N}} \alpha_{\mathrm{T}}(\mathrm{i}) \qquad (3.32)$$

III.2.6.2 Solution au problème 2 : « Décodage »

Il s'agit de trouver dans le modèle la suite d'états qui maximise :

$$P(\mathrm{Q}|\mathrm{O},\lambda) \Leftrightarrow P(\mathrm{Q}\,,\mathrm{O}|\lambda) \qquad (3.32)$$

Pour trouver le meilleur chemin $Q = (q_1,q_2,...,q_T)$ pour une suite d'observations $O=(O_1,O_2,...,O_T)$, on définit $\delta_t(i)$ qui est la probabilité du meilleur chemin amenant à l'état s_i à l'instant t, en étant guidé par les t premières observations :

$$\delta_t(i) = max_{q_1,\dots,q_{t-1}} P(q_1, q_2, \dots, q_t = s_i, O_1, O_2, \dots, O_t|\lambda) \qquad (3.33)$$

Par récurrence, on calcule

$$\delta_{t+1}(j) = max_i\left[\delta_t(i)a_{ij}\right]b_j(O_{t+1}) \qquad (3.34)$$

On garde la trace de la suite d'états qui donne le meilleur chemin amenant à l'état s_i à t dans un tableau ψ.

a) Algorithme de Viterbi

- Initialisation

$$\delta_1(i) = \pi_i b_i(O_1) \quad 1 \le i \le n,$$

$$\Psi_1(i) = 0\,; \qquad (3.35)$$

- Induction

$$\delta_1(i) = max_{1\le i\le n}\left[\delta_{t-1}(i)a_{ij}\right]b_j(O_t) \quad 1 \le j \le n,$$

$$\Psi_1(i) = arg\ max_{1\le i\le n}\left[\delta_{t-1}(i)a_{ij}\right] \quad 1 \le j \le n, \qquad (3.36)$$

- Terminaison

$$P^* = max_{1\le i\le n}[\delta_T(i)],$$

$$q_T^* = arg\ max_{1\le i\le n}[\delta_T(i)]; \qquad (3.37)$$

- Chemin obtenu « Backtracking »

$$q_t^* = \Psi_{t+1}(q_{t+1}^*) \quad T-1 \geq t \geq 1. \qquad (3.38)$$

La fonction "argmax" permet de mémoriser l'indice i avec lequel la valeur $\delta_{t-1}(i).a_{ij}$ est maximale et la complexité de cet algorithme est $O(N^2T)$.

III.2.6.3 Solution au problème 3 : « Apprentissage »

Le but de l'apprentissage est de déterminer les paramètres (A, B, π) qui maximisent la Probabilité de la suite d'observations $P\ (O/\lambda)$.
L'idée employée ici est d'utiliser des procédures de ré-estimation par punition -récompense :

- Choisir un ensemble initial de paramètres λ_0;
- Calculer λ_n à partir de λ_{n-1};
- Répéter ce processus jusqu'à un critère de fin.

Partant de λ_n, λ_{n+1}doit vérifier :

$$\prod_r P(O^r \mid \lambda_{n+1}) \geq \prod_r P(O^r \mid \lambda_n)$$

Ceci montre que λ_{n+1} doit améliorer la probabilité d'observations, ce qui revient à définir une fonction F telle que :

$$\lambda_{n+1} = F(\lambda_n)$$

L'approche la plus simple pour définir F consiste à faire des statistiques sur l'utilisation des transitions et des distributions. Ceci revient à calculer des fréquences d'utilisation à partir de l'ensemble d'apprentissage. Si l'ensemble est important, ces fréquences fournissent une bonne approximation des probabilités a posteriori utilisables alors comme paramètres du modèle pour l'itération suivante.

La méthode d'apprentissage va donc consister à partir d'un modèle initial aléatoire, à estimer ses paramètres comme indiqué ci-dessus, puis à recommencer cette estimation (”réestimation”) jusqu'à obtenir une certaine convergence. Les paramètres à déterminer différent selon que la distribution des observations soit continue ou discrète, on distingue :

- Pour le cas discret, $\lambda = (A, B, \pi)$: On estime les a_{ij}, les $b_j(k)$ et les π_i.

- Pour le cas continue, $\lambda = (A, (C_{jk}, \mu_{jk}, \sum_{jk}), \pi)$, On estime les a_{ij}, les $\pi_i$et les paramètres de chaque Gaussienne [3].

III.2.7 Les différentes structures du modèle de Markov caché

Il existe deux principaux types des HMMs selon les transitions entre les états de la chaîne de Markov : le modèle ergodique et le modèle gauche-droite.

- **Le modèle ergodique:**

C'est un modèle dit sans contraintes où toutes les transitions d'un état vers un autre sont permises ; c'est à dire que tous les états peuvent être atteints de n'importe quel état de départ. La figure suivante présente un exemple de ce type :

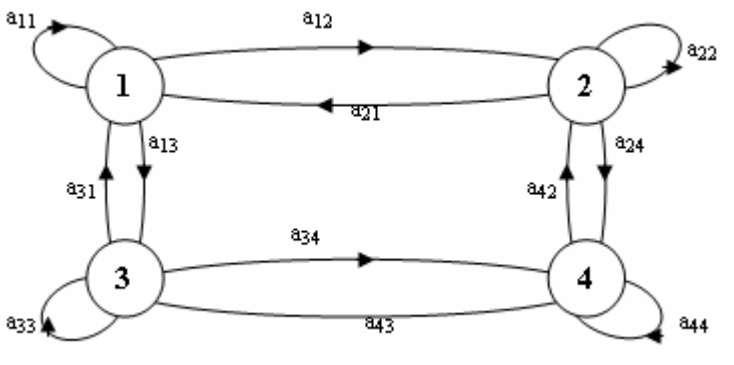


Figure 3.2 Le modèle ergodique

- **Le modèle gauche-droite**

Une des éléments qui influence également sur la qualité de reconnaissance quand on utilise le modèle de Markov Caché est leur structure. En effet, dans la plupart des applications ainsi que des études qui portent sur la reconnaissance de la parole, en raison de la propriété de la parole, on utilise souvent le modèle Gauche-Droite, celui dont la matrice de transition A possède la contrainte suivante:

$$aij=0 \quad \text{Pour } j<i$$

Dans ce type, en distingue le modèle parallèle et le modèle séquentiel :

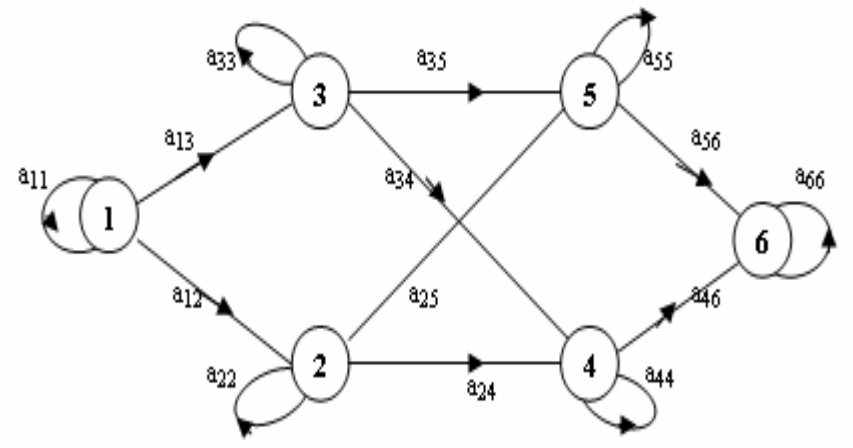


Figure 3.3 Le modèle parallèle

Voici le schéma du model séquentiel

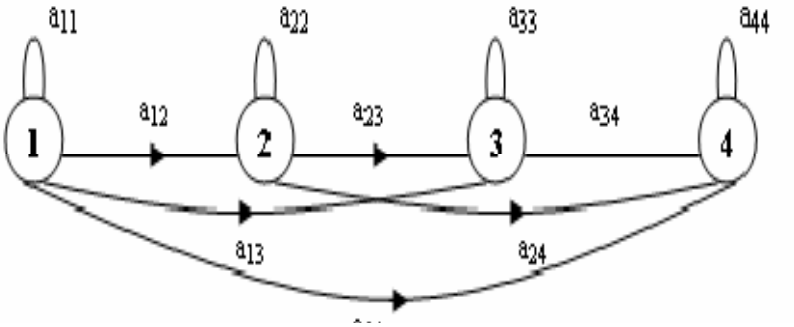


Figure 3.4 Le modèle sequential

III.3 D'autre techniques pour la reconnaissance de parole

III.3.1 La comparaison dynamique

Compte tenu de la forte variabilité inter-locuteur et même intra-locuteur, les prononciations d'un même mot se réalisent acoustiquement de manière fort différente entre autres, on observe une distorsion temporelle qui implique que les échelles temporelles des deux occurrences du même mot ne coïncident pas. On ne peut pas donc comparer point â point les formes acoustiques, et il est nécessaire de procéder à un alignement dit temporel.

Cette comparaison s'effectue par programmation dynamique. Elle est fondée sur les travaux de R. Bellman en 1957 pour la recherche de la trajectoire optimale. Pour formuler un système de pondération simple, celle-ci a été reprise par Sakoe et Shiba.

Dans un système de reconnaissance de mots isolés, fondé sur cette méthode, un ensemble de références est constitué lors de la phase d'apprentissage; pour chaque mot de vocabulaire, est acquise une (ou plusieurs) prononciation, chacune d'elles est paramétrée et donne naissance à une référence ou suite de vecteurs acoustiques R. Lors de la phase de reconnaissance, pour chaque prononciation à reconnaître T on calcule toutes les distances entre l'observation T et les références R par comparaison dynamique. Le mot reconnu est celui qui correspond à la référence pour laquelle la distance à T est de coût minimal.

III.3.2 Le réseaux de neurone [17] [22]

L'une des alternatives à l'utilisation d'HMMs en reconnaissance est le recours à des réseaux neuronaux. Un réseau de neurones est une interconnexion de cellules simples (neurones). Chaque celIule possède plusieurs entrées et une sortie. Le signal de sortie peut être la somme pondérée (éventuellement seuillée) des signaux collectés en entrée [9].

L'utilisation de ces réseaux est largement répandue dans les domaines devant résoudre des problèmes de classification et de reconnaissance des formes (traitement d'image, de signature sonar, ...).

Les réseaux de neurones (ANN, Artificial Neural Network) possèdent des propriétés très appréciées en RAP. Leur apprentissage est discriminant (ils permettent d'améliorer la reconnaissance d'une classe et simultanément de rejeter les autres classes). ils ne nécessitent pas d'hypothèses sur les propriétés statistiques des données en entrée(contrairement aux HMMs qui les modélisent par des PDFs).

Dans le cas des ANNs appliqués à la reconnaissance de la parole (mot ou tout autre unité acoustique), on utilisera le plus souvent des perceptrons multicouches. Plus généralement, on combinera le perceptron avec un algorithme d'alignement de type DTW, les distances locales utilisées lors de la DTW étant les sorties de FANN . En plus de leur utilisation dans le problème de reconnaissance, les ANNs peuvent aussi servir à prétraiter le signal de parole et à extraire des paramètres discriminants. En effet, les coefficients de pondération des couches cachés d'un ANN forment une série de paramètres caractérisant l'entrée. L'architecture d'un réseau de neurones à retard (TDNN, Time-Delay Neural Network) est décrite dans [27].

La particularité d'un neurone de TDNN réside dans le fait que ses entrés à un instant sont constitués de données issues de l'instant présent mais aussi du passé et du futur. L'objectif est d'intégré des schémas temporels dans l'ensemble des données que doit généraliser le réseau de

neurones. Un tel réseau combine la robustesse et le pouvoir discriminant des réseaux de neurones avec une architecture invariante par rapport au temps afin de former un identifieur de phonème très performant.

III.3.3 Les systèmes hybride [10] [11]

L'hybridation est souvent associée à l'utilisation des réseaux de neurones artificiels ou à des HMM pour faciliter la construction ou l'apprentissage de l'autre méthode. Cependant, cette hybridation peut également être vue comme une combinaison des deux méthodes en tirant avantage des forces de chacune. Le réseau de neurones et sa capacité d'avoir une vue générale de la séquence peut servir comme prétraitement d'une séquence, tandis que le HMM et sa capacité de structuration locale de l'information permet de raffiner la prédiction. On pourrait ici parler plutôt de méthode hybride ou d'hybridation de méthodes, au lieu d'architecture hybride, car l'hybridation est dans la cascade d'analyse des données, c'est-à-dire dans l'utilisation combinée de méthodes différentes [22] [26].

III.4 APPLICATIONS

Dans le domaine de la reconnaissance de la parole, on distingue trois grands types d'applications [3];

- les systèmes de commande vocale
- les systèmes de compréhension
- les machines à dicter

III.4.1 Les Commandes Vocale

Il existe aujourd'hui un nombre important de produits fiables sur le marché qui permettent de contrôler l'environnement au moyen d'une entrée vocale.

Les applications multiples et variées vont du jouet gadget à l'outil de travail sophistiqué. Voilà quelques exemples ;

- Commandes de la voiture
- Jeux vidéo

- Noms de gares SNCF

- Aides aux handicapées ces applications sont soit du type aide aux personnes ayant un organe défaillant hors la voix (aveugle, problèmes moteurs etc..), soit du type outil de rééducation pour les malentendants.

- Reconnaissance des chiffres comme le projet cabine vocale (pour composer un numéro de téléphone il suffit de prononcer la suite des chiffres).

Pour ces applications la taille du vocabulaire est limitée (elle ne dépasse pas quelques centaines de mots). Bien qu'une souplesse soit donnée à l'utilisateur quant au choix du vocabulaire, il est recommandé de choisir des mots contrastés pour réduire le risque d'ambiguïté. Ces systèmes sont d'autant plus performants (jusqu'à 99Vc de taux de reconnaissance) que les mots sont bien différentiables par leur longueur ou leur transcription phonétique. Certains de ces systèmes sont multilocuteurs, et ne nécessitent pas d'apprentissage préalable.

III.4.2 Les Systèmes De Compréhension

Les systèmes de compréhension se caractérisent par un vocabulaire limité et par une sémantique fermée.

Les applications généralement choisies sont de type par exemple comme l'interrogation d'une base de données, les standards téléphoniques automatisés qui donnent des renseignements météo, ou même permettent de réserver des places. Ces systèmes sont connectés à des modules d'interprétation de message reconnu, dont le but est de réagir soit par l'émission d'une réponse vocale soit par une action mécanique sur l'environnement après prise de décision. De ce fait, la performance de ces systèmes doit être jugée sur la base du nombre de phrases reconnues. Beaucoup de grands systèmes ont été conçus autour du projet ARPA lancé en 1977 aux USA. Aujourd'hui, plusieurs laboratoires travaillent sur le projet DARPA (CMU, MIT, BBN, SRI .). Par exemple, le système SPHINX développé au CMU, suppose l'emploi d'un petit vocabulaire (les milles mots du corpus "Ressources Management Data Base"). Il a par contre le mérite de permettre aux locuteurs une parole continue et ne nécessite pas d'apprentissage préalable. Le taux de reconnaissance sur les mots est de l'ordre de 96%.

III.4.3 Les Systèmes De Dictée Vocale

Les machines à dicter ont pour but de retranscrire un texte dicté par un locuteur devant un microphone aussi bien qu'une secrétaire, c'est à dire, en respectant au mieux les règles d'usage et

d'accord orthographique propres à la langue utilisée. La compréhension des phrases n'est nullement requise.

On peut remarquer que ce domaine occupe un lieu important, à la frontière de l'oral et de l'écrit. De fait, les registres de langue traités ne sont pas ceux du langage parlé, mais plutôt ceux de l'écrit.

En fonction de l'application envisagée, seront dictés des rapports, des articles de journaux, des lettres administratives ... Il en résulte une complexité moindre que s'il fallait retranscrire des dialogues à l'état brut. Dans le vif d'une conversation, les phrases grammaticales se mêlent aux phrases incomplètes, tandis que fourmillent hésitations, reprises, retours en arrière, ou autres répétitions. Historiquement, l'équipe de recherche IBM dirigée par F. Jelinek, est la première à avoir montré qu'un système à grand vocabulaire (Tangora 5000 mots en1995) pouvait tenir dans une petite boite portable.

Par la suite, l'ensemble des grands systèmes développés se sont inspirés du système Tangora. Avec un taux de réussite supérieure à 95% pour un vocabulaire de 20 000 mots, Tangora tend à devenir un système multi-lingue existant pour l'Anglais, l'Italien, le Français et l'Allemand. Le système Dragon fonctionne en mots isolés avec un vocabulaire de base de 16000 mots extensible à 30 000 mots. Un de ses points forts est sa capacité d’adaptation.

Aux USA, le second grand système vendu est la machine Kurzweil (1 000 à 10 000 mots). Le voice terminal de Kurzweil (KVT) ne s'est pas vraiment détaché de la commande vocale. Ses concepteurs affirment qu'il offre la possibilité de dicter en mots isolés un texte. Cependant les spécialistes ne lui confrère pas le statut de système de dictée.

En France, le système de dictée développé au LIMSI (5 000 à 10 000 mots) autour du circuit MPCD a abouti au produit DATAVOX (5 000 mots) commercialisé par la société VECSYS. Le taux de reconnaissance publié est de 95Vc pour un locuteur masculin. Le système Halmet est une maquette développée parallèlement, il peut traiter un vocabulaire de 7 000 mots comme mots isolés .

Le système développé à l'INRS (Bell Northern) par M. Lenning fonctionne en anglais avec une capacité de 86 000 mots. On trouve aussi des systèmes dédiés aux langues asiatiques comme celui réalisé pour le mandarin, qui peut traiter 60 000 mots.

III.4.4 Domaines connexes

D'autres types d'applications font également appel aux technologies déployées dans le domaine de la reconnaissance vocale. Leur but n'est cependant pas d'obtenir la transcription sous forme de texte:

a) **Identification de la langue**

Il s'agit ici de déterminer automatiquement la langue d'un utilisateur d'application vocale. Il est ainsi possible d'aiguiller l'utilisateur vers un opérateur parlant la même langue ou vers un module de dialogue adapté.

b) Identification et vérification du locuteur

L'identification consiste à déterminer l'auteur d'un signal de parole, parmi un ensemble de personnes ayant préalablement participé à une phase d'entraînement. Le nombre de décisions envisageables est au moins égal à la taille de la population. Par conséquent, les performances d'une tâche d'identification se dégraderont avec la taille de cette population.

La vérification, quant à elle, consiste à authentifier ou à rejeter un locuteur proclamant son identité. Dans ce cas, uniquement deux décisions peuvent être envisagées: acceptation ou rejet. Les performances d'une tâche de vérification seront donc a priori insensibles à la taille de la population. La vérification vocale du locuteur peut être couplée à d'autres approches de vérification dans le cadre d'applications de sécurité et de vérification d'identité: « vérification d'empreintes digitales », « vérification sur base d'une image du visage »

c) Segmentation en locuteurs (''Speaker Tracking'')

Dans le cadre d'applications de transcription et d'indexation automatique, il peut être utile de déterminer automatiquement les tours de parole et le nombre d'intervenants. Ces données enrichissent l'index automatique. Elles permettent de plus d'envisager une adaptation du système de reconnaissance vocale aux différents locuteurs, augmentant ainsi l'efficacité du système.

III.5 Conclusion

Nous avons présenté dans ce chapitre une description détaillée de l’un des méthodes statistiques probabilistes les plus utilisées pour la reconnaissance automatique de la parole qui sont les HMMs ainsi que les différents algorithmes liés au calcul des probabilités d’émissions et à l’estimation des paramètres du modèle en effectuant l’opération d’apprentissage. D’autre modèle

ont était présenté brièvement dans ce chapitre avec une description des applications possible de la RAP.

Nous verrons dans le chapitre suivant une nouvelle méthode probabiliste dans le domaine de la reconnaissance automatique de la parole.

Chapitre VI

Le Modèle d'arbre pour la reconnaissance automatique de la parole

Pour le cas multilocuteurs, vocabulaire limité, mot isolé et dépendant de locuteur, les HMMs reste un système performant. La détection des caractéristiques statistiques des mots et des unités peut être réalisée par les HMMs pour des locuteurs différents même dans le cas du grand vocabulaire. Notre contribution consiste à concevoir un système probabiliste compétitif aux HMMs.

IV.1 Base de données

Deux Bases de données ont été employées pour l'évaluation des différentes expériences.

a) Base Arabe

Cette base de données est le corpus des chiffres de 0 à 9 et 8 mots présentent les commandes vocales d'une calculatrice simple réalisée au niveau du laboratoire de recherche automatique et signaux d'Annaba **LASA** par la participation de 92 locuteurs (de 1 jusqu'à 46 des hommes, de 47 jusqu'à 92 des femmes).

Les données sont prélevées à une fréquence d'échantillonnage de 11025 kHz et numérisées à la résolution de 16 bits. Un sous-ensemble de la base de données est utilisé dans nos expériences composées d'un petit vocabulaire de dix chiffres isolés de (0 à 9) prononcés en arabe par 20 hommes et 20 femmes, chaque locuteur prononce 10 fois chaque mot de vocabulaire. 25% de la base est pour la tache de test (100 exemplaire (5x10 homes et 5X10 femmes)).

b) Base Japonaise (*Benchmark*) [27]

Neuf personnes masculins ont des difficultés de prononciation prononcent deux voyelles japonaises / AE / successivement. Pour chaque énoncé, 12 degrés de prédictions linéaires ont été appliqué, pour obtenir une série en temps discret avec 12 coefficients de cepstre LPC.

Cela signifie qu'un énoncé par un locuteur formé d'une série chronologique dont la longueur est comprise entre 7-29 et chaque point d'une série chronologique est de 12 coefficients.

Le nombre de la série temporelle est de 640 au total. 270 séries chronologiques pour l'apprentissage et l'autre ensemble de 370 séries chronologiques pour le test.

IV.1 Extraction des caractéristiques (Paramétrisation)

Une fois que le son a été émis par le locuteur, il est capté par un microphone. Le signal vocal est ensuite numérisé à l'aide d'un convertisseur analogique-numérique. Comme la voix humaine est constituée d'une multitude de sons, souvent répétitifs, le signal peut être compressé pour réduire le temps de traitement et l'encombrement en mémoire. L'analyse peut alors commencer. La première étape consiste à paramétrer le signal vocal du locuteur. Cela permet d'obtenir une " empreinte " caractéristique du son, sur laquelle on pourra ensuite travailler pour la reconnaissance. Pour cela, il existe plusieurs méthodes comme vus au chapitre II.

Nous utilisons MFCCS et LPC, Ces deux méthodes sont les plus couramment utilisées. Ces méthodes ont la propriété d'intégrer des connaissances du modèle auditif humain.

L'extraction des coefficients MFCCs est basée sur l'analyse par banc de filtres qui consiste à filtrer le signal par un ensemble de filtres passe-bande. L'énergie en sortie de chaque filtre est attribuée à sa fréquence centrale. Le tableau4.1 montre certains des paramètres du système adopté pour cette tâche.

Paramètre	Valeur
fréquence d'échantillonnage	11025 Hz, 16 bits
Preemphased	0.97
Type de fenêtre	Hamming

Tableau 4.1 Paramètres du Système

IV.2 Discrétisation des vecteurs

Le résultat de la paramétrisation est une série de vecteurs, caractérisent les différentes propriétés spectrales du signal. La tache de discrétisation peut se faire par la quantification des vecteurs à l'aide de la quantification vectorielle (VQ). Celle-ci est une représentation potentielle et efficace de l'information spectrale dans le signal de parole. Il est basé sur la génération d'un code de taille k à partir d'un ensemble de vecteurs de données.

Le principe de la quantification vectorielle est de partitionner l'espace de représentation en associant à chaque zone un représentant. Celui-ci est soit le centre de gravité de la partition (*centroïde)* (la moyenne des éléments de la partition), soit le vecteur le plus proche des autres (prototype). L'habitude veut que l'on parle dans les deux cas de prototypes.

L'ensemble des prototypes constitue le dictionnaire (*codebook*) de l'espace de représentation discrétisé. Divers algorithmes appartenant à la classification automatique non supervisée ont été proposés pour la construction du dictionnaire des prototypes. Chaque vecteur de paramètres est ensuite associé au prototype du dictionnaire dont il est le plus proche au sens d'une certaine distance (classiquement une distance euclidienne). La taille du dictionnaire permet un

contrôle de la distorsion introduite par la substitution d'un vecteur par son prototype. La contrepartie à cette perte de précision dans la modélisation des données est un coût de calcul faible.

Dans notre modélisation nous avons utilisé L'algorithme *k-means*, cet algorithme est l'algorithme de clustering le plus connu et le plus utilisé, du fait de sa simplicité de mise en œuvre. Il partitionne les données d'un signal en K clusters. Contrairement à d'autres méthodes dites hiérarchiques, qui créent une structure en « arbre de clusters » pour décrire les groupements, *k-means* ne crée qu'un seul niveau de clusters. L'algorithme renvoie une partition des données, dans laquelle les objets à l'intérieur de chaque cluster sont aussi proches que possible les uns des autres et aussi loin que possible des objets des autres clusters. Chaque cluster de la partition est défini par ses objets et son centroïde.

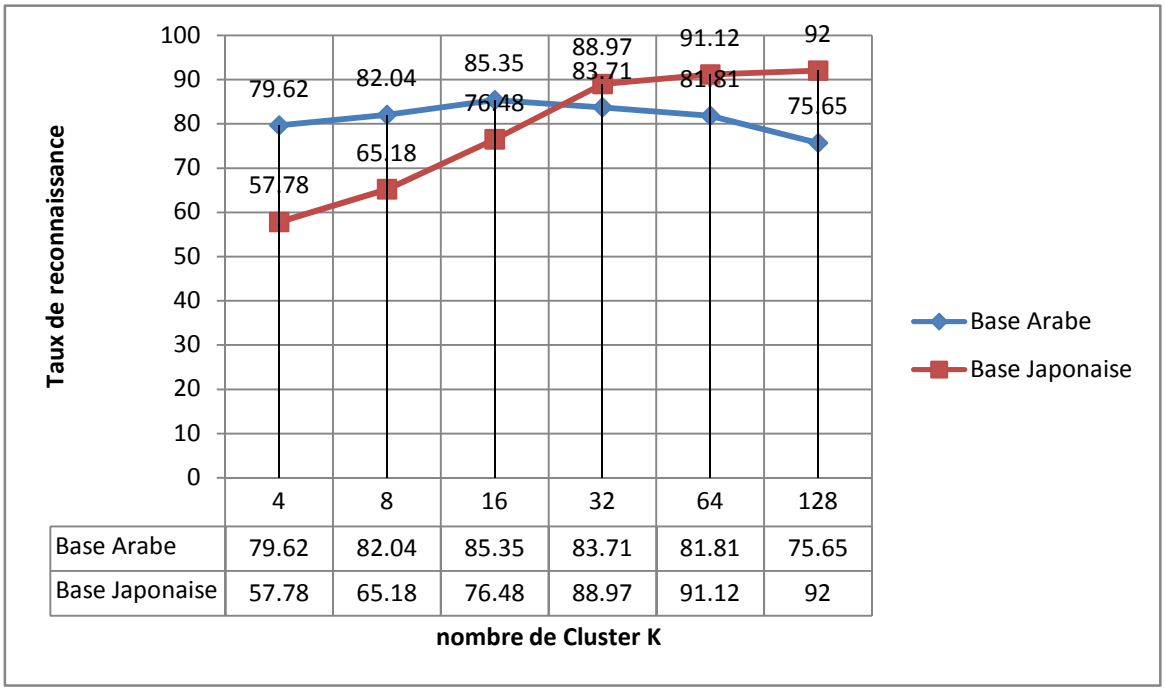
Le *k-means* est un algorithme itératif qui minimise la somme des distances entre chaque objet et le centroïde de son cluster. La position initiale des centroïdes conditionne le résultat final, de sorte que les centroïdes doivent être initialement placés le plus loin possible les uns des autres de façon à optimiser l'algorithme.

K-means change les objets de cluster jusqu'à ce que la somme ne puisse plus diminuer. Le résultat est un ensemble de clusters compacts et clairement séparés, sous réserve qu'on ait choisi la bonne valeur K du nombre de clusters.

Les principales étapes de l'algorithme *k-means* sont :

1. Choix aléatoire de la position initiale des K clusters.
2. (Réaffecter les objets à un cluster suivant un critère de minimisation des distances (généralement selon une mesure de distance euclidienne).
3. Une fois tous les objets placés, recalculer les K centroïdes.
4. Réitérer les étapes 2 et 3 jusqu'à ce que plus aucune réaffectation ne soit faite.

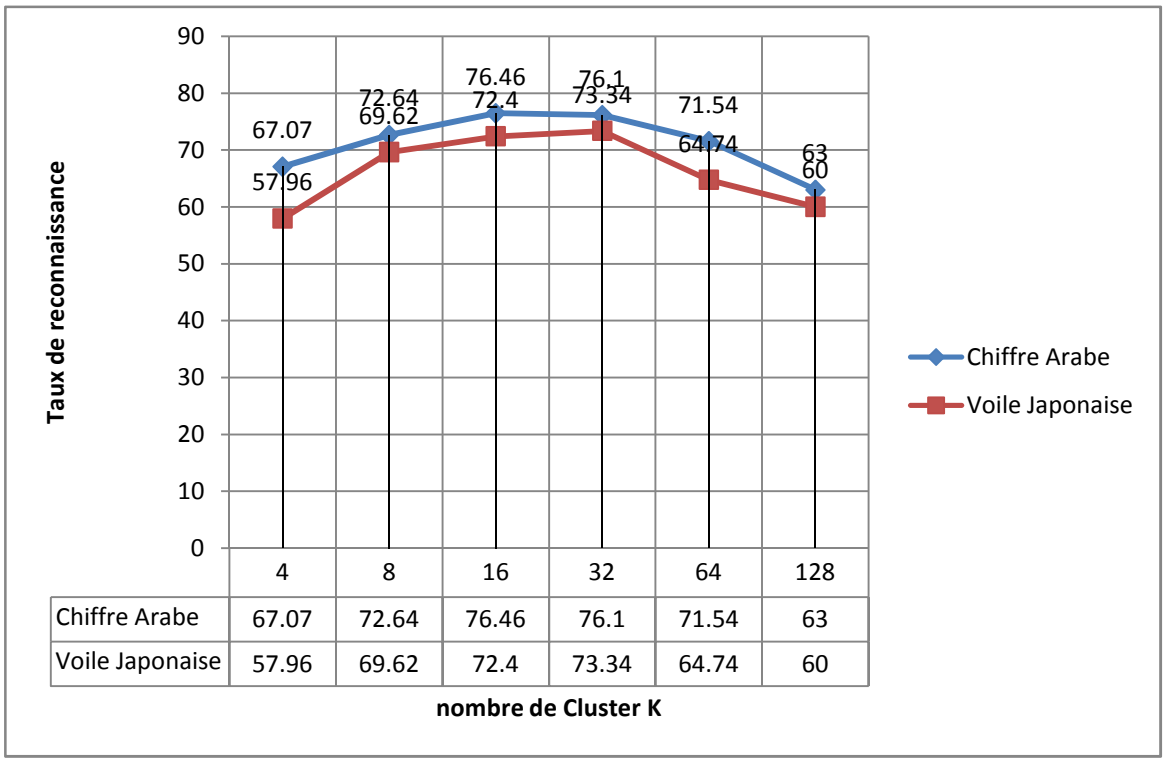
Pour choisir le nombre des clusters nous avons effectué une validation croisée (). La figure 4.1 montre les résultats obtenu pour différentes valeurs de K en utilisant les bases d'apprentissage et le modèle d'arbre. Suit aux résultats obtenu, la valeur K=16 est choisi pour la base des chiffres arabes et K=64 pour la base des voyelles japonaises.



	4	8	16	32	64	128
Base Arabe	79.62	82.04	85.35	83.71	81.81	75.65
Base Japonaise	57.78	65.18	76.48	88.97	91.12	92

Figure 4.1 Résultat de la validation croisée pour TM Vs la taille du dictionnaire

La figure 4.2 montre les résultats en utilisant les HMMs comme modéle pour l'évaluation du nombre de cluster, la valeur K=16 est choisi pour la base des chiffres arabes et K= 32 pour la base des voyelles japonaises.



	4	8	16	32	64	128
Chiffre Arabe	67.07	72.64	76.46	76.1	71.54	63
Voile Japonaise	57.96	69.62	72.4	73.34	64.74	60

Figure 4.2 Résultat de la validation croisée pour DHMMs Vs la taille du dictionnaire

IV.3 Les modèles probabilistes indexés par des arbres

IV.4.1 Formulation du problème

Soit D l'ensemble des mots prononcé et x_d un vecteur de n-dimension représente un mot prononcé. La Classe du tel mot est C_d avec $C_d = j$ si x_d appartient à la classe j avec $j = 0, \dots, S$ et S le nombre totale des classes. Supposons qu'on connaît la probabilité joint de la distribution $P(x_d, C_d)$ du vecteur (x_d, C_d). l'analyse bayésienne nous dit : quelle que soit la fonction du coût que l'utilisateur pourrait penser, il est nécessaire de connaitre la distribution postérieur $P(x_d | C_d)$.

L'information utile est la probabilité postérieure de la marginale d'un vecteur d'un mot prononcé. Pour chaque vecteur la quantité $P(C_d = j | x_d)$ represente la probabilité d'apatenance de ce vecteur x_d à la classe$C_d = j, j = 0, \dots S$. Pratiquement et pour l'ensemble des vecteur $x = (x_1, \dots, x_n)$ le modèle $P(x, C_d)$ est inconnus. En revanche nous disposons une base de données des mots prononcés. Cette collection des mots prononcées et de différents locuteurs. L'ensemble d'échantillons est noté $\{(x^{(1)}, C^{(1)}), \dots, (x^{(N)}, C^{(N)})\}$ et pour chaque $1 \leq i \leq N$, $x^{(i)}$ est le vecteur qui represente le mot prononcé et $C^{(i)}$ la classe de ce vecteur. Nous supposons que l'ensemble des données sont indépendant l'un de l'autre avec une distribution$P(x, C_d)$. La collection des données est réferer plus tard par la base d'apprentissage.

Notre objectif est de trouvé pour chaque classe des mots un graphe non cyclique (arbre) modélise $P(x, C_d = j)$ noté $P_j(x)$, et de construire un classifier probabiliste.

IV.4.2 Modèle d'arbre

Dans cette section, nous introduisons le modèle de l'arbre. Soit V désigne un ensemble de n variables aléatoires discrètes. Pour chaque variable aléatoire, $v \in V$, soit $\delta(v)$ représente sa range, $x_u \in \delta(v)$ une valeur particulière. $x = (x_1, \dots, x_n)$ Représente une attribution au variable dans V. Prenons $G(V, E)$, un graphe complet non orienté correspondant aux n variables, où E est un ensemble d'arêtes. Deux sommets voisin u et v sont notés$u \sim v$. Proposition [15]

Si le graphe G est un arbre, un graphe sans cycle connectés dont nous le note T, la paramétrisation de l'arbre est de la manière suivante: Pour $u, v \in V$ et $(u, v) \in E$, soit $q_{T_{uv}}$ la distribution probabiliste joint pour u et v. Nous avons besoin de ces distributions d'être compatible avec la marginalisation, notons par $q_{T_u}(x_u)$ le marginale de $q_{T_{uv}}(x_u, x_v)$,

ou $q_{T_{vu}}(x_v, x_u)$, pour chaque $v \neq u$. Nous allons maintenant affecter un graphe $G(V, E)$ à la distribution q_T comme suit :

$$q_T(x) = \prod_{(u\sim v)\in T} \frac{q_{T_{uv}}(x_u, x_v)}{q_{T_u}(x_u)\, q_{T_v}(x_v)} \prod_{u\in V} q_{T_u}(x_u) \qquad (4.1)$$

IV.4.2.1 Apprentissage du modèle

Le problème de l'apprentissage est formulé comme suit:

Etant donné un ensemble d'observations$X = \left(x^{(1)}, \ldots, x^{(N)}\right)$, , nous voulons trouver pour chaque classe des mots j, un arbre T_j dans lequel la probabilité de distribution est efficace. L'apprentissage du modèle est fait par la maximisation de la log-vraisemblance pour les données d'apprentissage pour chaque classe.

Chow et Liu [13] a montré que la maximisation du pois de l'arbre de recouvrement (MWST) en utilisant l'information mutuelle I_{uv} comme poids de l'arête(u, v), maximise aussi la probabilité de la distribution pour chaque classe j.

L'algorithme est résumé dans le tableau 4.2.

Algorithme Chow_ Lui (P_X)

Input: Distribution P_X représente un domaine V

Procédure MWST (Weights) qui donne comme sortie l'arbre de recouvrement maximal représentant V

1. Calcul de la distribution marginale P_{X_v}, $P_{X_{uv}}$ pour $u, v \in V$
2. Calcul de la valeur de l'information mutuelle I_{uv} pour $u, v \in V$
3. $T_j = MWST(\{I_{uv}\})$
4. Met $q_{j_{uv}} \equiv P_{X_{uv}}\ pour\ u, v \in V$

Tableau 4.2 Algorithme de Chow et Liu et l'estimation des paramètres

IV.4.2.2 L'inférence

On cherche attribué un vecteur $x = (x_1, \dots, x_n)$ à une tel classe, represente un mot prononcé. L'erreur de classification peut se minimiser on choisissant $\text{Argmax}_j\big(P(C_d = j|x)\big)$. revenant au théorème de Bayse

$$P(C_d = j|x) = \frac{P(x|C_d = j)P(C_d = j)}{P(x)} \qquad (4.2)$$

Nous avons aussi,

$$P(x) = \sum_{j=0}^{j=S} P(x|C_d = j)P(C_d = j) \qquad (4.3)$$

Avec

$$P(x|C_d = j) \approx \prod_{(u\sim v)\in T_j} \frac{q_{j_{uv}}(x_u, x_v)}{q_{j_u}(x_u)\, q_{j_v}(x_v)} \prod_{u\in V} q_{j_u}(x_u)$$

$$= q_j(x) \qquad (4.4)$$

$$\forall\, i, j = (0, \dots, S)\ \ P(C_d = j) \approx P(C_d = i)$$

Alors

$$P(C_d = j|x) \approx \frac{q_j(x)\, P(C_d = j)}{P(x)} \qquad (4.5)$$

Tous les éléments de "(4.3)" et "(4.4)" sont déjà obtenus dans l'étape d'apprentissage.

IV.5 Résultat expérimentaux

IV.5.1 Modèle d'arbre et DHMM

Pour l'évaluation du système markovienne nous avons utilisé un système classique discret à 5 états et de structure présenté dans la figure 4.3. Le tableau 4.3 montre les résultats obtenus via le résultat du modèle d'arbre pour les deux bases de données.

	Taux de reconnaissance du TMD %	Taux de reconnaissance du DHMM %
Base des chiffres Arabes	91.11	81.61
Base des voyelles Japonais	94.40	85.36

Tableau 4.3 Résultats des deux systèmes de reconnaissance (DTM et HMMs)

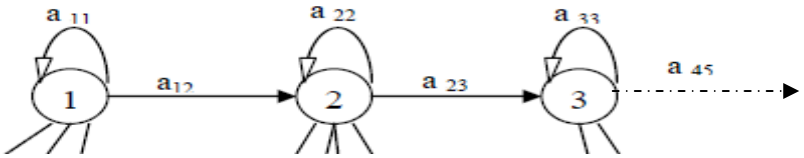


Figure 4.3 La structure de Chaine de Markov Utilisé

Les tableaux 4.4 et 4.5 montrent les résultats de classification pour chaque classe pour chacune des bases de données.

Classes de la base Arabe	Taux De reconnaissance du TMD %	Taux De reconnaissance du DHMMs%
0	76.70	68 .06
1	99.20	89.7
2	98.05	98.90
3	91.00	82.63
4	79.75	58.16
5	95.15	84.30
6	90.10	95.36
7	85.80	90.40
8	97.50	75.86
9	97.80	72.73
Total	91.11	81.61

Tableau 4.4 Résultats de classification pour chaque classe pour la base de données Arabe

Classes de la base Des VJs	Taux De Reconnaissance du TMD %	Taux De reconnaissance du DHMMs %
1	93.34	81.23
2	93.14	92.51
3	94.80	91.23
4	96.60	88.50
5	99.65	86.79
6	99.72	93.61
7	94.91	80.12
8	90.70	82.17
9	86.81	72.08
Total	94.40	85.36

Tableau 4.5 Résultats de classification pour chaque classe pour la base VJs

IV.5.2 Modèle d'arbre avec une structure arborisant prédéfinit

Le problème du modèle d'arbre est sa complexité qu'est d'ordre exponentiel, ainsi que le problématique de trouver l'arbre de recouvrement maximal qui est un problème Np-complet.

Suite à la nature des données vocale qu'est une série temporelle nous proposons une structure d'arbre prédéfinit adéquate avec cette nature (Figure 4.4). Cette proposition rend la complexité du modèle d'ordre linéaire et nous permet d'avoir une méthode très rapide pour la reconnaissance automatique de la parole.

Le tableau 4.6 montre le résultat obtenu avec un arbre optimal et notre structure d'arbre proposé.

	Taux de reconnaissance avec l'arber optimum %	Taux de reconnaissance avec l'arbre prédéfinit%
Base des chiffres Arabes	91.11	91.10
Base des voyelles Japonais	94.40	94.08

Tableau 4.6 Résultat de modèle d'arbre avec une structure d'arbre prédéfinit Vs la structure d'arbre optimale

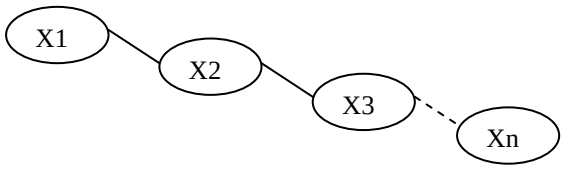


Figure 4.4 La structure de l'arbre prédéfinit pour le modèle d'arbre

IV.5.3 discussion et Conclusion

On remarque selon les tableaux 4.3, 4.4 et 4.5 que le meilleur taux de reconnaissance est obtenu par l'utilisation de notre modèle pour les deux bases de test.

Le tableau 4.6 montre des résultats proches entre les deux modèles d'arbre le premier avec un arbre pré structuré et l'autre avec un arbre optimum.

Les résultats obtenus montrent que:

- Le Modèle de distribution arborescente est un modèle probabiliste statistique compétitif au HMMs.

- L'utilisation de notre structure prédéfinit de l'arbre résoudre le problème de la complexité et rend le modèle une méthode très rapide tout en gardant un taux de reconnaissance proche de celui du modèle avec un arbre optimum.

Conclusion générale

La reconnaissance de la parole est actuellement traitée par une modélisation probabiliste markovienne. Le recours aux modèles de Markov cachés a permis de résoudre un grand nombre des difficultés inhérentes à la reconnaissance des formes segmentales.

Toutefois. Les modèles de Markov discriminants sont habituellement obtenus par de nouvelles itérations d'apprentissage des modèles initiaux à l'aide d'un critère discriminant. Contrairement à ces approches qui sont révélées coûteuses en temps de calcul et d'utilisation complexe pour un apport finalement restreint, nous avons choisi d'introduire une nouvelle formule dans les modèles probabilistes indexés par un arbre TDM (*Tree Distribution Model*).

Les résultats montrent que les performances de TDM sont meilleurs que les DHMMs standards. De nombreuses voies restent à explorer pour une meilleure utilisation du MDT.

Nous avons proposé comme perspective :

- D'améliorer l'algorithme de *clustering* par l'utilisation d'autre algorithme et autre distance
- Extensier le modèle pour traiter des données non discrétisées.

Bibliographie

[1] R. Boite, H. Bourlad, T. Dutoit, J. Hancq, and H. Leich , Traitement de la parole, presses polytechniques et universitaires romandes . , France, décembre 1999.

[2] M. Kunt,: Techniques modernes de traitement numérique des signaux, Presses polytechniques et universitaires Romandes, 1991.

[3] B. Houcine, " CONTRIBUTION A LA RECONNAISSANCE DE LA PAROLE PAR APPROCHE HYBRIDE DTW/HMMM". Thèse Doctorat Institut d'électronique, Université d'Annaba, 2008.

[4] S. Stephane, An algorithm for automatic formant extraction using linear prediction spectra , IEEE transaction on acoustics, speech, and signal processing, volassp -22. April 1974.

[5] M. Djemili, " reconnaissance de mots isolés arabes par DTW & HMM". Mémoire de Magister, Institut d'électronique, Université d'Annaba, 2001.

[6] A. Sakina Reconnaissance de la parole par HMM, Mémoire de magister, institut d'électronique, université d'Annaba, 2004.

[7] S. FURUI, MEMBER, IEEE Cepstral Analysis Technique for Automatic Speaker Verification, IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. ASSP-29, NO. 2, APRIL 1981.

[8] Rabiner, L.R.mA tutorial on hidden Markov models and selected applications in speech recognition,Proceedings of the IEEE Volume 77, Issue 2, Feb. 1989 Page(s):257 – 286.

[9] N. Morgan and H. Bourlard, "Neural networks for statistical recognition of speech," Proc. IEEE, vol. 83, no. 5, pp. 742–772, May 1995.

[10] Pujol, P.; Pol, S.; Nadeu, C.; Hagen, A.; Bourlard, H.; Comparison and combination of features in a hybrid HMM/MLP and a HMM/GMM speech recognition system, Speech and Audio Processing, IEEE Transactions on Volume 13, Issue 1, Jan. 2005 Page(s):14 – 22.

[11] Lopes, C.; Perdigao, F.;]A discriminative training method applied to a hybrid ANN/HMM phoneme recognizer Signal Processing, 2008. ICSP 2008. 9th International Conference on 26-29 Oct. 2008 Page(s):1981 – 1984.

[12] Essa, E.M.; Tolba, A.S.; Elmougy, S.; A comparison of combined classifier architectures for Arabic Speech Recognition, Computer Engineering & Systems, 2008. ICCES 2008. International Conference on 25-27 Nov. 2008 Page(s):149 – 153.

[13] C. Chow, C. Liu, "Approximating discrete probability distributions with dependence trees", Fifteenth IEEE Transactions on Information Theory 14 (3) (1968) 462–467, May.

[14] S.Ioffe, D.Forsyth, “Mixtures of trees for object recognition”, Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference.

[15] J. Pearl, “Probabilistic reasoning in intelligent systems: networks of plausible inference”. Morgan Kaufmann, 1988.

[16] L.Xue, J.Yin, Z.Ji, L.Jiang, “A Particle Swarm Optimization for Hidden for Hidden Markov Model Training”, in Proceedings of the 8th International Conference on Signal Processing, 2006, pp.16-20.

[17] M. D. Richard and R. P. Lippmann, “Neural network classifiers Estimate Bayesian a posteriori probabilities,” in Proc. Conf. Neural Computation 3, 1991, pp. 461–483.

[18] J.K. Baker, “Stochastic modeling for automatic speech understanding”, Speech Recognition, Academic Press, New York, 1975.

[19] F. Jelinek, “Continuous speech recognition by statistical methods”, Proc. IEEE 64, 1976, pp.250-256.

[20] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B.Dunn, "Speaker verification using adapted gaussian mixture models", Digital Signal Processing, vol. 10, no. 1-3, 2000.

[21] M.Al-Zabibi, “An Acoustic–Phonetic Approach in Automatic Arabic Speech Recognition”, the British Library in Association with UMI, 1990.

[22] Alotaibi,Y.A, “Spoken Arabic digits recognizer using recurrent neural networks”, Signal Processing and Information Technology, 2004. Proceedings of the Fourth IEEE International.

[23] Khalid Saeed, Mohammad Kheir Nammous, “Speech-and-Speaker Identification System: Feature Extraction, Description, and Classification of Speech-Signal Image”, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 54, NO. 2, APRIL 2007.

[24] S. El Fkihi, M. Daoudi, D. Aboutajdine, “Probability Approximation Using Best-Tree Distribution for Skin Detection”, Advanced Concepts for Intelligent Vision Systems, University of Antwerp, Antwerp,Belgium, September 18–21, 2006, pp. 767–775

[25] Sanaa El Fkihi , Mohamed Daoudi, Driss Aboutajdine ,”The mixture of K-Optimal-Spanning-Trees based probability approximation: Application to skin detection”, Elsevier 2008.

[26] Fengqin Yang; Changhai Zhang, An Effective Hybrid Optimization Algorithm for HMM,
Natural Computation, 2008. ICNC '08. Fourth International Conference on Volume 4, 18-20 Oct. 2008 Page(s):80 – 84.

[27] The UCI KDD Archive Information and Computer Science University of California, Irvine Irvine, CA 92697-3425

Publication

Tree Distribution Classifier for Automatic Spoken Arabic Digit Recognition*

N. HAMMAMI, M. SELLAMI
Department of Computer Science, LRI, Algeria
nacereddine.hammami@gmail.com, sellami@lri-annaba.net

***Accepté au ICITST-2009 Cosponsored by IEEE London UK**

Abstract

In this work we propose a novel method for automatic discrete speech recognition composed from two steps. In a first step, discrete speech features are extracted by means of Mel Frequency Cepstral Coefficients (MFCCs) followed by vector quantization (VQ). Then in a second step, the obtained features are fed to a Tree distribution classifier which provides the class-label associated with each feature by approximating the true class probability by means of an optimal spanning tree model. The experimental results obtained on a spoken Arabic digit dataset confirmed the promising capabilities of the proposed approach.

1. Introduction

Automatic Speech Recognition (ASR) is gaining a growing role for a variety of applications, such as hand-free operation and control (as in cars and airplanes), automatic query answering, telephone communication with information systems, automatic dictation (speech-to-text transcription), government information systems, etc. In fact, speech communication with computers, PCs, and household appliances is envisioned to be the dominant human machine interface in the near future.

In the literature of speech recognition many methods have been proposed. Generally, these are based on Hidden Markov Models (HMM) [1], neural networks [2], statistical analysis and vector quantization [3], and Gaussian mixture models (GMM) [4]. By contrast to other languages the Arabic language had limited number of research efforts [6] [7], although it is one of the oldest languages in the world and the fifth widely used language nowadays [5].

To deal with this issue, we propose in this paper an automatic method for discrete recognition of spoken Arabic digits using a Tree distribution classifier. The choice of the discrete model is computationally efficient and represents a powerful base form by selecting appropriate and well trained observation symbols and incorporating parameter smoothing. Moreover, it performs very well for several tasks such as large vocabulary isolated word recognition. In this model type, the extracted feature vector is mapped into one of a set of prototype vectors through a vector quantization process.

Prototype vectors are constituted during the training phase using a clustering algorithm. After this step, the derived features are fed to a Tree distribution classifier, which provides the class-label associated with each feature by approximating the true class probability by means of an optimal-spanning-Tree model. The latter are increasingly and successfully used to deal with the probability estimation in different pattern recognition problems such as skin detection [8] [9] and object detection [11]. The experimental results obtained on a spoken Arabic digit dataset proved the promising capabilities of the proposed approach.

The remainder of this paper is organized as follows: in section two, we describe the feature extraction method. Section three details the Tree distribution classifier model and section four is devoted to experiments and results. Finally, conclusions and perspectives are drawn in section five.

2. Feature Extraction

In the signal analysis phase the input speech signal is transformed into feature vectors containing spectral and/or temporal information using Mel Frequency Cepstral coefficients (MFCCs) [6]. Table 1 shows some of the system parameters adopted for such task. The

result of the feature extraction is a series of vectors, characteristic of the time varying spectral properties of the speech signal. These can then be mapped into discrete vectors by quantizing them using vector quantification (VQ). The latter is a potentially efficient representation of spectral information in the speech signal. It is based on the generation of a code of size *M* from a training set of vectors of size *L*. To this end, we purpose to adopt the well known *k-means* clustering algorithm, which is summarized in the following steps:

Initialization: we choose an arbitrarily *M* vectors to represent the initial set of code words in the codebook. Once the codebook of vectors has been obtained, the mapping between the observation vectors and codebook indices becomes a simple nearest neighbor computation, i.e. the observation vector is assigned the index.

TABLE1. System Parameters

Parameter	Value
Sampling rate	11025 Hz, 16 bits
Preemphased	0.97
Window type	Hamming

Centroid update: we update the codeword for each index using the centroid of the training vector assigned to the index. The distance used is the Euclidian distance, whose minimum value is used to update the centroid. If we consider $c(k)$ as the current centroid of the k^{th}cluster and $v(k)$ a vector in the cluster then:

$$D_{min} = \min_{1 \le n \le N} \left[\left[\sum_{k=1}^{K} (C_n(k) - v(k))^2 \right]\right] \qquad (1)$$

3. Tree Distribution Model

Consider D the set of spoken digits and x_d the *n*-dimensional vector representing the spoken Arabic digits. The class of the spoken digit is C_d with $C_d = j$ if x_d belongs to class j with $j = 0, \dots, 9$. Let us assume that we know the joint probability distribution $P(x_d, C_d)$ of the vector (x_d, C_d). Then the Bayesian analysis tells us that, whatever the cost function the user might think of, all that is needed is the a-posterior distribution $P(x_d | C_d)$.

The useful information is contained in the one spoken digit vector marginal of the a-posterior probability. That is for each spoken digit vector, the quantity $P(C_d = j | x_d)$ $q$uantifying the belief for the appurtenance of the spoken digit vector x_d to the class $C_d = j, j = 0, \dots, 9$. In practice for $x = (x_1, \dots, x_n)$ the model $P(x, C_d)$ is unknown. Instead, we have spoken Arabic digits database. It is a collection of pronounced Arabic digit (zero to nine) from dependent speaker.

The collection samples noted $\{(x^{(1)}, C^{(1)}), \dots, (x^{(N)}, C^{(N)})\}$ where for each $1 \le i \le N$, $x^{(i)}$ is a vector representation of the spoken digit and $C^{(N)}$ is the associated class. We assume that the samples are independent each other with the distribution $P(x, C_d)$. The collection of samples is referred later as the training data. Our objective is to find for each class a non oriented acyclic graph (tree) modeling $P(x, C_d = j)$ noted $P_j(x)$ and construct a probabilistic classifier.

A. Tree model

In this section we introduce the Tree model. Let V denotes a set of *n* discrete random variables of interest. For each random variable$v \in V$, let $\delta(v)$ represent its range, $x_u \in \delta(v)$ a particular value. $x = (x_1, \dots, x_n)$ denotes an assignment to the variables in V .

Let's consider a complete non oriented graph $G(V, E)$ corresponding to the *n* variables, where E is a set of edges. Two neighbor vertices u and v are noted $u \sim v$.

Proposition

If the graph G was a tree; a connected graph without loops which we note T, we parameterize a tree in the following way: For $u, v \in V$ and $(u, v) \in E$, let $q_{T_{uv}}$ denote a joint probability distribution on *u* and *v*. We require these distributions to be consistent with respect to marginalization, denoting by $q_{T_u}(x_u)$the marginal of$q_{T_{uv}}(x_u, x_v)$, or$q_{T_{vu}}(x_v, x_u)$, with respect to x_u for any $v \neq u$.

We now assign a distribution q_T to the graph $G(V, E)$ as follows [12]:

$$q_T(x) = \prod_{(u \sim v) \in T} \frac{q_{T_{uv}}(x_u, x_v)}{q_{T_u}(x_u)\, q_{T_v}(x_v)} \prod_{u \in V} q_{T_u}(x_u) \qquad (2)$$

B. Learning of tree distribution

The learning problem is formulated as follows: given a set of observations $X = (x^{(1)}, \dots, x^{(N)})$, we want to find for each digit class j, $j = 0, \dots, 9$ one tree T_j in which the distribution probability is efficient.

We learn the model by maximizing the log-likelihood for the training data for each class. Chow and Liu [10] showed that the maximum weight spanning tree (MWST) using mutual information I_{uv} as the weight for the edge(u, v), maximizes the likelihood over tree distributions q_j for each class j. The algorithm is summarized on Table 2.

TABLE 2. The Chow and Liu Algorithm for Maximum Likelihood Estimation of Tree Structure and Parameters

Algorithm Chow_ Lui $(\boldsymbol{P_X})$

Input: *Distribution* P_X *over domain* V

Procedure MWST (Weights) that outputs a maximum weight spanning tree over V

1 Compute marginal distributions P_{X_v}, $P_{X_{uv}}$

for $u, v \in V$

2 Compute mutual information values I_{uv}

for $u, v \in V$

3 $T_j = MWST(\{I_{uv}\})$

4 Set $q_{j_{uv}} \equiv P_{X_{uv}}$ *for* $u, v \in V$

C. Inference

We would denote the class of a vector $x = (x_1, \dots, x_n)$, which represents a spoken Arabic digit. The expected classification error can be minimized by choosing $Argmax_j\big(P(C_d = j|x)\big)$. According to Bayes's theorem:

$$P(C_d = j|x) = \frac{P(x|C_d = j)P(C_d = j)}{P(x)} \quad (3)$$

Moreover,

$$P(x) = \sum_{j=0}^{j=9} P(x|C_d = j)P(C_d = j) \quad (4)$$

In which

$$P(x|C_d = j) \approx \prod_{(u\sim v)\in T_j} \frac{q_{j_{uv}}(x_u, x_v)}{q_{j_u}(x_u)\, q_{j_v}(x_v)} \prod_{u\in V} q_{j_u}(x_u)$$

$$= q_j(x) \quad (5)$$

$$\forall\, i, j = (0, \dots, 9)\ \ P(C_d = j) \approx P(C_d = i)$$

Therefore

$$P(C_d = j|x) \approx \frac{q_j(x)\, P(C_d = j)}{P(x)} \quad (6)$$

All the elements of "(4)" and "(5)" are previously computed in learning setup.

4. Experimental Results

A. Dataset Description

The experiments were performed using the Arabic digit corpus database from the national laboratory of automatic and signals at the University of Badji-Mokhtar in Annaba, Algeria. This data base was created from all ten Arabic digits. A number of 40 individual (20 males and 20 females) Arabic native speakers were asked to utter all digits ten times. Hence, the database consists of 10 repetitions of every digit produced by each speaker. Depending on this, the database consists of 4000 tokens (10 digits x 10 repetitions x 40 speakers). In this research, speaker-independent mode is considered.

B. First experiment

In this experiment, before applying the tree model to the above dataset, we perform feature extraction by means of VQ as described previously. The result of the clustering is a codebook of 16 (optimum size obtained using 2-fold cross validation on the training set). Figure 1 shows accuracy results for different values of k.

The discretized vectors with 16-means are the final features used throughout, and it's the input-set that is used in classification step. Table 3 shows the classification results obtained by the proposed Tree model. As can be seen, the overall accuracy of the system is 90.35%.

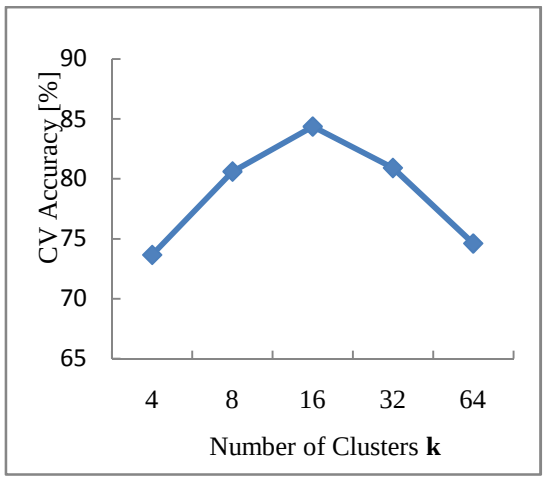


Figure 1. Cross Validation (CV) Accuracy Versus the Number of Clusters

C. Second Experiment

In this experiment, we assess the sensitivity of the Tree classifier with respect to the training set size. For such purpose, we reduce the training set and repeat the experimental scenario of experiment one. The obtained results shown in Table 4 confirm the robustness of the tree classier with respect to the training set size.

TABLE 3. Recognition by Dependence Tree Model

Arabic Digits Classes	Dependence Tree Model Success Rate %
0	91.00
1	99.00
2	91.50
3	88.00
4	81.50
5	94.50
6	84.50
7	89.50
8	92.50
9	91.00
OA	90.35

TABLE 4. Sensitivity for the Size of Training Data Set

(Test set is fixed at 2000 samples)

Training data size	Dependence Tree Model Success Rate %
400	72.25
800	81.45
1200	87.40
1600	87.75
2000	90.35

This result compared to result obtained by alternative methods [6] [7] showed the benefit of using tree distribution model.

5. Conclusion

In this paper, we have presented a Tree distribution model for discrete speech recognition. The experimental results obtained on spoken Arabic digits confirm the promising capabilities of the proposed approach. Future developments adopted for more than one tree in classifier design, will hopefully lead to more robust classification results.

6. References

[1] X.Huang, A. Acero, and H. Hon, “Spoken Language Processing”, *Prentice Hall PTR*, 2001.

[2] J.P Hosom, R.Cole and M.Fanty, "Speech Recognition Using Neural Networks at the Center for Spoken Language Understanding", *NSFGraduate Research Traineeships project*, Center for Spoken Language Understanding (CSLU), Oregon Graduate Institute of Science and Technology, USA, Jul. 1999.

[3] F. Bimbot and L. Mathan. "Second-order statistical measures for text independent speaker identification", In Proc. *ESCA Workshop on Automatic Speaker Recognition Identification and Verification*, pp. 51-54, 1994.

[4] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B.Dunn, "Speaker verification using adapted gaussian mixture models", *Digital Signal Processing*, vol. 10, no. 1-3, 2000.

[5] M.Al-Zabibi, “An Acoustic–Phonetic Approach in Automatic Arabic Speech Recognition”, *the British Library in Association with UMI*, 1990.

[6] Alotaibi,Y.A, “Spoken Arabic digits recognizer using recurrent neural networks”, *Signal Processing and Information Technology*, 2004. Proceedings of the Fourth IEEE International.

[7] Khalid Saeed, Mohammad Kheir Nammous, “Speech-and-Speaker Identification System: Feature Extraction, Description, and Classification of Speech-Signal Image”, *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, VOL. 54, NO. 2, APRIL 2007.

[8] S. El Fkihi, M. Daoudi, D. Aboutajdine, “Probability Approximation Using Best-Tree Distribution for Skin Detection”, *Advanced Concepts for Intelligent Vision Systems*, University of Antwerp, Antwerp,Belgium, September 18–21, 2006, pp. 767–775

[9] Sanaa El Fkihi , Mohamed Daoudi, Driss Aboutajdine ,”The mixture of K-Optimal-Spanning-Trees based probability approximation: Application to skin detection”, *Elsevier* 2008.

[10] C. Chow, C. Liu, “Approximating discrete probability distributions with dependence trees”, *Fifteenth IEEE Transactions on Information Theory* 14 (3) (1968) 462–467, May.

[11] S.Ioffe, D.Forsyth, “Mixtures of trees for object recognition”, *Computer Vision and Pattern Recognition*, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference.

[12] J. Pearl, “Probabilistic reasoning in intelligent systems: networks of plausible inference”. *Morgan Kaufmann*, 1988.

Tree Distribution Classifier for Automatic Spoken Arabic Digit Recognition

N. HAMMAMI, M. SELLAMI
Department of Computer Science, LRI, Algeria
nacereddine.hammami@gmail.com, sellami@lri-annaba.net

Abstract

In this work we propose a novel method for automatic discrete speech recognition composed from two steps. In a first step, discrete speech features are extracted by means of Mel Frequency Cepstral Coefficients (MFCCs) followed by vector quantization (VQ). Then in a second step, the obtained features are fed to a Tree distribution classifier which provides the class-label associated with each feature by approximating the true class probability by means of an optimal spanning tree model. The experimental results obtained on a spoken Arabic digit dataset confirmed the promising capabilities of the proposed approach.

1. Introduction

Automatic Speech Recognition (ASR) is gaining a growing role for a variety of applications, such as hand-free operation and control (as in cars and airplanes), automatic query answering, telephone communication with information systems, automatic dictation (speech-to-text transcription), government information systems, etc. In fact, speech communication with computers, PCs, and household appliances is envisioned to be the dominant human machine interface in the near future.

In the literature of speech recognition many methods have been proposed. Generally, these are based on Hidden Markov Models (HMM) [1], neural networks [2], statistical analysis and vector quantization [3], and Gaussian mixture models (GMM) [4]. By contrast to other languages the Arabic language had limited number of research efforts [6] [7], although it is one of the oldest languages in the world and the fifth widely used language nowadays [5].

To deal with this issue, we propose in this paper an automatic method for discrete recognition of spoken Arabic digits using a Tree distribution classifier. The choice of the discrete model is computationally efficient and represents a powerful base form by selecting appropriate and well trained observation symbols and incorporating parameter smoothing. Moreover, it performs very well for several tasks such as large vocabulary isolated word recognition. In this model type, the extracted feature vector is mapped into one of a set of prototype vectors through a vector quantization process.

Prototype vectors are constituted during the training phase using a clustering algorithm. After this step, the derived features are fed to a Tree distribution classifier, which provides the class-label associated with each feature by approximating the true class probability by means of an optimal-spanning-Tree model. The latter are increasingly and successfully used to deal with the probability estimation in different pattern recognition problems such as skin detection [8] [9] and object detection [11]. The experimental results obtained on a spoken Arabic digit dataset proved the promising capabilities of the proposed approach.

The remainder of this paper is organized as follows: in section two, we describe the feature extraction method. Section three details the Tree distribution classifier model and section four is devoted to experiments and results. Finally, conclusions and perspectives are drawn in section five.

2. Feature Extraction

In the signal analysis phase the input speech signal is transformed into feature vectors containing spectral and/or temporal information using Mel Frequency Cepstral coefficients (MFCCs) [6]. Table 1 shows some of the system parameters adopted for such task. The result of the feature extraction is a series of vectors, characteristic of the time varying spectral properties of the speech signal. These can then be mapped into discrete vectors by quantizing them using vector quantification (VQ). The latter is a potentially efficient representation of spectral information in the speech signal. It is based on the generation of a code of size M from a training set of vectors of size L. To this end, we purpose to adopt the well known *k-means* clustering algorithm, which is summarized in the following steps:

Initialization: we choose an arbitrarily M vectors

to represent the initial set of code words in the codebook. Once the codebook of vectors has been obtained, the mapping between the observation vectors and codebook indices becomes a simple nearest neighbor computation, i.e. the observation vector is assigned the index.

TABLE1. System Parameters

Parameter	Value
Sampling rate	11025 Hz, 16 bits
Preemphased	0.97
Window type	Hamming

Centroid update: we update the codeword for each index using the centroid of the training vector assigned to the index. The distance used is the Euclidian distance, whose minimum value is used to update the centroid. If we consider $c(k)$ as the current centroid of the k^{th}cluster and $v(k)$ a vector in the cluster then:

$$D_{min} = \min_{1\le n\le N} \left[\left[\sum_{k=1}^{K} (C_n(k) - v(k))^2 \right]\right] \qquad (1)$$

3. Tree Distribution Model

Consider D the set of spoken digits and x_d the *n*-dimensional vector representing the spoken Arabic digits. The class of the spoken digit is C_d with $C_d = j$ if x_d belongs to class j with $j = 0, \dots, 9$. Let us assume that we know the joint probability distribution $P(x_d, C_d)$ of the vector (x_d, C_d). Then the Bayesian analysis tells us that, whatever the cost function the user might think of, all that is needed is the a-posterior distribution $P(x_d | C_d)$.

The useful information is contained in the one spoken digit vector marginal of the a-posterior probability. That is for each spoken digit vector, the quantity $P(C_d = j | x_d)$ *q*uantifying the belief for the appurtenance of the spoken digit vector x_d to the class $C_d = j, j = 0, \dots, 9$. In practice for $x = (x_1, \dots, x_n)$ the model $P(x, C_d)$ is unknown. Instead, we have spoken Arabic digits database. It is a collection of pronounced Arabic digit (zero to nine) from dependent speaker.

The collection samples noted $\{(x^{(1)}, C^{(1)}), \dots, (x^{(N)}, C^{(N)})\}$ where for each $1 \le i \le N$, $x^{(i)}$ is a vector representation of the spoken digit and $C^{(N)}$ is the associated class. We assume that the samples are independent each other with the distribution $P(x, C_d)$. The collection of samples is referred later as the training data. Our objective is to find for each class a non oriented acyclic graph (tree) modeling $P(x, C_d = j)$ noted $P_j(x)$ and construct a probabilistic classifier.

A. Tree model

In this section we introduce the Tree model. Let V denotes a set of n discrete random variables of interest. For each random variable$v \in V$, let $\delta(v)$ represent its range, $x_u \in \delta(v)$ a particular value. $x = (x_1, \dots, x_n)$ denotes an assignment to the variables in V.

Let's consider a complete non oriented graph $G(V, E)$ corresponding to the *n* variables, where E is a set of edges. Two neighbor vertices u and v are noted $u \sim v$.

Proposition

If the graph G was a tree; a connected graph without loops which we note T, we parameterize a tree in the following way: For $u, v \in V$ and $(u, v) \in E$, let $q_{T_{uv}}$ denote a joint probability distribution on u and v. We require these distributions to be consistent with respect to marginalization, denoting by $q_{T_u}(x_u)$the marginal of$q_{T_{uv}}(x_u, x_v)$, or$q_{T_{vu}}(x_v, x_u)$, with respect to x_u for any $v \neq u$.

We now assign a distribution q_T to the graph $G(V, E)$ as follows [12]:

$$q_T(x) = \prod_{(u\sim v)\in T} \frac{q_{T_{uv}}(x_u, x_v)}{q_{T_u}(x_u)\, q_{T_v}(x_v)} \prod_{u\in V} q_{T_u}(x_u) \qquad (2)$$

B. Learning of tree distribution

The learning problem is formulated as follows: given a set of observations $X = (x^{(1)}, \dots, x^{(N)})$, we want to find for each digit class j, $j = 0, \dots, 9$ one tree T_j in which the distribution probability is efficient.

We learn the model by maximizing the log-likelihood for the training data for each class. Chow and Liu [10] showed that the maximum weight spanning tree (MWST) using mutual information I_{uv} as the weight for the edge(u, v), maximizes the likelihood over tree distributions q_j for each class j. The algorithm is summarized on Table 2.

TABLE 2. The Chow and Liu Algorithm for Maximum Likelihood Estimation of Tree Structure and Parameters

Algorithm Chow_ Lui $(\boldsymbol{P_X})$

Input: *Distribution* P_X *over domain V*

Procedure MWST (Weights) that outputs a maximum weight spanning tree over V

1 Compute marginal distributions P_{X_v}, $P_{X_{uv}}$

for $u, v \in V$

2 Compute mutual information values I_{uv}

for $u, v \in V$

3 $T_j = MWST(\{I_{uv}\})$

4 Set $q_{j_{uv}} \equiv P_{X_{uv}}$ *for* $u, v \in V$

C. Inference

We would denote the class of a vector $x = (x_1, \dots, x_n)$, which represents a spoken Arabic digit. The expected classification error can be minimized by choosing $Argmax_j\big(P(C_d = j|x)\big)$. *A*ccording to Bayes's theorem:

$$P(C_d = j|x) = \frac{P(x|C_d = j)P(C_d = j)}{P(x)} \quad (3)$$

Moreover,

$$P(x) = \sum_{j=0}^{j=9} P(x|C_d = j)P(C_d = j) \quad (4)$$

In which

$$P(x|C_d = j) \approx \prod_{(u\sim v)\in T_j} \frac{q_{j_{uv}}(x_u, x_v)}{q_{j_u}(x_u)\, q_{j_v}(x_v)} \prod_{u\in V} q_{j_u}(x_u)$$

$$= q_j(x) \quad (5)$$

$$\forall\, i, j = (0, \dots, 9)\ \ P(C_d = j) \approx P(C_d = i)$$

Therefore

$$P(C_d = j|x) \approx \frac{q_j(x)\, P(C_d = j)}{P(x)} \quad (6)$$

All the elements of "(4)" and "(5)" are previously computed in learning setup.

4. Experimental Results

A. Dataset Description

The experiments were performed using the Arabic digit corpus database from the national laboratory of automatic and signals at the University of Badji-Mokhtar in Annaba, Algeria. This data base was created from all ten Arabic digits. A number of 40 individual (20 males and 20 females) Arabic native speakers were asked to utter all digits ten times. Hence, the database consists of 10 repetitions of every digit produced by each speaker. Depending on this, the database consists of 4000 tokens (10 digits x 10 repetitions x 40 speakers). In this research, speaker-independent mode is considered.

B. First experiment

In this experiment, before applying the tree model to the above dataset, we perform feature extraction by means of VQ as described previously. The result of the clustering is a codebook of 16 (optimum size obtained using 2-fold cross validation on the training set). Figure 1 shows accuracy results for different values of k.

The discretized vectors with 16-means are the final features used throughout, and it's the input-set that is used in classification step. Table 3 shows the classification results obtained by the proposed Tree model. As can be seen, the overall accuracy of the system is 90.35%.

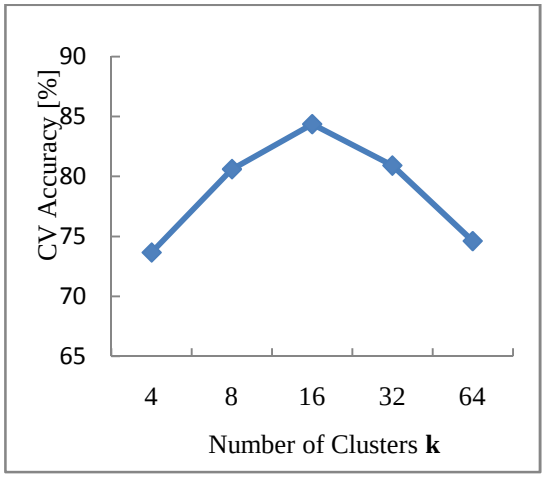


Figure 1. Cross Validation (CV) Accuracy Versus the Number of Clusters

C. Second Experiment

In this experiment, we assess the sensitivity of the Tree classifier with respect to the training set size. For such purpose, we reduce the training set and repeat the experimental scenario of experiment one. The obtained results shown in Table 4 confirm the robustness of the tree classier with respect to the training set size.

TABLE 3. Recognition by Dependence Tree Model

Arabic Digits Classes	Dependence Tree Model Success Rate %
0	91.00
1	99.00
2	91.50
3	88.00
4	81.50
5	94.50
6	84.50
7	89.50
8	92.50
9	91.00
OA	90.35

TABLE 4. Sensitivity for the Size of Training Data Set

(Test set is fixed at 2000 samples)

Training data size	Dependence Tree Model Success Rate %
400	72.25
800	81.45
1200	87.40
1600	87.75
2000	90.35

This result compared to result obtained by alternative methods [6] [7] showed the benefit of using tree distribution model.

5. Conclusion

In this paper, we have presented a Tree distribution model for discrete speech recognition. The experimental results obtained on spoken Arabic digits confirm the promising capabilities of the proposed approach. Future developments adopted for more than one tree in classifier design, will hopefully lead to more robust classification results.

6. References

[1] X.Huang, A. Acero, and H. Hon, "Spoken Language Processing", *Prentice Hall PTR*, 2001.

[2] J.P Hosom, R.Cole and M.Fanty, "Speech Recognition Using Neural Networks at the Center for Spoken Language Understanding", *NSFGraduate Research Traineeships project*, Center for Spoken Language Understanding (CSLU), Oregon Graduate Institute of Science and Technology, USA, Jul. 1999.

[3] F. Bimbot and L. Mathan. "Second-order statistical measures for text independent speaker identification", In Proc. *ESCA Workshop on Automatic Speaker Recognition Identification and Verification*, pp. 51-54, 1994.

[4] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B.Dunn, "Speaker verification using adapted gaussian mixture models", *Digital Signal Processing*, vol. 10, no. 1-3, 2000.

[5] M.Al-Zabibi, "An Acoustic–Phonetic Approach in Automatic Arabic Speech Recognition", *the British Library in Association with UMI*, 1990.

[6] Alotaibi,Y.A, "Spoken Arabic digits recognizer using recurrent neural networks", *Signal Processing and Information Technology*, 2004. Proceedings of the Fourth IEEE International.

[7] Khalid Saeed, Mohammad Kheir Nammous, "Speech-and-Speaker Identification System: Feature Extraction, Description, and Classification of Speech-Signal Image", *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, VOL. 54, NO. 2, APRIL 2007.

[8] S. El Fkihi, M. Daoudi, D. Aboutajdine, "Probability Approximation Using Best-Tree Distribution for Skin Detection", *Advanced Concepts for Intelligent Vision Systems*, University of Antwerp, Antwerp,Belgium, September 18–21, 2006, pp. 767–775

[9] Sanaa El Fkihi , Mohamed Daoudi, Driss Aboutajdine ,"The mixture of K-Optimal-Spanning-Trees based probability approximation: Application to skin detection", *Elsevier* 2008.

[10] C. Chow, C. Liu, "Approximating discrete probability distributions with dependence trees", *Fifteenth IEEE Transactions on Information Theory* 14 (3) (1968) 462–467, May.

[11] S.Ioffe, D.Forsyth, "Mixtures of trees for object recognition", *Computer Vision and Pattern Recognition*, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference.

[12] J. Pearl, "Probabilistic reasoning in intelligent systems: networks of plausible inference". *Morgan Kaufmann*, 1988.