MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU

FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

MEMOIRE DE MAGISTER

En vue de l'obtention du diplôme de Magister en Electronique

Option télédétection

Présenté par :

M[r] BERBECHE Kamal

Intitulé :

Modèles de Markov Cachés : Application à La Reconnaissance Automatique de la Parole.

Devant le Jury d'examen composé de :

Mr Laghrouche Mourad	Professeur à l'UMMTO	Président
Mr Haddab Salah	Maître de conférences A à l'UMMTO	Rapporteur
Mr Hammouche Kamal	Professeur à l'UMMTO	Examinateur
Mme Ameur Zohra	Professeur à l'UMMTO	Examinateur
Mr Lazri Mourad	Maître de Conférences B à l'UMMTO	Examinateur

REMERCIEMENTS

La réalisation de ce mémoire en vue de l’obtention du diplôme de Magister en électronique a été rendue possible grâce au soutien de plusieurs personnes à qui je voudrai témoigner ma reconnaissance, leurs disponibilités et leurs compétences m'ont permis de franchir beaucoup d’obstacles, qu’ils trouvent ici le témoignage de ma gratitude et mes remerciements les plus sincères.

Je voudrais tout d'abord adresser tous mes remerciements à mon directeur de mémoire Monsieur HADDAB Salah, Maitre de Conférences A à l’université de Tizi Ouzou pour son immense patience, sa grande disponibilité et ses conseils qui ont contribué grandement à la réalisation de ce travail. Qu’il trouve ici l’expression de ma profonde gratitude.

J’exprime mes sincères remerciements et ma profonde gratitude à Mr Laghrouche Mourad, professeur à l’Université Mouloud MAMMERI de Tizi-ouzou pour l’honneur qu’il me fait en présidant ce jury.

Je tiens à remercier chaleureusement Mme Ameur Zohra, professeur à l’Université Mouloud MAMMERI de Tizi-ouzou, pour avoir accepté de faire partie du jury.

Également, j’exprime ma profonde gratitude à Mr Hammouche Kamal, professeur à l’Université Mouloud MAMMERI de Tizi-ouzou, pour avoir accepté de faire partie du jury.

J’adresse mes vifs remerciements à Mr Lazri Mourad, Maitre de conférance B à l’Université Mouloud MAMMERI de Tizi-ouzou, pour avoir accepter aussi de faire partie du jury.

Mes remerciements et ma gratitude aux responsables, chercheurs du laboratoire LAMPA.

Je désir aussi remercier les enseignants du département électronique de l’université de tizi-ouzou qui m’ont fourni les outils et les connaissances nécessaires à la réussite de mes études universitaires.

Je voudrais exprimer ma plus haute reconnaissance à mes parents et à toute ma famille pour leur soutien, leur aide et leur patience.

Finalement, je n’oublierai pas de citer tous mes amis et collègues qui m'ont toujours soutenu et encouragé tout au long de cette démarche.

TABLES DES MATIÈRES

Table des matières

CHAPITRE III : LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

CHAPITRE IV : LES MODÈLES DE MARKOV CACHÉS

GLOSSAIRE

LPC : Coefficients de prédiction linéaire.

LPCC : Coefficients cepstrales de prédiction linéaire.

MFCC : Coefficients cepstrales à échelle fréquentielle de Mel.

DCT : Transformée en cosinus discrète.

PLP : Prédiction linéaire perceptuelle.

RASTA : Analyse spectrale relative.

JRASTA: Analyse spectral relative au bruit additif.

MRA: Analyse à resolution multiple.

ANN: Artificiel Neural Network, Réseau de Neurones artificiel.

GMM: Gaussian mixture model, model de mixture de gaussiennes.

MMC: Model de Markov Cachés.

HMM: Hidden markov Model.

MAP: Maximum à posteriori.

ML: Maximum de vraissamblance.

EM: ExpectationMaximisation.

SEM: Expectation Maximisation Stochastique.

ICE: Estimation Conditionnelle itérative.

MIM: Maximisation de l'information mutuelle.

INTRODUCTION GÉNÉRALE

Le traitement de la parole est, aujourd'hui, une composante fondamentale des sciences de l'ingénieur. Située au croisement du traitement du signal numérique et du traitement du langage (c'est-à-dire du traitement de données symboliques), cette discipline scientifique a connu, depuis les années 60, une expansion fulgurante, liée au développement des moyens et des techniques de télécommunications.

L'importance particulière du traitement de la parole s'explique par la position privilégiée de la parole comme vecteur d'information dans notre société humaine.

L'extraordinaire singularité de cette science, qui la différencie fondamentalement des autres composantes du traitement de l'information, tient, sans aucun doute, au rôle fascinant que joue le cerveau humain à la fois dans la production et dans la compréhension de la parole et à l'étendue des fonctions qu'il met en oeuvre.

Après plus de soixante années de recherches et de développement industriel, les performances des systèmes de reconnaissances automatiques de la parole (RAP) se sont considérablement améliorées, permettant d'aborder des domaines d'application de complexité croissante. Les travaux actuels les plus avancés concernent des systèmes de dialogue via le téléphone, la reconnaissance de la parole spontanée ou la transcription d'émissions de radio ou télévision. Les performances obtenues dépendent beaucoup du type de tâche considérée (taille et difficulté du vocabulaire, nombre et diversités des locuteurs, conditions d'enregistrement). Le traitement automatique de la parole a été, dès l'origine fortement, tributaire de l'évolution technologique.

S'il n'est pas en principe de parole sans cerveau humain pour la produire, l'entendre, et la comprendre, les techniques modernes de traitement de la parole tendent cependant à produire des systèmes automatiques qui se substituent à l'une où l'autre de ces fonctions.

Ainsi, les analyseurs de parole cherchent à mettre en évidence les caractéristiques du signal vocal tel qu'il est produit, ou parfois tel qu'il est perçu, mais jamais tel qu'il est compris. D'un autre coté les reconnaisseurs ont pour mission de décoder l'information portée par le signal vocal à partir de données fournies par l'analyse. On distingue fondamentalement deux types de reconnaissance, en fonction de l'information que l'on cherche à extraire du signal vocal : La reconnaissance du locuteur, avec pour objectif est reconnaitre la personne qui parle, et la reconnaissance de la parole, où l'on s'attache plutôt à reconnaitre ce qui est dit.

On classe également les reconnaisseurs en fonction des hypothèses simplificatrices sous lesquelles ils sont appelés à fonctionner :

En reconnaissance du locuteur, on fait la différence entre la vérification et l'identification du locuteur, selon que le problème est de vérifier que la voix analysée correspond bien à la personne qui est sensée la produire, ou qu'il s'agisse de déterminer qui, parmi un nombre fini et préétabli de locuteur, a produit le signal analysé.

Par ailleurs, on distingue la reconnaissance du locuteur dépendante du texte, reconnaissance avec texte dicté, et reconnaissance indépendante du texte. Dans le premier cas, la phrase à prononcer, pour être reconnu, est fixée dès la conception du système ; elle est fixée lors du test dans le deuxième cas, et n'est pas précisée dans le troisième.

On parle également de reconnaisseur de parole monolocuteur, multilocuteur, ou indépendant du locuteur, selon qu'il a été entrainé à reconnaitre la voix d'une personne, d'un groupe fini de personnes, ou qu'il est, en principe, capable de reconnaitre n'importe qui.

On distingue enfin reconnaisseur de mots isolés, reconnaisseur de mots connectés, et reconnaisseur de parole continue, selon que le locuteur sépare chaque mot par un silence, qu'il prononce de façon continue une suite de mots prédéfinis, ou qu'il prononce n'importe quelle suite de mots de façon continue.

Dans ce mémoire, nous consacrons le premier chapitre aux généralités sur la parole, sa production et perception chez l'être humain, son acquisition et ses traitements et analyse.

Dans le second chapitre, nous décrivons les paramètres acoustiques pertinents du signal de parole. Par la suite, le chapitre trois est consacré à la description des systèmes de reconnaissance automatique de la parole.

Dans le quatrième chapitre, nous allons introduire les Modèles de Markov Cachés et leurs algorithmes, critères qui sont à la base des systèmes modernes de reconnaissance automatique de la parole.

Nous terminons, dans le chapitre Cinq, par décrire l'application réalisée qui consiste à la reconnaissance automatique de la parole sous matlab et sous HTK.

CHAPITRE I : GÉNÉRALITÉS SUR LA PAROLE

Introduction

L'information portée par le signal de parole peut être analysée de bien des façons. On distingue, généralement, plusieurs niveaux de description non exclusifs : Acoustique, phonétique et bien d'autres [1].

Dans ce chapitre nous allons, dans un premier temps, décrire les processus de production et de perception auditive de la parole puis nous donnerons un aperçu sur les notions de phonétique. Nous terminerons par la conversion de la parole en signal électrique et nous rappellerons quelques outils de base utilisés en traitement de signaux acoustiques.

I.1. Production de la parole

La parole peut être décrite comme le résultat de l'action volontaire et coordonnée d'un certain nombre de muscles des appareils respiratoires et articulatoires [1]. Cette action se déroule sous le contrôle du système nerveux central qui reçoit, en permanence, des informations par rétroaction auditive et par les sensations kinesthésiques [2].

I.1.1. Le processus de production

De façon simple, on peut résumer le processus de production de la parole comme un système dans lequel une ou plusieurs sources excitent un ensemble de cavités. La source sera soit générée au niveau des cordes vocales soit au niveau d'une constriction du conduit vocal.

Dans le premier cas, la source résulte d'une vibration quasi-périodique des cordes vocales et produit ainsi une onde de débit quasi-périodique.

Dans le second cas, la source sonore est soit un bruit de friction soit un bruit d'explosion qui peut apparaître s'il y a un fort rétrécissement dans le conduit vocal où si un brusque relâchement d'une occlusion du conduit vocal s'est produit.

L'ensemble de cavités situées après la glotte, dites les cavités supraglottiques, vont ainsi être excitées par la ou les sources et "filtrer" le son produit au niveau de ces sources. Ainsi, en changeant la forme de ces cavités, l'homme peut produire des sons différents. Les acteurs de cette mobilité du conduit vocal sont communément appelés les articulateurs.

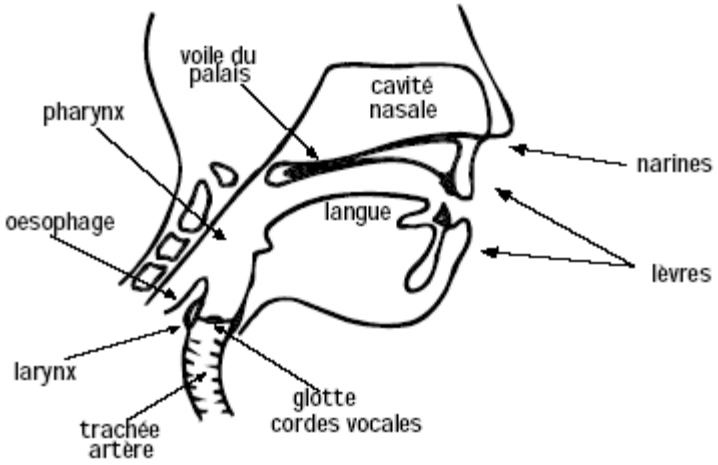


Fig.1.1-L'appareil phonatoire.

On peut donc résumer le processus de production de la parole en trois étapes essentielles

- La génération d'un flux d'air qui va être utilisé pour faire naître une source sonore (au niveau des cordes vocales ou au niveau d'une constriction du conduit vocal) C'est le rôle de la soufflerie.
- La génération d'une source sonore sous la forme d'une onde quasi-périodique résultant de la vibration des cordes vocales ou/et sous la forme d'un bruit résultant d'une constriction ou d'un brusque relâchement ou occlusion du conduit vocal : C'est le rôle de la source vocale.
- La mise en place des cavités supraglottiques (conduits nasal et vocal) pour obtenir le son désiré (c'est principalement le rôle des différents articulateurs du conduit vocal).

I.1.2. Les différentes étapes de production de la parole

L'appareil respiratoire fournit l'énergie nécessaire à la production de sons, en poussant de l'air à travers la tranchée-artère. Au sommet de celle-ci se trouve le larynx où la pression de l'air est modulée avant d'être appliquée au conduit vocal. Le larynx est un ensemble de muscles et de cartilages mobiles qui entourent une cavité située à la partie supérieure de la tranchée (fig.1.1). Les cordes vocales sont en faite deux lèvres symétriques placées en travers du larynx. Ces lèvres peuvent fermer complètement le larynx et, en s'écartant progressivement, déterminer une ouverture triangulaire appelée glotte. L'air y passe librement pendant la respiration et la voix chuchotée, ainsi que pendant la phonation des sons non voisés. Les sons voisés résultent au contraire d'une vibration périodique des cordes vocales.

Le larynx est d'abord complètement fermé, ce qui accroit la pression en amont des cordes vocales, et les force à s'ouvrir, ce qui fait tomber la pression, et permet aux cordes vocales de se refermer ; des impulsions périodiques de pression sont ainsi appliquée, au conduit vocal, composé des cavités pharyngienne et buccale pour la plupart des sons. Lorsque la luette est en position basse, la cavité nasale vient s'y ajouter en dérivation.

Dans la suite de cette section, nous allons définir au mieux les organes intervenants dans ce processus.

I.1.3. Les organes de production de la parole

La parole est essentiellement produite par deux types de sources vocales. La première, plus sonore, est celle qui prend naissance au niveau du larynx suite à la vibration des cordes vocales. La seconde, moins sonore, prend naissance au niveau d'une constriction du conduit vocal ou lors d'un relâchement brusque d'une occlusion du conduit vocal. On parlera dans ce cas de sources de bruit.

I.1.3.1. Le larynx

Le larynx est un organe situé dans le cou qui joue un rôle crucial dans la respiration et dans la production de parole. Le larynx (fig.1.2) est plus spécifiquement situé au niveau de la séparation entre la trachée artère et le tube digestif, juste sous la racine de la langue. Sa position varie avec le sexe et l'âge : il s'abaisse progressivement jusqu'à la puberté et il est sensiblement plus élevé chez la femme.

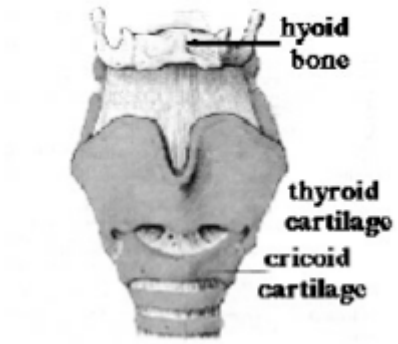


Fig.1.2-Schéma du larynx

Il est constitué d'un ensemble de cartilages, il est constitué d'un ensemble de cartilages entourés de tissus mous. La partie la plus proéminente du larynx est formée du thyroïde. La

partie antérieure de cartilage est communément appelée la "pomme d'Adam". On trouve, juste au dessus du larynx, un os en forme de 'U' appelé l'os hyoid. Cet os relie le larynx à la mandibule par l'intermédiaire de muscles et de tendons qui joueront un rôle important pour élever le larynx pour la déglutition ou la production de parole.

La partie inférieure du larynx est constituée d'un ensemble de pièces circulaires, le cricoïde, sous lequel on trouve les anneaux de la trachée artère.

Le larynx assure ainsi trois fonctions essentielles :

- Le contrôle du flux d'air lors de la respiration
- La protection des voies respiratoires
- La production d'une source sonore pour la parole

- **Les muscles du larynx**

Les mouvements du larynx sont contrôlés par deux groupes de muscles. On distingue ainsi les muscles intrinsèques, qui contrôlent le mouvement des cordes vocales et des muscles à l'intérieur du larynx, et les muscles extrinsèques, qui contrôlent la position du larynx dans le cou.

La figure I.3 nous représente les muscles intrinsèques. Les cordes vocales sont ouvertes par une paires de muscles (les muscles cricoaryténoide postérieur) qui sont situés entre la partie arrière du cricoïde et le cricoarytenoïde.

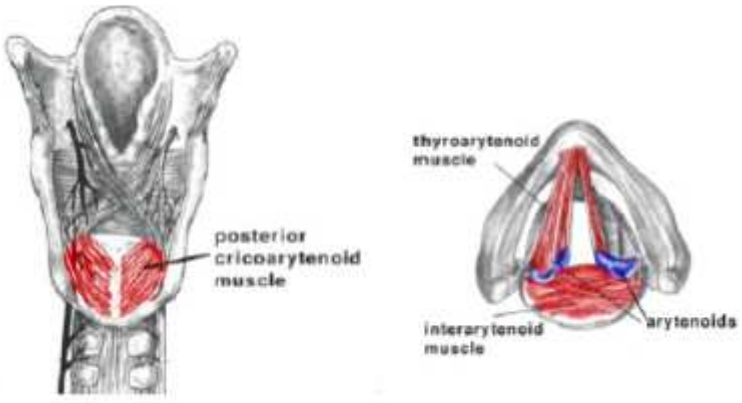


Fig.1.3 Schéma des muscles intrinsèques du larynx

- **Les cordes vocales**

Les cordes vocales situées au centre du larynx ont un rôle fondamental dans la production de la parole.

Elles sont constituées de muscles recouverts d'un tissus assez fin couramment appelé la muqueuse. Sur la partie arrière de chaque corde vocale, on trouve une petite structure faite de cartilages : Les aryténoides. De nombreux muscles y sont rattachés qui permettent de les écarter pour assurer la respiration.

Durant la production de parole, les aryténoides sont rapprochés (voir figure I.3). Sous la pression de l'air provenant des poumons, les cordes vocales s'ouvrent puis se referment rapidement. Ainsi, lorsqu'une pression soutenue de l'air d'expiration est maintenue, les cordes vocales vibrent et produisent un son qui sera par la suite modifié dans le conduit vocal pour donner lieu à un son voisé. Ce processus de vibration des cordes vocales est décrit un peu plus en détail ci-aprés.

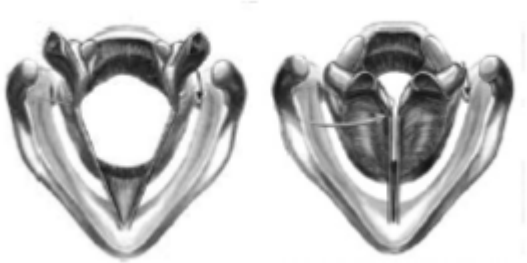


Fig.1.4 Les cordes vocales en position ouvertes durant la respiration (à gauche) et fermés pour la production de parole (à droite)

Plusieurs muscles aident pour fermer et tendre les cordes vocales. Les cordes vocales sont elles même constituées d'un muscle, le thyroaryténoide. Un autre muscle, l'interaryténoide , permet de rapprocher ces deux cartilages. Le muscle cricoaryténoide latéral qui est lui aussi situé entre l'aryténoide et le cartilage cricoïde sert à la fermeture du larynx.

Le muscle cricothyroide va du cartilage cricoide jusqu'au cartilage thyroide. Lorsqu'il se contracte, le cartilage cricoid bascule en avant et tend les cordes vocales ce qui résultera à un élèvement de la voix.

Les muscles extrinsèques n'affectent pas le mouvement des cordes vocales mais élèvent ou abaissent le larynx dans sa globalité.

La figure 1.5 donne une vue schématique d'une coupe verticale du larynx. Sur ce schéma, les cordes vocales sont ici clairement séparées, comme elles seraient durant la respiration. On peut également remarquer au-dessus des cordes vocales, des tissus ayant pour principal rôle d'éviter le passage de substances dans la trachée durant la déglutition : ce sont les fausses cordes vocales. Il est important de noter qu'elles ne jouent aucun rôle lors de la phonation. Le cartilage mou en forme grossière de langue qui se trouve au-dessus est appelé l'épiglotte et a également un rôle pour protéger l'accès de la trachée lors de la déglutition.

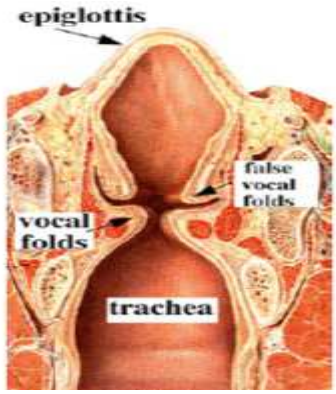


Fig.1.5-Vue longitudinale du larynx

I.1.3.2. Les cavités supraglottiques

D'autres organes situés au dessus de la glottes (organes supraglottiques) interviennent également, meme à un degré moindre, dans la production du son. On distingue, ainsi :

- **Le conduit vocal**

Considéré comme un tube acoustique de section variable qui s'étend de la glotte jusqu'aux lèvres. Pour un adulte, le conduit vocal mesure environ 17 cm. Sa forme varie en fonction du mouvement des articulateurs qui sont les lèvres, la mâchoire, la langue et le velum. Ces articulateurs sont brièvement décrits ci-dessous.

- **Le conduit nasal**

Le conduit nasal est un passage auxiliaire pour la transmission du son. Il commence au niveau du velum et se termine aux niveaux des fosses nasales. Pour un homme adulte, cette cavité mesure environ 12 cm.

Le couplage acoustique entre les deux cavités est contrôlé par l'ouverture au niveau du velum (figure I.1). On notera que le velum -ou voile du palais- est largement ouvert. Dans ce

cas, on aura la production d'un son nasal. Dans le cas contraire, lorsque le velum ferme le conduit nasal le son produit sera dit non-nasal.

D'autre organes, dits articulateurs, joue également un rôle chacun en ce qui le concerne. Les articulateurs sont :

- **La langue**

La langue est une structure frontière, appartenant à la fois à la cavité buccale pour sa partie dite mobile et au glosso-pharynx pour sa partie dite fixe, qui appliquée contre le palais ou les dents constituent un organe vibratoire accessoire, intervenant dans la formation des consonnes. Elle a donc de l'importance pour la phonation.

On comprend que la langue est un articulateur fondamental puisque sa position est déterminante dans le conduit vocal.

- **La mâchoire**

La mâchoire possède un nombre de degrés de liberté plus faible et étant un corps rigide ne peut pas se déformer comme la langue. Néanmoins, la mâchoire peut non seulement s'ouvrir et se fermer, mais peut également s'avancer ou effectuer des mouvements de rotation.

Son rôle dans la parole n'est cependant pas primordial dans la mesure où il est possible en bloquant la mâchoire de parler de façon très intelligible.

- **Les lèvres**

Les lèvres sont situées à l'extrémité du conduit vocal et comme pour la langue, elles possèdent une grande mobilité en raison des nombreux muscles impliqués dans leur contrôle. Les points de jonction des lèvres supérieure et inférieure s'appellent les commissures et jouent un grand rôle dans la diplomatie (pour le sourire, bien sur...).

Au point de vue acoustique, c'est l'espace intérolabial qui est important. On peut observer différents mouvements importants pour la phonation dont :

- l'occlusion (les lèvres sont fermées)
- La protrusion (les lèvres sont avancées vers l'avant)

- l'élévation et l'abaissement de la lèvre inférieure
- l'étirement, l'abaissement ou l'élévation des commissures

I.1.4. Les sons de la parole par l'approche production

Dans ce qui suit, nous allons s'intéresser aux différentes classes de sons au niveau phonétique tout en expliquant comment ces sons sont produits.

- **Notions de phonétique**

La parole, qu'elle qu'en soit la langue, est constituée d'un nombre finis d'éléments sonores distinctifs. Ces éléments forment les unités linguistiques élémentaires et ont la propriété de changer le sens d'un mot. Ces unités élémentaires sont appelés phonèmes [3].

Un phonème est donc la plus petite unité phonique fonctionnelle, c'est-à-dire distinctive. Il n'est pas défini sur un plan acoustique, articulatoire, ou perceptuel, mais bien sur le plan fonctionnel. Les phonèmes n'ont pas d'existence indépendante : Ils constituent un ensemble structuré dans lequel chaque élément est intentionnellement différent de tous les autres, la différence étant à chaque fois porteuse de sens. La liste des phonèmes pour la plupart des langues européennes a été établie dès la fin du 19è siècle.

Les phonèmes peuvent ainsi être vus comme les éléments de base pour le codage de l'information linguistique.

Cependant, ces phonèmes peuvent se regrouper en classes dont les éléments partagent des caractéristiques communes. On parlera ici de "traits distinctifs".

- **Trait distinctif :** Un trait distinctif est l'expression d'une similarité au niveau articulatoire, acoustique ou perceptif des sons concernés.

Par exemple, pour les voyelles on distinguera 4 traits distinctifs :

- La nasalité : la voyelle a été prononcée à l'aide du conduit vocal et du conduit nasal suite à l'ouverture du velum.
- Le degré d'ouverture du conduit vocal
- La position de la constriction principale du conduit vocal, cette constriction étant réalisée entre la langue et le palais.
- La protrusion des lèvres.

De même, les consonnes seront classées à l'aide de 3 traits distinctifs :

- Le voisement : la consonne a été prononcée avec une vibration des cordes vocales
- Le mode d'articulation (on distinguera les modes occlusif, fricatif, nasal, glissant ou liquide).
- La position de la constriction principale du conduit, souvent appelée lieu d'articulation qui contrairement aux voyelles n'est pas nécessairement réalisé avec le corps de la langue.

En fait, les phonèmes (qui peuvent donc être décrits suivant leurs traits distinctifs) sont des éléments abstraits associés à des sons élémentaires. Bien entendu, les phonèmes ne sont pas identiques pour chaque langue et le /a/ du français n'est pas totalement équivalent au /a/ de l'anglais. Ainsi, est née l'idée de définir un alphabet phonétique international (alphabet IPA) qui permettrait de décrire les sons et les prononciations de ces sons de manière compacte et universelle.

Il existe d'autres façons d'organiser les sons, par exemple en opposant les sons sonnants (voyelles), les consonnes nasales, les liquides ou les glissantes » aux sons obstruants « occlusives, fricatives ».

- **Les voyelles**

Les voyelles sont typiquement produites en faisant vibrer ses cordes vocales. Le son de telle ou telle voyelle est alors obtenu en changeant la forme du conduit vocal à l'aide des différents articulateurs. Dans un mode d'articulation normal, la forme du conduit vocal est maintenue relativement stable pendant quasiment toute la durée de la voyelle.

- **Les consonnes**

Comme pour les voyelles, les consonnes vont pouvoir être regroupées en traits distinctifs. Contrairement aux voyelles, elles ne sont pas exclusivement voisées (même si les voyelles prononcées en voix chuchotée sont, dans ce cas également, non voisées) et ne sont pas nécessairement réalisées avec une configuration stable du conduit vocal.

- **Les consonnes voisées**

On parlera de consonnes voisées lorsqu'elles sont produites avec une vibration des cordes vocales « comme par exemple /b/ dans "bol" où les cordes vocales vibrent avant le

relâchement de la constriction ». Lorsqu'en plus du voisement, une source de bruit est présente due à une constriction du conduit vocal, on pourra parler de consonnes à excitation mixte « c'est le cas par exemple du /v/ dans "vent" ».

– **Les fricatives**

Sont produites par un flux d'air turbulent prenant naissance au niveau d'une constriction du conduit vocal. On distingue plusieurs fricatives suivant le lieu de cette constriction principale :

- Les labio-dentales, pour une constriction réalisée entre les dents et les lèvres « comme pour le /f/ dans "foin" ».
- Les dentales, pour une constriction au niveau des dents « comme pour le /t/ anglais dans "thin" »
- Les alvéolaires, pour une constriction juste derrière les dents « comme pour le /s/ dans "son" ».

En fait, suivant les langues, en regardant plusieurs langues, on s'aperçoit que quasiment tous les points d'articulations du conduit vocal peuvent être utilisés pour réaliser des fricatives.

C'est d'ailleurs l'une des difficultés de l'apprentissage des langues étrangères car il n'est pas aisé d'apprendre à réaliser des sons qui demandent de positionner la langue à des endroits inhabituels.

– **Les plosives**

Elles sont caractérisées par une dynamique importante du conduit vocal. Elles sont réalisées en fermant le conduit vocal en un endroit. L'air provenant des poumons crée alors une pression derrière cette occlusion qui est ensuite soudainement relâchée suite au mouvement rapide des articulateurs ayant réalisé cette occlusion. De même, que pour les fricatives, l'un des traits distinctifs entre les plosives est le lieu d'articulation. Pour les plosives, on aura ainsi :

- Les labiales, pour une occlusion réalisée au niveau des lèvres.
- Les dentales, pour une occlusion au niveau des dents.
- Les vélo-palatales, pour une occlusion au niveau du palais.

En plus du lieu d'articulation, les plosives peuvent également être voisées ou non voisées.

Ainsi, une dentale voisée « /d/ » se distinguera uniquement par la présence de voisement « vibration des cordes vocales du /t/ » qui est prononcée avec le même lieu d'articulation.

- **Les consonnes nasales**

Elles sont en général voisées et sont produites en effectuant une occlusion complète du conduit vocal et en ouvrant le vélum permettant au conduit nasal d'être l'unique résonateur. Comme pour les autres consonnes, on aura, suivant le lieu d'articulation :

- Les labiales, pour une occlusion du conduit vocal réalisée au niveau des lèvres.
- Les dentales, pour une occlusion du conduit vocal au niveau des dents.
- Les vélo-palatales, pour une occlusion du conduit vocal au niveau du palais.

- **Les glissantes et les liquides**

Cette classe de consonnes regroupe des sons qui ressemblent aux voyelles. Les liquides sont d'ailleurs parfois appelées semi consonnes ou semi-voyelles. Les glissantes et les liquides, en général, voisées et non nasales.

- Les glissantes, comme leur nom l'indique, sont des sons en mouvement et précédent toujours une voyelle « ou un son vocalique ».
- Les liquides « ou semi-voyelles » sont des sons tenus, très similaires aux voyelles mais en général avec une constriction plus conséquente et avec l'apex de la langue plus relevé.

I.2. Audition-perception des sons de parole

Le son, et en particulier la parole, étant le moyen de communication privilégié pour l'être humain, nous ne pourrons pas décrire le phénomène sans aborder la notion d'audition, c'est-à-dire de réception et d'interprétation du son. L'organe de perception du son est l'oreille.

I.2.1. Structure de l'oreille

L'oreille est séparée en 3 parties principales comme indiqué sur le schéma de l'appareil auditif de la figure1.6

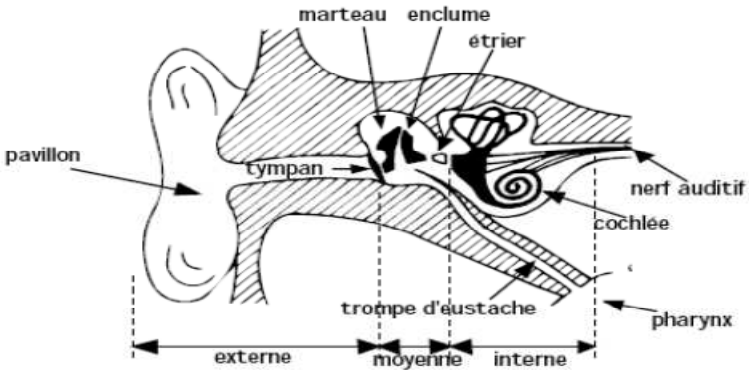


Fig.1.6- L'appareil auditif

- **L'oreille externe** : Allant du pavillon au tympan et réalisant une conduction aérienne.
- **L'oreille moyenne** : Constituée de 3 osselets « le marteau, l'enclume et l'étrier » s'étend du tympan à la fenêtre ovale et réalise une adaptation d'impédance pour transmettre les ondes acoustiques aériennes reçues au niveau de l'oreille externe vers l'oreille interne.
- **L'oreille interne** : dans laquelle se trouve la cochlée. La cochlée joue un rôle primordial dans la perception des sons. En effet, un son parvenant au pavillon de l'oreille sera transformé en vibration au niveau de l'entrée de la cochlée.

I.2.2. Principe de perception auditive

La parole peur être décrite comme le résultat de l'action volontaire et coordonnée d'un certain nombre de muscles. Cette action se déroule sous le contrôle du système nerveux central qui reçoit en permanence des informations par rétroaction auditive et par les sensations kinesthésiques, ce principe est présenté sur la figure 1.7.

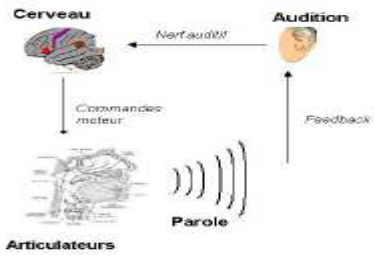


FIGURE I.10 – Système de production et feedback auditif

Fig.1.7- Système de production et la rétroaction auditif.

Les ondes sonores sont recueillies par l'appareil auditif, ce qui provoque les sensations auditives. Ces ondes de pression sont analysées dans l'oreille interne qui envoie au cerveau l'influx nerveux qui en résulte. Le phénomène physique induit alors un phénomène psychique grâce à un mécanisme physiologique complexe [3].

Le mécanisme de l'oreille interne (marteau, étrier, enclume) permet une adaptation d'impédance entre l'air et le milieu liquide de l'oreille interne. Les vibrations de l'étrier sont transmises au liquide de la cochlée. Celle-ci contient la membrane basilaire qui transforme les vibrations mécaniques en impulsions nerveuses. La membrane s'élargit et s'épaissit au fur et à mesure que l'on se rapproche de l'apex de la cochlée.

Les fibres nerveuses aboutissent à une région de l'écorce cérébrale, appelée aire de projection auditive, et située dans le lobe temporal. En cas de lésion de cette aire, on peut observer des troubles auditifs. Les fibres nerveuses auditives afférentes « de l'oreille au cerveau » et efférentes « du cerveau vers l'oreille » sont partiellement croisées : chaque moitié du cerveau est mise en relation avec les deux oreilles internes.

. Entre l'arrivée des signaux vibratoires aux oreilles et la sensation du son dans le cerveau, a lieu le phénomène de traitement des signaux par le système nerveux. Cela signifie que la vibration physique de l'air ne parvient pas de façon brute au cerveau. Elle est transformée, Comme décrit sur la figure 1.8.

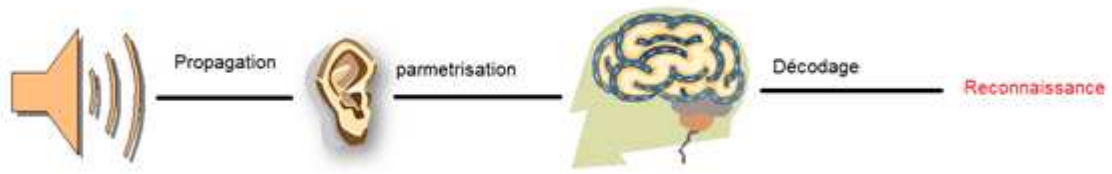


Fig.1.8-Perception et analyse du son par l'être humain.

Il reste très difficile de nos jours de dire comment l'information auditive est traitée par le cerveau. On a pu par contre étudier comment elle était finalement perçue, dans le cadre d'une science spécifique appelée psychoacoustique [4]. Sans vouloir entrer dans trop de détails sur la contribution majeure des psychoacousticiens dans l'étude de la parole, il est intéressant d'en connaître les résultats les plus marquants.

Ainsi, l'oreille ne répond pas également à toutes les fréquences. La figure 1.9 présente le champ auditif humain, délimité par la courbe de seuil de l'audition et celle du seuil de la

douleur. Sa limite supérieure en fréquence (~16000 Hz, variable selon les individus) fixe la fréquence d'échantillonnage maximale utile pour un signal auditif (~ 32000 Hz).

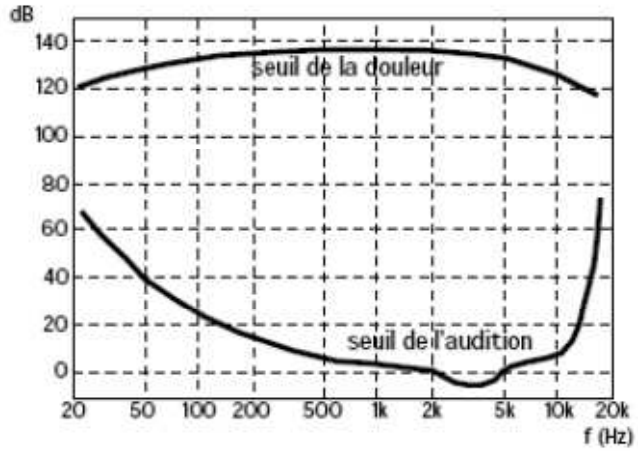


Fig.1.9- Champs auditif humain.

A l'intérieur de son domaine d'audition, l'oreille ne présente pas une sensibilité identique à toutes les fréquences. La figure 1.10, fait apparaître les courbes d'égale impression de puissance auditive - physiologie auditive (aussi appelée ***sonie***, exprimée en ***sones***) en fonction de la fréquence. Elles révèlent un maximum de sensibilité dans la plage [500 Hz, 10 kHz], en dehors de laquelle les sons doivent être plus intenses pour être perçus.

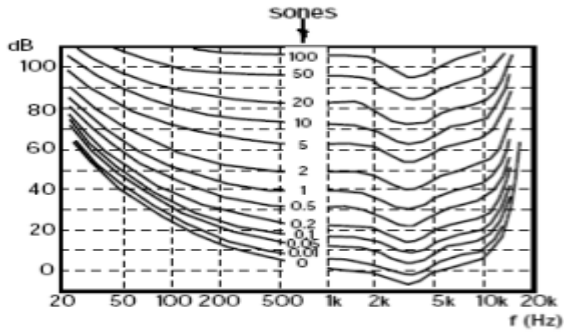


Fig.1.10

I.3. Traitement de la parole

De façon générale, le traitement du signal est un ensemble de méthodes et de techniques agissant sur un signal électrique afin d'en extraire l'information désirée. Ce signal doit traduire le plus fidèlement possible le phénomène physique à étudier.

La parole apparait physiquement comme une variation de l'air causée et émise par le système articulatoire. C'est un phénomène physique acoustique qui prend une forme analogique.

La phonétique acoustique étudie ce signal en le transformant dans un premier temps en signal électrique grâce au transducteur approprié : le microphone (lui-même associé à un préamplificateur).

De nos jours, le signal électrique résultant est le plus souvent numérisé. Il peut alors être soumis à un ensemble de traitements, dans le but d'en extraire les informations et les paramètres pertinents en rapport avec l'application. Ainsi, la conversion du phénomène de parole en signal électrique nécessite les opérations suivantes.

I.3.1. Numérisation

La numérisation du signal de parole est à présent assurée par un convertisseur analogique- numérique (**CAN**)

Cette opération, schématisée à la figure1.11, requiert successivement : un filtrage de garde, un échantillonnage, une quantification et un codage.

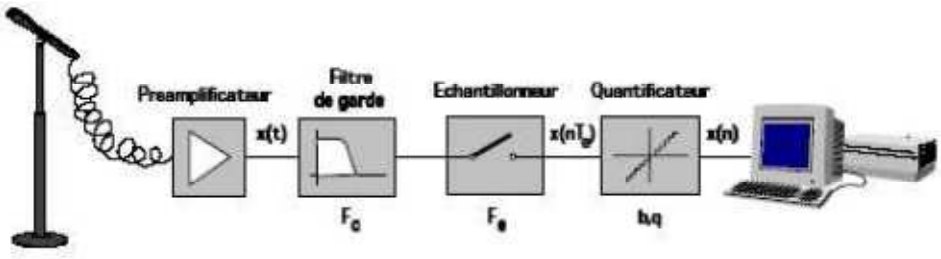


Fig.1.11- Enregistrement numérique d'un signal acoustique.

I.3.2. L'échantillonnage

L'échantillonnage transforme le signal à temps continu $\boldsymbol{x(t)}$ en signal à temps discret $\boldsymbol{x(nT_e)}$ défini aux instants d'échantillonnage, multiples entiers de la période d'échantillonnage $\boldsymbol{T_e}$; celle-ci est elle-même l'inverse de la fréquence d'échantillonnage $\boldsymbol{f_e}$.

En ce qui concerne le signal vocal, le choix de $\boldsymbol{f_e}$ résulte d'un compromis. Son spectre peut s'étendre jusqu'à 12kHz. Il faut donc en principe choisir une fréquence $\boldsymbol{f_e}$ **é**gale à 24kHz au moins pour satisfaire raisonnablement au théorème de shannon. Cependant, le cout d'un traitement numérique, filtrage, transmission, ou simplement enregistrement peut être réduit d'une façon notable si l'on accepte une limitation du spectre par un filtrage préalable. C'est le rôle du filtre de garde, dont la fréquence de coupure $\boldsymbol{f_c}$ est choisie en fonction de la fréquence d'échantillonnage retenue.

I.3.3. La Quantification

Cette étape consiste à approximer les valeurs réelles des échantillons selon une échelle de n niveaux appelée échelle de quantification.

Parmi le continuum des valeurs possibles pour les échantillons($\boldsymbol{nT_e}$)**,** la quantification ne retient qu'un nombre fini $\boldsymbol{2^n}$ de valeurs, espacées du pas de quantification **δ**. Le signal numérique résultant est noté $\boldsymbol{x(n)}$**.** La quantification produit une erreur de quantification qui normalement se comporte comme un bruit blanc, le pas de quantification est donc imposé par le rapport signal à bruit à garantir. Aussi adopte-t-on pour la transmission téléphonique une loi de quantification logarithmique et chaque échantillon est représenté sur 8 bits. Par contre, la quantification du signal musical exige en principe une quantification linéaire sur 16 bits.

I.3.4. Le Codage

C'est la représentation binaire des valeurs quantifiées qui permet le traitement du signal sur machine.

I.4. Analyse du signal de parole

Une fois numérisé, le signal de parole peut être traité de différentes façons suivant les objectifs visés. Le nombre de techniques possible étant très vaste, nous allons, dans ce qui suit, citer les outils relatifs au signal de parole.

I.4.1. Analyse temporelle

Le signal de parole est un signal quasi-stationnaire. Cependant, sur un horizon de temps supérieur, il est clair que les caractéristiques du signal évoluent significativement en fonction des sons prononcés comme illustré sur la figure ci-dessous.

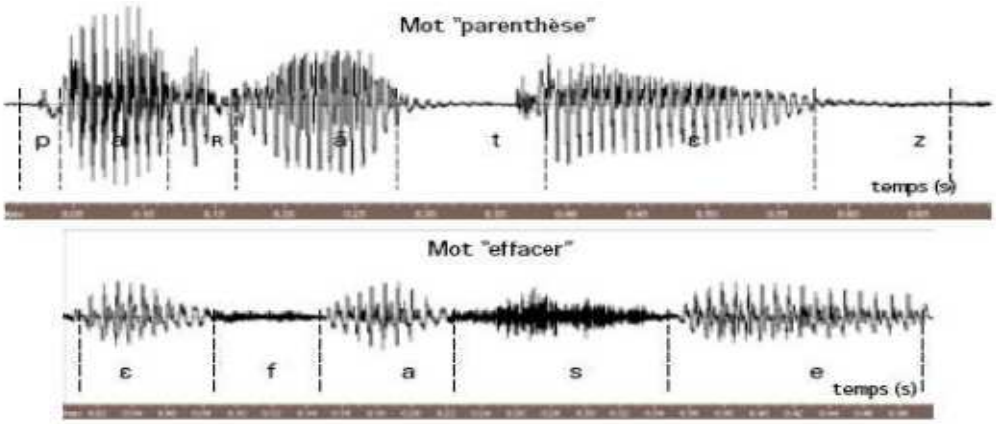


Fig.1.12- Représentation temporelle(Audiogramme) de signaux de parole.

La première approche pour étudier le signal de parole consiste à observer la forme temporelle du signal. On peut à partir de cette forme temporelle en déduire un certain nombre de caractéristiques qui pourront être utilisées pour le traitement de la parole. Il est, par exemple, assez clair de distinguer les parties voisées, dans lesquelles on peut observer une forme d'onde quasi-périodique, des parties non voisées dans lesquelles un signal aléatoire de faible amplitude est observé. De même, on peut voir que les petites amplitudes sont beaucoup plus représentées que les grandes amplitudes ce qui pourra justifier des choix fait en codage de la parole.

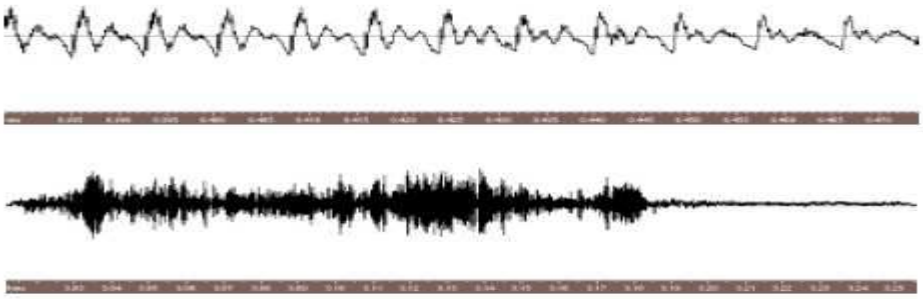


Fig.1.13- Exemple de son voisé (haut) et non voisé (bas).

I.4.2. Analyse fréquentielle

Une seconde approche pour caractériser et représenter le signal de parole est d'utiliser une représentation spectrale.

On peut classer en deux grandes catégories les méthodes de traitement du signal :

- **les méthodes générales** : valables pour tout signal évolutif dans le temps, en particulier les analyses spectrales.
- **les méthodes se référant à un modèle** : un modèle de production du signal vocal ou un modèle d'audition.

- **Méthodes générales**

Les méthodes spectrales occupent une place prépondérante en analyse de la parole : l'oreille effectue, entre autres, une analyse fréquentielle du signal qu'elle perçoit ; de plus, les sons de la parole peuvent être assez bien décrits en termes de fréquences.

La transformée de Fourier permet d'obtenir le spectre d'un signal, en particulier son spectre fréquentiel, c'est-à-dire sa représentation amplitude-fréquence.

La figure 1.14 illustre la transformée de Fourier d'une tranche voisée et celle d'une tranche non voisée. Les parties voisées du signal apparaissant sous la forme de successions de pics spectraux marqués, dont les fréquences centrales sont multiples de la fréquence fondamentale. Par contre, le spectre d'un signal non voisé ne présente aucune structure particulière. La forme générale de ces spectres, appelée enveloppe spectrale, présente elle-même des pics et des creux qui correspondent aux résonnances et aux anti-résonnances du conduit vocal et sont appelés formants et anti-formants.

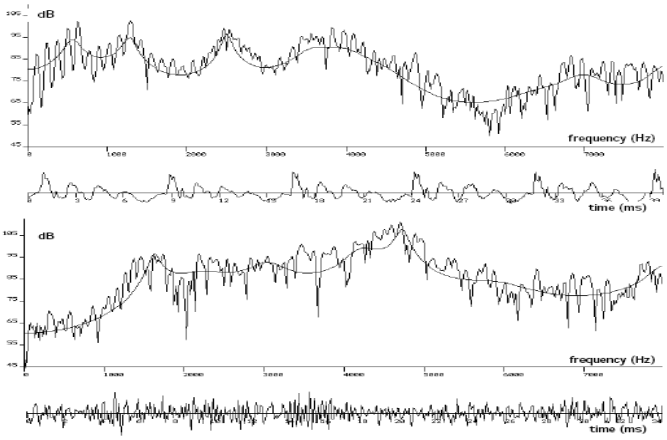


Fig.1.14- Evolution de la transformée de Fourier discrète du [a] et du [ʃ] de « baluchon ».

La parole étant un phénomène non stationnaire, il importe de faire intervenir le temps comme troisième variable dans la représentation. Clairement, la représentation la plus répandue est le spectrogramme.

- **Spectrogramme**

Le spectrogramme permet de donner une représentation tridimensionnelle d'un son dans laquelle l'énergie par bande de fréquences est donnée en fonction du temps [1].

Plus précisément, le spectrogramme représente le module de la transformée de Fourier discrète calculé sur une fenêtre temporelle plus ou moins longue. La transformée de Fourier discrète **TFD X(k)** de la ième fenêtre du signal de parole $\boldsymbol{x(n)}$ est donnée par :

$$\boldsymbol{X_i(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi nk/N}} \qquad (1)$$

L'amplitude du spectre y apparait sous la forme de niveaux de gris dans un diagramme en deux dimensions temps-fréquence, comme on peut le remarquer sur les Spectrogramme de la

figure 1.16. On parle de spectrogramme à large bande ou à bande étroite selon la durée de la fenêtre de pondération. Les spectrogrammes à bande large sont obtenus avec des fenêtres de pondération de faible duré ; ils mettent en évidence l'enveloppe spectrale du signal, et permettent par conséquent de visualiser l'évolution temporelle des formants. Les périodes voisées y apparaissent sous la forme de bandes verticale plus sombres.

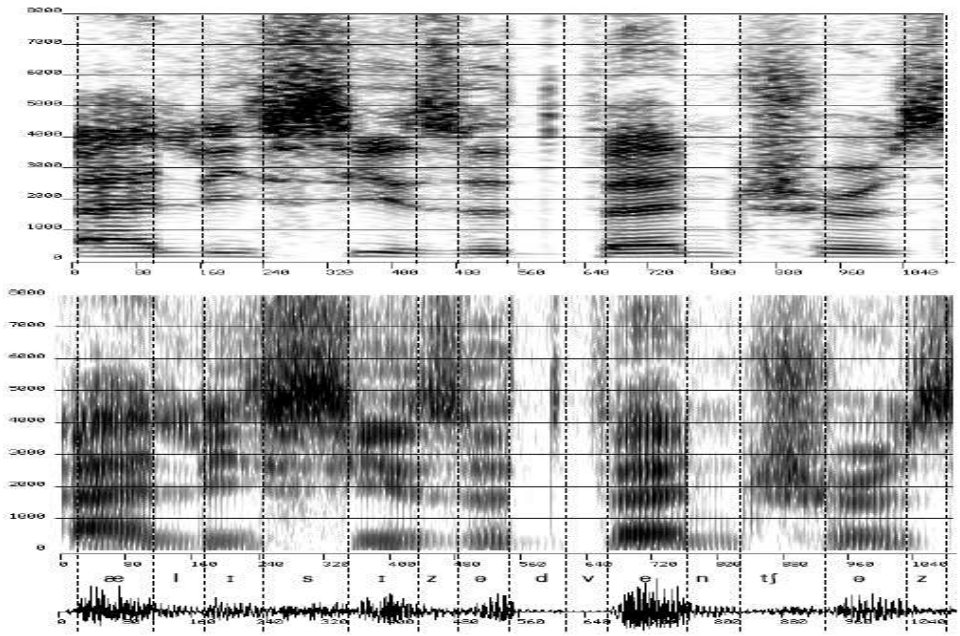


Fig.1.16- Spectrogramme à large bande (en bas), à bande étroite (en haut), et évolution temporelle de la phrase anglaise « Alice's adventures », échantillonnée à 11.25 kHz (calcul avec fenêtre de hamming de 10 et 30 ms respectivement).

- **Caractéristique du signal de parole**

Le signal de parole est un vecteur acoustique porteur d'informations d'une grande complexité.

- **Traits acoustiques :**

Les traits acoustiques du signal de parole sont liés à sa production.

- ✓ **La fréquence fondamentale**

C'est Le premier trait acoustique, c'est la fréquence de vibration des cordes vocales. Pour les sons voisés, la fréquence fondamentale correspond à la fréquence du cycle d'ouverture/fermeture des cordes vocales.

- ✓ **Le spectre de fréquence**

C'est le deuxième trait acoustique dont dépend principalement le timbre de la voix. Il résulte du filtrage dynamique du signal en provenance du larynx ou signal glottique par le conduit vocal.

- ✓ **L'énergie**

Le dernier trait acoustique est l'énergie correspondant à l'intensité sonore. L'énergie de la parole est liée à la pression de l'air en amant du larynx. Elle est habituellement plus forte pour les segments voisés de la parole que pour les segments non voisés.

Chaque trait acoustique est intimement lié à une caractéristique perceptuelle

- ✓ **Le timbre**

Le timbre est une caractéristique permettant d'identifier une personne à la simple écoute de sa voix. Il provient en particulier de la résonance dans la poitrine, la gorge, la cavité buccale et le nez. Le timbre dépend fortement de la corrélation entre la fréquence fondamentale et les harmoniques qui sont les multiples de cette fréquence.

- ✓ **Le pitch**

Les variations de la fréquence fondamentale définissent le pitch qui constitue la perception de la hauteur (où les sons s'ordonnent de grave à aigu). Seuls les sons quasi-périodiques (voisés) engendrent une sensation de hauteur tonale bien définie.

- ✓ **Intensité**

L'intensité d'un son, appelée aussi volume, permet de distinguer un son fort d'un son faible. Elle correspond à l'amplitude de l'onde acoustique. Pour le son, onde de compression, cette grandeur est la pression.

- **Méthodes avec modélisation**

Dans cette catégorie, les méthodes dites de Codage Prédictif Linéaire LPC [1] ont été largement utilisées pour l'analyse de la parole. Elles font référence à un modèle du système de phonation, que l'on représente en général comme un tuyau sonore à section variable. L'analyse LPC est utilisée essentiellement en codage et en synthèse de la parole.

- **Méthodes cepstrales**

Une méthode d'analyse du signal vocal fondée sur une modélisation est actuellement très répandue en reconnaissance automatique de la parole : il s'agit de l'analyse cepstrale [5].

La plupart des systèmes actuels de reconnaissance de parole, utilisent un ensemble de paramètres appelés MFCC (Mel Frequency Cepstrum Coefficients) dont le principe d'obtention repose sur l'analyse cepstrale.

Cette méthode, appelée aussi analyse homomorphique, a pour but de séparer dans le signal vocal les contributions respectives de la source du signal à savoir la vibration des cordes vocales et du conduit vocal dont les fréquences de résonance conduisent notamment aux formants des voyelles.

Fig.1.16-Principe de l'analyse homomorphique.

La figure1.17 montre les phases d'obtention de coefficients MFCC à partir d'un signal. Ces coefficients sont robustes car, d'une part, ils assurent comme il vient d'être dit une séparation entre la fonction de transfert du conduit vocal et les caractéristiques du fondamental de la voix, et, d'autre part, ils sont peu sensibles à la puissance acoustique du signal analysé.

Fig.1.17-Principe de calcul des coefficients *MFCC*

- **Modèles d'oreille**

Une famille de méthodes d'analyse de parole s'inspire des données de la psycho-acoustique et de la physiologie de l'audition humaine telles que courbes d'isosonie, bandes critiques de l'oreille, phénomènes non linéaires (saturation, masquage de sons, etc.), contrôle de gain, filtrage cochléaire, etc.

Les modèles d'oreille [5], sont utilisés pour obtenir une représentation fréquentielle de la parole. On les trouve dans des systèmes de reconnaissance de parole, notamment en présence de bruits.

- **Analyse perceptive**

En présence de bruit important, les méthodes d'analyse traditionnelles ont du mal à extraire les caractéristiques représentatives de la parole. De nombreuses méthodes ont été proposées pour améliorer cette situation. Elles se fondent sur différentes méthodes, notamment sur des propriétés de la perception auditive [5]. Un bon exemple est l'analyse RASTA-PLP, utilisée avec succès en reconnaissance de parole dans du bruit. Cette méthode intègre plusieurs opérations inspirées de données perceptives.

- **Analyse par ondelettes**

Parmi les travaux menés pour améliorer les techniques d'analyse de signaux, l'analyse par ondelettes [2], présente un intérêt certain. Ce type d'analyse permet d'obtenir une représentation temps-fréquence locale d'un signal comme alternative au spectre de Fourier. L'intérêt, pour des signaux non stationnaires comme la parole, est de pouvoir mener une analyse multi-résolution des phénomènes correspondant à des échelles de temps et de fréquence différentes.

L'analyse par ondelettes a été appliquée à de nombreux types de signaux (biomédicaux, sismiques, etc.). Dans le cas de la parole, les applications actuelles concernent la synthèse, le codage, la suppression de bruit, etc. Peu de travaux ont trait à la reconnaissance.

Dans le chapitre suivant nous allons présenter les méthodes d'analyses et d'extraction, les plus utilisés pour le signal de parole dont le but de la reconnaissance automatique de la parole.

CHAPITRE II : LES PARAMÈTRES PERTINENTS DU SIGNAL DE PAROLE

Introduction

Le signal de parole est trop redondant et variable pour être utilisé directement dans un système de reconnaissance automatique de la parole. Il est donc nécessaire d'en extraire l'information pertinente afin de caractériser et d'identifier le contenu linguistique. Le signal de parole est représenté, en général, dans le domaine fréquentiel montrant l'évolution temporelle de son spectre. Ce domaine est approprié pour la reconnaissance puisque l'on peut raisonnablement considérer que les propriétés du spectre restent stationnaires durant des intervalles de temps d'environ une dizaine de ms (valeur adoptée de manière classique).

Les systèmes de reconnaissance intègrent un module de paramétrisation dont le rôle est de créer des vecteurs de paramètres acoustiques résultant de l'analyse spectrale du signal de parole. La plupart des techniques de paramétrisation consistent à décrire l'enveloppe du spectre à court terme dans le domaine fréquentiel. D'autres techniques peuvent être utilisées comme l'analyse en ondelette.

Dans ce chapitre, nous allons présenter les méthodes; les plus utilisées, les plus récentes et les variantes améliorées; d'extraction des paramètres acoustiques pertinents de la parole pour la reconnaissance automatique de la parole, sujet de ce travail de mémoire.

II.1. Coefficients cepstraux de prédiction linéaire

La prédiction linéaire est une technique issue de l'analyse de la production de la parole permettant d'obtenir des coefficients de prédiction linéaire (Linear Prediction Coefficients – LPC). Des paramètres cepstraux LPCC [7]. (Linear Prediction Cepstral Coefficients) sont ensuite calculés à partir de ces coefficients.

Dans ce cadre d'analyse, le signal de parole x est considéré comme la conséquence de l'excitation du conduit vocal par un signal provenant des cordes vocales. La prédiction s'appuie sur le fait que les échantillons de parole adjacents sont fortement corrélés, et que, par conséquent, l'échantillon sn peut être estimé en fonction des p échantillons précédents.

Par prédiction linéaire, on obtient donc une estimation du signal :

$$\hat{x}_n = \sum_{i=1}^{p} a_i . x_{n-i}$$

Où les a_i sont des coefficients constants sur une fenêtre d'analyse. La définition devient exacte si on inclut un terme d'excitation :

$$x_n = \sum_{i=1}^{p} a_i . x_{n-i} + G e_n$$

Où e est le signal d'excitation et G le gain de l'excitation. La transformée en Z de cette égalité donne :

$$GE(z) = \left(1 - \sum_{i=1}^{p} a_i z^{-i}\right) X(z)$$

D'où :

$$H(z) = \frac{X(z)}{E(z)} = \frac{G}{(1 - \sum_{i=1}^{p} a_i z^{-i})} = \frac{G}{A(z)}$$

Cette équation peut être interprétée comme suit : Le signal x est le résultat de l'excitation du filtre tout pôle $H(z) = \frac{G}{A(z)}$ par le signal d'excitation e.

Les coefficients a_i sont les coefficients qui minimisent l'erreur quadratique moyenne :

$$E_n = \sum_m (G . e_{n+m})^2 = \left[\sum_m \left(x_{n+m} - \sum_{i=1}^{p} a_i x_{n+m-i}\right)\right]^2$$

À partir de ces échantillons prédis, on peut calculer les paramètres cepstraux. Le cepstre est le résultat de la transformée de Fourier inverse appliquée au logarithme de la transformée de Fourier du signal de parole. Les paramètres cepstraux c_i sont les coefficients du développement de Taylor du logarithme du filtre tout pôle :

$$ln\left[\frac{G^2}{|A(z)|^2}\right] = \sum_{-\infty}^{+\infty} c_i z^{-i}$$

Ce qui donne :

$$c_0 = \ln(G^2)$$

$$c_m = a_m + \sum_{=i-p}^{i-1} \frac{k}{i} c_k a_{i-k} \quad avec\ i = p, \ldots, N_c$$

$$c_{-m} = c_m$$

Les paramètres cepstraux ont l'avantage d'être peu corrélés entre eux. Cela permet d'utiliser des matrices de covariances diagonales pour leur moment de second ordre, et ainsi gagner beaucoup de temps lors du décodage. Les différentes étapes de l'analyse **LPCC** sont détaillées dans la figure 2.1

Comme dit précédemment, ce modèle provient de l'analyse de la production de la parole. D'autres formes d'analyses qui tiennent compte du mode de perception auditive de la parole plutôt que du mode de production sont présentées dans les sections suivantes.

II.2. L'analyse en banc de filtre

L'analyse par banc de filtres [8] est une technique initialement utilisée pour le codage du signal de parole. Elle produit des paramètres cepstraux (Mel-Frequency Cepstral Coefficients) -MFCC. Le signal de parole est analysé à l'aide de filtres passe-bande permettant d'estimer l'enveloppe spectrale en calculant l'énergie dans les bandes de fréquences considérées.

Les bandes de fréquences des filtres sont espacées logarithmiquement selon une échelle perceptive afin de simuler le fonctionnement du système auditif humain. Les échelles perceptives les plus utilisées sont celles de Mel et de Bark [8]. Plus la fréquence centrale du filtre est basse, plus la bande passante du filtre est étroite. Augmenter la résolution pour les basses fréquences permet d'extraire plus d'information dans ces zones où elle est plus dense.

Il est possible d'utiliser directement les coefficients obtenus à la sortie des filtres pour la reconnaissance de la parole, cependant, d'autres coefficients plus discriminants, plus robustes au bruit ambiant et surtout décorrélés entre eux sont préférés : les coefficients cepstraux. Un ensemble de M coefficients cepstraux, généralement entre 10 et 15, sont calculés en effectuant un liftrage (filtrage dans le domaine cepstral) du spectre en puissance d'un signal selon la transformée en cosinus discret (Discrete Cosinus Transform – DCT) :

$$_{i} = \sum_{m=0}^{M} S_m \cdot \left(\frac{\pi}{N_f}\left(m + \frac{1}{2}\right) i\right) \qquad \textbf{pour } i = 0, \dots, M-1$$

Où N_f est le nombre de filtres utilisé.

Le coefficient c_0 correspond à l'énergie moyenne de la trame. De manière générale, on ne le prend pas en compte afin de rendre les MFCC peu sensibles à la puissance acoustique du signal de parole.

Les différentes étapes de l'analyse **MFCC** sont détaillées dans la figure 2.1.

II.3. Analyse par prédiction linéaire perceptuelle

L'analyse par Prédiction Linéaire Perceptuelle [9] (Perceptual Linear Prediction – PLP) repose sur un modèle de perception de la parole. Les différentes étapes de l'analyse PLP sont détaillées dans la figure 2.1.

Elle est basée sur le même principe que l'analyse prédictive et intègre trois caractéristiques de la perception :

- **Intégration des bandes critiques** : la prédiction linéaire produit la même estimation de l'enveloppe spectrale pour toute la zone de fréquences utiles, ce qui est en contradiction avec le fonctionnement de l'appareil perceptif humain. En effet, l'oreille humaine a la faculté d'intégrer certaines zones de fréquences en bande appelées bandes critiques. Les bandes critiques sont réparties selon l'échelle de Bark, dont la relation avec la fréquence est définie par :

$$\boldsymbol{f = 600\sinh\left(\frac{z}{6}\right)}$$

avec f la fréquence en Hertz et z la fréquence en Bark.

La nouvelle densité spectrale est échantillonnée selon cette nouvelle échelle, ce qui augmente la résolution pour les basses fréquences.

- **Préaccentuation pas courbe d'isosonie** : cette caractéristique provient de la psychoaccoustique qui a montré que l'intensité sonore d'un son pur perçue par l'appareil auditif varie avec la fréquence de ce son. Ainsi, dans l'analyse PLP, afin de prendre en compte la manière dont l'appareil auditif perçoit les sons, la densité spectrale doit être multipliée par une fonction de pondération non linéaire. Cette fonction peut être estimée en utilisant l'abaque sur laquelle sont reportées les lignes isosoniques. Ces lignes correspondent à la trajectoire d'égale intensité sonore pour différentes fréquences d'un son pur. En pratique, cette préaccentuation est remplacée par l'application du filtre passe-haut dont la transformée en $\boldsymbol{Z}$ est :

$$\boldsymbol{(1 - 0.95z^{-1})}.$$

- **Loi de Stevens** : l'intégration des bandes critiques et la préaccentutation ne suffisent pas à faire correspondre l'intensité mesurée et l'intensité subjective (appelée sonie). La loi de Stevens donne la relation entre ces deux mesures :

$$\textbf{sonie} = (\textbf{intensité})^{0,33}$$

Les **PLP** sont basés sur le spectre à court terme du signal de parole, comme les coefficients **LPC**. Cela signifie que le signal est analysé sur une fenêtre glissante de courte durée. En général, on utilise une fenêtre de longueur 10 à 30 ms. que l'on décale de 10 ms pour chaque trame.

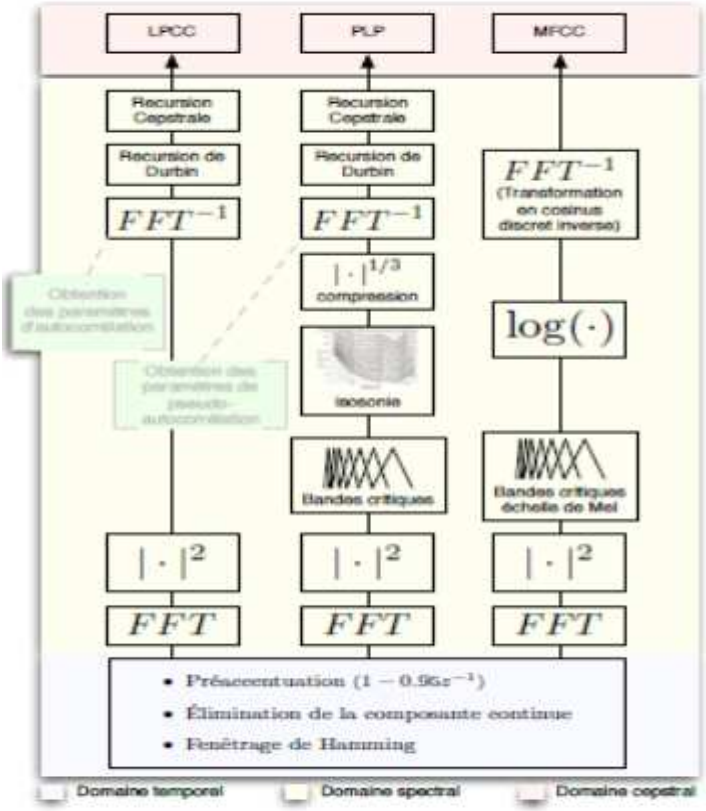


Fig. 2.1-Comparaison de trois analyses du signal : LPCC, PLP et MFCC

II.4. Méthodes RASTA- PLP et JRASTA- PLP

Afin d'augmenter la robustesse des paramètres PLP, on peut envisager l'analyse spectrale relative **RASTA** [10] (Relative Spectral), présentée comme une façon de simuler l'insensibilité de l'appareil auditif humain aux stimuli à variation temporelle lente. Cette technique traite les composantes de parole non linguistiques, qui varient lentement dans le temps, dues au bruit convolutif (log-RASTA) et au bruit additif (J-RASTA). En pratique, RASTA effectue un filtrage passe-bande sur le spectre logarithmique ou sur le spectre

compressé par une fonction non linéaire. L'idée principale est de supprimer les facteurs constants dans chaque composante du spectre à court-terme avant l'estimation du modèle tout-pôle. L'analyse RASTA est souvent utilisée en combinaison avec les paramètres PLP. Les étapes d'une analyse RASTA-PLP sont décrites dans la figure 2.3.

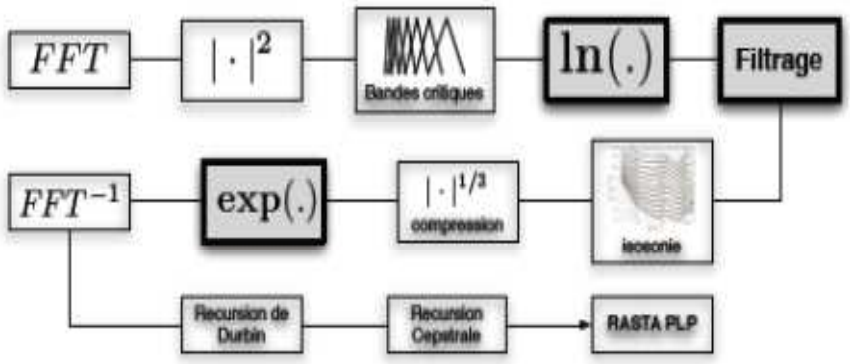


Fig.2.3 Analyse RASTA PLP

Les étapes grisées sont celles qui font la spécificité du traitement RASTA. La différence entre RASTA et J-RASTA se situe au niveau du logarithme (4ème étape) :

$\boldsymbol{ln}\ (\boldsymbol{x})$ Pour RASTA et $\boldsymbol{ln}\ (\mathbf{1} + \boldsymbol{jx})$ pour J-RASTA.

II.5. Analyse à résolution multiple

L'analyse à résolution multiple (**Multi Resolution Analysis – MRA**) [11], effectue une analyse en ondelettes d'une fenêtre de signal audio. Cela consiste à faire passer le signal dans un arbre de filtres passe-bas et passe-haut, à la sortie desquels l'énergie à court terme est calculée « voir figure 2.4 ». À chaque niveau de l'arbre, le signal est entièrement décrit, mais dans une résolution fréquentielle et temporelle différente. Comme on peut le constater, la disposition des filtres n'est pas intuitive, car il faut prendre en compte le phénomène de repliement spectral qui recopie dans les basses fréquences le signal haute fréquence inversé. Ensuite, il faut regrouper les énergies calculées aux feuilles de l'arbre pour former les trames qui seront utilisées dans le système de reconnaissance de la parole.

Considérons une fenêtre de taille N échantillons, qui se déplacent de M échantillons.Pour **MRA**, les valeurs utilisées pour N sont 256 (32 ms) ou 384 (48 ms), et M est fixé à 80 échantillons (soit 10 ms). À noter que ce front-end a été développé pour des applications téléphoniques. Le nombre d'échantillons obtenus dans les nœuds de l'arbre

diminue quand on descend dans l'arbre, mais l'intervalle temporel associé aux échantillons filtrés reste inchangé.

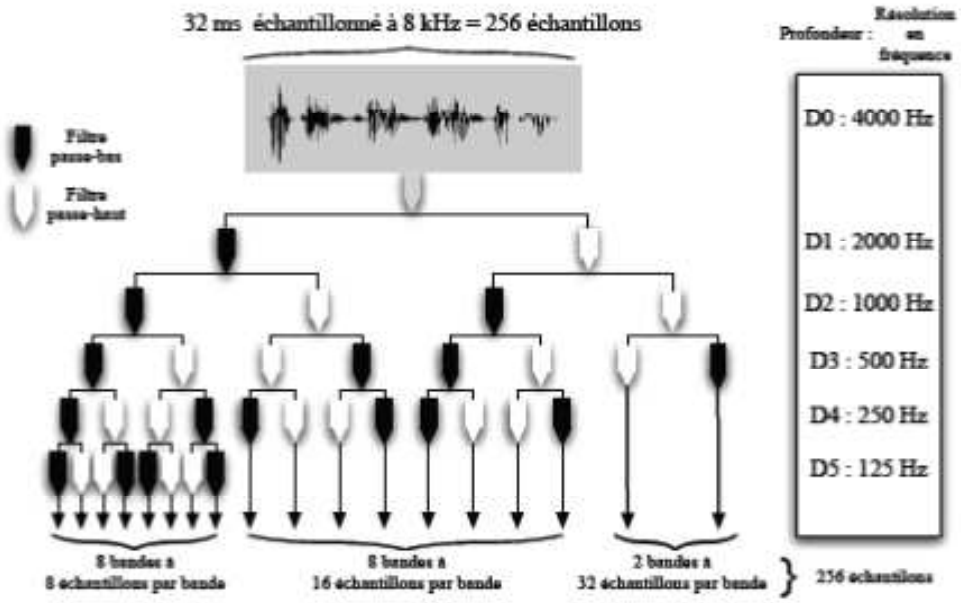


Fig. 2.4-Principe de l'analyse à résolution multiple.

Selon le principe d'indétermination d'Heisenberg, il existe une relation entre la résolution temporelle et la résolution fréquentielle des échantillons dans les différentes sous-bandes. Sur la base de ce principe, le produit de la résolution en temps et celle en fréquence ne doit pas être inférieur à un certain seuil. Étant donné qu'à chaque niveau de l'arbre, la résolution fréquentielle est divisée par deux « cf. figure 2.4 », on peut considérer des intervalles temporels d'intégration différents pour chaque niveau de l'arbre. Pour cela, on utilise l'extracteur de paramètres sur le même nombre d'échantillons à chaque niveau, ce qui a pour conséquence de diviser l'intervalle temporel par deux. Pour les 8 premières bandes (de 0 à 1 kHz) on utilise les 8 échantillons disponibles. Pour les 8 bandes suivantes (de 1kHz à 3 kHz) on n'utilise que les 8 échantillons centraux sur les 16 disponibles. Enfin, pour les deux dernières bandes (de 3 kHz à 4 kHz) on utilise seulement 10 échantillons sur les 32 disponibles. Tout ceci est détaillé dans le tableau 2.1

Niveau de l'arbre	Résolution en fréq. [Hz]	Intervalle temporel en ms (éch.)	
		32 (N=256)	48 (N=384)
1	4000	10 (80)	10 (80)
2	2000	10 (40)	10 (40)
3	1000	10 (20)	10 (20)
4	500	10 (10)	12 (12)
5	250	16 (8)	24 (12)
6	125	32 (8)	48 (12)

TAB. 2.1-Correspondance entre résolution fréquentielle et temporelle pour l'analyse MRA

À la sortie de ces filtres, on doit appliquer une opération d'extraction de paramètres acoustiques sur les échantillons filtrés obtenus. Notons ci les échantillons d'un nœud de l'arbre, et N leur nombre. Cette opération est appelée intégration. Les opérateurs disponibles pour l'intégration sont nombreux, les plus utilisés sont :

- L'énergie moyenne par échantillon :

$$\boldsymbol{E = \frac{1}{N}\sum_{i=1}^{N} c_i^2}$$

- La norme p :

$$\boldsymbol{X = \frac{1}{N}\sum_{i=1}^{N}|c_i|^p \quad avec \quad p = 1,2,3}$$

- L'entropie moyenne :

$$\boldsymbol{H = \frac{1}{N}\sum_{i=1}^{N} c_i^2 . log c_i^2}$$

- L'opérateur teager :

$$\boldsymbol{T = \frac{1}{N-2}\sum_{i=2}^{N-1}\left(c_i^2 - c_{i-1}.c_{i+1}\right)}$$

- La dimension théorique « combinaison de l'entropie moyenne et l'énergie moyenne » :

$$\boldsymbol{TD = E.exp^{\frac{-H}{2}}}$$

Les paramètres MRA ont la particularité de ne pas décrire l'enveloppe spectrale du signal, mais plutôt de représenter le signal en termes d'énergie présente dans chaque bande de fréquences et d'utiliser la redondance de représentation de ce signal de parole à chaque niveau de l'arbre. L'intérêt de considérer de tels paramètres est qu'on peut supposer que

l'information qu'ils contiennent sera différente de celle fournie par les représentations cepstrales.

II.6. Méthodes Acoustiques hybrides

Ces paramètres sont calculés à partir de paramètres discriminants obtenus à l'aide d'un réseau de neurones. Les systèmes de reconnaissance automatique de la parole utilisent en général des modèles à base de **GMMs** pour estimer les distributions de vecteurs de paramètres décorrélés qui correspondent à des unités acoustiques de courte durée « syllabes, phonèmes, phonèmes en contexte, ... ». En comparaison, les systèmes hybrides **ANN/MMC** [12] utilisent des réseaux de neurones entraînés de manière discriminante pour estimer les distributions de probabilité des unités étant donné les observations acoustiques.

Cette approche consiste à combiner des paramètres discriminants issus d'un réseau de neurones avec une modélisation des distributions par **GMMs**. Le réseau de neurones génère les probabilités postérieures des unités qui sont ensuite transformés pour être utilisés comme paramètres d'entrée pour le modèle **MMC/GMM** qui est alors appris de manière conventionnelle. Les transformations sur les distributions de probabilité sont de différentes sortes. Les réseaux de neurones produisent directement des probabilités a posteriori contrairement aux mixtures de gaussiennes. Étant donné que les probabilités postérieures ont une distribution très biaisée, il est avantageux de les transformer en prenant leur logarithme par exemple. Une alternative à cela est d'omettre la dernière non-linéarité à la sortie du réseau de neurones. Cette non linéarité, le softmax, correspond à normaliser les exponentiels « ce qui est très proche de prendre le logarithme des probabilités ». Les vecteurs de probabilités postérieures ont tendance à posséder une valeur élevée, correspondant au phonème prononcé, et les autres basses. Les réseaux de neurones n'ont pas la contrainte d'utiliser des paramètres acoustiques décorrélés comme les **MMCs**. Cependant, il s'avère que la transformation de Karhunen-Loeve, plus connue sous le nom d'analyse en composante principale « Principal Component Analysis **– PCA** » est utile pour décorréler les paramètres, vraisemblablement parce qu'elle augmente la correspondance entre les paramètres et les modèles à base de mixture de gaussiennes. Les principaux résultats obtenus avec ce genre de technique sont présentés dans plusieurs travaux.

II.7. Autres paramètres acoustiques

Dans le but d'accroitre la robustesse des systèmes de reconnaissance automatique de la parole, Beaucoup d'autres paramètres acoustiques ont été développés afin, le plus souvent, de compléter et combiner les paramètres existants(combinaison de paramètres acoustique).

Conclusion

Dans ce chapitre, nous avons décris les méthodes, les plus utilisées d'extraction des paramètres acoustiques pertinents en termes d'efficacité et de performances pour les systèmes de Reconnaissance Automatique de la parole de la parole.

Le chapitre suivant sera consacré au sujet de la reconnaissance Automatique de la parole : à la description des principes de base qui constituent les systèmes de reconnaissance et la difficulté relative de la mise en œuvre de ces systèmes.

CHAPITRE III : LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

Introduction

Le problème de la reconnaissance automatique de la parole consiste à extraire, à l'aide d'un ordinateur, l'information lexicale contenue dans un signal de parole.

Depuis plus de deux décennies, des recherches intensives dans ce domaine ont été accomplies par de nombreux laboratoires internationaux. Des progrès importants ont été accomplis grâce au développement d'algorithmes puissants ainsi qu'aux avancées en traitement du signal. Différents systèmes de reconnaissance de la parole ont été développés, couvrant de vastes domaines tel que la reconnaissance de quelques mots clés sur lignes téléphoniques, les systèmes à dicter vocaux, les systèmes de commande et contrôle sur PC, et allant jusqu'aux systèmes de compréhension du langage naturel.

Dans ce chapitre, nous allons décrire et éclairer au mieux la complexité inhérente à la mise en ouvre d'un système de reconnaissance automatique de la parole, objet de notre travail, puis nous allons définir les principes, les approches et les techniques qui sont à la base de la plupart de ces systèmes.

III.1. Niveaux de complexité de la RAP

Pour bien appréhender le problème de la reconnaissance automatique de la parole, il est bon d'en comprendre les différents niveaux de complexité.

Le signal de parole est un des signaux les plus complexes : En plus de la complexité physiologique inhérente au système phonatoire et des problèmes de coarticulation qui en résultent, le conduit vocal varie également très fort d'une personne à l'autre.

La mesure de ce signal de parole est fortement influencée par la fonction de transfert (comprenant les appareils d'acquisition et de transmission, ainsi que l'influence du milieu ambiant).

Il y a d'abord le problème de la variabilité intra et inter-locuteurs. Le système peut être dépendant du locuteur (optimisé pour un locuteur bien particulier) ou indépendant du locuteur (pouvant reconnaître n'importe quel utilisateur).

Evidemment, les systèmes dépendants du 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 cependant pas toujours possible. Par exemple, dans le cas d'applications téléphoniques, les systèmes doivent pouvoir être utilisés par n'importe qui et doivent donc être indépendants du locuteur.

Bien que la méthodologie de base reste la même, Cette indépendance au locuteur est cependant 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 capable de s'adapter rapidement (de façon supervisée ou non supervisée) au nouveau locuteur.

Par ailleurs, un système peut être destiné à reconnaitre des mots isolés ou de la parole continue. Il est plus simple de reconnaître des mots isolés bien séparés par des périodes de silence que de reconnaître la séquence de mots constituant une phrase. En effet, dans ce dernier cas, non seulement la frontière entre mots n'est plus connue mais, de plus, 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 - un exemple simple et bien connu étant les liaisons du français).

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 naturel avec 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.

La taille du vocabulaire et son degré de confusion sont également des facteurs importants. Les petits vocabulaires sont évidemment plus faciles à reconnaître que les grands vocabulaires, étant donné que dans ce dernier cas, les possibilités de confusion augmentent.

- **Robustesse d’un système**

Un système est dit robuste s’il est capable de fonctionner proprement dans des conditions difficiles. En effet, de nombreuses variables peuvent affecter significativement les performances des systèmes de reconnaissance:

- Bruits d'environnement tels que bruits additifs stationnaires ou non stationnaires

(par exemple, dans une voiture ou dans une usine).

- Acoustique déformée et bruits « additifs » corrélés avec le signal de parole utile (par exemple, distorsions non linéaires et réverbérations).
- Utilisation de différents microphones et différentes caractéristiques (fonctions de transfert) du système d'acquisition du signal (filtres), conduisant généralement à du bruit de convolution.
- Bande passante fréquentielle limitée (par exemple dans le cas des lignes téléphoniques pour lesquelles les fréquences transmises sont naturellement limitées).
- Elocution inhabituelle ou altérée, comprenant entre autre: l'effet Lombard, (qui désigne toutes les modifications, souvent inaudibles, du signal acoustique lors de l'élocution en milieu bruité), le stress physique ou émotionnel, une vitesse d'élocution inhabituelle, ainsi que les bruits de lèvres ou de respiration.

Certains systèmes peuvent être plus robustes que d'autres à l'une ou l'autre de ces perturbations, mais en règle générale, les reconnaisseurs de parole actuels restent encore trop sensibles à ces paramètres.

III.2. Approche et techniques de reconnaissance automatique de la parole

III.2.1. Approche par la normalisation temporelle

Les premiers succès en reconnaissance vocale ont été obtenus dans les années 70 à l'aide d'un paradigme de reconnaissance de mots. L'idée, très simple dans son principe, consiste à faire prononcer un ou plusieurs exemples de chacun des mots susceptibles d'être reconnus, et à les enregistrer sous forme de vecteurs acoustiques (typiquement : un vecteur de coefficients LPC ou assimilés toutes les 10 ms). L'étape de reconnaissance proprement dite consiste alors à analyser le signal inconnu sous la forme d'une suite de vecteurs acoustiques similaires, et à comparer la suite inconnue à chacune des suites des exemples préalablement enregistrés. Le mot (reconnu) sera alors celui dont la suite de vecteurs acoustique (spectrogramme) ressemble le mieux à celle du mot inconnu. Ce principe de base n'est cependant pas implémentable directement : Un même mot peut en effet être prononcé d'une infinité de façons différentes, en changeant le rythme de l'élocution. Il en résulte des spectrogrammes plus ou moins

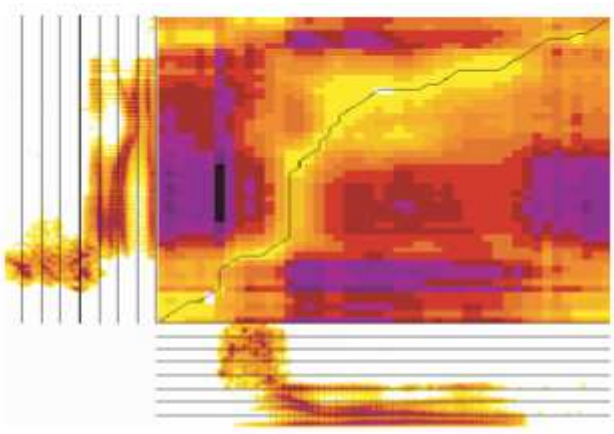
distordus dans le temps. La superposition du spectrogramme inconnu aux spectrogrammes de base doit dès lors se faire en acceptant une certaine (élasticité) sur les spectrogrammes candidats.

Une solution à ce problème d'élasticité ou recalage temporel fait appel aux technique de la programmation dynamique est formalisée mathématiquement par un algorithme désormais bien connu : L'algorithme **DTW** (Dynamic Time Warping) [13].

Les définitions de distances locales peuvent également être adaptées de façon à tenir compte du type de caractéristiques acoustiques utilisées (distance euclidienne, distance de Mahalanobis[14], distance d'Itakura[1] ou de l'importance relative des différentes composantes).Cette méthode donne d'excellents résultats. On démontre qu'elle fournit la solution optimale du problème.

Le principe de comparaison dynamique est illustré sur la figure **III.1.** Il consiste à rechercher la mise en correspondance optimale entre deux formes. Cette dernière est matérialisée par le chemin de recalage donné sur la figure **III.1**.

Le recalage temporel peut également être effectué à l'aide de modèles stochastiques présentés dans le paragraphe suivant et qui sont maintenant utilisés dans la plupart des systèmes.



Exemple de chemin de recalage (en noir) obtenu pour la comparaison de deux occurrences du mot « Zéro » représentées par leurs spectrogrammes. La matrice donne la distance entre deux éléments des spectrogrammes. Plus la couleur est sombre, plus la distance entre les deux éléments correspondants est grande. On constate que l'algorithme de programmation dynamique trouve le cheminement optimal correspondant à la suite d'éléments les plus proches.

Fig. III.1-Principe de la programmation dynamique

III.2.2. Approche par modélisation stochastique

Dans le paragraphe précédent, nous avons montré comment on pouvait effectuer par programmation dynamique l'intégration temporelle de distances locales, permettant en même temps de normaliser les variations temporelle des unités de parole. Cette approche conduit également à une segmentation automatique de la phrase en termes de segments de références.

Il y a cependant plusieurs limitation liées à l'approche DTW, cette approche requiert souvent une détection automatique de début et fin, ce qui est déjà une source de problèmes. De plus, si on essaie d'adapter la définition de distance locale, il est souvent difficile, sans outils mathématiques puissants, d'en comprendre les effets au niveau du critère global que l'on s'est donné à minimiser. Finalement, étant donné que la parole est beaucoup plus que la simple concaténation d'éléments linguistiques (par exemple, des mots ou des phonèmes) bien définis, il est nécessaire de pouvoir modéliser les variabilités et les dépendances de chaque unité en fonction de son contexte. Comme nous le verrons par la suite au chapitre 4, l'entrainement de distributions statistiques représente la meilleure approche pour modéliser la variabilité observée sur des exemples réels.

Pour toutes ces raisons, les modèles statistiques [15] sont maintenant très utilisés dans les problèmes de reconnaissance de séquences complexes telles que le signal de parole. De plus, l'introduction d'un formalisme statistique permet l'utilisation de plusieurs outils mathématiques très puissants (l'algorithmeEM (IV.5.2)) pour déterminer les paramètres par entrainement, et pour effectuer la reconnaissance et la segmentation automatique de mots et de parole continue. Ces outils mathématiques sont maintenant largement utilisés et constituent aujourd'hui l'approche dominante en reconnaissance de la parole.

Pour la plupart de ces systèmes de reconnaissance, la parole est supposée avoir été générée selon un ensemble de distributions statistiques. Par définition, une distribution unique ne peut générer qu'un processus stationnaire. Etant donné que la parole est constituée de plusieurs sons différents, il est nécessaire de considérer plusieurs distributions. Chaque distribution est modélisée par un ensemble de paramètres qui seront déterminés sur base d'un ensemble d'entrainement de façon à minimiser la probabilité d'erreur. Pendant la reconnaissance, nous recherchons alors, à travers l'espace de toutes les séquences de distributions possibles (dans les limites de contraintes phonologiques et, éventuellement,

syntaxiques), la séquence de modèles (et donc de la phrase (mot) associée) qui maximise la probabilité a posteriori par exemple.

- **Modèle acoustique MMC**

Selon le formalisme des modèles de Markov cachées (MMC) (chap.IV), le signal de parole est supposé être produit par un automate stochastique fini construit à partir d'un ensemble d'états stationnaires régis par des lois statistiques. En d'autres mots, le formalisme des modèles MMC suppose que le signal de parole est formé d'une séquence de segments stationnaires, tous les vecteurs associés à un même segment stationnaire étant supposés avoir générés par le même état MMC. Chaque état de cet automate est caractérisé par une distribution de probabilité décrivant la probabilité d'observation des différents vecteurs acoustiques.

Les transitions entre états sont instantanées. Elles sont caractérisées par une probabilité de transition. Ainsi chaque état du modèle permet de modéliser un segment de parole stationnaire, la séquence d'état permet quant à elle de modéliser la structure temporelle de la parole comme une succession d'état stationnaires. À cet effet les modèles utilisés en reconnaissance automatique de la parole sont généralement du type gauche-droite où les transitions possibles sont soit des boucles sur un même état, soit le passage à un état suivant (à droite). L'aspect séquentiel du signal de parole est ainsi modélisé.

Comme unité linguistique (chaque phonème ou chaque mot) est donc modélisé par un ou plusieurs états stationnaires, les mots sont ensuite construits en termes de séquences de phonèmes et les phrases en termes de séquences de mots. Chaque état stationnaire est représenté par les paramètres de fonctions statistiques invariables, par exemple la moyenne et la variance d'une distribution gaussienne où des GMM.

- **Modèle de Markov Caché MMC (Hidden Markov Model HMM)**

Il est caractérisé par un double processus stochastique :

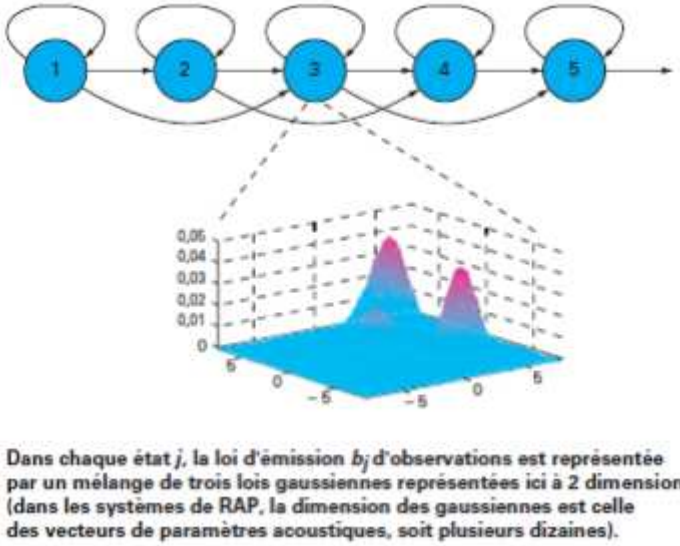
- un processus interne : non observable
- un processus externe : observable

Ces deux chaînes se combinent pour former le processus stochastique.

La chaîne interne : est une chaîne de Markov que l'on suppose à chaque instant dans un état où la fonction aléatoire correspondante engendre un segment élémentaire (de l'ordre de

10 ms ou plus), représenté par un vecteur de paramètres, de l'onde acoustique observée. Un observateur extérieur ne peut voir que les sorties de ces fonctions aléatoires, sans avoir accès aux états de la chaîne sous-jacente, d'où le nom de modèle caché.

Un des grands intérêts des MMC réside dans l'automatisation de l'apprentissage des différents paramètres et distributions de probabilités du modèle à partir de données acoustiques représentatives de l'application considérée, essentiellement les probabilités de transition d'un état du MMC à un autre état et surtout les lois d'émission. Ces lois d'émissions (probabilités) sont en général représentées sous forme d'une somme de fonctions gaussiennes (parfois plusieurs (GMM), permettant de mieux approcher la loi réelle du phénomène), comme l'illustre la figure III.2. Cet apprentissage est assuré par des algorithmes itératifs d'estimation des paramètres, notamment l'algorithme de Baum-Welch (IV.5.2), cas particulier de l'algorithme EM (Expectation-Maximisation) fondé sur le principe de maximum de vraisemblance.



Dans chaque état j, la loi d'émission b_j d'observations est représentée par un mélange de trois lois gaussiennes représentées ici à 2 dimensions (dans les systèmes de RAP, la dimension des gaussiennes est celle des vecteurs de paramètres acoustiques, soit plusieurs dizaines).

Fig. III.2- modèle de markov à cinq états

Les MMC peuvent être utilisés de plusieurs façons en RAP, selon l'importance de l'application (taille du vocabulaire et type de parole : mots isolés ou parole continue).

Pour la reconnaissance de mots isolés, il est possible de modéliser chaque mot par un MMC, La reconnaissance revient alors à calculer la vraisemblance de la suite d'observations acoustiques constituant le mot à reconnaître par rapport à chacun des modèles. Le modèle présentant la plus grande vraisemblance d'avoir émis cette suite d'observations fournit le mot reconnu. L'algorithme permettant d'optimiser ce calcul est à nouveau fondé sur la programmation dynamique, mais dans un cadre stochastique, l'algorithme de Viterbi (IV.4.4).

Pour la reconnaissance de la parole continue, l'utilisation de modèles globaux pour chaque mot pose divers problèmes : espace mémoire de stockage, volume de données acoustiques nécessaires pour l'apprentissage de tous les MMC. La solution adoptée est d'utiliser des MMC pour représenter les unités phonétiques.

Ces unités peuvent être de nature variée : phonèmes, diphones, syllabe, fenone, allophones.

Les modèles de mots sont construits par concaténation des modèles analytiques élémentaires correspondant aux transcriptions phonétiques de ces mots. Pour mettre au point des MMC aussi indépendants du locuteur que possible, il est nécessaire d'augmenter le nombre de paramètres des MMC.

Les solutions disponibles sont de deux types :

- Les multi-modèles **:** le principe est de représenter le même mot par plusieurs MMC correspondant à différentes classes de locuteurs.
- Les mélanges de densités de probabilité **:** au lieu de représenter la probabilité d'émission d'un segment de parole pour une loi de probabilité (une gaussienne), on utilise un mélange de lois gaussiennes permettant de mieux approcher la loi réelle du phénomène acoustique.

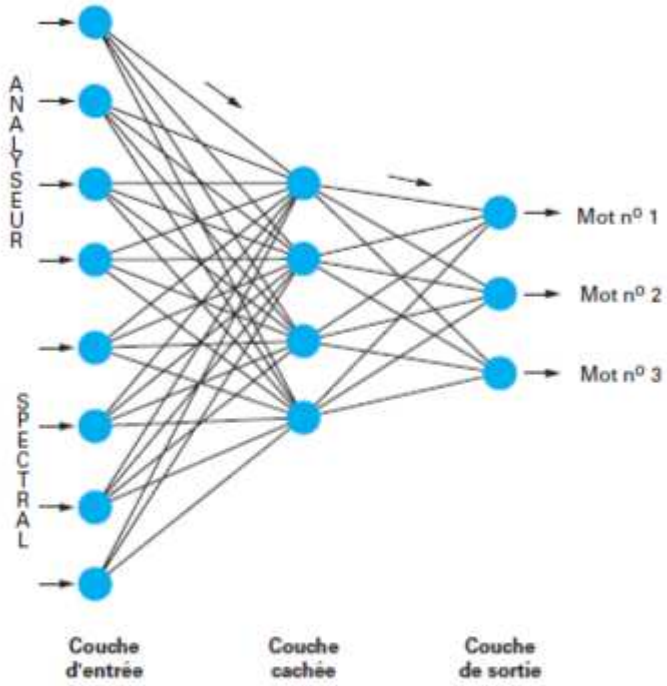
III.2.3. Approche par modèles neurométriques

L'utilisation de modèles connexionnistes, ou réseaux neuronaux, fondés sur une modélisation plus ou moins réaliste du cortex humain, s'est récemment répandue et a permis d'obtenir des résultats intéressants en RAP comme dans d'autres domaines de la perception. Ces modèles sont constitués par l'interconnexion d'un très grand nombre de processeurs élémentaires inspirés du fonctionnement du neurone. Plusieurs types ont été utilisés dans différents domaines du traitement de la parole (reconnaissance, débruitage de parole, vérification du locuteur, etc.) que nous allons décrire brièvement ci-dessous.

- **Perceptrons multicouches** avec apprentissage par rétropropagation du gradient d'erreur.
  - La prise en compte du temps, problème majeur en parole, est impossible dans le modèle de base. Des variantes ont été proposées pour pallier cet inconvénient :

perceptrons contextuels et perceptrons à entrée récurrente ; perceptrons à retard temporel TDNN (Time Delay Neural Network).

La figure III.3 illustre le fonctionnement d'un perceptron avec une couche cachée.



La couche de neurones d'entrée reçoit les données concernant le mot à reconnaître provenant par exemple d'un analyseur spectral Chaque neurone de la couche de sortie est associé à la reconnaissance d'un mot du vocabulaire. Toutes les connexions entre neurones sont pondérées.

Fig. III.3-Perceptron à une couche cachée pour la reconnaissance de mots

Les réseaux neuronaux (essentiellement perceptrons multicouches) sont presque exclusivement utilisés en reconnaissance de la parole comme frontal de MMC ; un réseau neuronal est alors entraîné pour fournir à un MMC des valeurs de probabilités nécessaires à son fonctionnement. De telles architectures hybrides stochastiques/ neuronales-ANN/HMM [16] se classent parmi les plus performantes dans les tests de systèmes de reconnaissance à très grands vocabulaires.

L'hybridation d'un MMC avec un réseau neuronal est intéressante du fait des propriétés discriminantes du réseau neuronal. L'hybridation d'un MMC avec d'autres classifieurs discriminants s'est révélée intéressante en RAP, notamment les SVM.

Pour ajouter des paramètres neuronaux, aux paramètres calculés à partir du signal de parole par une des méthodes exposées au chapitre 2, notamment les paramètres cepstraux. On utilise dans ce cas la capacité d'un réseau neuronal à modéliser une distribution de

probabilités quelconque par apprentissage à partir d'exemples. De tels paramètres, associés aux paramètres MFCC, sont actuellement les plus performants en reconnaissance de la parole continue.

III.2.4. Approche Bayésienne

La quasi-totalité des systèmes de reconnaissance de parole continue actuels se fondent sur une approche statistique et plus précisément sur la théorie de la décision bayésienne [15]. Le principe, illustré sur la figure III.4, peut être résumé comme suit.

Le signal de parole est analysé par une des méthodes présentées au chapitre 2. Un mot ou une phrase en entrée du système est ainsi représenté comme une suite $\boldsymbol{x}$ de vecteurs de paramètres. La reconnaissance revient à trouver la suite de mots $\boldsymbol{W}$, formée de n mots, n > 1 n'étant pas connu a priori, dont la probabilité conditionnelle $\boldsymbol{P(W/x)}$ connaissant l'entrée $\boldsymbol{x}$ est maximale.

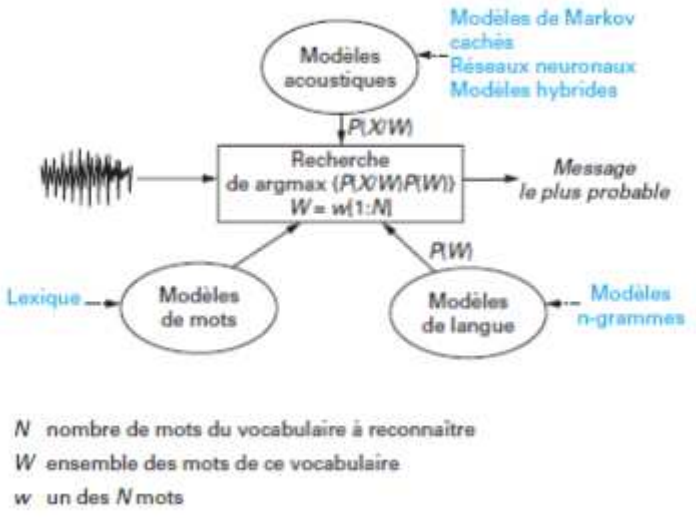


Fig. III.4-Principe de la reconnaissance bayésienne

$\boldsymbol{P(x/W)}$ est la probabilité d'observer la séquence de vecteurs $\boldsymbol{x}$ **ou** $(\boldsymbol{X})$ lorsque la suite de mots $\boldsymbol{W}$ est prononcée. Cette probabilité est donnée par un modèle acoustique, le plus souvent un modèle MMC.

$\boldsymbol{P(W)}$ est la probabilité de la suite de mots $\boldsymbol{W}$ dans le langage utilisé. Elle est fournie par un modèle de langage ML.

Conclusion

Dans ce chapitre nous avons décrit le principe de la Reconnaissance Automatique de la Parole tout en essayant de mettre en évidence les niveaux de complexités majeurs relatifs à la RAP, ainsi que ; sans détailler, cité Les raisons souvent rencontrées qui peuvent affecter les performances de ces systèmes. nous avons définis, également les approches, les principes et les techniques utilisées dans le domaine de la RAP.

Le chapitre suivant sera consacré à la définition des Modelès de Markov Cachés , les algorithmes d'entrainements et de reconnaissance qui ont contribuer grandement à, la théorie et la recherches sur les Modèles de Markov cachés et qui leurs ont permis de s'appliquer et de s'imposer dans beaucoup de domaines .

De nos jours, les MMC, sont un outil largement utilisé dans beaucoup de domaines, incontournable en termes d'efficacités et performances dans le domaine de la reconnaissance automatique de la parole.

CHAPITRE IV : LES MODÈLES DE MARKOV CACHÉS

Introduction

Les modèles de Markov cachés sont des outils statistiques permettant de modéliser des phénomènes stochastiques. Ces modèles sont utilisés dans de nombreux domaines [17] tels que la reconnaissance et la synthèse de la parole, la biologie, l'ordonnancement, l'indexation de documents, la reconnaissance d'images, la prédiction de séries temporelles,… Pour pouvoir utiliser ces modèles efficacement, il est nécessaire d'en connaitre les principes.

Ce chapitre a pour objectif d'établir les principes, les notations utiles et les principaux algorithmes qui constituent la théorie des modèles de Markov cachés (MMC).

A cet effet, nous commençons en présentant un historique des étapes les plus marquantes dans la construction de cette théorie. Après avoir défini ce que sont les chaines de Markov, nous verrons que pour mieux modéliser les phénomènes étudiés, il est nécessaire de considérer un modèle ayant un pouvoir d'expression supérieur. Les modèles de Markov cachés (MMC) en font partie. Nous présentons alors les MMC. La suite s'attache à présenter les algorithmes classiques des MMC pour le décodage\seguementation où la reconnaissance: Forward, Backward et de Viterbi. La dernière section de ce chapitre est consacrée aux différents critères utilisables classiquement pour l'apprentissage de MMC. Finalement, nous terminons chapitre par plusieurs remarques sur les critères d'apprentissage.

IV.1. Historique

Les modèles de Markov cachés ont une longue histoire derrière eux. En 1913, les premiers travaux sur les chaines de markov pour l'analyse du langage permettent à A.A. Markov de concevoir la théorie des chaines de Markov [18]. De 1948 à 1951, Shannon conçoit la théorie de l'information en utilisant les chaines de Markov [19].

Dès 1958, les modèles probabilistes d'urnes [20], le calcul direct du maximum de vraisemblance [21] et l'observation de la suite d'états dans une chaine de Markov [22], sont réalisés. Mais ce n'est qu'a partir de 1966 avec les travaux de L.E. Baum [23], que les algorithmes basiques pour l'estimation des états et des paramètres des modèles, pour les modèles de Markov cachés, sont mis au point. À partir de 1980, ces modèles sont étendus afin d'intégrer la notion de durée variable [25] et densités de probabilités continues multivariables. Les travaux de A. J. Viterbi [26] et G. D. Forney [26] ont permis de construire un algorithme efficace et dont la complexité est linéaire, par apport à la longueur de la suite

d’observations, pour le calcul de la séquence d’états cachés. En 1970, les termes « modèles de Markov cachés » ou « chaines de Markov cachées » (hidden Markov models) mis au point par L. P. Neuwirt afin de remplacer l’appellation « fonction probabiliste d’une chaine de markov » utilisée jusque là [27].

À partir de 1975, les modèles de Markov cachés ont commencé à être utilisés dans de nombreux domaines, parmi lesquelles la reconnaissance automatique de la parole [28]. Les premiers travaux sur les modèles de Markov cachés pour la reconnaissance automatique de la parole ont été menés en parallèle par le groupe IBM composé de L. R. Bahl et F. Jelinek [29] et par J. K. Baker au CMU [30]. Ces travaux ont permis de découvrir les capacités des modèles de Markov cachés pour la reconnaissance de la parole.

Dans les années 1980, les modèles de Markov cachés incorporant des réseaux de neurones apparaissent [31]. Depuis lors, ces nouveaux modèles ont été très largement utilisés pour la reconnaissance de mots isolés [32],pour la reconnaissance de mots enchainés [33], pour la reconnaissance de la parole continue [34] ou pour la localisation de mots dans une phrase [35].

À partir des années 1990, sont mises en œuvre les premières applications à la reconnaissance d’images [36] et de l’écriture apparaissent [37].

Récemment, les modèles de Markov cachés ont même été utilisés pour l’ordonnancement de taches [38] et les technologies [39].

Les modèles de markov cachés sont une famille d’outils mathématiques probabilistes parfaitement adaptés à la modélisation de séquences temporelles. Il existe plusieurs types de modèles de markov cachés afin de mieux répondre à des problèmes spécifiques. Dans le cadre de notre travail et plus particulièrement de ce chapitre, nous nous intéresserons principalement aux modèles de markov cachés discrets du premier ordre, que nous abrégerons par la suite en MMC. Pour pouvoir présenter les MMC, il est nécessaire de commencer par présenter les modèles de Markov et les propriétés qui leurs sont associées.

IV.2. Les chaines de Markov discrètes

En calcul des probabilités, on définit une variable aléatoire « v. a. » réelle comme une fonction mesurable $\boldsymbol{X}: \boldsymbol{\Omega} \rightarrow \mathbb{R}$. $\boldsymbol{\Omega}$ est appelé l'univers. Dans de nombreux cas de figures, $\boldsymbol{\Omega}$ est l'ensemble des réels $\mathbb{R}$, l'ensemble des entiers positifs $\mathbb{N}$ ou un de leurs sous-ensembles.

- **Processus stochastique** Un processus stochastique est une famille $\{\boldsymbol{X_t}\}_{\boldsymbol{t \in T}}$ de v. a. définies sur $\boldsymbol{\Omega}$

$(\boldsymbol{X_t}: \boldsymbol{\Omega} \rightarrow \mathbb{R})$

L'ensemble $\mathbb{T}$ représente souvent la notion de temps mais il peut également correspondre à la notion de position spatiale en dimension 2 ou à toute autre notion en autant de dimensions que nécessaire. Dans le cas où $\mathbb{T}$ représente la notion de temps et si $\mathbb{T}$ est discret, on parle de processus stochastique en temps discret, tandis que le processus est dit en temps continu, lorsque $\mathbb{T}$ est continu. Les états d'un processus stochastique défini par les v. a. $\boldsymbol{X_t}: \boldsymbol{\Omega} \rightarrow \mathbb{R}$ pour tout $\boldsymbol{t} \in \mathbb{T}$ sont les valeurs prises par ces v. a. lorsque $\boldsymbol{t}$ varie. On note $\mathbb{S}$ l'ensemble des « états » du processus.

A. A. Markov fut le premier à étudier et poser les bases mathématiques permettant l'étude des chaines qui portent son nom. La définition de ces chaines est la suivante :

- **Condition d'une chaine de Markov :** Un processus $\{\boldsymbol{S_t}\}_{\boldsymbol{t} \in \mathbb{T}}$ $(\boldsymbol{S_t}: \boldsymbol{\Omega} \rightarrow \mathbb{S})$ est une chaine de markov s'il vérifie les trois conditions suivantes :

$\mathbb{T}$ est dénombrable ou fini. Dans ce cas et pour simplifier les notations ultérieures, il est toujours possible de prendre $\mathbb{T} \subseteq \boldsymbol{N^*} = \{\boldsymbol{1}, \boldsymbol{2}, \ldots\}$. Cette condition signifie que le processus ne change de valeur qu'à des instants déterminés a priori.

L'ensemble $\mathbb{S}$ des états du processus est dénombrable. Dans la suite, nous supposerons également que $\mathbb{S}$ est fini. Nous pouvons alors définir $\mathbb{S} = \{\boldsymbol{s_1}, \ldots, \boldsymbol{s_N}\}$ cet ensemble.

Le processus est associé à une fonction de probabilité $\boldsymbol{P}$ vérifiant la propriété markovienne : « la probabilité que le processus soit dans un état particulier à un instant t ne dépend que de l'état dans lequel se trouve le processus au temps $\boldsymbol{t-1}$ ». Soit $\boldsymbol{Q} = (\boldsymbol{q_t})_{\boldsymbol{t} \in \mathbb{T}}$ une suite d'états du processus $(\boldsymbol{q_t} \in \mathbb{S})$. La propriété de Markov vérifie la relation suivante, pour toute suite d'états $\boldsymbol{Q}$ et pour tout instant $\boldsymbol{t} \in \mathbb{T}$:

$$\boldsymbol{P(S_t = q_t / S_{t-1} = q_{t-1}, \dots, S_1 = q_1) = P(S_t = q_t / S_{t-1} = q_{t-1})} \qquad (12)$$

La probabilité $\boldsymbol{P(S_t = q_t / S_{t-1} = q_{t-1})}$ correspond à la probabilité de transition de l'état $\boldsymbol{q_{t-1}}$ à l'instant $\boldsymbol{t-1}$ vers l'état $\boldsymbol{q_t}$ à l'instant **t**.

- **Homogénéité d'une chaine de Markov :** Une chaine de Markov est homogène (dans le temps) si et seulement si les probabilités de transition ne dépendent pas du temps t « les probabilités de transition sont stationnaires », c'est-à-dire que pour tout $(\boldsymbol{t}, \boldsymbol{t'}) \in \mathbb{T}^{\boldsymbol{2}}$, on a :

$$\boldsymbol{P\left(S_{t+1} = s_j / S_t = s_i\right) = P(S_{t'+1} = s_j / S_{t'} = s_i)}$$

On note $\boldsymbol{a_{i,j}}$ cette probabilité.

Une chaine de Markov homogène est donc totalement définie par la donnée des états, des probabilités des états initiaux **Π** et des probabilités des transitions entre états A avec :

$$\boldsymbol{\Pi = \begin{pmatrix} \pi_1 \\ \vdots \\ \pi_N \end{pmatrix} = (\pi_1, \dots, \pi_N)' \qquad et \qquad \pi_i = P(S_1 = s_i)}$$

$$\boldsymbol{A = (a_{i,j})_{1 \le i,j \le N} \; et \; a_{i,j} = P(S_{t+1} = s_i / S_t = s_j)}$$

- **Vecteurs et matrices stochastiques :**

Un vecteur $\boldsymbol{V = (v_1, \dots, v_N)}$ de dimension *N* « ou, de manière équivalente, son transposé » est stochastique si et seulement si :

Pour tout $\boldsymbol{i = 1 \dots N, 0 \le v_i \le 1, \sum_{i=1}^{N} v_i = 1}$.

Une matrice $\boldsymbol{M = (m_{i,j})_{1 \le i,j \le N}}$ de dimension $\boldsymbol{N * N}$ est dite stochastique si et seulement,

si Pour tout $\boldsymbol{i = 1 \dots N}$ $\boldsymbol{et}$ $\boldsymbol{j = 1 \dots N, 0 \le m_{i,j} \le 1}$,

Pour tout $\boldsymbol{i = 1 \dots N, \sum_{j=1}^{N} m_{i,j} = 1}$.

- **Caractéristique d'une chaine de Markov :**

Une matrice est stochastique si et seulement si les lignes qui la composent sont des vecteurs stochastiques.

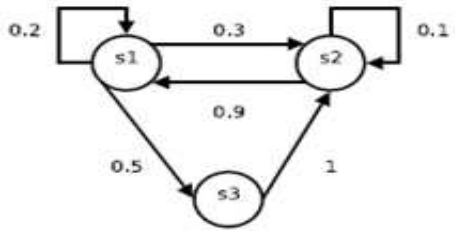
Le système est forcément dans un et un seul état particulier au départ donc Π est un vecteur stochastique. $\boldsymbol{A}$ est une matrice stochastique car, en partant dans un état $\boldsymbol{s_i}$ à l'instant $\boldsymbol{t}$, le processus transite forcément vers l'un des $\boldsymbol{N}$ états du système au temps $\boldsymbol{t+1}$.

A tout couple formé d'un vecteur stochastique $\boldsymbol{V}$ de dimension $\boldsymbol{N}$ et d'une matrice stochastique $\boldsymbol{M}$ de dimensions $\boldsymbol{N*N}$, il est possible d'associer une chaine de Markov caractérisée par le couple $(\boldsymbol{V},\boldsymbol{M})$.

- **Représentation graphique d'une chaine de Markov** : Une chaine de Markov peut être représentée graphiquement. Pour cela, on associe à la chaine de Markov $\{\boldsymbol{S_t}\}_{\boldsymbol{t}\in\mathbb{T}}$ un graphe $\boldsymbol{G}$ dont l'ensemble des sommets est en bijection avec l'ensemble des états $\mathbb{S}$ et dont l'ensemble des arcs $\boldsymbol{U}$ (orientés dans le sens des transitions) est défini par

$$(\boldsymbol{s_i},\boldsymbol{s_j}) \in \boldsymbol{U} \leftrightarrow \boldsymbol{a_{i,j}} > 0$$

Afin de simplifier les notations, l'ensemble des sommets du graphe $\boldsymbol{G}$ est représenté par l'ensemble $\mathbb{S}$. La figure **IV.5** présente la représentation graphique associée à la chaine de Markov $(\boldsymbol{\Pi},\boldsymbol{A})$.



$$A = \begin{pmatrix} 0.2 & 0.3 & 0.5 \\ 0.9 & 0.1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \qquad \Pi = \begin{pmatrix} 0.4 \\ 0.2 \\ 0.4 \end{pmatrix}$$

Fig. IV.5- Représentation graphique de la chaine de Markov $(\boldsymbol{\Pi},\boldsymbol{A})$

- **IV.2.1 Les modèles de Markov cachés discrets (MMC) (HMM)**

Les chaines de Markov peuvent servir à modéliser de nombreux processus stochastiques.

Cependant, dans certains cas, ces modèles ne permettent pas d'exprimer le comportement du système avec suffisamment de précision. Pour améliorer cette précision, les modèles de Markov cachés ont été développés.

Un modèle de Markov caché discret correspond à la modélisation de deux processus stochastiques : un processus caché parfaitement modéliser par une chaine ce Markov discrète et un processus observé dépendant des états du processus caché.

Soit $\mathbb{S} = \{\boldsymbol{s_1}, \dots, \boldsymbol{s_N}\}$ l'ensemble des $\boldsymbol{N}$ états cachés du système. Soit $\boldsymbol{S} = (\boldsymbol{S_1}, \dots, \boldsymbol{S_T})$ un $\boldsymbol{T}$-uple de v. a. définies sur $\mathbb{S}$. Soit $\mathbb{V} = \{\boldsymbol{v_1}, \dots, \boldsymbol{v_M}\}$ l'ensemble des $\boldsymbol{M}$ symboles émissibles par le système. Soit $\boldsymbol{V} = (\boldsymbol{v_1}, \dots, \boldsymbol{v_M})$ un $\boldsymbol{T}$-uple de v. a. définies sur $\mathbb{V}$.

Un modèle de Markov caché discret du premier ordre est alors défini par les probabilités suivantes :

- Les probabilités d'initialisation des états cachés : $\boldsymbol{P(S_1 = s_i)}$
- Les probabilités de transition entre états cachés : $\boldsymbol{P(S_t = s_j / S_{t-1} = s_i)}$
- Les probabilités d'émission des symboles dans chaque état caché :

$$\boldsymbol{P(V_t = v_j / S_t = s_i)}.$$

Si le modèle de Markov caché est stationnaire alors les probabilités de transition entre états cachés et les probabilités d'émission des symboles dans chaque état caché sont indépendantes du temps $\boldsymbol{t} > 1$.

On peut alors définir, pour tout $\boldsymbol{t} > 1$ quelconque, $\boldsymbol{A} = (\boldsymbol{a_{i,j}})_{\boldsymbol{1 \le i,j \le N}}$ avec

$\boldsymbol{a_{i,j}} = \boldsymbol{P(S_t = s_j / S_{t-1} = s_i)}, \boldsymbol{B} = (\boldsymbol{b_i(j)})_{\boldsymbol{1 \le i \le N, 1 \le j \le M}}$ avec $\boldsymbol{b_i(j)} = \boldsymbol{P(V_t = v_j / S_t = s_i)}$ et $\boldsymbol{\Pi} = (\boldsymbol{\pi_1}, \dots, \boldsymbol{\pi_N})'$ avec $\boldsymbol{\pi_i} = \boldsymbol{P(S_1 = s_i)}$. Un modèle de markov caché stationnaire du premier ordre $\boldsymbol{\lambda}$ est donc totalement défini par le triplet $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi})$. Par la suite, nous utiliserons la notation $\boldsymbol{\lambda} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi})$ et nous emploierons le terme MMC pour des modèles de Markov

cachés stationnaires du premier ordre. Les relations de dépendance entre les différentes variables aléatoires d'un MMC sont schématisées par la figure . Dans cette représentation, les flèches partent de la v. a. qui conditionne et se terminent au niveau de la variable aléatoire conditionnée. Dans la figure IV.7, seules les transitions au temps $\boldsymbol{t-1}$, $\boldsymbol{t}$ et $\boldsymbol{t+1}$ sont représentées.

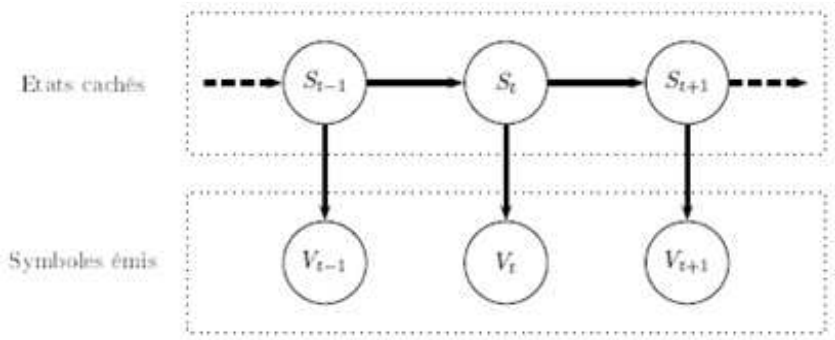


Fig. IV.7-Relation de dépendance entre les variables aléatoires d'un *MMC*

On note $\boldsymbol{Q} = (\boldsymbol{q_1}, \dots, \boldsymbol{q_T}) \in \boldsymbol{S^T}$ une séquence d'états cachés et $\boldsymbol{O} = (\boldsymbol{o_1}, \dots, \boldsymbol{o_T}) \in \mathbb{V}^{\boldsymbol{T}}$ une séquence de symboles observés. La probabilité de réalisation de la séquence d'états cachés $\boldsymbol{Q}$ et de la séquence d'observation $\boldsymbol{O}$ par rapport au ***MMC*** $\boldsymbol{\lambda}$ est alors

$$\boldsymbol{P(V = O, S = Q/A, B, \Pi)}$$

Ou plus simplement

$$\boldsymbol{P(V = O, S = Q/\lambda)}$$

En utilisant les dépendances des probabilités conditionnelles, on déduit que :

$$\boldsymbol{P(V = O, S = Q/\lambda) = P(V = O/S = Q, \lambda)P(S = Q/\lambda)}$$

De plus,

$$\boldsymbol{P(V = O/S = Q, \lambda) = \prod_{t=1}^{T} P(V_t = o_t/S_t = q_t, \lambda)}$$

$$\boldsymbol{P(S = Q/\lambda) = P(S_1 = q_1/\lambda) \prod_{t=1}^{T-1} P(S_{t+1} = q_{t+1}/S_t = q_t, \lambda)}$$

A partir d'un **MMC** $\boldsymbol{\lambda}$, d'une séquence d'états cachés $\boldsymbol{Q}$ et d'une séquence d'observations $\boldsymbol{O}$, il est possible de calculer l'adéquation entre le modèle $\boldsymbol{\lambda}$ et les deux séquences $\boldsymbol{Q}$ et $\boldsymbol{O}$.

Pour cela, il suffit de calculer la probabilité $\boldsymbol{P(V = O, S = Q/\lambda)}$. Cette dernière correspond à la probabilité que la séquence d'observations $\boldsymbol{O}$ ait effectivement été engendrée par le modèle $\boldsymbol{\lambda}$ en suivant la séquence d'états cachés $\boldsymbol{Q}$.

Lorsque la séquence d'états cachés n'est pas connue, il est possible d'évaluer la vraisemblance d'une séquence d'observation $\boldsymbol{O}$ par rapport à un modèle $\boldsymbol{\lambda}$. La vraisemblance correspond à la probabilité $\boldsymbol{P(V = O/\lambda)}$ que la séquence d'observations ait été engendrée par le modèle pour l'ensemble des séquences d'états cachés possibles. On remarque alors que la formule suivante est vérifiée :

$$\boldsymbol{P(V = O/\lambda) = \sum_{Q \in S^T} P(V = O, S = Q/\lambda)}$$

L'utilisation des MMC, nécessite la résolution de plusieurs problèmes principaux : le calcul de la vraisemblance, le décodage / segmentation de séquence d'observations et l'apprentissage.

IV.3. calcul de la vraisemblance

Comme nous l'avons vu précédemment, calculer la vraisemblance d'une séquence de $\boldsymbol{T}$ observations par apport à un ***MMC*** $\boldsymbol{\lambda}$ consiste à évaluer la probabilité $\boldsymbol{P(V = O/\lambda)}$. Ce calcul peut s'effectuer en utilisant differentes méthodes, dans ce qui suit, nous allons expliquer les principes de chaque méthode et de présenter l'algoithme de calcul correspendant.

IV.3.1. L'algorithme Forward

Pour présenter rapidement cet algorithme, il est nécessaire de définir les variables Forward [28] **(**pour tout $\boldsymbol{i = 1 \dots N}$ et $\boldsymbol{t = 2 \dots T}$ **)** :

$$\begin{cases} \boldsymbol{\alpha_1(i) = P(V_1 = o_1, S_1 = s_i/\lambda)} \\ \boldsymbol{\alpha_t(i) = P(V_1 = o_1, \dots, V_t = o_t, S_t = s_i/\lambda)} \end{cases}$$

On remarque alors que la relation de récurrence suivante est vérifiée pour tout $\boldsymbol{t = 1 \dots T - 1}$ et $\boldsymbol{j = 1 \dots N}$.

$$\boldsymbol{\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{i,j}}$$

De plus, on a $\boldsymbol{P(V = O/\lambda) = \sum_{i=1}^{N} \alpha_T(i)}$. L'algorithme Forward est alors donné par l'algorithme 1.1. La complexité de cet algorithme est en $\boldsymbol{O(N^2 T)}$.

Pour $i = 1$ à N **faire**
  $\alpha_1(i) = \pi_i b_i(o_1)$
Fin pour
Pour $t = 1$ à $T - 1$ **Faire**
  Pour $j = 1$ à N **Faire**
    $\alpha_{t+1}(j) = (\sum_{i=1}^{N} \alpha_t(i) a_{ij}(o_{t+1})$
  Fin Pour
Fin pour

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

Algorithme IV.1 : Algorithme Forward

Cet algorithme permet de calculer la vraisemblance d'une séquence d'observations. Cependant, dans la pratique, des problèmes de précision numérique apparaissent à l'implémentation rendant l'algorithme Forward inutilisable. Une solution consiste à opérer un ré-échelonnement des valeurs [28]. Pour cela, on définit deux ensembles de variables $\alpha'_t(i)$ et $\tilde{\alpha}_t(i)$ **(** pour tout $i = 1 \ldots N$ et $t = 2 \ldots T$ **)** par :

$$\begin{cases} \alpha'_1 = \alpha_1(i) \\ \alpha'_t(i) = \sum_{j=1}^{N} \tilde{\alpha}_{t-1}(j) a_{i,j}. b_j(o_t) \end{cases}$$

On définit c_t (pour tout $t = 1 \ldots T$) le coefficient de normalisation à 1de la somme des $\tilde{\alpha}_t(i)$ ($i.e. \sum_{i=1}^{N} \tilde{\alpha}_t(i) = 1$) par $c_t = \frac{1}{\sum_{i=1}^{N} \alpha'_t(i)}$. On pose $\tilde{\alpha}_t(i) = c_t \alpha'_t(i)$ avec $C_t = \prod_{k=1}^{t} c_k$.

On montre par récursivité que

$$\tilde{\alpha}_t(i) = C_t \alpha_t(i)$$

Or, par définition, on a $\sum_{i=1}^{N} \tilde{\alpha}_t(i) = 1$ d'où :

$$P(V = O/\lambda) = \frac{1}{C_T}$$

L'algorithme Forward avec ré-échelonnement (également nommé rescaling) [28] est donné par l'algorithme **IV.2**. Sa complexité est identique à celle de l'algorithme Forward, c'est-à-dire $O(N^2 T)$, cependant l'algorithme nécessite plus d'opérations. De plus, la valeur prise par $P(V = O/\lambda)$ est très petite et est considérée la plupart du temps comme étant nulle dans les représentations en nombres réels sur les machines. Par conséquent, on considère plus facilement son logarithme, qui s'obtient par :

$$\ln P(V = O/\lambda) = \ln \frac{1}{C_T} = -\sum_{t=1}^{T} \ln c_t$$

Pour $i = 1$ à N **Faire**
  $\alpha'_1(i) = \pi_i b_i(o_1)$
Fin pour
$c_1 = \frac{1}{\sum_{i=1}^{N} \alpha'_1(i)}$
Pour $i = 1$ à N **Faire**
  $\tilde{\alpha}_1(i) = c_1 \alpha'_1(i)$
Fin Pour
Pour $t = 1$ à $T - 1$ **Faire**
  Pour $j = 1$ à N **Faire**
    $\alpha'_{t+1}(j) = \left(\sum_{i=1}^{N} \tilde{\alpha}_t(i) a_{ij}\right) b_j(o_{t+1})$
  Fin Pour
  $c_t = \frac{1}{\sum_{j=1}^{N} \alpha'_t(j)}$
  Pour $j = 1$ à N **Faire**
  $\tilde{\alpha}_{t+1}(j) = c_t \alpha'_{t+1}(j)$

  Fin Pour
Fin pour

$$P(V = O/\lambda) = \frac{1}{\prod_{t=1}^{T} c_t}$$

Algorithme IV.2 :Algorithme Forward avec ré-échelonnement

IV.3.2. L'algorithme Backward

Bien que le problème du calcul de la vraisemblance soit résolu, nous allons également présenter l'algorithme Backward [36] qui permet aussi de calculer la vraisemblance et qui surtout sera nécessaire dans les sections ultérieurs, notamment pour l'apprentissage. Les variables Backward sont définies par

(pour tout $i = 1 \ldots N$ et $t = 1 \ldots T - 1$) :

$$\begin{cases} \beta_T(i) = 1 \\ \beta_t(i) = P(V_{t+1} = o_{t+1}, \ldots, V_T = o_T / S_t = s_i / \lambda) \end{cases}$$

Pour tout $i = 1 \ldots N$ et $t = 1 \ldots T - 1$, les relations suivante sont vérifiées :

$$\beta_t(i) = \sum_{i=1}^{N} a_{i,j} \beta_{t+1}(i) b_i(o_{t+1})$$

$$P(V = O/\lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i)$$

L'algorithme Backward, de même complexité que l'algorithme Forward, est donné par l'algorithme **IV.3**.

Pour $\boldsymbol{i = 1}$ à $\boldsymbol{N}$ **Faire**
$\boldsymbol{\beta_T(i) = 1}$
Fin Pour
Pour $\boldsymbol{t = T - 1}$ à $\boldsymbol{1}$ **Faire**
Pour $\boldsymbol{i = 1}$ à $\boldsymbol{N}$ **Faire**

$$\boldsymbol{\beta_t(i) = \sum_{j=1}^{N} a_{ij} \beta_{t+1}(j) b_j(o_{t+1})}$$

Fin Pour
Fin Pour

$$\boldsymbol{P(V = O/\lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i)}$$

Algorithme IV.3 :Algorithme Backward

Tout comme l'algorithme Forward, l'algorithme Backward souffre de problème de précision numérique. Par conséquent, il est nécessaire d'utiliser le ré-échelonnement des variables Backward. Pour cela, on définit l'ensemble de variables $\widetilde{\boldsymbol{\beta}}_{\boldsymbol{t}}(\boldsymbol{i})$ par **(** pour tout $\boldsymbol{i = 1 \ldots N}$ et $\boldsymbol{t = 1 \ldots T - 1}$ **)** :

$$\begin{cases} \widetilde{\boldsymbol{\beta}}_{\boldsymbol{T}}(\boldsymbol{i}) = \boldsymbol{c_T} \\ \widetilde{\boldsymbol{\beta}}_{\boldsymbol{t}}(\boldsymbol{i}) = \boldsymbol{c_t} \sum_{j=1}^{N} \boldsymbol{a_{i,j} b_j(o_{t+1}) \widetilde{\beta}_{t+1}(j)} \end{cases}$$

On pourra remarquer que les coefficients c_t sont ceux calculés précédemment pour l'algorithme Forward avec ré-échelonnement.

En définissant $\boldsymbol{D_t = \prod_{k=t}^{T} c_k}$, il est possible de montrer les relations suivantes :

$$\boldsymbol{D_1 = C_T = \frac{1}{P(V=O/\lambda)}}$$

$$\boldsymbol{C_t D_{t+1} = C_T = \frac{1}{P(V=O/\lambda)}}$$

$$\widetilde{\boldsymbol{\beta}}_{\boldsymbol{t}}(\boldsymbol{i}) = \boldsymbol{D_t \beta_t(i)}$$

L'algorithme Backward avec ré-échelonnement [28] est donné par l'algorithme **IV.4**. Sa complexité est identique à celle de l'algorithme Backward, c'est-à-dire $\boldsymbol{O(N^2 T)}$. On remarque également que le calcul de $P(V = O/\lambda)$ par cet algorithme offre peu d'intérêt, car il nécessite de connaitre les coefficients $\boldsymbol{c_t}$ de l'algorithme Forward avec ré-échelonnement.

Pour $\boldsymbol{i = 1}$ à $\boldsymbol{N}$ **Faire**

$\widetilde{\boldsymbol{\beta}}_{\boldsymbol{T}}(\boldsymbol{i}) = \boldsymbol{c}_{\boldsymbol{T}}$

Fin Pour

Pour $\boldsymbol{t = T - 1}$ à $\boldsymbol{1}$ **Faire**

Pour $\boldsymbol{i = 1}$ à $\boldsymbol{N}$ **Faire**

$\widetilde{\boldsymbol{\beta}}_{\boldsymbol{t}}(\boldsymbol{i}) = \boldsymbol{c}_{\boldsymbol{t}} \sum_{\boldsymbol{j=1}}^{\boldsymbol{N}} \boldsymbol{a}_{\boldsymbol{i,j}} \widetilde{\boldsymbol{\beta}}_{\boldsymbol{t+1}}(\boldsymbol{j}) \boldsymbol{b}_{\boldsymbol{j}}(\boldsymbol{o}_{\boldsymbol{t+1}})$

Algorithme IV.4 :Algorithme ***Backward avec ré-échelonnement***

IV.3.3. Probabilités déductibles

A partir des variables Forward et Backward, avec ou sans ré-échelonnement, il nous est d'ores et déjà possible d'exprimer deux probabilités utiles.

$$\boldsymbol{P(V = O, S_t = s_i/\lambda) = \alpha_t(i)\beta_t(i)}$$

$$= \frac{\widetilde{\boldsymbol{\alpha}}_{\boldsymbol{t}}(\boldsymbol{i})\widetilde{\boldsymbol{\beta}}_{\boldsymbol{t}}(\boldsymbol{i})}{\boldsymbol{P(V = O/\lambda)}}$$

$$\boldsymbol{P\big(V = O, S_t = s_i, S_{t+1} = s_j/\lambda\big) = \alpha_t(i)a_{i,j}b_j(o_{t+1})\beta_{t+1}(j)}$$

$$= \frac{\widetilde{\boldsymbol{\alpha}}_{\boldsymbol{t}}(\boldsymbol{i})\boldsymbol{a}_{\boldsymbol{i,j}}\boldsymbol{b}_{\boldsymbol{j}}(\boldsymbol{o}_{\boldsymbol{t+1}})\widetilde{\boldsymbol{\beta}}_{\boldsymbol{t+1}}(\boldsymbol{j})}{\boldsymbol{P(V = O/\lambda)}}$$

IV.3.4. Décodage/segmentation de séquences d'observations

Le décodage ou la segmentation de séquences d'observations consiste à trouver la séquence d'états cachés qui a engendré une séquence d'observations. Deux approches sont possibles. La première consiste à rechercher, à chaque instant, l'état qui a le plus probablement engendré le symbole observé. La deuxième approche consiste à trouver la séquence complète d'états cachés qui a le plus probablement engendré la séquence d'observations.

IV.3.4.1. Etats cachés les plus probables à chaque instant

Dans cette approche, on cherche la séquence $\boldsymbol{Q^* = (q_1^*, \dots, q_T^*) \in \mathbb{S}^T}$ vérifiant, pour tout $\boldsymbol{t = 1 \dots T}$, l'équation :

$$\boldsymbol{q_t^* = \arg max_{i=1\dots N} P(V = O, S_t = s_i/\lambda)}$$

Il est donc nécessaire, d'après la formule 1.1, de calculer en premier lieu les variables Forward et Backward. Malgré sa formulation simple, le recherche de l'état caché le plus probable à chaque instant a une complexité en $\boldsymbol{O(N^2T)}$. De plus, la séquence $\boldsymbol{Q^*}$ obtenue peut être inconsistante, dans le sens où $\boldsymbol{P(V = O, S = Q^*) = 0}$. En effet, il est possible que la transition entre deux états $\boldsymbol{s_i}$ et $\boldsymbol{s_j}$ existe dans la séquence $\boldsymbol{Q^*}$, alors que la probabilité $\boldsymbol{a_{i,j}}$ est nulle.

IV.3.4.2. Algorithme de viterbi

La recherche de la séquence d'états cachés $\boldsymbol{Q^*}$ qui le plus probablement engendré une séquence d'observations $\boldsymbol{O}$ consiste à résoudre

$$\boldsymbol{Q^* = \arg max_{Q \in \mathbb{S}^T} P(V = O, S = Q/\lambda)}$$

L'algorithme permettant de résoudre ce problème est l'algorithme de Viterbi [28].

On définit

$$\boldsymbol{\delta_t(i) = max_{(q_1,\dots,q_{t-1} \in \mathbb{S}^{t-1})}\{P(S_1 = q_1, \dots, S_{t-1} = q_{t-1}, S_t = s_i, V_t = o_1, \dots, V_t = o_t/\lambda)}$$

La probabilité du meilleur chemin partiel amenant à l'état caché $\boldsymbol{s_i}$ au temps $\boldsymbol{t}$ et $\boldsymbol{\psi_t(i)}$ le meilleur chemin amenant à l'état $\boldsymbol{s_i}$ au temps $\boldsymbol{t}$ à partir du temps $\boldsymbol{t - 1}$.

L'algorithme de Viterbi est alors donné par l'algorithme 5. Sa complexité est $\boldsymbol{O(N^2T)}$.

Pour $i = 1$ à N **Faire**
$\delta_1(i) = \pi_i b_i(o_1)$
Fin Pour
Pour $t = 2$ à T **Faire**
Pour $j = 1$ à N **Faire**
$\psi_t(j) = argmax_{1\le i\le N}\{\delta_{t-1}(i)a_{ij}\}$

$\delta_t(j) = max_{1\le i\le N}\left\{\delta_{t-1}(\psi_t(j))a_{\psi_{t(j),j}}b_j(o_t)\right\}$
Fin Pour
Fin Pour
$q_T^* = arg\ max_{1\le i\le N}\{\delta_T(i)\}$
$P(V = O, S = Q^*) = max_{1\le i\le N}\{\delta_T(i)\} = \delta_T(q_T^*)$
Pour $t = T - 1$ à **Faire**
$q_t^* = \psi_{t+1}(q_{t+1}^*)$

Algorithme. IV.5 :Algorithme de Viterbi

Tout comme les algorithmes Forward et Backward, cet algorithme souffre de problèmes liés à l'implémentation. Pour les résoudre, il est également nécessaire de mettre en place une stratégie de ré-échelonnement. A cet effet, on définit

$\widetilde{\delta}_t(i) = max_{(q_1,\dots,q_{t-1})\in\mathbb{S}^{t-1}} \ln P(S_1 = q_1, \dots, S_{t-1} = q_{t-1}, S_t = s_i, V_1 = o_1, \dots, V_t = o_t/\lambda)$

L'algorithme de Viterbi avec ré-échelonnement [28] est alors donné par l'algorithme 6. Sa complexité est $O(N^2T)$, cependant, le calcul des logarithmes peut d'avérer plus couteux.

Pour $i = 1$ à N **Faire**
$\widetilde{\delta}_1(i) = ln\ \pi_i + lnb_i(o_1)$
Fin Pour
Pour $t = 2$ à T **Faire**
Pour $j = 1$ à N **Faire**
$\widetilde{\psi}_t(j) = argmax_{1\le i\le N}\{\widetilde{\delta}_{t-1} + ln\ a_{ij}\}$
$\widetilde{\delta}_t(j) = max_{1\le i\le N}\{\widetilde{\delta}_{t-1}(i) + lna_{ij}\} + lnb_j(o_t)$
$= \widetilde{\delta}_{t-1}(\widetilde{\psi}_t(j) + lna_{\widetilde{\psi}_t(j),j} + lnb_j(o_t)$
Fin Pour
Fin Pour
$q_T^* = argmax_{1\le i\le N}\{\widetilde{\delta}_T(i)\}$
$lnP(V = O, S = Q^*) = max_{1\le i\le N}\{\widetilde{\delta}_T(i)\} = \widetilde{\delta}_T(q_T^*)$
Pour $t = T - 1$ à 1 **Faire**
$q_t^* = \widetilde{\psi}_{t+1}(q_{t+1}^*)$
Fin Pour

Algorithme IV.6 :Algorithme de Viterbi avec ré-échelonnement

IV.4. Apprentissage des modèles de Markov cachés

Apprendre un MMC c'est ajuster les paramètres du modèle de manière à maximiser un certain critère. Différents critère sont disponibles dans la littérature. Nous n'allons pas tous les recenser, mais nous allons présenter les plus importants et les plus couramment utilisés.

IV.4.1 Apprentissage étiqueté

Pour effectuer un apprentissage étiqueté, également connu dans la littérature comme l'apprentissage de Viterbi, on dispose de deux informations : la séquence d'observations $\boldsymbol{O}$ et la séquence d'états cachés $\boldsymbol{Q}$ qui a engendré la séquence précédente. Le critère que l'on cherche à maximiser est $\boldsymbol{P(V = O, S = Q/\lambda)}$. Pour le maximiser, il suffit de compter les différentes transitions du système. Habituellement, avec ce type d'apprentissage, on ne considère pas une seule séquence d'observations à la fois, mais plusieurs. Notons $\{\boldsymbol{O^1}, \dots, \boldsymbol{O^K}\}$ les séquences d'observations, $\{\boldsymbol{Q^1}, \dots, \boldsymbol{Q^K}\}$ les séquences d'états associées et $\{\boldsymbol{T^1}, \dots, \boldsymbol{T^K}\}$ les longueurs des séquences. Dans ce cas, on utilise toujours le comptage des différentes transitions du système, mais en considérant toutes les séquences simultanément de manière indistincte. L'algorithme d'apprentissage étiqueté est donné par l'algorithme **IV.7**.

Sa complexité est $\boldsymbol{O(N^2 + NM + T)}$ en désignant par $\boldsymbol{T}$, la longueur totale des séquences d'observations considérées.

$\forall \boldsymbol{i = 1 \dots N, x_i = 0}$
$\forall \boldsymbol{i = 1 \dots N, j = 1 \dots N, y_{i,j} = 0}$
$\forall \boldsymbol{i = 1 \dots N, j = 1 \dots M, z_{i,j} = 0}$
Pour $\boldsymbol{k = 1}$ **à** $\boldsymbol{K}$ **Faire**
  Incrémenter $\boldsymbol{x_{q_1^k}}$
  Pour $\boldsymbol{t = 1}$ **à** $\boldsymbol{T^k}$ **Faire**
  Incrémenter $\boldsymbol{y_{q_t^k}, o_t^k}$
  Si $\boldsymbol{t < T^k}$ **Alors**
    Incrémenter $\boldsymbol{z_{q_t^k}, q_{t+1}^k}$
  Fin Si
  Fin Pour
Fin Pour

$$\forall \boldsymbol{i = 1 \dots N, \pi_i = \frac{x_i}{\sum_r^N x_r}}$$
$$\forall \boldsymbol{i = 1 \dots N, j = 1 \dots N, a_{i,j} = \frac{y_{i,j}}{\sum_{r=1}^N y_{i,r}}}$$
$$\forall \boldsymbol{i = 1 \dots N, j = 1 \dots M, b_i(j) = \frac{z_{i,j}}{\sum_{r=1}^M z_{i,r}}}$$

Algorithme IV.7 : Apprentissage étiqueté

Lorsque le nombre de séquences d'observations ou de séquences d'états cachés ou tout simplement le nombre d'apparitions d'un ou plusieurs motifs est trop réduit, l'apprentissage est souvent peu efficace, car le modèle n'arrive pas à généraliser ce qu'il doit reconnaitre. En effet, de nombreuses probabilités sont très petites, voire nulles. Un moyen de résoudre ces problèmes consiste à effectuer un lissage lors de l'estimation des probabilités. En notant $\boldsymbol{c} > 0$ le coefficient de lissage, l'algorithme est donné par l'algorithme **IV.8**.

Dans cet algorithme, le coefficient de lissage est identique pour toutes les probabilités, mais rien n'empêche de le choisir différent pour chacune d'elles, afin d'inclure des connaissances expertes dans l'apprentissage.

$\forall \boldsymbol{i} = \boldsymbol{1} \dots \boldsymbol{N}, \boldsymbol{x_i} = \boldsymbol{0}$
$\forall \boldsymbol{i} = \boldsymbol{1} \dots \boldsymbol{N}, \boldsymbol{j} = \boldsymbol{1} \dots \boldsymbol{N}, \boldsymbol{y_{i,j}} = \boldsymbol{0}$
$\forall \boldsymbol{i} = \boldsymbol{1} \dots \boldsymbol{N}, \boldsymbol{j} = \boldsymbol{1} \dots \boldsymbol{M}, \boldsymbol{z_{i,j}} = \boldsymbol{0}$
Pour $\boldsymbol{k} = \boldsymbol{1}$ à $\boldsymbol{K}$ **Faire**
  Incrémenter $\boldsymbol{x_{q_1^k}}$
  Pour $\boldsymbol{t} = \boldsymbol{1}$ à $\boldsymbol{T^k}$ **Faire**
    Incrémenter $\boldsymbol{y_{q_t^k}, o_t^k}$
    Si $\boldsymbol{t} < \boldsymbol{T^k}$ **Alors**
      Incrémenter $\boldsymbol{z_{q_t^k}, q_{t+1}^k}$
    Fin Si
  Fin Pour
Fin Pour

$$\forall \boldsymbol{i} = \boldsymbol{1} \dots \boldsymbol{N}, \boldsymbol{\pi_i} = \frac{\boldsymbol{c} + \boldsymbol{x_i}}{\boldsymbol{N_c} + \sum_{r=1}^{N} \boldsymbol{x_r}}$$

$$\forall \boldsymbol{i} = \boldsymbol{1} \dots \boldsymbol{N}, \boldsymbol{j} = \boldsymbol{1} \dots \boldsymbol{N}, \boldsymbol{a_{i,j}} = \frac{\boldsymbol{c} + \boldsymbol{y_{i,j}}}{\boldsymbol{N_c} + \sum_{r=1}^{N} \boldsymbol{y_{i,r}}}$$

$$\forall \boldsymbol{i} = \boldsymbol{1} \dots \boldsymbol{N}, \boldsymbol{j} = \boldsymbol{1} \dots \boldsymbol{M}, \boldsymbol{b_i(j)} = \frac{\boldsymbol{c} + \boldsymbol{z_{i,j}}}{\boldsymbol{M_c} + \sum_{r=1}^{M} \boldsymbol{z_{i,r}}}$$

Algorithme IV.8 : Apprentissage étiqueté avec lissage

IV.4.2 Maximisation de la vraisemblance

Le critère de maximum de vraisemblance consiste à trouver le modèle $\boldsymbol{\lambda}^*$ maximisant la probabilité $\boldsymbol{P}(\boldsymbol{V} = \boldsymbol{O}/\boldsymbol{\lambda})$**[36]**. En général, il n'est pas possible de trouver ce modèle optimal. Néanmoins, pour tenter de résoudre ce problème, il existe principalement deux méthodes : utiliser l'algorithme Expectation-Maximisation, ou utiliser une descente de gradient.

IV.4.2.1. Introduction à l'algorithme Expectation-Maximisation

L'algorithme Expectation-Maximisation (EM) est une méthode générale d'optimisation en présence d'information incomplète. L'algorithme permet, à partir d'un modèle initial $\boldsymbol{m'}$, de trouver un modèle $\boldsymbol{m}$ qui augmente la vraisemblance. Dans cette section, nous ne démontrerons pas l'algorithme **EM**. Nous nous contenterons juste d'exposer les principes et formules qui nous seront nécessaires par la suite. Le lecteur intéressé trouvera dans [40] un exposé plus complet de la méthodologie de l'algorithme **Expectation-Maximisation**.

Particulièrement bien adapté à des probabilités, l'algorithme **EM** repose sur deux hypothèses simples :

maximiser $\boldsymbol{P(X = x/M = m)}$ est équivalent à maximiser $\boldsymbol{\ln P(X = x/M = m)}$;

l'introduction de variables non observées ou cachées $\boldsymbol{Y}$ définies sur $\mathbb{Y}$ dans l'expression de la vraisemblance permet d'effectuer les calculs plus facilement.

Dans le cas de variables aléatoires discrètes, on définit $\boldsymbol{\Gamma_x(m, m')}$, [39] par :

$$\boldsymbol{\Gamma_x(m, m') = \sum_{y\in\mathbb{Y}} P(Y = y/X = x, M = m') \ln P(X = x, Y = y/M = m)}$$

$$\boldsymbol{= E_{y\in\mathbb{Y}}[\ln P(X = x, Y = y/M = m)/X = x, M = m']}$$

Avec $\mathbb{E}_{\mathbb{Y}}[\boldsymbol{f}]$ l'espérance mathématique de $\boldsymbol{f}$ sur l'ensemble $\mathbb{Y}$.

L'algorithme EM [40] consiste donc à construire, à partir d'un modèle initial $\boldsymbol{m_1}$, une suite de modèles $(\boldsymbol{m_t})_{\boldsymbol{t>0}}$ vérifiant

$$\boldsymbol{\Gamma_x(m_{t+1}, m_t) \geq \Gamma_x(m_t, m_t)}$$

Une condition suffisante est alors de rechercher le modèle m_{t+1} qui maximise la fonction $\boldsymbol{\Gamma_x(m_{t+1}, m_t)}$. La suite $(\boldsymbol{m_t})_{\boldsymbol{t>0}}$ vérifie, pour tout $\boldsymbol{t > 1}$ et $\boldsymbol{m_{t+1} \neq m_t}$, la relation

$$\boldsymbol{P(X = x/M = m_{t+1}) > P(X = x/M = m_t)}$$

L'un des plus célèbres applications de l'algorithme EM est l'algorithme Baum-Welch permettant l'apprentissage des MMC .

IV.4.2.2. L'algorithme de Baum-Welch

Dans le cas des MMC, on cherche à maximiser $\boldsymbol{P(V = O/\lambda)}$ où $\boldsymbol{O}$ désigne une séquence de $\boldsymbol{T}$ observations. En appliquant l'algorithme EM à la maximisation de cette probabilité**.** On est amené à maximiser $\boldsymbol{\Gamma(\lambda, \lambda')}$, avec $\boldsymbol{\lambda = (A, B, \Pi)}$ le nouveau modèle et $\boldsymbol{\lambda'}$ le modèle connu (ou actuel) :

$$\boldsymbol{\Gamma_o(\lambda, \lambda') = \sum_{Q \in \mathbb{S}^T} P(S = Q/V = O, \lambda') \ln P(V = O, S = Q/\lambda)}$$

En effectuant les différents calculs, on obtient :

$$\boldsymbol{\pi_i = P(S_1 = s_i/O\,, \lambda')}$$

$$\boldsymbol{a_{i,j} = \frac{\sum_{t=1}^{T-1} P(S_t = s_i, S_{t+1} = s_j/V = O, \lambda')}{\sum_{t=1}^{T-1} P(S_t = s_i/V = O, \lambda')}}$$

$$\boldsymbol{b_i(j) = \frac{\sum_{t=1}^{T} P(S_t = s_i/V = O, \lambda')\delta(o_t = j)}{\sum_{t=1}^{T} P(S_t = s_i/V = O, \lambda')}}$$

Les formules de ré-estimation obtenues ci-dessus peuvent s'interpréter de la façon suivante

$$\boldsymbol{\pi_i = \text{probabilité d'etre dans l'état } s_i \text{ à l'instant } t = 1}$$

$$\boldsymbol{a_{i,j} = \frac{\text{nombre de transitions de l'état } s_i \text{ à l'instant t=1}}{\text{nombre de fois où l'on quitte l'état } s_i}}$$

$$\boldsymbol{b_j(k) = \frac{\text{nombre d'apparition simultanées de l'état } s_j \text{ et du symbole } v_k}{\text{nombre d'apparitions de l'état } s_j}}$$

On peut alors remarquer que le principe reste similaire à celui de l'apprentissage étiqueté du paragraphe IV.4.1, à la différence que l'étiquetage s'effectue en probabilité avant ré-estimation.

L'algorithme de Baum-Welch [23] est donné par l'algorithme 9. Sa complexité est $\boldsymbol{O(N^2T + NMT)}$.

D'une manière naïve, les probabilités utilisées pour la ré-estimation des matrices peuvent être obtenues par les algorithmes Forward et Backward. Cependant, toujours pour des problèmes d'implémentation numériques, on utilise plutôt leurs versions utilisant les algorithmes avec ré-échelonnement.

Choisir un model initial λ_0

$$t = 0$$

Répéter

$$t \leftarrow t + 1$$

Calculer les variables Forward et Backward pour le model λ_{t-1}

Calculer Π $de\ \lambda_t$

Calculer $A\ de\ \lambda_t$

Calculer $B\ de\ \lambda_t$

Tant que $\left(P(V = O/\lambda_t) > P(V = O/\lambda_{t-1})\right)\ et\ t < t_{max}$

Algorithme IV.9 : Algorithme de Baum-Welch

IV.4.2.3. Descente de gradient

La deuxième méthode permettant d'optimiser la vraisemblance $P(V = O/\lambda)$ consiste à utiliser la descente de gradient. L'utilisation de la descente de gradient avec des MMC pose un problème de taille, les contraintes de stochasticité doivent être respectées par les paramètres du modèle.

A. Changement de l'espace de représentation

Une solution simple consiste à re-paramétrer les MMC avec des variables prenant leurs valeurs dans l'espace réel à l'aide des équations suivantes :

$$\forall\ 1 \leq i, j \leq N, \quad a_{i,j} = \frac{\exp(x_{i,j})}{\sum_{k=1}^{N} \exp(x_{i,k})}$$

$$\forall\ 1 \leq i \leq N, 1 \leq j \leq M,\ b_i(j) = \frac{\exp(y_{i,j})}{\sum_{k=1}^{M} \exp(y_{i,k})}$$

$$\forall\ 1 \leq i \leq N, \quad \pi_i = \frac{\exp(Z_i)}{\sum_{k=1}^{N} \exp(Z_k)}$$

Si l'on suppose que les coefficients des matrices stochastiques sont strictement positifs, alors il existe au moins une solution à ces équations. Un MMC λ est alors parfaitement défini par les trois matrices stochastiques $\lambda = (A, B, \Pi)$ ou les trois matrices réelles $\lambda = (X, Y, Z)$.

Dans le cas où tous les coefficients ne sont pas strictement positifs, il est toujours possible de fixer le coefficient nul à une valeur très petite, mais non nulle, de manière à ne pas trop déformer le modèle.

Il est intéressant de noter que les paramètres $\boldsymbol{x_{i,j}}, \boldsymbol{y_{i,j}}$ et $\boldsymbol{z_i}$ ne sont pas uniques. En effet, l'ajout d'une constante commune à chaque bloc de variables stochastique donne des valeurs vérifiant également les équations. Pour passer des coefficients $\boldsymbol{a_{i,j}}, \boldsymbol{b_i(j)}$ et $\boldsymbol{\pi_i}$ aux coefficients $\boldsymbol{x_{i,j}}, \boldsymbol{y_{i,j}}$ et $\boldsymbol{z_i}$, il suffit alors d'utiliser les formules suivantes :

$$\boldsymbol{x_{i,j} = \ln a_{i,j}}$$

$$\boldsymbol{y_{i,j} = \ln b_i(j)}$$

$$\boldsymbol{z_i = \ln \pi_i}$$

L'utilisation de ce paramétrage pour le calcul des dérivées partielles pose cependant problème lorsque l'une des probabilités du modèle est nulle. Une solution couramment utilisée est d'imposer que les coefficients soient non nuls. Une autre solution consiste à définir l'opérateur $\boldsymbol{LN}$ en imposant une valeur $\boldsymbol{\epsilon > 0}$ proche de zéro et une valeur $\boldsymbol{HV}$ négative et grande en valeur absolue, telle que

$$\boldsymbol{LN(x)} = \begin{cases} \boldsymbol{HV} & \textbf{si } \boldsymbol{x \leq \epsilon} \\ \boldsymbol{\ln(x)} & \textbf{sinon} \end{cases}$$

En remplaçant l'opérateur **ln** dans les équations du paragraphe A, il est possible de ne pas imposer de contraintes de stricte positivité aux coefficients du modèle. Cependant, il faut avoir à l'esprit que cette transformation n'est pas réversible.

B. Calcul du gradient

Soit $\boldsymbol{L(\lambda) = \ln P(V = O/\lambda)}$. Maximiser la vraisemblance $\boldsymbol{P(V = O/\lambda)}$ est équivalent à maximiser le logarithme de la vraisemblance $\boldsymbol{L(\lambda)}$. Calculons le gradient de $\boldsymbol{L(\lambda)}$ par rapport aux paramètres de $\boldsymbol{\lambda}$ au point $\boldsymbol{u}$.

$$\boldsymbol{\frac{\partial L(\lambda)}{\partial \lambda}(u) = \frac{1}{P(V=O/u)} \frac{\partial P(V=O/\lambda)}{\partial \lambda}(u)}$$

Maximiser $\boldsymbol{L(\lambda)}$ nécessite donc de calculer les dérivées partielles de $\boldsymbol{P(V = O/\lambda)}$ par rapport aux différents paramètres du modèle.

$$\boldsymbol{\frac{\partial P(V=O\lambda)}{\partial x_{i,j}} = \sum_{t=2}^{T} \alpha_{t-1}(i) a_{i,j} b_j(o_t) \beta_t(j) - a_{i,j} \sum_{t=2}^{T} \alpha_{t-1}(i) \beta_{t-1}(i)}$$

$$\frac{\partial P(V=O/\lambda)}{\partial y_{i,j}} = \sum_{t=1}^{T} \beta_t(i)\alpha_t(i)(k(o_t = j) - b_i(j))$$

$$\frac{\partial P(V=O/\lambda)}{\partial z_i} = \pi_i b_i(o_1)\beta_1(i) - \pi_i P(V = O/\lambda)$$

$$\frac{1}{P(V=O/\lambda)} \frac{\partial P(V=O/\lambda}{\partial x_{i,j}} = \sum_{t=2}^{T} \alpha_{t-1}(i) a_{i,j} b_j(o_t)\tilde{\beta}_t(j) - a_{i,j} \sum_{t=2}^{T} \frac{\tilde{\alpha}_{t-1}(i)\tilde{\beta}_{t-1}(i)}{c_{t-1}}$$

$$\frac{1}{P(V=O/\lambda)} \frac{P(V=O/\lambda)}{\partial y_{i,j}} = \sum_{t=1}^{T} \frac{\tilde{\beta}_t(i)\tilde{\alpha}_t(i)(k(o_t=j)-b_i(j))}{c_t}$$

$$\frac{1}{P(V=O/\lambda)} \frac{P(V=O/\lambda)}{\partial z_i} = \pi_i b_i(o_1)\tilde{\beta}_1(i) - \pi_i$$

A partir de ce gradient, il est possible d'utiliser n'importe quelle descente de gradient telles que celles décrites dans [**41**]*

Cependant, il faut garder à l'esprit que ce calcul est couteux en temps machine. Sa complexité est $O(N^2T + NMT)$.

On notera que, l'algorithme de Baum-Welch ou la descente de gradient, les deux méthodes nécessitent un modèle initial à améliorer. Ces approches simples possèdent un gros inconvénient : elles sont sensibles au point de départ et elles convergent vers des optima locaux de la vraisemblance. Il existe de nombreuses variantes de l'algorithme ***EM*** construites pour pallier à certains de ces inconvénients.

- **Stochastic Expectation Maximisation**

L'algorithme **SEM** (Stochastic Expectation Maximisation) [42] peut également être utilisé pour effectuer l'apprentissage de MMC. L'algorithme SEM est une variante stochastique de l'algorithme EM, beaucoup moins sensible au point de départ. À partir d'un modèle initial λ_0, il consiste à engendrer en probabilité, selon la loi de λ_0, une séquence d'états cachés ayant engendré la séquence d'observations. À partir de cette séquence d'états cachés, un apprentissage étiqueté est réaliser afin d'obtenir un nouveau modèle λ_1. La procédure est réitérée plusieurs fois.

Cette méthode possède deux avantages importantes par rapport à l'algorithme EM : la convergence est rapide et l'algorithme SEM est peu sensible au modèle initial. Cependant, la méthode possède également deux gros désavantages : elle est moins efficace que l'algorithme

EM (Baum-Welch) en présence de séquences d'observations trop courtes et la suite des modèles obtenus ne converge pas ponctuellement « on n'a pas $\boldsymbol{P(V = O/\lambda_n) \leq P(V = O/\lambda_{n+1})}$ mais uniquement globalement ».

Le lecteur pourra remarquer que cet algorithme est proche de l'algorithme de **segmental k-means** de la section 9: seul le mode de génération de la séquence d'états cachés change.

- **Estimation Conditionnelle Itérative : ICE**

L'algorithme d'Estimation Conditionnelle Itérative (Iterative Conditional Estimation-**ICE**) est une méthode d'optimisation en présence de données cachées proposée dans [47]. Le principe de ICE consiste à utiliser un estimateur des paramètres du modèle calculé à partir des informations complétées, c'est-à-dire à partir de la séquence d'observations et d'une séquence d'états cachés. Il a été montré que la meilleure approximation du modèle au sens de l'erreur quadratique moyenne est l'espérance conditionnelle. Ainsi, dans des cas particulier des MMC, l'algorithme ***ICE*** permet d'aboutir aux mêmes formules de ré-estimation que l'algorithme de Baum-Welch

- **Les autres variantes**

D'autres variantes de l'algorithme EM sont disponibles dans la littérature. Il est possible de citer l'algorithme de **SAEM** [43], qui est un intermédiaire entre **EM** et **SEM**, ou **MCEM** [44], qui utilise de manière intense la génération de séquences d'états et les méthodes de Monte-Carlo.

IV.5. Critère du maximum a posteriori (MAP)

Le critère de maximum a posteriori **MAP** trouve son intérêt dans la théorie de la décision bayésienne. Jusqu'à maintenant, nous avons considéré des critères d'optimisation des modèles utilisant la règle de décision suivante :

Si $\boldsymbol{P(V = O/\lambda_1} > P(V = \boldsymbol{O/\lambda_2})$ alors $\boldsymbol{O}$ a été le plus probablement émise par le modèle $\boldsymbol{\lambda_1}$.

Bien que la notion de « plus probablement émise » soit couramment utilisée afin de dire « appartient », c'est-à-dire, pour notre exemple, que la séquence d'observation $\boldsymbol{O}$ appartient à la classe modélisée par $\boldsymbol{\lambda_1}$, rien ne garantit que ce choix soit optimal.

Un moyen de garantir un choix optimal, au moins en théorie, est d'utiliser la théorie de la décision bayésienne [45]. Le critère de décision utilisé est alors

$\boldsymbol{P(\lambda_1 / V = O)} > P(\boldsymbol{\lambda_2 / V = O})$, alors $\boldsymbol{O}$ appartient à la classe $\boldsymbol{\lambda_1}$

Ce critère pose problème, car nous ne savons pas comment exprimer ces probabilités. Cependant, en transformant ces probabilités, on obtient :

$$\boldsymbol{P(\lambda / V = O)} = \frac{\boldsymbol{P(V=O/\lambda)P(\lambda)}}{\boldsymbol{P(V=O)}}$$

Où $\boldsymbol{P(\lambda/V = O)}$ est la probabilité a posteriori du modèle $\boldsymbol{\lambda}$ connaissant la séquence d'observations $\boldsymbol{O}$, $\boldsymbol{P(\lambda)}$ est la probabilité a priori, d'apparition du modèle $\boldsymbol{\lambda}$ et $\boldsymbol{P(V = O)}$ est la probabilité a priori d'apparition de la séquence d'observations $\boldsymbol{O}$.

Le critère *MAP* possède un avantage certain sur le critère de maximum de vraisemblance :

Les probabilités a priori permettent de modéliser le déséquilibre éventuel dans l'apparition des séquences d'observations.

La première remarque que l'on peut faire est que les probabilités $\boldsymbol{P(V = O)}$ peuvent être ignorées car, dans la règle de décision bayésienne, elles peuvent être simplifiées.

L'apprentissage des modèles avec le critère MAP dépend très fortement des objectifs visés.

Lorsque les modèles sont appris séparément l'objectif est alors de maximiser la probabilité $\boldsymbol{P(\lambda / V = O)}$, c'est-à-dire, après simplification, maximiser $\boldsymbol{P(V = O / \lambda)P(\lambda)}$. Si la probabilité $\boldsymbol{P(\lambda)}$ s'exprime indépendamment des valeurs prises par les matrices stochastiques qui le définissent, alors les deux probabilités peuvent être apprises séparément. Pour la probabilité $\boldsymbol{P(V = O/\lambda)}$, il suffit d'utiliser le critère de maximum de vraisemblance et pour la probabilité $\boldsymbol{P(\lambda)}$ on utilise généralement une estimation statistique de l'apparition de ce modèle.

Dans le cas où l'expression de $\boldsymbol{P(\lambda)}$ dépend des valeurs prises par les paramètres du modèle, il n'est pas possible d'utiliser le critère de maximum de vraisemblance.

Une solution consiste alors à utiliser une descente de gradient afin de maximiser le critère $\boldsymbol{\ln P(V = O/\lambda) + \ln P(\lambda)}$ à condition que $\boldsymbol{P(\lambda)}$ soit différentiable. Lorsque le critère devient

plus complexe, ou lorsqu'il utilise plusieurs modèles ou séquences d'observations en simultané, la même démarche peut être utilisée : s'il est possible d'optimiser séparément les deux types de probabilités, il faut les traiter séparément. Dans le cas où cela n'est pas possible, le moyen le plus courant d'effectuer l'apprentissage consiste à utiliser la descente de gradient. Pour certains critères, il n'est pas possible d'éliminer les probabilités $\boldsymbol{P(V = O)}$, il est alors nécessaire de les modéliser et de les inclure dans la descente de gradient.

IV.6. Maximisation de l'information mutuelle

L'un des buts principaux de l'apprentissage de MMC est d'effectuer une classification. En effet, on cherche souvent, à partir d'une observation $\boldsymbol{O}$, à décider de manière automatique à quelle autre observation elle ressemble le plus et surtout à décider à quelle classe de séquences d'observations elle appartient réellement.

- **Exemple illustratif** On considère un système d'identification biométrique basé sur la photographie du visage. Initialement, le système possède au moins une photographie de chaque personne à reconnaitre. Chaque photographie est modélisée par un MMC après qu'elle ait été transformée par un procédé quelconque en séquence d'observations. Si une personne se présente devant la caméra, le système va prendre une photographie, la transformer en séquence et comparer les différentes vraisemblances avec les MMC appris. Le MMC qui permet d'obtenir la meilleure vraisemblance permet alors de dire que la personne est celle qui correspond à la photographie du MMC. En théorie, cette méthode fonctionne mais, en pratique, ce n'est pas toujours le cas. Si l'ensemble des photographies concerne des photographies de visages de personnes de même couleur de peau et de même couleur de cheveux, alors il ya de grande chances pour que les modèles reconnaissent bien l'ensemble des visages, car la modélisation des visages sera quasi identique. Une solution consiste à effectuer l'apprentissage des MMC avec un autre critère que la vraisemblance. Le critère de prédilection pour cette tache est la maximisation de l'information mutuelle MIM, Plusieurs variantes de la maximisation de l'information mutuelle existent : elles sont présentées ci-dessous.

IV.6.1. Maximisation de l'information mutuelle de la vraisemblance

La première forme du critère de **MIM** s'attache à différentier les modèles par leurs vraisemblances. A cet effet, on cherche à maximiser la vraisemblance de la séquence

d'observations à apprendre $\boldsymbol{O}$ mais également à minimiser la vraisemblance des séquences d'observations à ne pas reconnaitre $\boldsymbol{N^1, \dots, N^K}$. L'avantage de ce critère est qu'il laisse le MMC modéliser ce qui est caractéristique, tout en acceptant de moins bien modéliser ce qui ne l'est pas.

Ce critère peut prendre plusieurs formes. La forme présentée ci-après est celle décrite dans [28]. Cette forme est intéressante, car elle permet de gérer facilement les problèmes de précision numérique.

$$\boldsymbol{mim(\lambda)} = \frac{\boldsymbol{P(V=O/\lambda)}}{\prod_{k=1}^{K} \boldsymbol{P(V=N^k/\lambda)}}$$

Comme nous pouvons le voir, maximiser cette expression entraine la maximisation de la vraisemblance $\boldsymbol{P(V = O/\lambda)}$ et la minimisation des vraisemblances $\boldsymbol{P(V = N^k/\lambda)}$.

On remarque alors que maximiser $\boldsymbol{mim(\lambda)}$ est équivalent à maximiser son logarithme népérien. On définit alors

$$\boldsymbol{L(\lambda) = \ln\ mim(\lambda) = \ln P(V = O\ /\ \lambda) - \sum_{k=1}^{K} \ln P(V = N^k\ /\ \lambda)}$$

Pour optimiser ce critère, il est alors possible d'utiliser une descente de gradient, Il est donc nécessaire de calculer le gradient de $\boldsymbol{L(\lambda)}$. Ce dernier est donné par l'équation suivante :

$$\frac{\boldsymbol{\partial L(\lambda)}}{\boldsymbol{\partial \lambda}} = \frac{\boldsymbol{1}}{\boldsymbol{P(V=O/\lambda)}} \frac{\boldsymbol{\partial P(V=O/\lambda)}}{\boldsymbol{\partial/\lambda}} - \sum_{k=1}^{K} \frac{\boldsymbol{1}}{\boldsymbol{P(V=N^k\lambda}} \frac{\boldsymbol{\partial P(V=N^k\lambda)}}{\boldsymbol{\partial \lambda}}$$

Or ce gradient n'est autre qu'une combinaison linéaire des gradients calculés à la section 6. Pour cela, on note $\boldsymbol{\tilde{\alpha}_t(i)}$, $\boldsymbol{\tilde{\beta}_t(i)}$, $\boldsymbol{c_t}$ les variables Forward et Backward avec ré-échelonnement et les coefficients de ré-échelonnement calculés pour la séquence d'observation $\boldsymbol{O}$. On note $\boldsymbol{\tilde{\alpha}_t^k(i)}$, $\boldsymbol{\tilde{\beta}_t^k(i)}$, et $\boldsymbol{c_t^k}$ les variables Forward et Backward avec ré-échelonnement $\boldsymbol{N^k}$. Alors, en reprenant les équations de la section précédente, on obtient

$$\frac{\boldsymbol{\partial L(\lambda)}}{\boldsymbol{\partial x_{i,j}}} = \sum_{t=2}^{T} \boldsymbol{\tilde{\alpha}_{t-1} a_{i,j} b_j(o_t) \tilde{\beta}_t(j)} - \boldsymbol{a_{i,j}} \sum_{t=2}^{T} \frac{\boldsymbol{\tilde{\alpha}_{t-1}(i) \tilde{\beta}_{t-1}(i)}}{\boldsymbol{c_{t-1}}}$$

$$- \sum_{k=1}^{K} \left(\sum_{t=2}^{T} \boldsymbol{\tilde{\alpha}_{t-1}^k a_{i,j} b_j\left(n_t^k\right) \tilde{\beta}_t^k(j)} - \boldsymbol{a_{i,j}} \sum_{t=2}^{T} \frac{\boldsymbol{\tilde{\alpha}_{t-1}^k(i) \tilde{\beta}_{t-1}^k(i)}}{\boldsymbol{c_{t-1}^k}} \right)$$

$$\frac{\boldsymbol{\partial L(\lambda)}}{\boldsymbol{\partial y_{i,j}}} = \sum_{t=1}^{T} \frac{\boldsymbol{\tilde{\beta}_t(i) \tilde{\alpha}_t(i) (k(o_t=j) - b_i(j))}}{\boldsymbol{c_t}}$$

$$-\sum_{k=1}^{K}\left(\sum_{t=1}^{T}\frac{\tilde{\beta}_t^k(i)\tilde{\alpha}_t^k(i)\left(k\left(n_t^k=j\right)-b_i(j)\right)}{c_t^k}\right.$$

Et

$$\frac{\partial L(\lambda)}{\partial z_i}=\pi_i b_i(o_1)\tilde{\beta}_1(i)-\pi_i$$

$$-\sum_{k=1}^{K}\left(\pi_i b_i\left(n_1^k\right)\tilde{\beta}_1^k(i)-\pi_i\right)$$

Il suffit alors d'utiliser la technique de la descente de gradient.

IV.6.2. Maximisation de l'information mutuelle du MAP

Le critère de maximisation de l'information mutuelle pour le critère de décision ***MAP*** décrit dans [47] conduit à la minimisation du critère $I(\lambda, O)$ avec $\mathbb{L}$ l'ensemble des L MMC $\{\lambda_1, \ldots, \lambda_L\}$ représentant les L classes et $\mathbb{O}$ l'ensemble des séquences d'observations à apprendre. On définit $c(O_i) \in [1..L]$ le numéro de la classe associée à la séquence d'observations O_i. Le critère est :

$$I(\mathbb{L}, \mathbb{O}) = H(\mathbb{L}) - I(\mathbb{L} \parallel \mathbb{O})$$

Avec

$$H(\mathbb{L}) = -\sum_{\lambda_i \in \lambda} P(\lambda_i) \ln P(\lambda_i)$$

Et

$$I(\mathbb{L} \parallel \mathbb{O}) = \sum_{\lambda_i \in \lambda} \sum_{O_j \in O} P(V = O_j, \lambda_i) \ln \frac{P(V=O_j, \lambda_i)}{P(\lambda_i)P(V=O_j)}$$

Si l'on considère que les probabilités $P(V = O_i)$ sont constantes et que les probabilités $P(\lambda_i / V = O_j)$ sont nulles, sauf quand $i = j$, alors minimiser $I(\mathbb{L}, \mathbb{O})$ est équivalent à maximiser [47] :

$$\tilde{I}(\mathbb{L} \parallel \mathbb{O}) = \sum_{\lambda_i \in \lambda} \sum_{O_j \in C_i} \frac{P(V=O_j/\lambda_j)P(\lambda_i)}{\sum_{\lambda_i \in \lambda} P(V=O_j/\lambda_i)P(\lambda_i)}$$

Avec $C_i = \{O_j \in O / c(O_j) = i\}$.

La maximisation de $\tilde{I}(\mathbb{L} \parallel \mathbb{O})$ peut alors être réalisée grâce à une descente de gradient ou grâce à l'algorithme de Baum-Welch.

IV.7. Le critère de segmental k-means

Parmi l'ensemble des critères utilisés pour l'apprentissage de MMC, le critère de segmental k-means se détache des autres. En effet, pour ce critère, on cherche à optimiser la probabilité $\boldsymbol{P(V=O,S=Q^*/\lambda)}$ avec $\boldsymbol{Q^*}$ la séquence d'états cachés qui a le plus probablement engendré la séquence telle que calculée par l'algorithme de Viterbi. Une des grandes difficultés de ce critère est qu'il n'est ni dérivable, ni même continu. Par conséquent, les méthodes s'appuyant sur l'algorithme EM ou les descentes de gradient ne sont pas utilisables. Cependant, il existe quand même un moyen d'ajuster les paramètres d'un modèle de manière à maximiser cette probabilité. Cet algorithme appelé segmental k-means repose sur deux algorithmes décrits précédemment : l'algorithme de viterbi et l'apprentissage étiqueté.

Son principe est simple :

À partir d'un modèle initial et de la séquence d'observations $\boldsymbol{O}$, on recherche la séquence d'états cachés qui a le plus probablement été suivie pour générer $\boldsymbol{O}$ à l'aide de l'algorithme de Viterbi. Cette recherche permet d'étiqueter la séquence $\boldsymbol{O}$ et par conséquent de la segmenter ;

Une fois étiquetée, la séquence $\boldsymbol{O}$ est alors apprise par comptage des transitions effectives entre les états et les émissions de symboles. Cette étape peut alors être considérée comme un k-means consistant à ré-estimer les « centres des classes » ;

- Le nouveau modèle est alors utilisé comme modèle initial et les deux opérations précédentes sont répétées tant que nécessaire.

Il a pu être montré [46] que l'algorithme de segmental k-means « algorithme **III.10** » permettait d'augmenter la probabilité $\boldsymbol{P(V=O,S=Q^*/\lambda)}$ de manière itérative et qu'il convergeait vers un maximum local du critère considéré. Lorsque l'on utilise ce critère, il faut faire attention à sa formulation. L'augmentation itérative de la probabilité consiste à trouver un modèle $\boldsymbol{\lambda_{k+1}}$ à partir d'un modèle $\boldsymbol{\lambda_k}$ tel que

$$\boldsymbol{P(V=O,S=Q_k^*/\lambda_k)} < P(V=O,S=\boldsymbol{Q_{k+1}^*/\lambda_{k+1}})$$

Et non pas tel que $\boldsymbol{P(V=O,S=Q_k^*/\lambda)} < P(V=O,S=\boldsymbol{Q_k^*}\,/\,\boldsymbol{\lambda_{k+1}})$ avec $\boldsymbol{Q_k^*}$ la séquence de Viterbi obtenue avec le modèle λ_k et Q_{k+1}^* la séquence de Viterbi obtenue avec le modèle $\boldsymbol{\lambda_{k+1}}$. En effet, la séquence de Viterbi change lorsque le modèle est modifié.

Choisir un MMC initial $\boldsymbol{\lambda_0}$
$\boldsymbol{t = 0}$
Répéter
$\boldsymbol{t = t + 1}$
$\boldsymbol{Q^* = Viterbi(O, \lambda_{t-1})}$
Estimer $\boldsymbol{\lambda_t}$ à partir de $\boldsymbol{O}$ et $\boldsymbol{Q^*}$
Tant que $\boldsymbol{P(V = O, S = Q_t^* / \lambda_t) > P(V = O, S = Q_{t-1}^* / \lambda_{t-1}}$

Algorithme IV.10 : Algorithme de segmental k-means

L'algorithme de segmental k-means peut également être utilisé avec plusieurs séquences d'observations. Pour cela, il suffit de considérer le critère d'apprentissage

$$\prod_{k=1}^{K} \boldsymbol{P(V = Q^k, S = Q^{k*} / \lambda)}$$

L'algorithme consiste alors à appliquer l'algorithme de Viterbi à chacune des séquences d'observations et à utiliser l'apprentissage étiqueté de toutes ces séquences simultanément, comme décrit précédemment.

Cet algorithme est parfois utilisé en raison de sa rapidité en lieu et place de l'algorithme de Baum-Welch, en considérant l'hypothèse suivante : les probabilités complétées $\boldsymbol{P(V = O, S = Q/\lambda)}$ sont nulles ou négligeables pour toutes les séquences d'états, à l'exception de celle de la séquence $\boldsymbol{Q^*}$ de Viterbi associée. Par conséquent, maximiser, $\boldsymbol{P(V = O/\lambda)}$ est équivalent à maximiser $\boldsymbol{P(V = O, S = Q^*/\lambda)}$. Bien qu'il soit possible de trouver des modèles pathologiques contredisant cette hypothèse, dans la pratique, l'hypothèse est souvent confirmée.

IV.8. Minimisation du taux d'erreur de classification

Ce critère a pour objectif de minimiser le taux d'erreur de classification avec une décision soumise au critère MAP. Pour le décrire brièvement, on considère un ensemble $\mathcal{E} = \{(\boldsymbol{O_i}, \boldsymbol{c(O_i)})\}$ de séquences d'observations ainsi que le numéro de la classe qui leur sont associés. Soit $\{\boldsymbol{1}, \dots, \boldsymbol{L}\}$ les L classes à apprendre et $\{\boldsymbol{\lambda_1}, \dots, \boldsymbol{\lambda_L}\}$ les MMC associés.

IV.8.1. Première approche

Dans cette première approche, on a abouti au critère suivant, après plusieurs transformations et approximations :

$$\boldsymbol{\tau_e = 1 - \frac{1}{N}\sum_{m=1}^{L}\sum_{O_i\in,c(O_i)=m}\frac{P(V=O_i/\lambda_m)P(\lambda_m)}{\sum_{l=1}^{L}P(V=O_i/\lambda_l)P(\lambda_l)}}$$

Si l'on suppose que les classes apparaissent avec la même probabilité $\boldsymbol{i.e.P(\lambda_i) = \frac{1}{L}}$ pour tout $\boldsymbol{i = 1 \ldots L}$, alors minimiser τ_e revient à maximiser $\tilde{\tau}_e$ avec

$$\boldsymbol{\tilde{\tau}_e = \frac{1}{N}\sum_{m=1}^{L}\sum_{O_i\in\mathcal{E},c(O_i)=m}\frac{P(V=O_i/\lambda_m)}{\sum_{l=1}^{L}P(V=O_i/\lambda_l)}}$$

Il suffit alors d'utiliser une descente de gradient sur l'ensemble des paramètres $\{\boldsymbol{\lambda_1, \ldots, \lambda_L}\}$ pour maximiser $\boldsymbol{\tilde{\tau}_e}$ et donc minimiser $\boldsymbol{\tau_e}$.

Pour que l'approximation effectuée soit valable, il est nécessaire que les modèles initiaux utilisés par la descente de gradient aient été obtenus par la maximisation de $\boldsymbol{\prod_{O_i\in\mathcal{E}} P(V = O_i/\lambda_c(O_i))}$ par un des algorithmes de maximisation de la vraisemblance décrit précédemment.

Bien que ce critère semble intéressant, il n'obtient pas toujours de bons résultats. En effet, minimiser $\boldsymbol{\tau_e}$ ne garantit aucunement que ce taux sera faible sur un ensemble d'observations autre que celui utilisé pour l'apprentissage.

IV.8.2. Deuxième approche

D'autres définissent le critère de minimisation des erreurs de classification (minimum classification error **MCE)** sous la forme

$$\boldsymbol{J = \frac{1}{|E|}\sum_{O_i\in O}\Delta\left(max_{\lambda_j\in\lambda_J\neq c(O_j)}\ln\frac{P(V=O_i/\lambda_j}{P(V=O_i/\lambda_{c(O_i)}}\right)}$$

Avec $\boldsymbol{\Delta(z) = \begin{cases} 1 & si\ z \geq 1 \\ 0 & sinon \end{cases}}$

Ce critère n'est pas dérivable, ni même continu. Pour l'approcher sous forme continue, il suffit d'utiliser une sigmoïde $\boldsymbol{[1 + e^{-z}]^{-1}}$ à la place de $\boldsymbol{\Delta(z)}$ et l'opérateur **softmax** $\boldsymbol{\ln\sum_k e^{zk}}$ à la place de $\boldsymbol{max_k z_k}$. Le critère $\boldsymbol{J}$ peut alors être approché [47] par :

$$\boldsymbol{\tilde{J} = \frac{1}{|E|}\sum_{O_i\in O}\frac{1}{1+\exp\left(\ln P(V=O_i/\lambda_{c(O_i)})-\ln\sum_{\lambda_j\in\lambda_J\neq c(O_i)}P(V=O_i/\lambda_j)\right)}}$$

Le critère $\boldsymbol{\tilde{J}}$ peut alors etre minimisé à l'aide d'une descente de gradient.

Dans la litératures, On trouve d'autre formes de critères de minimisation des erreurs de classification que Les deux forme de présentées ci-dessus.

IV.9. Remarques générales sur les critères d'apprentissage

Comme nous venons de le voir, de nombreux critères peuvent être considérés pour l'apprentissage de modèles de Markov cachés. Les critères que nous avons décrits dans ce chapitre ne sont pas les seuls envisageables, mais ce sont les plus couramment utilisés. De plus, la démonstration des algorithmes d'apprentissage fournit la majorité des outils nécessaires à la conception des algorithmes d'apprentissage pour tous les critères envisageables.

Tous les algorithmes d'apprentissage de ce chapitre n'ont pas la même complexité. Pour faciliter le choix à la fois du critère et de l'algorithme de résolution, nous avons construit le tableau suivant.

Critère	Algorithme	Complexité d'une itération
$P(V=O,S=Q/\lambda)$	apprentissage étiqueté	N^2+NM+T
$\prod_{k=1}^{K} P(V=O_k,S=Q_k/\lambda)$	apprentissage étiqueté	$N^2+NM+\sum_{k=1}^{K} T_k$
$P(V=O/\lambda)$	Baum-Welch	N^2T+NMT
$\prod_{k=1}^{K} P(V=O_k/\lambda)$	Baum-Welch	$N^2\sum_{k=1}^{K} T_k+NM\sum_{k=1}^{K} T_k$
$P(V=O/\lambda)$	Gradient	N^2T+NMT
$\prod_{k=1}^{K} P(V=O_k/\lambda)$	Gradient	$N^2\sum_{k=1}^{K} T_k+NM\sum_{k=1}^{K} T_k$
$\frac{P(V=O/\lambda)}{\prod_{k=1}^{K} P(V=O_k/\lambda)}$	Information mutuelle	$N^2(T+\sum_{k=1}^{K} T_k)$ $+NM(T+\sum_{k=1}^{K} T_k)$
$P(\lambda/V=O)$	MAP simple	N^2T+NMT
$P(\lambda/V=O)$	MAP complexe	Potentiellement très élevée
$P(V=O,S=Q^*/\lambda)$	*segmental k-means*	N^2T+NM
$\prod_{k=1}^{K} P(V=O_k,S=Q_k^*/\lambda)$	*segmental k-means*	$N^2\sum_{k=1}^{K} T_k+NM$

TAB. IV.1 complexité associée aux algorithmes en fonction des critères optimisés

$\{\boldsymbol{O}, \boldsymbol{O_1}, \dots, \boldsymbol{O_K}\}$ est l'ensemble des séquences d'observations de longueurs $\{\boldsymbol{T}, \boldsymbol{T_1}, \dots, \boldsymbol{T_k}\}$.

$\boldsymbol{N}$ est le nombre d'états cachés du ***MMC***. $\boldsymbol{M}$ est le nombre de symbole du ***MMC***.

Il est intéressant de remarquer que l'algorithme de segmental k-means peut être beaucoup plus rapide que l'algorithme de Baum-Welch. En effet, il est très courant que le nombre de symbole $\boldsymbol{M}$ soit beaucoup plus grand que le nombre des états cachés. Dans ces conditions, lorsque la longueur $\boldsymbol{T}$ de la séquence d'observations augmente, le terme dominant dans la complexité de l'algorithme de Baum-Welch est $\boldsymbol{NMT}$ tandis que pour l'algorithme de segmental k-means, ce terme dominant est $\boldsymbol{N^2T}$. Par conséquent, pour $\boldsymbol{M} > N$, l'algorithme de segmental k-means est plus rapide que l'algorithme de Baum-Welch lorsque la longueur de la séquence d'observations augmente.

L'algorithme de segmental k-means est donc parfois utilisé en lieu et place de l'algorithme de Baum-Welch, car plusieurs auteurs ont remarqué que la probabilité $\boldsymbol{P}(\boldsymbol{V} = \boldsymbol{O}, \boldsymbol{S} = \boldsymbol{Q}^* / \boldsymbol{\lambda})$ est élevée par rapport aux autres chemins d'états et qu'une grande majorité des chemins ont une probabilité très faible, voire nulle. Par conséquent, certains travaux émettent l'hypothèse que $\boldsymbol{P}(\boldsymbol{V} = \boldsymbol{O} \,/\, \boldsymbol{\lambda}) \approx \boldsymbol{P}(\boldsymbol{V} = \boldsymbol{O}, \boldsymbol{S} = \boldsymbol{Q}^* \,/\, \boldsymbol{\lambda})$.

Un autre point important est que la complexité du calcul du gradient est du même ordre que celle de l'algorithme de Baum-Welch. Cette propriété est intéressante car elle signifie, a priori, qu'il n'est pas forcément beaucoup plus couteux d'utiliser des critères tels que la maximisation de l'information mutuelle ou le critère MAP simple. En effet, on remarque que la complexité de l'optimisation de ces critères est linéaire en fonction de la longueur totale des séquences d'observations impliquées. Cependant, la descente de gradient peut nécessiter d'effectuer ce calcul plusieurs fois avant d'améliorer un modèle et par conséquent l'approche par descente de gradient est considérée comme étant relativement couteuse.

Conclusion

Nous avons présentés dans ce chapitre, les algorithmes pour la reconnaissance et d'apprentissage, et les critères de discrimination permettant, à partir d'un Modèle MC initial, de trouver un nouveau Modèle MC augmentant le critère sélectif pour reconnaissance de la parole.

Dans la plupart des systèmes de reconnaissances MMC actuelles, le but de l'entrainement acoustique est de trouver l'ensemble des paramètres acoustiques du MMC maximisant, sur l'ensemble des phrases d'entrainements, la vraisemblance des données étant donnée les modèles corrects associées (supposés connues pendant l'entrainement).

Le chapitre suivant, sera consacré à décrire notre mise en œuvre. La mise en pratique de notre étude sur la parole. Nous allons procéder à l'implémentation informatique des modèles de markov cachés pour la reconnaissance automatique de la parole (RAP). La première sous matlab et la second sous la plate forme HTK, l'une des plus utilisée actuellement dans le domaine de la RAP.

CHAPITRE V :

IMPLEMENTATION DE LA RECONNAISSANCE AUTOMATIQUE PAR MMC

Introduction

Nous allons, dans cette dernière partie, mettre en application le traitement automatique de la parole pour la reconnaissance à base de MMC. Pour cela, nous commencerons par décrire la structure générale dans un système de reconnaissance de la parole et les différents blocs intervenant dans cette opération. Par la suite, nous effectuons deux application, et les résultats obtenues. Ce qui distingue ces deux application est que la première est effectuée sous matlab et la seconde sous la plate forme HTK [annaxe A]. Dans le premier cas l'approche utilisée est purement acoustique alors que dans le second (système triphone) avec dépendance contextuelle.

V.1. Objectif du travail :

Dans ce travail nous avons développé deux systèmes, le premier sous Matlab et le second sous HTK et où l'objectif est la reconnaissance automatique de la parole qui s'effectuera sous la base de données parole TIdigit constituée d'un ensemble d'apprentissage et d'un ensemble de test. La base **TI**digit [49] pour Texas Instruments digits est parmi les premières bases de données de parole destinées à des applications de traitement de la parole. Conçu initialement à des fins d'évaluation des algorithmes de reconnaissance de la parole indépendante du locuteur, elle contient 77 séquences de digits connectés prononcés par 326 locuteurs dont 114 femmes, 111 hommes, 51 filles et 50 garçons.

L'objectif que nous nous sommes fixé consiste à reconnaitre les chiffre un à onze prononcé par les différents locuteurs de la base de données. Par ailleurs, nous effectuons une comparaison des taux de reconnaissance entre le système conçu sous matlab (approche acoustique) et le système conçu sous HTK avec dépendance contextuelle.

V.2. Structure générale d'un Reconnaissance Automatique de la parole continue

Un système de RAP continue est un système destiné à reconnaitre des phrase plus au moins longue avec ses hésitation et ses liaisons…etc. Etant donné la complexité du problème, le formalisme de reconnaissance de la parole nécessite une décomposition en plusieurs opérations élémentaire qui sont les suivantes:

- Un module de traitement du signal et d'analyse acoustique (feature extraction) transformant le signal de parole en une séquence de vecteurs acoustiques (détaillé au chap. II).
- Un générateur d'hypothèses locales qui affectera une étiquette ou des hypothèses locales correspondant à chaque segment élémentaire de parole (associé à un ou plusieurs vecteurs acoustique). Ce générateur d'hypothèses locales portera, généralement, sur des modèles d'unités élémentaires de parole (typiquement des mots ou des phonèmes). Cette opération nécessite un entrainement sur une grande quantité d'exemples (enregistrement de nombreuses phrases) contenant plusieurs fois les différentes unités de parole dans des contexte variés.
- Un module d'alignement temporel (pattern matching) transformant les hypothèses locales en un score global sur la phrase prononcée. Ceci pourra être réalisé par l'algorithme de Déformation Temporelle Dynamique(DTW) ,Modèles de Markov Cachés (MMC).
- Un module syntaxique interagissant avec le module d'alignement temporel et qui forcera le reconnaisseur à intégrer les contraintes syntaxiques et éventuellement sémantiques et pragmatiques.

Le schéma synoptique d'un tel système est représenté ci-dessous.

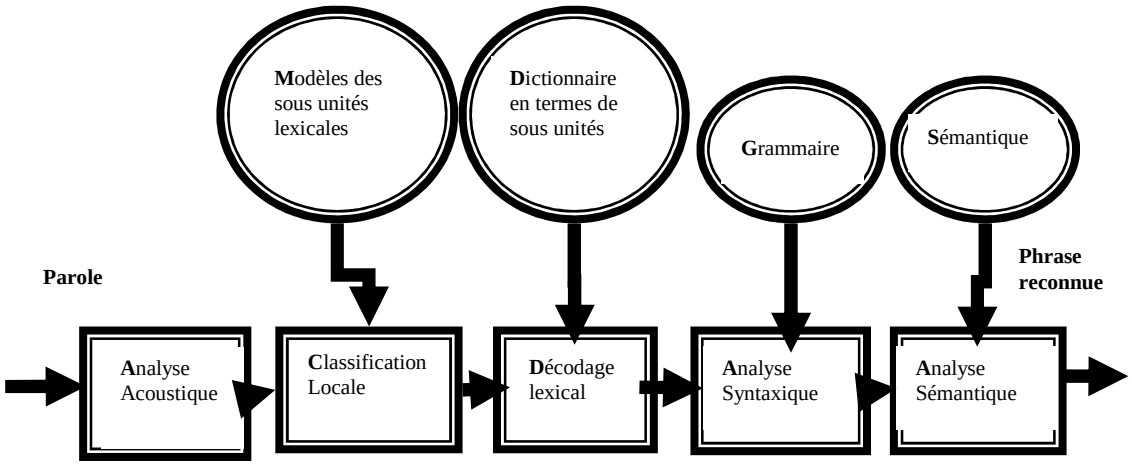


Fig. V.1-Schéma général d'un système de R.A.P.

V.3. Structure d'un Système de Reconnaissance Automatique de la parole continue par MMC

Le schéma précédant est une sructure générale qui ne tient pas compte de l'outil de traitement utilisé en l'occurrence les MMC. Dans le cas d'un système de reconnaissance par MMC, le schéma générale se présente comme suit :

- **Architecture générale**

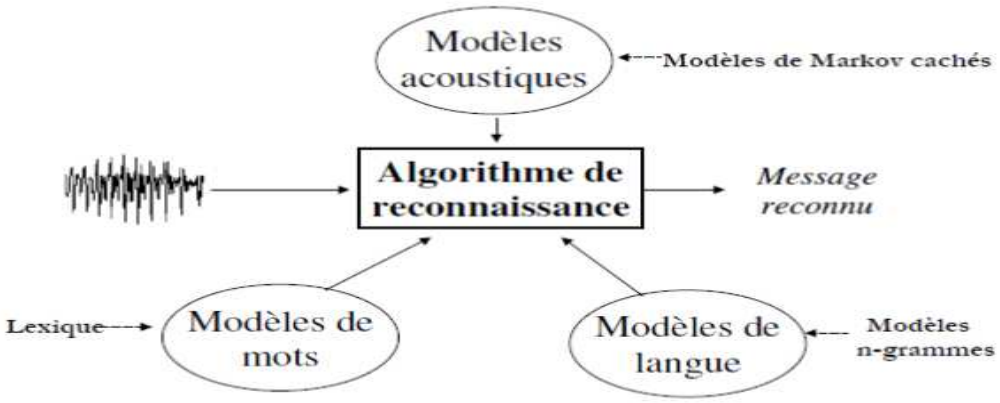


Fig. V.2-Schéma Générale de la R. A. P. par MMC.

Sous une vue plus détaillé, on retrouve les différents blocs qui se présente dans le schéma suivant.

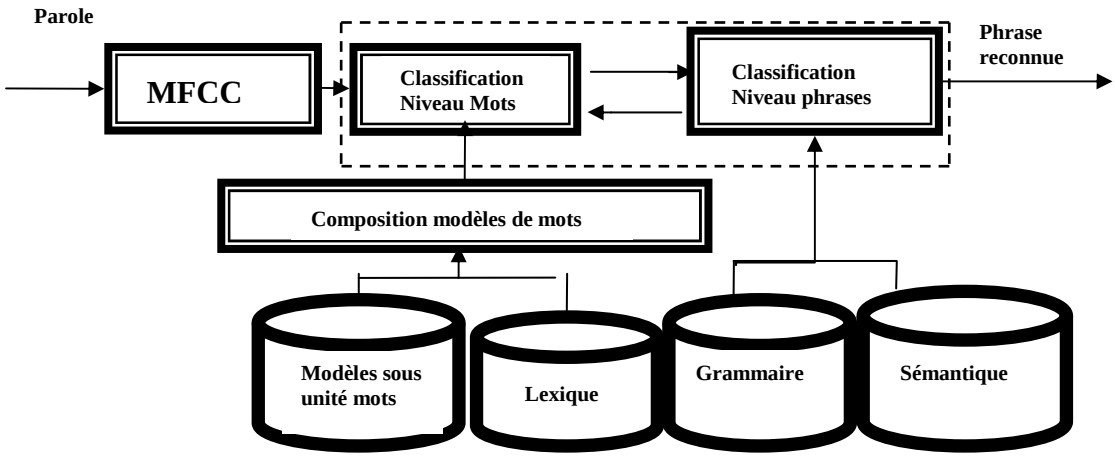


Fig. V.3-Schéma bloc détaillé de la R. A. P. par MMC .

Dans ce schéma, les blocs (grammaire, sémantique) font partie du modèle de language, par contre le cylindre (Modèles sous unité mots) fait référence au modèle acoustique, quand aux blocs de classification font partie des algorithme d'entrainement et de reconnaissance.

V.4. Première Application : Développement d'un Système de Reconnaissance de la parole par MMC sous Matlab.

V.4.1.Organigramme générale

Notre application est prévu pour la base TIdigit [49], qui est une base de données paroles constituée des chiffres un à onze en anglais et prononcée par plusieurs locuteurs, pour cela, nous avons prévu onze modèle($\lambda^1, \ldots \lambda^{11}$) un pour chaque chiffre qui seront entrainés avec les donnée d'apprentissage dont le but de la reconnaissance.

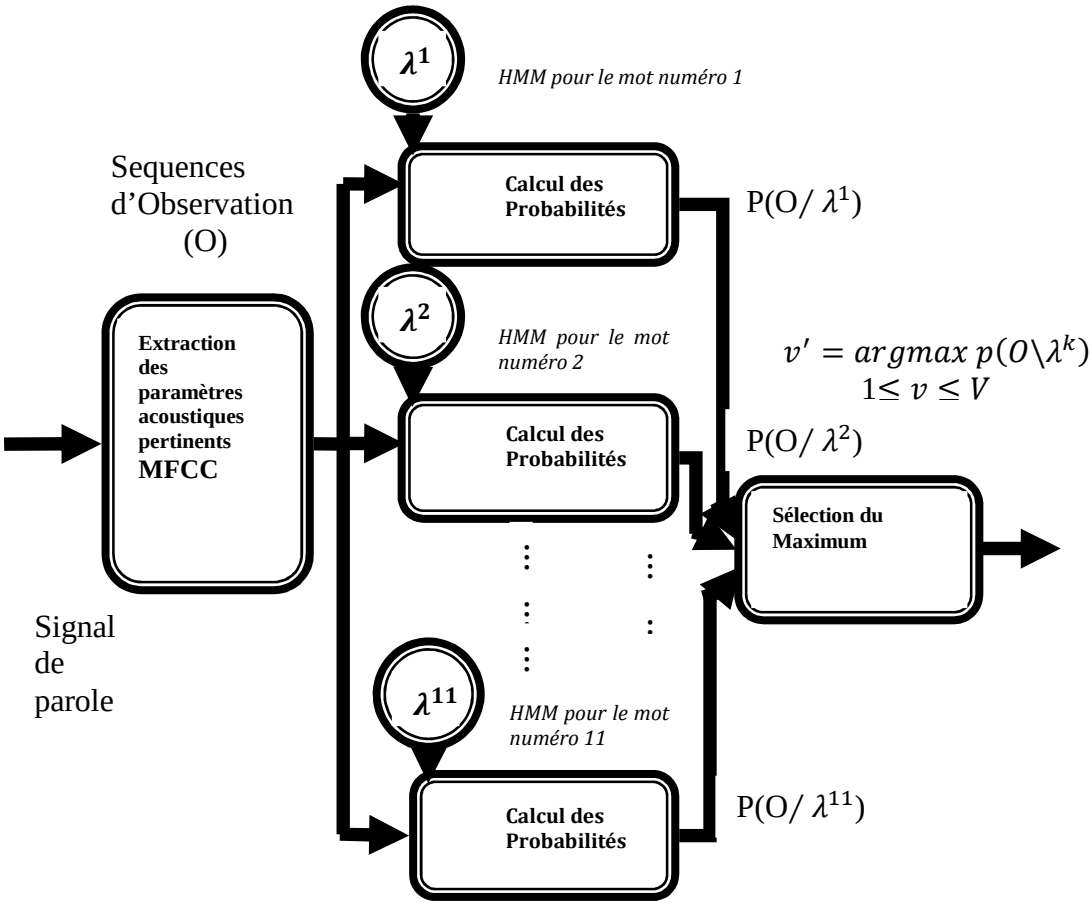


Fig. V.4 Schéma bloc d'un système de reconnaissance de parole isolés

L'entrainement s'effectuera avec l'algorithme de Baum welch détaillé en chapitre 4 et schématisé à la figure V.7. La reconnaissance quand à elle se fera avec l'algorithme de viterbi détaillé en chapitre 4.

V.4.2.Extraction des paramètres MFCC

La première étape de traitement des données parole et l'étape d'extraction des paramètres acoustique, qui dans notre cas est la Mel fréquency cepstral coéfficient (détaillée en annexe B). Le choix s'est porté sur les 13 premiers coefficients MFCC excepté le coefficient c_0 qui est substitué par le logarithme de l´énergie du signal. Pour chaque coefficient, on attribue une dérivée première (13 dérivées premières au total) ainsi qu'une dérivée seconde (13 dérivées secondes) pour prendre en compte la dynamique du signal. En somme, on obtient un vecteur acoustique de 39 coefficients correspondant à chaque trame du signal

- **Organigramme**

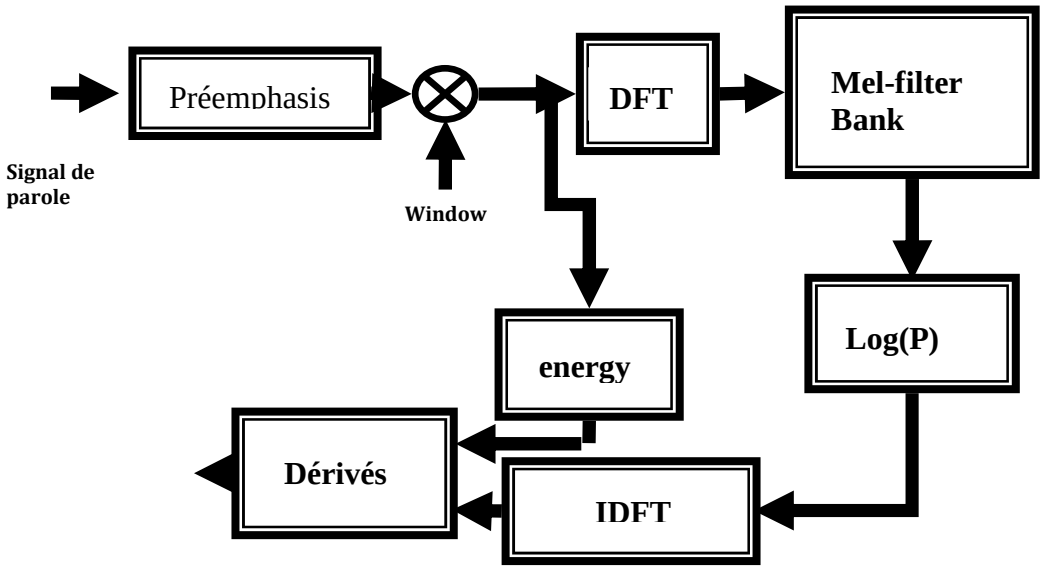


Fig. V.5 Schéma bloc de la paramètrisation MFCC

V.4.3. Le modèle HMM

Chacun des onze modèles MMC choisi est un modèle gauche droite à Cinque états parfaitement adapté et le plus utilisé du fait qu'il tient compte du caractère séquentiel de la parole. Chaque état émet des observations modélisées avec une simple gaussienne dans notre application.

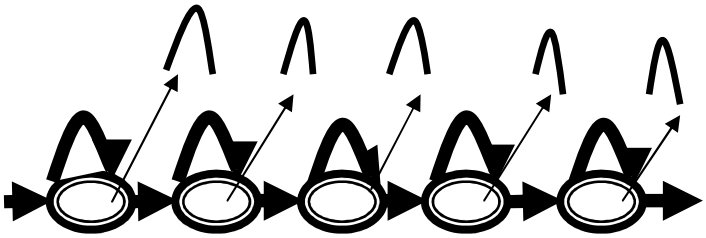


Fig. V.6 Modèle MMC

V.4.4. L'entrainement du modèle MMC

L'organigramme d'apprentissage (Baum welch) des modèles avec les données parole d'entrainement de la base Tidigit peut être schématisé comme suit. le critère discriminatoire est le critère du Maximum de vraisemblance (chapitre IV).

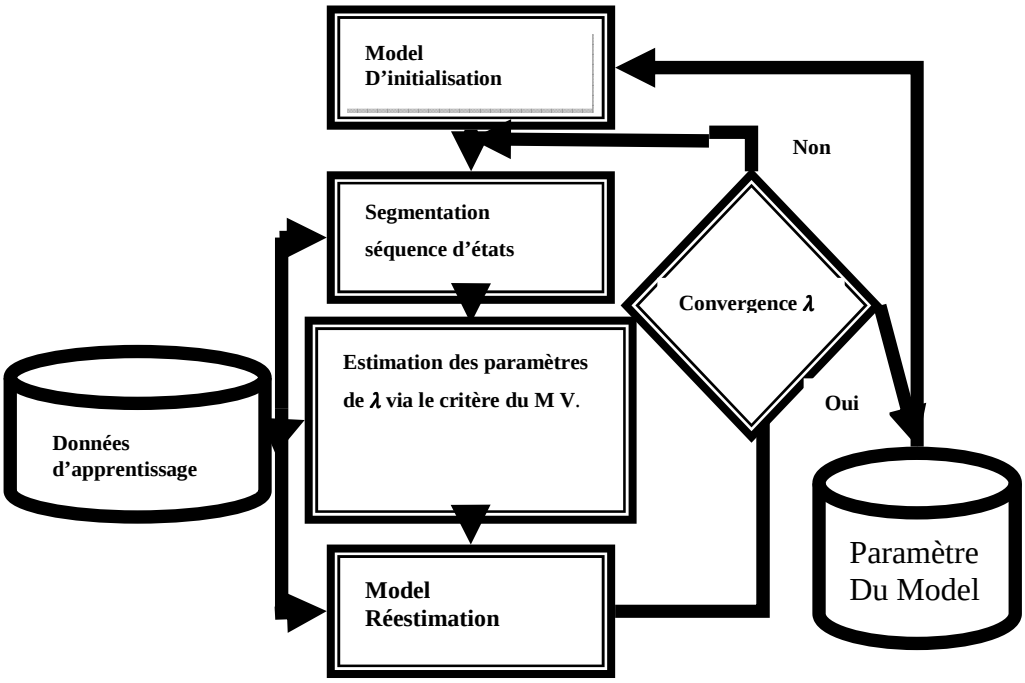


Fig. V.7 Schéma d'apprentissage de Baum-Welch

V.4.5 Tests et Résultats

Dans ce tableau nous présentons différent test qui sont fait en variant le nombre d'états du modèle MMC choisit puis en variant le nombre de données parole d'apprentissage. Les résultats obtenus sont résumés dans le tableau suivant :

Nbre d'états	Base d'apprentissege	Base de test	Résultats
4	**2090 mots (100%)**	**2484(100%)**	**91,83%**
4	**(50%)**	**(100%)**	**88,66%**
5	**(100%)**	**(100%)**	**93,04%**
5	**(50%)**	**(100%)**	**88,46%**
6	**(100%)**	**(100%)**	**93,97%**
6	**(50%)**	**(100%)**	**92,36%**

A partir de ces résultats on comprend le rôle primordial du processus d'apprentissage dans la reconnaissance automatique de la parole (plus la base est importante plus le taux de reconnaissance est meilleur) ce qui met en évidence l'intérêt et l'importance de la base des données parole.

V.5 Deuxième Application : Développement d'un système de Reconnaissance de la Parole sous HTK

Nous avons développé, dans le cadre de ce mémoire, deux systèmes, indépendants du locuteur et fondés sur les modèles de Markov cachés à partir de la plateforme HTK (Hidden Markov ToolKit) de l'Université de Cambridge [50] et sur la base de données de parole TIdigits [49]. La boîte à outils HTK est efficace, flexible (liberté du choix des options et possibilité d'ajout d'autres modules) et complète dans le sens où elle fournit une documentation très détaillée, le livre HTK [48], est une encyclopédie dans le domaine de reconnaissance de la parole.

Le premier système est un système monophone, le deuxième un système triphone. Le système monophone à l'inverse du système triphone, ces unités phonétiques (phonème) sont indépendantes, alors qu'elles sont dépendante dans le système triphone. L'intérêt est d'étudier l'impact de la nature de l'unité phonétique sur les performances de la reconnaissance de la parole.

Nous utiliserons la même base de donnée TIdigit et essaierons de reconnaitre les chiffres un à onze de cette base, puis nous comparerons les taux de reconnaissance à ceux obtenue par la première méthode sous matlab.

V.5.1 Système Monophone

Afin de concevoir notre système, on se base sur des unités acoustiques de type monophone indépandante. On commence par définir les ressources nécessaires dont on a besoin par la suite. On définit, alors, le modèle de langage, appelé aussi lexique ou grammaire **(TAB.V.2)**, qui décrit l'enchainement des mots. Ensuite, on construit le réseau de mots **(wdnet)** et le dictionnaire **(TAB. V.1)** respectivement, grâce aux outils HTK **HParse** et **HDMan** .

Pour la base de données TIdigits, qui est une base de chiffres en anglais, le vocabulaire est assez limité, d'où la simplicité de définir le dictionnaire et la grammaire **(TAB V.1 et TAB V.2).**

f ; k ; n ; r ;s ;t ;v ;w ;z ;sil ;ah ;ao ;ax ;ay ;eh ;ey ;ih ;iy ;ow ;th ;uw

```
eight ey t sil
five f ay v sp
four f ao r sp
nine n ay n sp
oh ow sp
one w ah n sp
seven s eh v ax n sp
sil
six s ih k s sp
three th r iy sp
two t uw sp
zero z ih r ow sp
```

TAB. V.1-Dictionnaire de la base Tidigits

```
$digit = one|two|three|four|five|six|seven|eight|nine|zero|oh ;
(sil <$digit> sil)
```

TAB. V.2-Grammaire de la base Tidigits

Soit un total de 21 phonèmes, une fois qu'on a défini le dictionnaire, la grammaire et la liste des phonèmes, on passe à la description des modèles de Markov cachés. On construit un modèle MMC pour chaque unité acoustique. La topologie MMC choisie est de type gauche-droit à 5 états dont les transitions autorisées sont décrites dans la figure (**Fig.V.8**) et initialisées dans la matrice de transition. La moyenne est initialisée à 0 et la variance à 1 (voir fichier

prototype d'initialisation (**TAB.V.5**)). Ces paramètres du modèle MMC seront réestimées par la suite lors de la phase d'apprentissage.

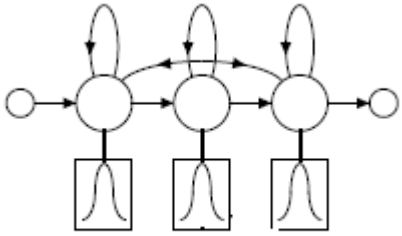


Fig.V.8 Modèle de Markov Cachés utilisé.
q_2, q_3 et q_4 sont émetteurs alors que l'état initial q_1 et final q_5 ne génèrent pas d'observations

Le fichier de configuration (TAB. V.3**)** config permet de définir les paramètres indispensables pour la phase de l'analyse acoustique. Ces coefficients sont extraits des fichiers **wav** et sur des fenêtres de 25ms grâce à l'outil **HCopy** en se servant du fichier de configuration comme paramètre d'entrée .

```
SOURCEFORMAT = WAV------> Format des signaux en entrée de la
phase d'analyse acoustique
TARGETKIND = MFCC-E-D-A ------> Type de paramétrisation utilisé
WINDOWSIZE = 250000.0 ------> Durée de la trame (25ms)
TARGETRATE = 100000.0 ------> Périodicité de la trame
PREEMCOEF = 0.97 ------> Coefficient de pré-accentuation
NUMCHANS = 26 ------> Nombre de canaux du banc de filtres Mel
NUMCEPS = 12 ------>Nombre de coefficients cepstraux MFCC
CEPLIFTER = 22 ------>Coefficient de lissage
```

TAB. V.3-Fichier de configuration pour la phase de l'analyse acoustique

```
~o <VecSize> 39 <MFCC_E_D_A>
~h "proto"
<BeginHMM>
<NumStates> 5-------> Nombre d'états HMM
<State> 2 <NumMixes> 1 -------> Nombre de gaussiennes
<Stream> 1
<Mixture> 1 1.0000
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<State> 3 <NumMixes> 1
<Stream> 1
<Mixture> 1 1.0000
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<State> 4 <NumMixes> 1
<Stream> 1
<Mixture> 1 1.0000
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<TransP> 5
0.000e+0 1.000e+0 0.000e+0 0.000e+0 0.000e+0 }
0.000e+0 4.000e-1 3.000e-1 3.000e+1 0.000e+0 }
0.000e+0 0.000e+0 6.000e-1 4.000e-1 0.000e+0 }-------> Matrice
0.000e+0 0.000e+0 0.000e+0 6.000e-1 4.000e-1 }     de transition
0.000e+0 0.000e+0 0.000e+0 0.000e+0 0.000e+0 }    entre les états
<EndHMM>
```

TAB.V.4-Fichier prototype d'initialisation

- **L'Apprentissage** : La phase d'apprentissage permet de constituer la base de données des modèles de référence du système. La qualité de cette modélisation conditionne en grande partie les résultats du système de reconnaissance de la parole. L'apprentissage est réalisé sous HTK en deux étapes majeures : l'initialisation et la ré-estimation. Pour cela, On utilise deux outils: **HCompV** et **HERest**. La phase d'initialisation des modèles MMC par l'outil **HCompV**, permet de mettre à jour la moyenne et la variance qui valent, avant cette étape, respectivement, 0 et 1. Cette mise à jour est réalisée sur l'ensemble des données du corpus d'apprentissage permettant d'aboutir, à la fin, à des valeurs globales qui seront clonées pour chaque état des modèles MMC.

Ensuite, on obtient dans le répertoire **hmm0** un nouveau fichier prototype contenant des valeurs globales de la moyenne et de la variance. On copie le contenu de ce fichier autant de fois qu'on a de phonèmes et on stocke le résultat du clonage dans un fichier macro nommé modèles**.mmf**. Tous les phonèmes seront ainsi initialisés aux mêmes valeurs de moyenne et de variance. A noter également que la mise à jour des variances est effectuée par défaut avec la commande **HCompV,** tandis que pour ré estimer la moyenne, l'option **-m** devient indispensable.

Le raffinement des modèles MMC consiste à ré estimer leurs paramètres (moyenne et variance) suivant l'algorithme de Baum Welch (chapitre IV) grâce à l'outil **HERest** (la ré estimation des modèles MMC contenu dans le répertoire **hmmi** est sauvegarde dans le répertoire **hmm i +1** à chaque itération **i**).

Ensuite, on généré un autre fichier **modeles0** dans un autre répertoire**.** Les modèles contenus dans ce fichier seront ré estimés suite à deux itérations de l'algorithme de Baum Welch représenté par l'outil **HERest**. Les derniers paramètres estimés, à ce stade, sont sauvegardés dans le répertoire **hmm7**.

Deux itérations de l'algorithme de Baum Welch permettent de ré estimer les modèles. Ainsi s'achève la phase d'apprentissage des modèles **MMC** avec une seule gaussienne.

- **Amélioration des modèles(GMM)** : Les modèles obtenus peuvent être améliorés par utilisation de densités de probabilités d'émission multi-gaussiennes au lieu de se contenter d'une simple loi normale. Cela permet d'´eviter certaines hypothèses grossières sur la forme de la densité si le nombre de gaussiennes est suffisant. En effet, le choix du nombre optimal de gaussiennes est un problème difficile. Un outil **d'HTK, HHEd** réalise l'augmentation du

nombre de gaussiennes, où on augmente progressivement le nombre de gaussiennes (1, 2, 4, 8, 12, 16). Chaque augmentation de gaussienne est suivie de deux ré estimations des modèles avec **HERest, HERest**.

Suite à cette procédure les modèles sont de plus en plus précis. Le seul inconvénient est la charge des calculs qui augmente à son tour.

V.5.2 Système triphone

Le premier système conçu est basé sur une modélisation par monophones, les modèles sont ainsi hors contexte. Or, un système plus robuste de reconnaissance de la parole continue devrait au moins envisager les effets de la co-articulation et de la vitesse d'élocution qui peuvent limiter son efficacité. souvent on considére que la production de la parole est parfaite et on oublie que le débit de la parole peut s'accélérer et que les organes phonatoires ne peuvent pas suivre car ils sont limités dans leur déplacement. Tout ceci provoque une certaine influence mutuelle suivant ou précédant les sons produits qui altère leurs formes en fonction du contexte gauche ou droit. D'où l'intérêt des modèles contextuels (diphones, triphones,...). Ceux-ci prennent en compte la source de variabilité du signal de parole permettant ainsi une meilleure modélisation, un gain significatif en précision de la transcription et ainsi de meilleures performances. Le seul inconvénient de telles approches est l'augmentation de la charge de calcul vu le très grand nombre de modèles contextuels existants. Suite à ces remarques, l'étape prochaine consiste à élaborer un système de reconnaissance de la parole dit contextuel car basé sur des triphones (contextes gauche et droit d'un phonème).

- **Conversion de la transcription :** On commence par convertir les transcriptions de phonèmes alignés **aligned.mlf**, du système monophone (V.5.1), en transcription par triphones avec l'outil **HLEd.**

Ensuite, on ré estime en deux itérations la moyenne et la variance des modèles avec l'algorithme Baum Welch toujours via l'outil **HERest.**

Pareil à la reconnaissance par monophones, on va procéder à l'augmentation progressive des gaussiennes jusqu'à en atteindre 16. Chaque augmentation sera suivie d'une phase de ré estimation des modèles par l'algorithme de Baum Welch.

- **La Reconnaissance :** Le processus de décodage consiste à comparer l'image de l'unité à identifier avec celles de la base de référence. Le module de décodage de la parole, **HVite**, utilise l'algorithme de Viterbi pour trouver la séquence d'états la plus probable correspondant aux paramètres observés et en déduire les unités acoustiques correspondantes. Le décodage est réalisé par l'algorithme de Viterbi sous la contrainte d'un réseau syntaxique et éventuellement d'un modèle de langage.

V.5.3 Analyse des résultats

	Monophones	Triphones
Base de test (Acc%)	**99,51%**	**99,47%**
Base d'apprentissage (Acc%)	**99,05%**	**99,23%**

TAB.V.5 Performance des système de reconnaissance à base de monophones et triphones sur la base de test et la base d'apprentissage du corpus Tidigits

D'après ce tableau, nos deux systèmes basés sur une paramétrisation de type MFCC, détaillée en Annexe B, sur une modélisation statistique de type HMM et sur une transcription avec et sans contexte, donnent de très bons résultats.

Les performances du système à base de triphones ne se distinguent pas nettement des performances du système à base de monophones, ceci peut être expliqué par le fait que les prononciations de la base de données TIdigits sont presque parfaites et les enregistrements ne modélisent pas les effets de coarticulation, Lombard, stress, sans pour autant oublier de signaler que cette base de parole est à vocabulaire réduit.

Conclusion

L'application sous HTK à base de monophone et triphone nous a permis d'avoir un meilleur taux de reconnaissance. Cela peut être expliqué par le fait que l'application développée sous matlab est purement acoustique (sans contrainte de langage) et quelle ne prend pas en compte la dépendance entre vecteurs acoustique (phonème) à l'inverse de l'application HTK.

Les MMC nous fournit une solution efficace du problème de la reconnaissance Automatique de la Parole et bénéficie d'algorithmes très efficaces pour la reconnaissance et

pour l'apprentissage Automatique. Cependant, Les hypothèses qui rendent l'optimisation des Modèles de Markov Cachés possible limitent toutefois leurs généralités et sans à l'origine de certaines de leurs faiblesses qui limitent les performances des systèmes de RAP (les données à l'entrée des MMCs sont supposées être statiquement indépendantes, la corrélation temporelle entre vecteurs acoustique est alors négligée. Aussi L'utilisation de MMCs de premier ordre repose sur l'hypothèse, que la parole est également un processus de Markov de premier ordre, rendant la modélisation et l'apprentissage de corrélations à long terme difficile). Beaucoup de variantes des MMCs (classique) présenté dans ce mémoire, existent, et sont appliquées dans les systèmes de reconnaissance Automatique de la parole. De nos jours L'Approche MMC est à la base de la plupart des systèmes de Reconnaissance modernes [50].

CONCLUSION GÉNÉRALE

Dans ce travail, notre objectif consiste à d'étudier le signal de parole afin de concevoir et de développer un système pour son traitement et sa reconnaissance. Pour concevoir notre système, nous avons étudié ceux déjà existants et avons choisi d'utiliser une plateforme qui nous a paru être la plus performante, la plus utilisée et celle qui a montré le plus ses preuves actuellement, qui est la plateforme HTK, Hidden Markov Toolkit, basée sur les modèles de Markov cachés.

Tout au long de ce travail nous avons abordé différents aspects tout aussi importants les uns que les autres. Nous avons commencé par comprendre le processus de génération de la parole par l'être humain puis nous nous sommes concentrés sur l'étude des différents moyens utilisés pour capter ce signal et le traiter. Par la suite, nous avons décrit les modèles de Markov cachés qui sont utilisés dans de nombreux domaine dont celui du traitement de la parole et avons finalement, choisi une plateforme basée sur ces modèles pour construire deux systèmes de reconnaissance automatique de la parole, le premier sous l'environnement Matlab et le second sous la plate forme HTK.

Nous avons, ainsi, réaliser notre système de reconnaissance de la parole sur la base de données parole TIdigit, notre base de travail et avons obtenue des taux de reconnaissance plus qu'appréciable, qui atteignant 99% dans le cas de l'utilisation de HTK.

Malgré ces avancées, les systèmes actuels sont encore imparfaits. Les problèmes à résoudre représentent un des défis les plus difficiles posés à l'intelligence artificielle. Un important effort de recherche est nécessaire, notamment sur le plan de la robustesse des méthodes de reconnaissance et de la conception de systèmes de dialogue. Les travaux à mener nécessitent un effort pluridisciplinaire de collecte de signal vocal, mais aussi de modélisation d'un ensemble de faits et de connaissances sur la langue naturelle et sur les mécanismes de la communication parlée. Nous avons vu qu'une modélisation stochastique permet de résoudre, en partie, le problème, mais il n'est pas exclu que l'utilisation de connaissances explicites revienne à l'ordre du jour à l'avenir.

Pour clore, nous espérons, par ce travail, avoir démontré l'importance du sujet et la nécessité de consacrer encore plus d'efforts et d'études pouvant nous rapprocher rapidement d'une solution performante que seule notre imagination pourrait limiter.

ANNEXES

ANNEXE A : MISE EN ŒUVRE DELA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE SOUS HTK.

Introduction

HTK est une boîte à outils de modèles de Markov cachés MMC, conçue pour la construction et la manipulation de ces modèles. Cette boîte est constituée d'un ensemble de modules bibliothèque et d'outils disponibles en codes sources C. Ces outils HTK sont conçus pour fonctionner en ligne de commande, généralement sous l'environnement linux avec le Shell C. Chaque outil a un nombre d'arguments obligatoires en plus d'arguments optionnels préfixés par le signe "–". Le chapitre "Référence section" de l'ouvrage htkbook [48] décrit en détail tous les outils de la boîte HTK ainsi que leurs arguments. Principalement, la boîte à outils HTK est utilisée pour la construction des systèmes RAP basés sur les modèles MMC dans un but de recherche scientifique. Généralement les deux processus indispensables pour le fonctionnement d'un RAP sont le processus d'apprentissage et celui de reconnaissance (ou décodage). La figure **A.1** illustre l'enchaînement de ces processus. Premièrement, les outils d'apprentissage HTK sont utilisés pour estimer les paramètres de l'ensemble des modèles MMC en utilisant des signaux de parole ainsi que leurs transcriptions associées. Ensuite, les signaux de parole inconnue sont transcrits en utilisant les outils de reconnaissance. Le lecteur peut consulter le livre htkbook pour plus de détails sur l'implémentation des systèmes RAP sous la plateforme HTK.

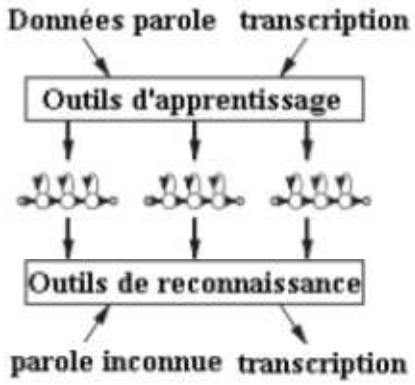


Fig.A.1-Processus d'un système de RAP

Pratiquement, la construction d'un système RAP se base sur 4 phases principales: préparation des données, apprentissage, test, analyse. La figure **A.2** illustre les différents outils HTK de chaque phase d'un système de Reconnaissance de la Parole continue.

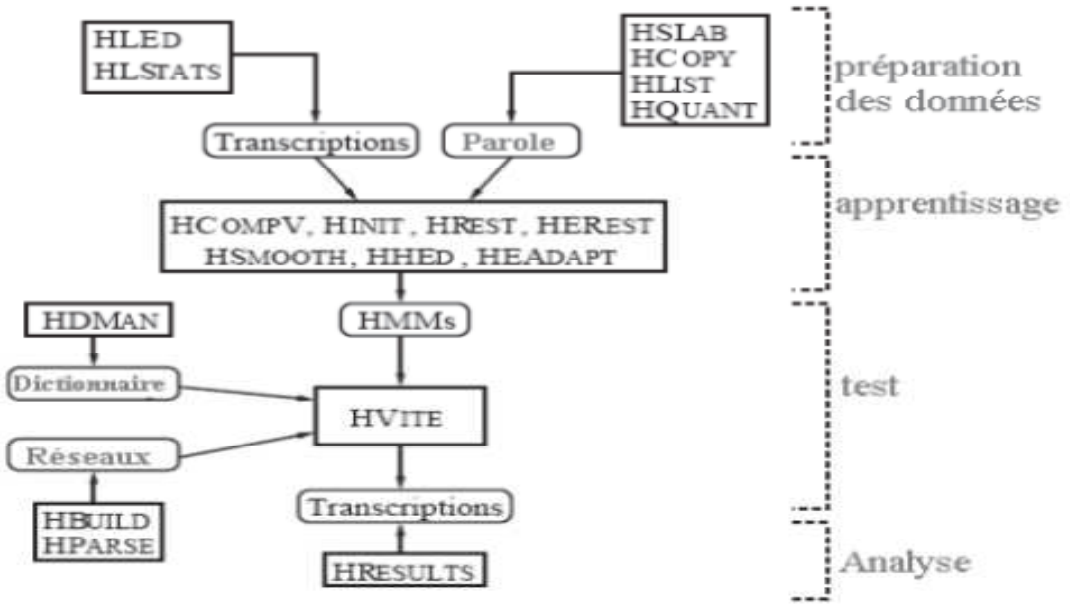


Fig.A.2 Différentes phases du système RAP sous HTK et outils associés

A.1. Outils de préparation de données

La construction d'un ensemble de modèles MMC exige un ensemble de fichiers de données de parole (signaux), ainsi que leurs transcriptions correspondantes. Souvent les données de parole sont récupérées à partir d'une base de données. Cette base doit être répartie en un corpus d'apprentissage et un corpus de test. Chacun de ces corpus contient un ensemble de fichiers texte contenant la transcription orthographique des phrases et un ensemble de fichiers de données contenant les échantillons des signaux correspondant aux fichiers texte. Avant d'être utilisées dans l'apprentissage, ces données doivent être converties en un format paramétrique approprié et ses transcriptions associées doivent être converties en format correct.Si les données de parole ne sont pas disponibles, alors l'outil **HSLab** peut être utilisé pour enregistrer la parole et l'étiqueter manuellement par n'importe quelle transcription (par phonème ou mot). Ainsi pour chaque phrase prononcée, on lui correspond un fichier signal (exemple d'extensions :.wav,.sig) et un fichier de transcription (extension.lab).

Cependant, avant d'effectuer ces transcriptions, un dictionnaire des mots doit être défini afin d'être utilisé dans la phase d'apprentissage et celle de test. Dans le cas d'un système basé sur des modèles HMM représentant des phonèmes, la construction du dictionnaire s'effectue par l'outil **HDMan**. De plus la grammaire de la tâche considérée doit être définie en utilisant l'outil **HParse.** Cet outil génère un réseau de mots définissant la grammaire considérée décris sur la figure **A.2.**

La dernière étape dans la phase de préparation des données est la conversion du signal de chaque phrase en une séquence de vecteurs acoustiques tel présenté sur la figure **A.3**. Cette conversion est effectuée par une analyse acoustique en utilisant l'outil **HCopie**. Différents types de paramètres acoustiques sont supportés par cet outil comme : LPC, LPCC, MFCC, PLP, FBANK (Log Mel-Filter Bank), MELSPEC (Linear Mel-Filter Bank), LPCEPSTRA (LPC Cepstral Coefficients), LPREFC (Linear Prediction Reflection Coefficients), USER (type défini par l'utilisateur).

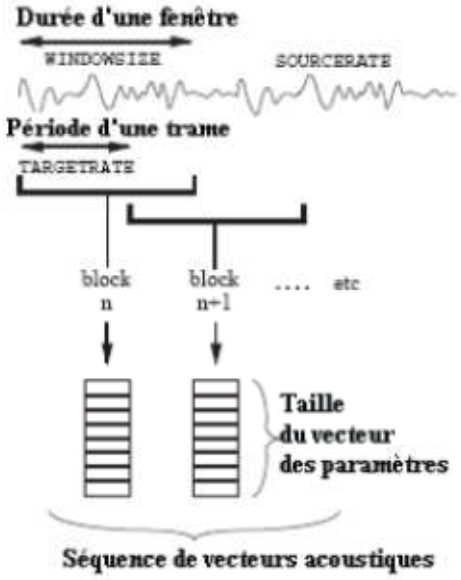


Fig.A.3-Processus de l'analyse acoustique

La ligne de commande pour l'exécution de **HCopy** s'écrit comme suit :

```
HCopy -T 1 -C config -S codetr.scp
```

La figure **A.4** montre le principe de fonctionnement de cet outil pour la conversion d'un ensemble de fichiers parole d'extension **.wav** en un ensemble de fichiers d'extension **.mfc** contenant des vecteurs de paramètres acoustiques **MFCC**. La liste de l'ensemble de ces fichiers est donnée dans un fichier appelé **codetr.dcp** dont un extrait est fourni :

```
root/training/corpus/sig/S0001.wav root/training/corpus/mfcc/S0001.mfc
root/training/corpus/sig/S0002.wav root/training/corpus/mfcc/S0002.mfc
root/training/corpus/sig/S0003.wav root/training/corpus/mfcc/S0003.mfc.....etc.
```

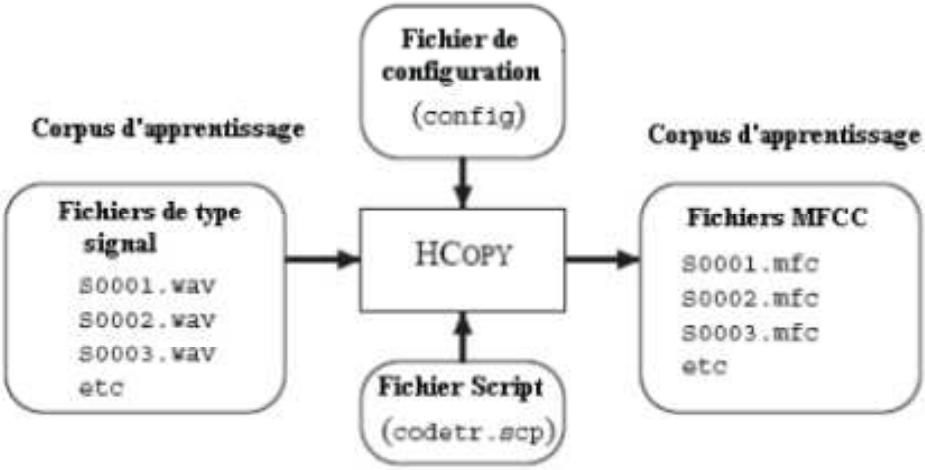


Fig.A.4 Principe de fonctionnement de l'outil HCopy

Cependant l'exécution de l'outil **HCopy** exige un fichier de configuration (config) pour définir les différents paramètres de l'analyse acoustique considérée. Voici un exemple de ce type de fichier associé à une analyse acoustique **MFCC** :

```
# Exemple d'un fichier de configuration pour une analyse acoustique MFCC
SOURCEFORMAT = HTK # donne le format des fichiers des signaux
TARGETKIND = MFCC_0_D_A # identificateur des coefficients à utiliser
# Unit = 0.1 micro-second :
WINDOWSIZE = 250000.0 # = 25 ms = longueur de la durée d'une trame
TARGETRATE = 100000.0 # = 10 ms = période des trames
NUMCEPS = 12 # nombre des coefficients MFCC(ici de c1 to c12)
USEHAMMING = T # utilisation de la fonction de Hamming pour le fenêtrage
des trames
PREEMCOEF = 0.97 # coefficient de prè-accentuation
NUMCHANS = 26 # nombre de canaux des bancs de filtres
CEPLIFTER = 22 # longueur de liftrage cepstral
# la fin
```

A.2. Outils d'apprentissage

La deuxième phase consiste à construire les modèles MMC des mots appartenant au dictionnaire de la tâche considérée. Premièrement, pour chaque mot, il faut définir un modèle prototype contenant la topologie choisie à savoir le nombre d'états du modèle, la disposition de transitions entre les états, le type de la loi de probabilité associée à chaque état. L'état initial et final de chaque modèle n'émettent pas des observations mais servent seulement à la connexion des modèles dans la parole continue. Les probabilités d'émissions associées aux états sont des mélanges de gaussiennes multivariées (GMM) dont les composantes sont les probabilités a priori définies chacune par une matrice de covariance et un vecteur de moyennes dans l'espace des paramètres acoustiques. La matrice de covariance peut être

choisie diagonale si l'on suppose l'indépendance entre les composantes des vecteurs acoustiques.

Ces modèles prototypes sont générés dans le but de définir la topologie globale des modèles HMM. Ainsi, l'estimation de l'ensemble des paramètres de chaque modèle MMC est le rôle du processus d'apprentissage. Les différents outils d'apprentissage sont illustrés dans la figure **A.5**.

Selon cette figure, deux chaines de traitement peuvent être envisagés pour l'initialisation des modèles MMC. La première chaîne tient en compte des signaux étiquetés en label de mot.

Dans ce cas, l'outil **HInit** extrait tous les segments correspondant au mot modélisé et initialise les probabilités d'émission des états du modèle au moyen de l'algorithme segmentale k-means. Ensuite l'estimation des paramètres d'un modèle est affinée avec **HRest**, qui applique l'algorithme optimal de Baum-Welch jusqu'à la convergence et ré estime les probabilités d'émission et de transition.

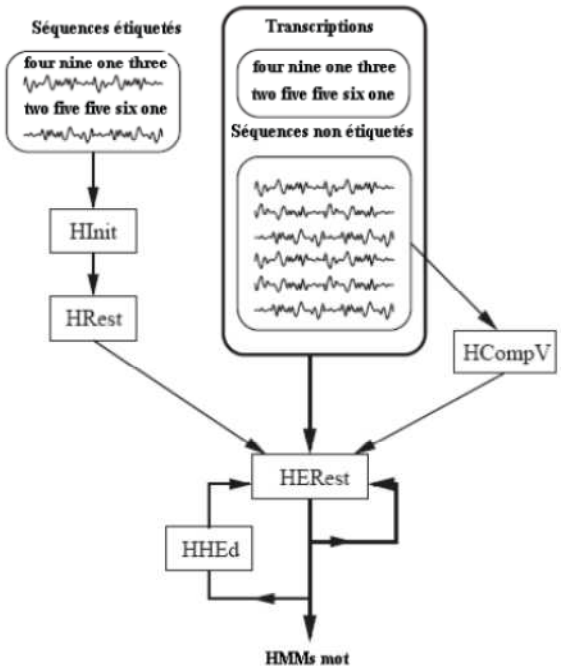


Fig.A.5 Outils d'apprentissage HTK

Dans la deuxième chaîne, les signaux ne sont pas étiquetés. Dans ce cas, tous les modèles MMC sont initialisés avec le même modèle dont les moyennes et les variances sont égales respectivement à la moyenne et la variance globales de tous les vecteurs acoustiques du corpus d'apprentissage. Cette opération est effectuée par l'outil **HCompV**.

Après l'initialisation des modèles, l'outil **HERest** est appliqué en plusieurs itérations pour ré estimer simultanément l'ensemble des modèles sur l'ensemble de toutes les séquences de vecteurs acoustiques non étiquetés. Les modèles obtenus peuvent être améliorés, en augmentant par exemple le nombre de gaussiennes servants à estimer la probabilité d'émission d'une observation dans un état. Cette augmentation est effectuée par l'outil **HHEd**. Les modèles doivent être ensuite ré estimés par **HRest** , **HERest.**

A.3. Outils de reconnaissance

La boîte HTK fournit un outil de reconnaissance appelé **HVite** qui permet la transcription d'une séquence de vecteurs acoustiques en une séquence de mots. Le processus de reconnaissance est illustré dans la figure **A.6**.

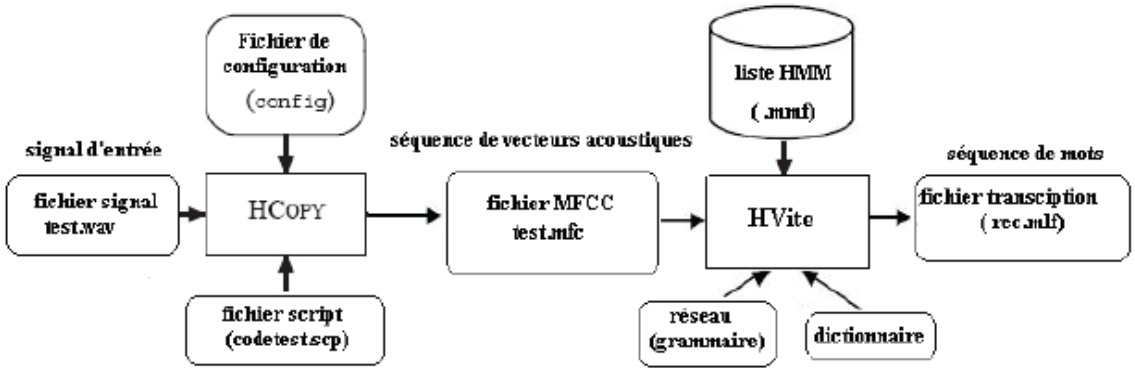


Fig.A.6 Processus de reconnaissance sous HTK

HVite utilise l'algorithme de Viterbi pour trouver la séquence d'états la plus probable qui génère la séquence d'observations (vecteurs acoustiques) selon un modèle MMC composite, ceci afin d'en déduire les mots correspondants. Le modèle composite permet la succession des modèles acoustiques en fonction du réseau de mots qui définit la grammaire de la tâche considérée.

Le résultat de décodage par l'outil **HVite** est enregistré dans un fichier d'extension **(.mlf)** contenant l'étiquetage en mots du signal d'entrée.

A.4. Outils d'évaluation

Généralement les performances des systèmes RAP sont évaluées sur un corpus de test contenant un ensemble de fichiers d'échantillons parole ainsi que leurs fichiers d'étiquetage associés. Les résultats de reconnaissance des signaux du corpus de test sont comparés aux étiquettes de référence par un alignement dynamique réalisé par **HResults**, afin de compter les étiquettes identifiées, omises, substituées par une autre, et insérées. Ces statistiques permettent de calculer le taux ou la précision de reconnaissance**.**

ANNEXE B :
LA PARAMÉTRISATION MFCC

Introduction

Parmi les paramétrisations les plus utilisés dans le domaine de la reconnaissance de la parole, les coefficients MFCC sont considérés comme étant les meilleurs paramètres qui peuvent caractériser une voix parmi d'autres et c'est cette paramétrisation que nous nous proposons de décrire dans cette dernière annexe de notre travail.

B.1. La paramétrisation par MFCC

La paramétrisation MFCC (Mel-Frequency Cepstral Coefficients) est la paramétrisation la plus répandue dans les systèmes de reconnaissance actuels. Nous donnons ci-dessous les principales étapes de cette para métrisation :

1. Fenêtrage du signal : Le signal de parole est séparé en trames de $\boldsymbol{N}$ échantillons, chaque trame étant séparée de $\boldsymbol{M}$ échantillons. Dans le cas courant où $\boldsymbol{M} < N$ on dira qu'il y a recouvrement (overlap en anglais) entre les trames. En pratique, la longueur $\boldsymbol{N}$ d'une trame est couramment choisie de façon à avoir des trames dont la durée est de l'ordre de 20 ms associé à un recouvrement entre trames de 50% correspondant à une valeur de $\boldsymbol{M} = {}^{\boldsymbol{N}}/_{\boldsymbol{2}}$. L'opération précédente consiste ainsi à appliquer une fenêtre rectangulaire de durée finie sur l'ensemble du signal. Pour réduire les effets dus aux discontinuités aux bords de la fenêtre, il est fréquent de pondérer une trame de longueur $\boldsymbol{N}$ par une fenêtre de pondération. L'une des fenêtres les plus utilisée est la fenêtre de Hamming. Cette opération donne la trame fenêtrée :

$$\boldsymbol{s_w(n) = \tilde{s}(n)\, w(n)}$$

Où $\boldsymbol{w(n) = 0.54 - 0.46\, cos\left(\frac{2\pi n}{N-1}\right)\ avec\ 0 \leq n \leq N-1}$

2. Calcul de la transformée de Fourier rapide (**FFT**) pour chaque trame du signal de parole.

3. Filtrage par un banc de filtres MEL : Cette opération permet d'obtenir à partir du spectre $\boldsymbol{S(k)}$ de chaque trame, un spectre modifié qui est en fait une suite de coefficients, noté $\boldsymbol{\tilde{S}(k)}$, représentant l'énergie dans chaque bande fréquentielle $\boldsymbol{k}$ (définies sur l'échelle Mel), pour $\boldsymbol{k = 1, \ldots, K}$. En pratique, on utilise des filtres triangulaires de largeur de bande constante et régulièrement espacées sur l'échelle Mel (On peut par exemple choisir un

espacement entre filtres de 150 mels et une largeur des filtres triangulaire prise à leur base de 300 mels).

4. Calcul des coefficients MFCC : Les coefficients MFCC sont alors obtenus en effectuant une transformée en cosinus discrète inverse du logarithme des coefficients $\tilde{S}(k)$:

$$\tilde{c}_n = \sum_{k=0}^{K} (\log \tilde{S}_k) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{k}\right] \qquad pour\ n = 1, 2, \dots\dots\dots, L.$$

Où L est le nombre de coefficients cepstraux désirés.

5. Pondération : En raison de la grande sensibilité des premiers coefficients cepstraux sur la pente spectrale générale et de la sensibilité au bruit des coefficients cepstraux d'ordre élevé, il est courant de pondérer ces coefficients pour minimiser cette sensibilité. Cette pondération pourra s'écrire sous la forme :

$$\hat{c}_m = w(m)c_m \qquad pour\ 1 \le m \le Q$$

Où Q est le nombre de coefficients cepstraux.

La fenêtre de pondération cepstrale est en fait un filtre passe bande dont un choix approprié peut être :

$$w(m) = \left[1 + \frac{Q}{2}\sin\left(\frac{\pi m}{Q}\right)\right] \qquad pour\ 1 \le m \le Q$$

Cette fenêtre tronque le nombre de coefficients et diminue le poids des premiers et derniers coefficients.

6. Calcul des dérivées temporelles Δ, Δ^2 : La représentation cepstrale donne une bonne représentation des propriétés fréquentielles locales du signal (i.e. pour une fenêtre de signal donnée). Une représentation améliorée peut être obtenue en incluant de l'information liée à l'évolution temporelle des coefficients cepstraux. Celle-ci peut être obtenue par exemple à l'aide des dérivées premières et secondes des coefficients cepstraux. Soit $c_m(t)$ les coefficients cepstraux obtenus à l'instant t (ou plus précisément à la fenêtre d'indice t). Cette suite est obtenue à des instants discrets et ainsi il est bien connu qu'un simple moyennage aux différences ne permet pas d'obtenir des estimations non bruitées. Ainsi, la dérivée est souvent obtenue en effectuant une moyenne sur un plus grand horizon temporelle sous la forme :

$$\Delta c_m \approx \mu \sum_{k=-K}^{K} k\, c_m(t+k)$$

Où μ est une constante de normalisation et $(2K+1)$ est le nombre de trames utilisées pour ce calcul.

Une implémentation classique de la paramétrisation MFCC consiste à prendre les 13 premiers coefficients cepstraux (en omettant l'énergie représentée par c_0) et à construire des vecteurs acoustiques de 39 éléments incluant les dérivées première (Δ)et seconde (Δ^2) de ces coefficients.

BIBLIOGRAPHIE

Bibliographie

[1] **R. Boite & all.,**Traitement de la parole, presses polytechniques et universitaires Romandes, Novembre 1999.

[2] **J P. Haton & all.,** Reconnaissance Automatique de la parole, Dunod

[3] **G. Von Bekesy**, Experiments in Hearing , McGraw-Hill, New York,1960.

[4] **E. Zwicker, R. Feldetkeller,** Psyachoacoustique, CENT-ENST, Collection technique et scientifique des télécommunications, Masson, Paris, 1981.

[5] **JP. Haton,** Reconnaissance Automatique de la Parole et dialogue oral homme-machine,

[7] **J. D. Markel et A. H. Gray Jr,** Linear Prediction of Speech. Communication and Cybernetics. Berlin Heidelberg New York : Springer-Verlag, 1976.

[8] **L. Rabinier et B H. Huang,** Fondamentals of speech Recognition, Englewood Cliffs, NJ.: Prentice Hall, 1993.

[9] **H. Hermansky,** Perceptual linear predictive (plp) analysis of speech. The Journal of the Acoustical Society of America 87, 1738–1752, 1990.

[10] **H. Hermansky et N. Morgan,** Rasta processing of speech. IEEE Transactions on Speech and Audio Processing 2(4), 578– 589,1994.

[11] **R. Gemello & all.,** Multiple resolution analysis for robust automatic speech recognition. Computer Speech and Language 20(1), 2–21, 2006.

[12] **H. Hermansky, D. Ellis, et S. Sharma,** Tandem connectionist feature extraction for conventional hmm systems. Dans les actes de IEEE International Conference on Acoustics, Speech and Language. Processing, Istanbul, Turkey, 1635–1638,2000.

[13] **T. Vintsyuk,** Speech discrimination by dinamique programming, Kibernetika, Vol. 4, pp,81-88, Jan-Fev, 1968.

[14] **P.C Mahalanobis,** On generalized distance in statistics, Procedings of the national Inst. Sci. (India), Vol. 12, pp. 49-55, 1936.

[15] **R. O. Duda &P. E. Hart,** Patern Classification and scene Analysis, Wiley, 1973.

[16]**N. Morgan & H. Bourlard**, Continuous Speech Recognition: An Introduction to the Hybrid HMM/Connectionist Approach, IEEE Signal Processing Magazine, Vol. 12, n°3, pp. 25-42, Mai 1995.

[17] **O. Cappé**, Ten years of Hmms, http://www.tsi.enst.fr/cappe/docs/hmmbib.html,2001.

[18] **A. A. Markov,** An example of statistical investigation in the text of ”Eugene oneygin” illustrating coupling of “test” in chains. In Processings of Academic Scientific St. Petersburg, IV, pages 153 162, 1913.

[19] **C. C. Shannon,** A mathematical theory of communications. Bell System Technology Journal, 27:379 423, 623, 656, 1948.

[20] **W. Feller**, An Introduction to probability theory and its applications, volume 1. John Willey, New York, 2nd edition, 1958.

[21] **H. O. Hartley**, Maximum likelihood estimation from Incomplete Data. Biometrics, 14:147 194, 1958.

[17]**O.Cappé,** Ten years of Hmms,http://www.tsi.enst.fr/cappe/docs/hmmbib.html,2001.

[18]**A. A.Markov,**An example of statistical investigation in the text of ”Eugene oneygin” illustrating coupling of “test” in chains. In Processings of Academic Scientific St. Petersburg, IV, pages 153 162, 1913.

[19]**C. C. Shannon,** A mathematical theory of communications. Bell System Technology Journal, 27:379 423, 623, 656, 1948.

[20]**W. Feller,** An Introduction to probability theory and its applications, volume 1. John Willey, New York, 2nd edition, 1958.

[21]**H. O. Hartley,** Maximum likelihood estimation from Incomplete Data. Biometrics, 14:147 194, 1958.

[22]**P. Billingsley**, Statistical inferance for Markov process, University of Chicago Press, Chicagoc, 1961.

[23]**L. E. Baum,** An inequality with applications to statistical estimation for probabilistic functions of Markov Process. Inequalities, 3 :1 8, 1972.

[24]**J. D. Furguson,** Variable duration models for speech. In Procedings of the Symposium on the Application of Hidden Markov Models to text and speech-IDA-CRD, Pages 8 15, Princeton NJ, 1980.

[25]**A. J. Viterbi,** Error Bounds for conventionnal codes and asymptotically optinium decoding algorithm. IEEE transactions on information theory, 13: 260 269, 1967.

[26]**Jr. Forney, G. D.,** The Viterbi Algorithme. In Procedings of IEEE, Vol. 61, pages 268 278, 1973

[27]**M. Slimane,** Les chaines de Markov cachés : définitions, algorithmes, architectures. Rapport interne n°260, Université François-Rabelais de Tours, Laboratoire d’Informatique, Tours, France, 2002.

[28]**L. R. Rabinier**, A tutorial on hidden Markov models and selected applications in speech recognition. In Procedings of the IEEE,Vol.77, pages 257 286, 1989.

[29]**L. R. Bahl and F. Jelinek**, Decoding For channels with insertions, deletions and substitutions, with applications to speech recognition. IEEE Transactions Theory, 21:404 411, 1975.

[30]**J. K. Baker**, Stochastic Modeling as a Means of Automatic Speech Recognition. PhD thesis, Carnegie-Mellon University, 1975.

[31]**H. Bourland and C. Wellekens,** Links Between Markov Models and multiplayer perceptrons. IEEE transactions on Pattern Analysis and Machine Inteligence, 12(10):1 4, 1990.

[32]**L. R. Rabinier & all.,** On the Application of vector Quantizisation and Hidden Markov Models to Speaker-Independant isolated word recognition. The Bell System Technical Journal, 62:1075 1105, 1983.

[33]**L. R. Rabinier and S. E. Levinson,** A Speaker independant, syntax directed connected word recognition system based on Hidden Markov models and level building. IEEE Transactions on Acoustics, Speech, Signal Precessing, 33(3):561 573, 1985.

[34]**L. R. Bahl & All.,** A Maximum Likelihood approach to continuous speech recognition. IEEE Transactions on Pattern Analysis and Machine Inteligence (PAMI),5(2): 197 190, 1983.

[35]**A. E. Rosemberg and A. M. Colla,** A Connected speech recognition system based on spotting diphone-like segments-preliminary results. In Proccessings of IEEE International Conference on Acoustics, Speech, Signal Precessing (ICASSP'87), Pages 85 87, Dallas, 1987.

[36]**F. Siamaria & A. Harter,** Parameterisation of Stochastics model for human Face Identification. In IEEE workshop on Applications of Computer Vision, Florida, 1994.

[37]**A. Kundu & P. Bahl,** Recognition of Handwritten script : a Hidden Markov model based approach. In International Conference on Acoustics, Speech, Signal Processing (ICASSP'88), Pages 928 931, 1988.

[38]**A. Soukhal & all.,** Application des Chaines de Markov cachées au problème d'ordonnancement dans une cellule robotisée. In Conférence Internationale sur la Productique (CIP'01), Pages 151 156, Algérie, 2001.

[39]**M. R. Amini,** Apprentissage automatique et recherche de l'information : Application à l'extraction d'information de surface et au résumé de texte. PhD thesis, Université Paris 6, 2001.

[40]**A. P. Dempster & all.,** Maximum-Likelihood from incomplete Data via the EM algorithm. Journal of the Royal Statistical Society B,39(1):1 39, 1977.

[41]**A. Ganapathiraju**, Discriminative techniques in hidden Markov models. Course paper, 1999.

[42]**G. Celeux & J. Diebolt**, L'algorithme SEM : un algorithme d'apprentissage probabiliste pour la reconnaissance des mélanges de densités. Revue de Statistique Appliquée, 34(2) :35 52, 1986.

[43] **G. Celeux & J. Diebolt,** Une version de type recuit simule de l'algorithme EM.

Technical Report RR-1123, INRIA-Rocquencourt, 1989.

[44]**O. Cappé & all.**, Simulation-based methods for blind maximum- likelihood filter identification. Signal Processing, 73: 3 25,1999.

[45]**M. Berthold & D. J. Hand**, Inteligent data analysis : an introduction. Springer-Verlag, 1998.

[46]**B. H. Juang & L. R. Rabinier**, The segmental k-means algorithm for estimating parameters of hidden Markov models. IEEE transactions on acoustics, speech and signal processing, 38(9):1639 1641, 1990.

[47]**L. Saul & M. Rahim**, Maximum likelihood and minimum classification error factor analysis for automatic speech recognition. IEEE Transactions on Speech and audio Precessing, 8(2): 115 125, 2000.

[48]**S. Young & all.,** The htk book (For htk version 3.4). Cambridge University Engenereering Departement, 2006.

[49] **R. G. Leonard & G. R. Doddington,** A Speaker-Independent Connected Digit Database :Tidigits. Texas Instruments, USA.

[50]**K. F. LEE**, Automatic Speech Recognition-The Development of the sphinx System, Kluwer Academic, Norwell Mass, 1989.