Table des matières

2. L'arabe standard et les dialectes du Maghreb

Chapitre 2 : Les Machines à Vecteurs Supports

Chapitre 3 : Identification Automatique des Langues (IAL)

Partie 2 : Vers un système d'identification des dialectes

Chapitre 4 : Système de réduction de données

Annexes

Table des figures

Chapitre 1

Chapitre 2

Chapitre 3

Chapitre 4

Chapitre 5

Notations

b	Biais d'un modelé linéaire
$\mathcal{B}(c, R)$	La plus petite boule englobante de centre c et de rayon R
$\mathcal{D}, \mathcal{D}_{\Phi}$	Echantillon d'apprentissage.
D	Dimension de l'espace caractéristique : $\Phi(x) \in \mathbb{R}^D$.
$\mathcal{F}$	Ensemble de fonctions de $\mathcal{X}$ dans $\mathcal{Y}$.
$\mathcal{H}$	Ensemble des hyperplans.
$K(.,.)$	Fonction noyaux
l	Taille de l'échantillon.
$\mathcal{L}$	Fonction de Lagrange.
m	Classe parmi plusieurs classes (Multi-classes)
$\mathcal{M}$	Nombre de classe dans un échantillon d'apprentissage
$\mathcal{N}(x\|\mu, \Sigma)$	Gaussienne de moyen μ et covariance Σ.
$p(x\|m = c)$	Probabilité d'observer x sachant que la classe est c.
$\mathcal{P}$	Distribution de probabilité génératrice des observations de $\mathcal{D}$
$\mathcal{R}$	Esperance du risque, dit aussi espérance de perte ou de l'erreur.
$\mathcal{R}_{emp}$	Risque empirique.
s	Ensemble des vecteurs supports.
w	Vecteur de poids d'un modèle SVM
$\mathcal{X} \subset \mathbb{R}^d$	Domaine des variables explicatives.
$x \in \mathbb{R}^d$	Vecteur de d variables explicatives.
$\mathrm{x}^{\mathrm{T}}\mathrm{y}$	Produit scalaire entre x et y
$\mathcal{Y}$	L'ensemble de toutes les classes
y	Classe de x, dite aussi label ou étiquette.

α_i	Multiplicateur de Lagrange
$\boldsymbol{\xi} \in \mathbb{R}^l$	Vecteur des variables d'écart à la marge
Φ	Expansion correspondant à un noyau de Mercer.
$\|.\|$	Norme
$\mathbf{1}$	Vecteur dont toutes les composantes valent 1.
$\mathbf{0}$	Vecteur dont toutes les composantes valent 0.

Abréviations

CMS	Cepstral Mean Substraction.
DAG	Decision Acyclic Graph
DFT	Discrete Fourier Transform
EM	Expectation-Maximization.
GMM	Gaussian Mixture Models.
HMM	Hidden Markov Model.
IPA	International Phonetic Alphabet.
KKT	Karush-Kuhn-Tucker condition.
LDC	Linguistic Data Consortium.
LM	Language Model
LPC	Linear Prediction Coefficient.
LPCC	Linear Prediction Cepstral Coefficient.
MFCC	Mel Frequency Cepstral Coefficient.
NAP	Nuisance Attribute Projection.
PLP	Perceptual Linear Prediction.
SRM	Structural Risk Minimization.
PPRLM	Parallel Phoneme Recognition followed by Language Modeling.
PRLM	Phoneme Recognition followed by Language Modeling
SDC	Shifted Delta Cepstral coefficients
SVM	Support Vector Machine.
SV	Support Vector.
UBM	Universal Background Model.
VC	Vapnik Chervonenkis dimension

Introduction Générale

Le but générale du traitement de la parole est d'extraire automatiquement d'un signal numérique de la parole des informations de haut niveau, c'est-à-dire facilement interprétables par l'être humain. Ce domaine de recherche a connu une progression rapide avec de nombreuses applications à la clé, dont la partie la plus visible a été l'apparition de logiciels de dictée vocale.

Ces systèmes visent à transcrire le message linguistique véhiculé par la parole dans un code graphique qui ait un sens pour l'homme : les mots. Le signal de la parole contient d'autres types d'information que ceux portant sur le message, comme l'identité de la langue ou de l'individu qui a prononcé le message. Cette problématique est le centre d'intérêt de la reconnaissance ou l'identification d'un dialecte, domaine qui connaît plusieurs sous-branches selon le contexte applicatif visé. Une formulation du problème, qui a fait l'objet d'une grande partie des recherches plus récentes dans le domaine, est l'identification automatique des dialectes

Une langue est définie comme un regroupement de dialectes partageant un vocabulaire similaire et ayant des systèmes phonologiques et grammaticaux similaires. Il existerait actuellement plus de 6000 langues parlées et plus de 10000 dialectes. Par exemple, dans le monde arabe on dénombre plusieurs dialectes parmi ces derniers nous nous intéressons aux dialectes du Maghreb.

Le but de l'identification automatique des dialectes est de reconnaître le dialecte parlé par un locuteur inconnu, parmi un ensemble fini de dialectes, pour des énoncés d'une durée limitée (entre 3 et 50 secondes habituellement).

La recherche dans le domaine de l'identification des dialectes s'est intensifiée depuis la décennie précédente. Les raisons de cette émergence sont liées à de nombreux développements de nature essentiellement applicative :

- l'expansion des communications parlées dans un cadre multilingue,
- l'augmentation des performances des systèmes de reconnaissance automatique de la parole,
- l'enregistrement et la mise à disposition de la communauté scientifique de corpus de dialectes conséquents.

L'identification automatique des dialectes peut permettre de confirmer ou d'infirmer les théories linguistiques portant sur les différences entre les langues. Ces théories reposent souvent sur un nombre limité d'expériences ; certaines sont parfois plus basées sur des

intuitions que sur des faits réels. Il est important de procéder à des expérimentations sur de plus larges bases de données, tâche rendue possible par le traitement automatique de la parole. Les retombées sur la connaissance des langues sont nombreuses.

A titre d'exemple, signalons deux conséquences immédiates:

- Les typologies linguistiques pourront être comparées aux typologies automatiques.
- De manière corrélée, une distance linguistique entre différents dialectes peut émerger et conduire à préciser la distance entre les dialectes d'une même langue.

Parmi les informations disponibles pour identifier un dialecte, les informations phonétiques et phonotactiques (caractéristiques des sons d'un dialecte et règles d'enchainement de ces sons) sont les plus fréquemment utilisées. Une des principales raisons est la maîtrise technique des modèles phonétiques et phonotactiques issues de la reconnaissance automatique de la parole, domaine bénéficiant de plusieurs décennies d'investissement intellectuel. Pourtant, des expériences en perception montrent que l'oreille humaine permet d'identifier les langues et dialectes à partir de leur seule prosodie mettant ainsi en avant le pouvoir discriminant de ces traits et l'intérêt manifeste de leur exploitation dans des systèmes d'identification automatique des dialectes.

Notre contribution consiste en la réduction de données caractéristiques de la parole afin que les séquences indésirables ne soient pas prises en compte par un système d'identification. Pour cela, nous avons pris une focalisation sur l'équivalence entre la formulation L2-SVM et la plus petite boule englobante (MEB - Minimal Enclosing Ball), et avons défini deux algorithmes moyennant les multi-classes SVM. Pour le développement du système d'identification de dialectes, nous avons intégré un module de réduction de données sur un système d'identification automatique de langues basé sur les modèles de Mélanges Gaussiens (GMM-Gaussian Mixture Model). Les résultats de nos expériences menées sur le système développé étaient très encourageants et satisfaisants.

Le mémoire présenté est structuré en cinq chapitres :

- **Chapitre 1** : Parole et Dialectes du Maghreb.

  Dans ce chapitre, est présentée une étude détaillée de l'appareil phonatoire humain pour permettre de connaitre toute les structures du système de production des sons. Cette base est nécessaire pour l'étude et la conception des systèmes de traitement de la parole. Au niveau linguistique, il est nécessaire de définir les propriétés fondamentales de la parole afin de mieux cerner les caractéristiques recueillies du signal de la parole pour tout

traitement sur un système quelconque de traitement de la parole. Sur le deuxième volet de ce chapitre, est donné un aperçu général sur la langue arabe standard et les dialectes du Maghreb ainsi que la relation qui existe entre eux.

- **Chapitre 2 :** Les Machines à Vecteurs Supports (SVMs).

Ce chapitre traite des Machines à Vecteurs Supports qui sont de puissants classifieurs inspirés par le principe de minimisation du risque structurel, et ils ont prouvé leur efficacité pour diverses tâches de classification. Ce chapitre détaille formellement les différentes hypothèses sur des types de donnée et des structures de classification. Par suite, le traitement des multi-classe SVM a montré leur capacité à résoudre des problèmes où les sorties définissent un ensemble d'éléments structuré.

- **Chapitre 3 :** Identification Automatique des Langues (IAL).

Ce chapitre présente un background sur les systèmes d'Identification Automatique des Langues et leurs conceptions structurelles à travers des études existantes dans à la littérature de recherche de ce domaine. Cela, a conduit à définir deux approches de base pour la classification, génératives et discriminante. La première regroupe des méthodes qui utilisent les données d'apprentissage pour modéliser les densités de probabilité et la deuxième regroupe une variété de méthodes statistiques qui utilisent les données d'apprentissage pour construire directement une correspondance entre les entrées et les sorties. Dans cette partie, il est mis en valeur les propriétés fondamentales de la parole décrites dans le premier chapitre, pour donner un détail sur les informations au niveau locution afin d'orienter toute conception d'un système d'identification des langues. On dénombre 4 types d'information : acoustique, phonétique, phono-tactique et prosodique. Ceci a permis de décrire les différents systèmes d'identification des langues ainsi que leurs composants.

- **Chapitre 4 :** Système de réduction de données.

Dans ce chapitre, nous avons établi un développement spécial sur la formulation de la norme L2-SVM et la formulation de la plus petite boule englobante (**M**inimale **E**nclosing **B**all-**MEB**) qui démontre la potentialité de faire une équivalence entre eux. L'exploitation de cette équivalence nous a conduits à développer une méthode de réduction des données moyennant les multi-classes SVM. Ceci nous a permis de concevoir deux algorithmes de réduction de données qui peuvent jouer le rôle d'un filtre de données où celles indésirables sont écartées.

- **Chapitre 5 :** Développement de systèmes d'identification de dialectes basé sur les Modèles de Mélanges de lois Gaussiennes.

Dans ce chapitre, nous avons exploité la réduction des données qui représentent les caractéristiques de la parole pour concevoir deux systèmes d'identification de dialecte basé sur les Modèles de Mélanges de lois Gaussiennes. Dans le premier système d'identification, la réduction est faite sur les caractéristiques acoustiques des signaux de la parole avant qu'elles soient modélisées par les Modèles de Mélanges de lois Gaussiennes. Tandis que pour le deuxième système d'identification, la réduction des données est opérée au niveau des modèles dans le modèle du monde (**U**niversal **B**ackground **M**odel-**UBM**). Ceci a été suivi par une expérimentation sur les deux systèmes.

Enfin, pour terminer une synthèse du travail de recherche réalisé est donnée en montrant les avantages d'une telle proposition, en décrivant ses points positifs et en suggérant plusieurs extensions dans les travaux futurs.

Partie 1

Parole, Dialectes et Identification Automatique des Langues: État de l'art

Chapitre 1

La parole et les dialectes du Maghreb

1 La production de la parole et ses propriétés

1.1 La physiologie de l'appareil phonatoire

Connaître les paramètres caractérisant un locuteur est nécessaire pour tout système de traitement de la parole. Pour cela, nous devons avoir une bonne compréhension du processus de production de la parole. Ce dernier est un mécanisme très complexe qui repose sur une interaction entre les systèmes neurologique et physiologique de l'être humain. La parole commence par une activité neurologique. Après que soient survenues l'idée et la volonté de parler, le cerveau dirige les opérations relatives à la mise en action des organes phonatoires. Le fonctionnement de ces organes est bien, quant à lui, de nature physiologique.

Une grande quantité d'organes et de muscles entrent en jeu dans la production des sons des langues naturelles qui définissent un appareil phonatoire, se divisant en trois parties dont chacune définit un groupe d'organes qui assument les fonctions essentielles suivantes dans l'acte de parole ou de phonation (cf. Fig. 1.1):

- ***Partie sub-glottique***, composée de l'appareil respiratoire (diaphragme, poumons, trachées), est une soufflerie qui fournit l'énergie et la quantité d'air nécessaire à la phonation en insufflant l'air vers la partie glottique.
- ***Partie glottique***, composée du larynx, est un organe vibrant où naît le son. Il contient les cordes vocales (replis tendus horizontalement qui, sous l'effet des muscles, jouent un rôle de valve vis-à-vis de l'air des poumons libérant ainsi un flux d'air vers la partie supra-glottique).
- ***Partie supra-glottique***, composée du conduit vocal, est formée des cavités orales (pharyngienne et buccale) à géométrie variable en fonction des éléments articulateurs (langue, mâchoire inférieure, lèvres) et des cavités nasales à géométrie fixe pouvant être couplées aux cavités orales par abaissement du voile du palais où s'effectue l'articulation proprement dite par les changements de forme du tractus vocal.

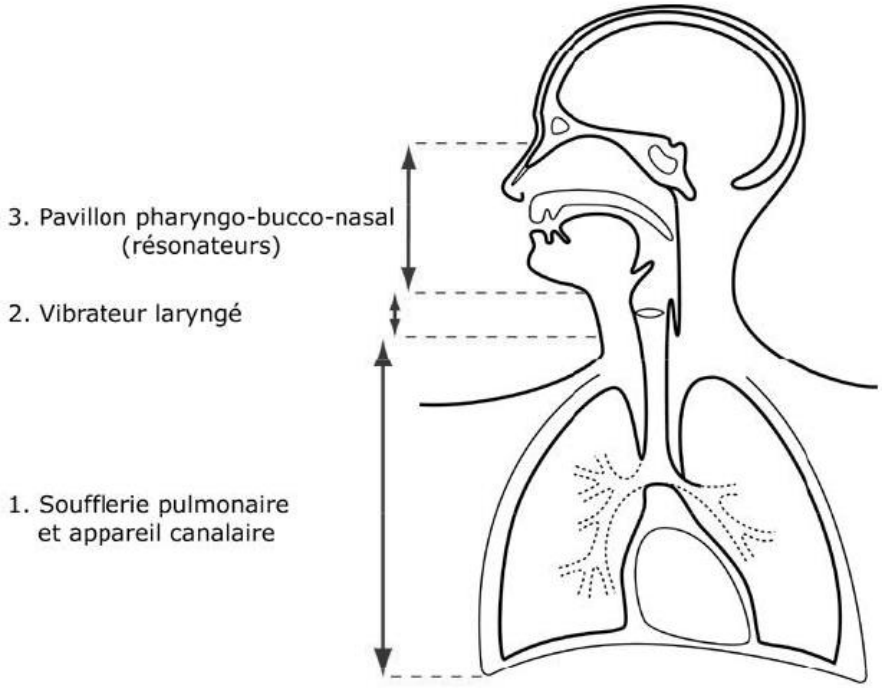


Fig. 1.1: Schéma de l'appareil phonatoire.

Aussi, nous pouvons assimiler l'appareil phonatoire humain à un système composé d'un ensemble de tubes (cf. Fig. 1.2) permettant de donner un aperçu simulé de la production de la parole.

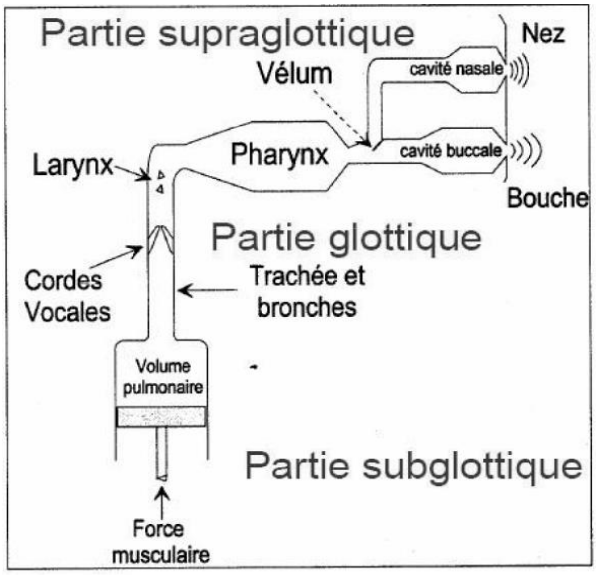


Fig. 1.2: L'appareil phonatoire humain schématisé.

1.2 La physiologie de la production de la parole

La production de la parole est composée de deux fonctions mécaniques de base : la phonation et l'articulation.

1.2.1 La phonation

La phonation est la production du signal acoustique par vibration des cordes vocales. La fréquence fondamentale moyenne F_0 de vibration des cordes vocales est située entre 140Hz et 240Hz pour les femmes, entre 100Hz et 150Hz pour les hommes. La mélodie de la voix résulte de cette vibration et se traduit phonétiquement par l'évolution de la fréquence fondamentale F_0.

Les poumons jouent un rôle de soufflerie qui propulse une colonne d'air ascendante dans la trachée artère. La colonne d'air pulsée traverse le larynx, qui constitue l'organe phonateur. L'espace entre les cordes vocales est appelé la glotte (cf. Fig. 1.2). La glotte s'ouvre lors de l'inspiration et se referme lors de la phonation permettant aux cordes vocales de vibrer sous l'effet de la dépression de part et d'autre de l'espace glottique ; ce qui génère un flux sonore appelé voisement. La production du voisement implique que les cordes vocales soient entièrement accolées et mises en vibration par le flux d'air ventilé par les poumons et véhiculé dans la trachée. Ainsi, la théorie acoustique de production de la parole distingue le mode «voisé», lorsque les cordes vocales vibrent périodiquement, et le mode «non voisé», lorsqu'elles ne vibrent pas. En réalité, le voisement se combine souvent avec une émission de bruit lors de la phonation (bruit d'aspiration, bruit de friction, bruit structurel, etc.).

Un exemple de signal de parole correspondant à la prononciation du mot **/sa/** est donnée à la figure 1.3. Le son du mot **/sa/** est représenté dans le domaine temporel : la première partie (de 0 à 80 ms) est non voisée, c'est un signal non périodique de faible énergie, quant à la dernière partie, elle représente un signal quasi-périodique avec une énergie plus grande, et est donc voisée.

1.2.2 L’articulation

L'air mis ou non en vibration poursuit son chemin à travers le conduit vocal et se propage ensuite dans l’atmosphère. La forme de ce conduit est déterminée par la position des articulateurs tels que la langue, la mâchoire, les lèvres ou le voile du palais, détermine le

timbre des différents sons de la parole. Le conduit vocal est ainsi considéré comme un filtre pour les différentes sources de production de parole telles que les vibrations des cordes vocales ou les turbulences engendrées par le passage de l'air à travers les constrictions du conduit vocal.

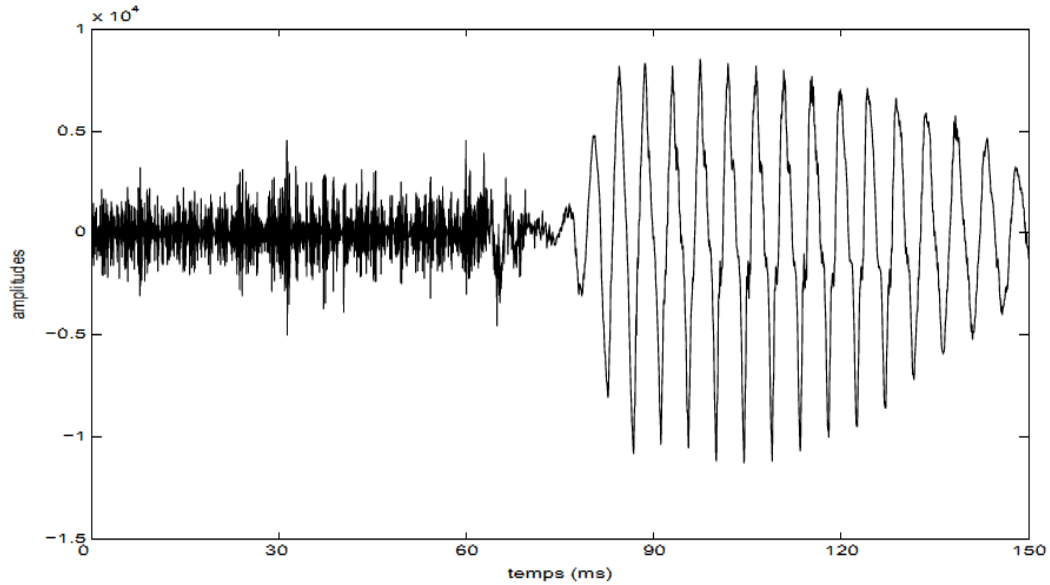


Fig. 1.3: Exemple de signal de parole (son du mot **/sa/**).

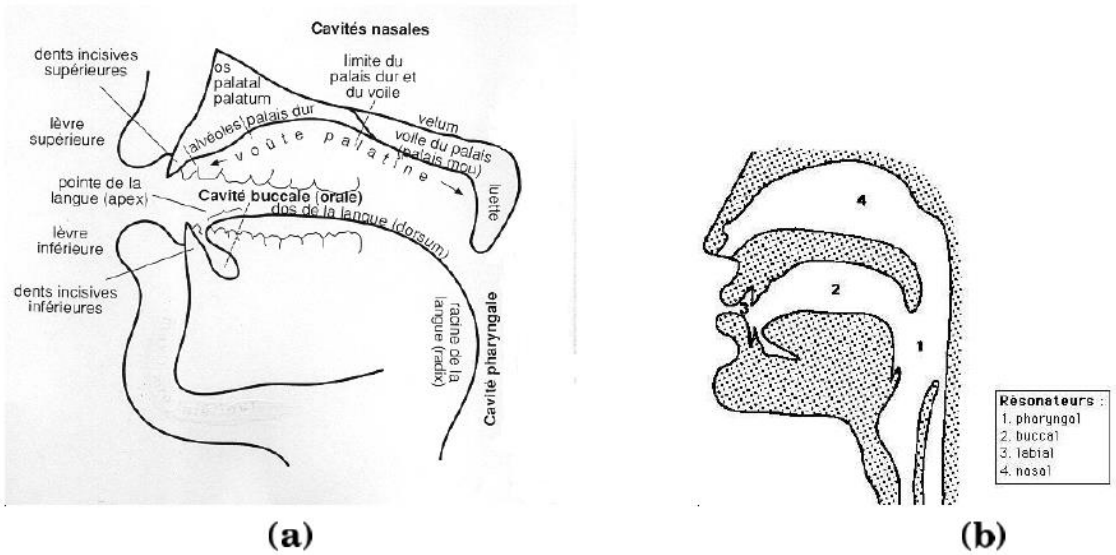


Fig. 1.4: (a) Les articulateurs de la parole, (b) Les cavités supra-glottiques [Henr (01)].

L'articulation inclut la modulation du signal acoustique par les articulateurs (principalement les lèvres, la langue et le palais) et la résonance de ce signal dans les cavités supra-glottiques (le pharynx, la bouche, les fosses nasales et la cavité labiale) (cf. Fig. 1.4). Les deux

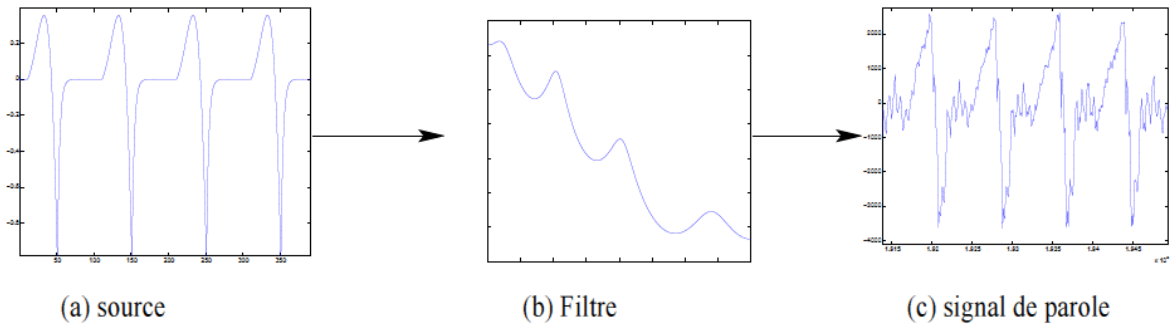
premières sont toujours sollicitées pour l'articulation des sons de la parole alors que les cavités nasale et labiale n'interviennent que pour la réalisation de sons spécifiques. Si les lèvres sont projetées vers l'avant et arrondies, un résonateur se forme effectivement à la sortie du canal buccal, le résonateur labial. Si au contraire, elles sont appliquées contre les dents, le résonateur labial ne se forme pas.

Les phonèmes ainsi produits, reflètent des unités distinctives minimales, qui peuvent être des consonnes sourdes, des consonnes voisées, ou des voyelles. Nous pouvons exciter le résonateur complexe de l'appareil phonatoire de différentes manières, le mode d'excitation étant fonction du phonème à produire.

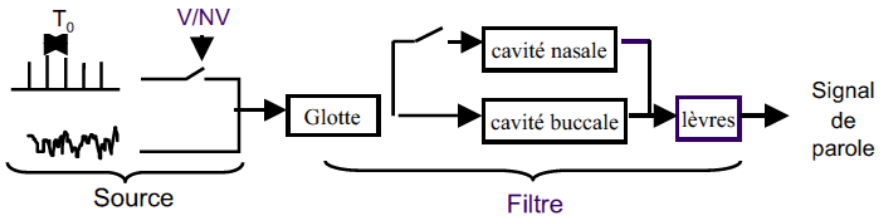
1.2.3 Le modèle de production de la parole

Le processus de production de la parole peut être représenté par le modèle source-filtre (cf. Fig. 1.5 (b)). Le signal de parole est modélisé comme la sortie d'un filtre linéaire variant dans le temps, qui simule les caractéristiques spectrales de la fonction de transfert du conduit vocal, excité par un signal source qui reflète l'activité des cordes vocales dans les zones voisées et le bruit de friction dans les zones non voisées. Quoique simpliste, cette représentation est capable de décrire la majorité des phénomènes de la parole qui a été à la base de nombreux codeurs et synthétiseurs de parole.

Une approximation classiquement employée consiste à considérer que le signal de source est constitué d'impulsions générées aux instants de fermeture de la glotte auxquelles s'ajoute un bruit blanc. Dans un tel modèle (cf. Fig. 1.5 (a)), le spectre de la partie "***Filtre***", appelée aussi enveloppe spectrale, est composée du spectre du filtre décrivant le conduit vocal auquel s'ajoute la partie lisse du spectre glottique. Suivant le modèle du signal glottique utilisé, cette partie lisse du spectre du signal glottique peut être modélisée par un modèle Auto-Regressive d'ordre 2 ou 4 [Fant (85), Klat (90)]. Certaines caractéristiques de ce modèle Auto-Regressive telles que la position du formant glottique et la pente spectrale sont d'ailleurs utilisées pour caractériser la qualité vocale du signal de parole [Henr (01)]. La partie "***Filtre***" ainsi modélisée est porteuse des informations relatives à "***l'empreinte***" vocale d'un locuteur ; c'est pourquoi elle est également dénommée timbre.



(a) Modèle-source filtre équivalent (Sons voisés)



(b) Modèle source-filtre

Fig. 1.5: (a) La production de la parole – (b) Le modèle source-filtre.

1.3 Les propriétés fondamentales de la parole

Le décodage d'un signal vocal en unités phonétiques étant fondé sur des connaissances fondamentales concernant la parole et son environnement (bruits parasites) ; il est important de bien connaître ces notions. De nombreuses descriptions acoustiques des phonèmes existent dans la littérature [Call (89)].

1.3.1 L'introduction du spectrogramme

L'information utile contenue dans le signal de parole étant véhiculée à la fois par les fréquences émises et par la structure temporelle du signal. Une représentation adéquate doit alors combiner les avantages de la représentation temporelle du signal de la représentation fréquentielle.

Une visualisation classiquement utilisée en traitement de la parole est le spectrogramme (le spectrogramme est obtenu par une transformée de Fourier discrète sur le signal échantillonné représenté d'une fenêtre glissante - généralement une fenêtre de Hamming de taille fixe -)

[Call (89)] qui est une représentation de l'énergie du signal en fonction du temps et de la fréquence : l'amplitude de l'énergie est donnée par le niveau de gris de chaque point.

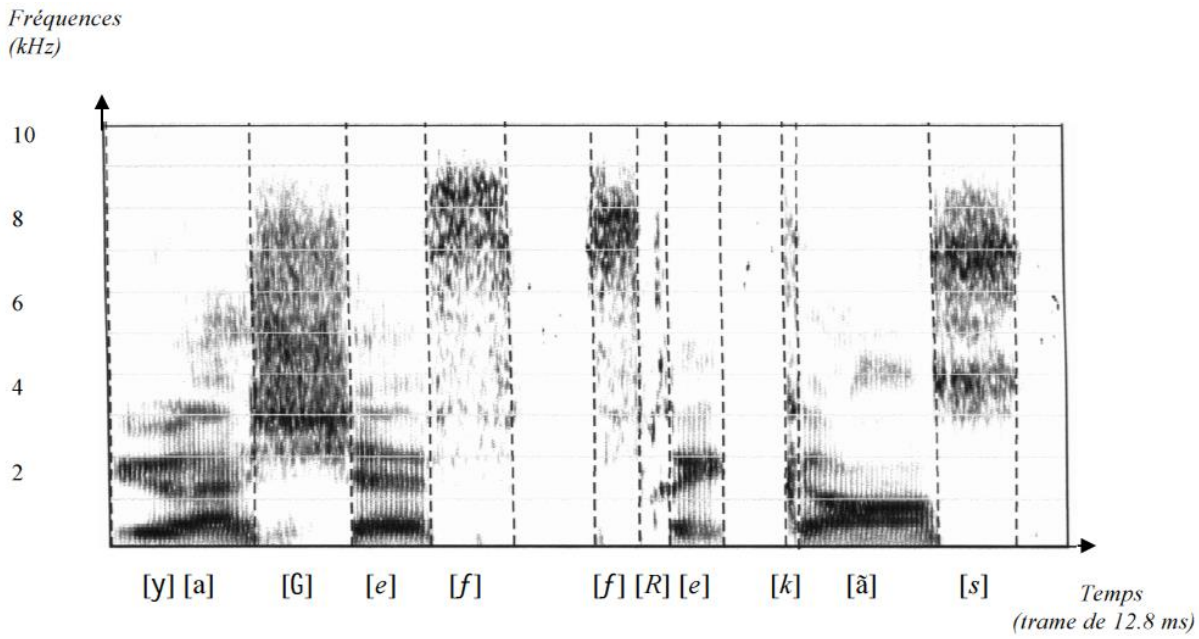


Fig. 1.6: Spectrogramme du signal de parole du mot énoncé « ***Uhf Fréquence*** ». Signal enregistré dans un environnement de laboratoire. Les zones sombres correspondent donc aux plages de fréquences plus fortement énergétiques et les formants apparaissent comme des bandes horizontales plus sombres.

Cette visualisation met en valeur les propriétés acoustiques et phonétiques des sons (durée, concentration et localisation de l'énergie, …). Une étude visuelle du spectrogramme facilitera l'identification des traits acoustiques des phonèmes.

1.3.2 La variabilité du signal de parole et ses perturbations

L'analyse du signal de parole est difficile. D'abord, à cause de la redondance en information du signal vocal, nécessaire pour résister aux perturbations du milieu ambiant qu'il faut réduire dans des proportions importantes et de façon pertinente. Ensuite, à cause de l'importante variabilité entre différentes réalisations d'un même phonème provoquée par :

- Les informations caractéristiques du locuteur contenues dans le signal vocal.
- L'influence contextuelle des phonèmes.
- Les influences extérieures au signal de parole.

Nous détaillons maintenant chacun de ces trois points qui sont tous très pénalisants dans le traitement automatique de parole.

1- Les variabilités dues aux informations caractéristiques du locuteur et portées par le signal de parole concernant [Junq (92)].

- Les *variabilités inter-locuteurs* : il existe un fort contraste entre les prononciations d'un même phonème provenant de locuteurs différents. Ceci peut être dû à la différence d'âge, de sexe, d'accent, de vitesse d'articulation, etc.
- Les *variabilités intra-locuteurs* : l'évolution de la voix d'un locuteur en fonction de la fatigue, de l'émotion, du stress et de l'attention, peut provoquer des différences importantes de prononciation d'un même phonème.

2- Les prononciations d'un même phonème dans des contextes différents, peuvent être extrêmement variables car la nature continue de la parole provoque les phénomènes suivants :

- Les *effets de coarticulation* : ce sont des phénomènes qui apparaissent aux frontières des phonèmes. Cela se traduit souvent par l'abrègement, voire la disparition de certains phénomènes ou par l'introduction d'une consonne de liaison.
- *L'influence contextuelle* : les phonèmes sont très influençables par leur contexte gauche et droit de la partie supra-glottique de l'appareil phonatoire. Cela se manifeste par une modification des réalisations acoustiques des sons.
- Les phonèmes sont plus ou moins *accentués* selon leur positions dans le mot.

3- Enfin, il faut considérer les perturbations de la parole dues

- Aux *difficultés techniques* : bruit ambiant, bruit de ligne téléphonique, etc.
- Au *pré-traitement du signal* : conversion analogique-numérique, filtrage, pré-amplification, etc.
- Au *microphone* : variations de position, type de téléphone, etc.

Mais, l'influence extérieure la plus pénalisante est le bruit parasite qui affecte le signal vocal. Dans ce cas, nous avons des bruits ressemblant le plus souvent à des consonnes fricatives et sont:

- *Des bruits de frictions* : papier sur la table, frottement contre le micro du téléphone, etc.
- *Des bruits de respiration.*

Le bruit de respiration

- occupe la bande spectrale de la parole,
- peut aléatoirement se superposer ou prolonger un mot,
- apparaît avec un niveau d'énergie élevé.

Les bruits de frictions

- occupent la bande spectrale de la parole avec une énergie située en hautes fréquences,
- sont perceptibles essentiellement dans les zones de silence,
- apparaissent avec un niveau d'énergie qui peut être élevé.

Même si un être humain est capable de distinguer très facilement ces types de bruit de la parole, il n'en va pas de même d'une machine. En effet, le comportement spectral des bruits de frictions et surtout de la respiration est comparable à celui de sons de la parole comme les consonnes fricatives qui ne contiennent aucune périodicité et dont l'énergie est située en haute fréquence.

1.4 Les sons et phonétiques

Les différents sons de la parole sont regroupés en classes phonétiques en fonction de leurs caractéristiques principales. Ces caractéristiques représentent des différences qui sont suffisamment importantes. Dans un spectrogramme, les différents sons sont visibles selon leur classe respective, de durée minimale de quelque milli-secondes sans aucune écoute de la phrase correspondante (cf. Fig. 1.6).

1.4.1 Les voyelles

Cette classe se caractérise principalement par le voisement qui crée des formants. Ces formants, qui sont des zones fréquentielles de forte énergie, correspondent à une résonance dans le conduit vocal de la fréquence fondamentale produite par les cordes vocales. Ces formants sont principalement ceux en basses fréquences qui peuvent se distinguer dans le spectre (cf. Fig. 1.5 (a)).

1.4.2 Les occlusives

Les phonèmes de cette classe se caractérisent par la fermeture du conduit vocal précédant un brusque relâchement. Dans la production de parole, nous avons deux parties successives: un silence correspondant à l'occlusion effective suivi d'une explosion au moment du relâchement. Les occlusives peuvent être voisées à la manière des voyelles ou sourdes ; c'est à dire non voisées.

1.4.3 Les fricatives

Dans cette classe, sont regroupés les sons produits par la friction de l'air dans le conduit vocal lorsque celui-ci est rétréci au niveau des lèvres, des dents ou de la langue. Cette friction produit un bruit de hautes fréquences et peut être voisée ou sourde.

1.4.4 Les semi-voyelles

Elles ont la structure acoustique des voyelles mais ne peuvent en jouer le rôle car elles ne sont que des transitions vers d'autres voyelles qui sont les véritables noyaux syllabiques.

1.4.5 Les liquides

Les liquides sont très similaires aux voyelles et aux semi-consonnes mais leur durée et leur énergie sont généralement plus faibles. Elles sont sonores.

1.4.6 Les nasales

Les phonèmes sont formés par passage de l'air dans le conduit vocal depuis les cordes vocales. Ce passage exclut normalement toute connexion du conduit normal, le conduit buccal, avec le conduit nasal. Les nasales sont produites de la même manière que les occlusives nasales mais l'air n'est pas comprimé dans le conduit vocal. Les nasales sont voisées.

1.4.7 Les diphtongues

Cette classe phonétique est propre à l'anglo-saxon et l'arabe. Les phonèmes qui composent cette classe se caractérisent par deux états formantiques stables ainsi que par la transition entre ces deux états.

1.4.8 Les emphatiques

L'arabe connaît une série de consonnes complexes, dites « *emphatiques* », basées sur un recul de la racine de la langue (créant ainsi une augmentation du volume de la cavité buccale) vers le fond de la bouche créant une prononciation simultanée du phonème au niveau du pharynx. On note même une certaine vélarisation ou prononciation simultanée du phonème au niveau du palais mou, le velum ou voile du palais.

2. L’arabe standard et les dialectes du Maghreb

2.1 La langue arabe standard

Parlé depuis le IIe siècle, la langue arabe est une des plus anciennes langues existantes au monde. L'arabe est la langue officielle de 22 pays et est très largement parlé à travers le monde. C'est également une langue officielle d'organisations telle que l'ONU.

La langue arabe appartient au groupe des langues sémitiques et c'est la langue du *Qur’an* (le livre saint islamique). Même si elle est souvent associée à l'Islam, la plupart des musulmans ne parlent pas Arabe. Cependant, elle a une influence grandissante dans les pays islamiques où les résidents utilisent beaucoup de vocabulaire emprunté à l'arabe.

La langue arabe peut sembler être une langue très difficile à maitriser quand vous voyez un texte écrit en arabe. Cependant, une fois que vous maitrisez l'alphabet, l'écriture et la prononciation, vous vous rendez compte que la construction de cette langue est au final très logique.

2.1.1 La caractéristique de la langue arabe standard

Les mots arabes des mêmes familles ont des racines communes. Une racine sémitique est une séquence de trois consonnes reliées à un concept. Pour créer un mot, on doit rajouter des voyelles et des consonnes qui ne sont pas des racines.

La langue arabe est composée de 28 consonnes et trois voyelles. Une fois que l'alphabet est maitrisé, sa lecture ne devrait pas être trop difficile.

Les consonnes sont divisées en quatre parties : sourdes, sonores, emphatiques et nasales. Aussi, voyelles et consonnes peuvent se présenter sous forme de géminées.

Table 1 montre les lettres arabes (alphabet) et leurs correspondances avec une description phonétique de chaque phonème incluant leur place d’articulation. Les symboles utilisés pour la représentation de cet alphabet sont d’IPA (International Phonetic Alphabet), LDC (Language Data Consortium), et Worldbet.

Table 2.1: Alphabet de la langue Arabe.

Letter	LDC Symbols	IPA Symbols	Worldbet Symbols	Description
ء (Hamza)		/ʔ/		Elle est intégrée dans une voyelle
ب (Beh)	/b/	/b/	[b]	Stop voisé bilabial plosive
ت (Teh)	/t/	/ts/	[ts]	Affricative dentale non voisé
ث (Theh)	/th/	/θ/	[T]	Fricative dentale non voisé
ج (Jeem)	/j/	/ʒ/	[Z]	Fricative pelato-alvéolaire voisé
ح (Hah)	/H/	/ħ/	[H]	Fricative glottal non voisé
خ (Khah)	/x/	/x/	[K]	Fricative velar non voisé
د (Dal)	/d/	/d/	[d]	Plosive dental voisé
ذ (Thal)	/Z/	/ð/	[D]	Fricative inter-dental voisé
ر (Reh)	/r/	/r/	[r]	Palato-alvéolaire voisé é
ز (Zain)	/z/	/z/	[z]	Fricative alvéolaire voisé
س (Seen)	/s/	/s/	[s]	Fricative alvéolaire non voisé
ش (Sheen)	/sh/	/ʃ/	[S]	Fricative palato-alvéolaire non voisé
ص (Sad)	/S/	/sˤ/	[sr]	Fricative alvéolaire vélarisé non voisé
ض (Dad)	/D/	/dˤ/	[dd]	Stop alvéolaire vélarisé voisé
ط (Tah)	/T/	/tˤ/	[tt]	Stop alvéolaire plosive non voisé
ظ (Thah)	/TH/	/ðˤ/		Fricative inter-dental vélarisé voisé
ع (Ain)	/C/	/ʔˤ/	[!]	Fricative pharyngal voisé
غ (Ghain)	/G/	/ɣ/	[G]	Fricative vélaire voisé
ف (Feh)	/f/	/f/	[f]	Fricative labiodental non voisé
ق (Qaf)	/q/	/q/	[q]	Stop uvulaire non voisé
ك (Kaf)	/k/	/k/	[k]	Stop vélaire non voisé
ل (Lam)	/l/	/l/	[l]	Latéral alvéolaire voisé
م (Meem)	/m/	/m/	[m]	Nasal bilabial voisé
ن (Noon)	/n/	/n/	[n]	Nasal alvéolaire voisé
هـ (Heh)	/h/	/h/	[h]	Fricative glottal non voisé
و (Waw)	/w/	/w/	[w]	Approximatif bilabial voisé
ي (Yeh)	/y/	/j/	[j]	Approximatif palatal voisé

Définition des différentes abréviations

Vélaire : Se dit des voyelles ou des consonnes articulées près du voile du palais.

Uvulaire : consonne dont le lieu d'articulation se situe à l'extrémité postérieure du palais mou au niveau de la luette.

Pharyngal : Se dit d'une consonne articulée en rapprochant la racine de la langue et la paroi arrière du pharynx.

Glottal : Emis par la glotte.

Alvéole : Consonne articulée avec la pointe de la langue au niveau des alvéoles des dents.

Dentale : Consonne dentale que l'on prononce en appuyant la langue sur les dents.

Bilabiale : Consonne labiale réalisée avec la participation des deux lèvres.

labiodentale : Se dit d'une consonne réalisée avec la lèvre inférieure et les incisives supérieures.

Palatale : Se dit d'une voyelle ou d'une consonne qui a son point d'articulation situé dans la région du palais dur .

Dans la langue arabe, il y a des consonnes que nous ne pouvons pas les trouver dans d'autres langues qui diffèrent dans leurs intonations [Gibb (02), Kirc (03)]. Parmi ces consonnes, nous avons deux classes: pharyngal et emphatique. Pour la première classe, nous notons quatre fricatives, /ʔˤ/(ع), /ħ/(ح), /ɣ/(غ), /x/(خ) et un stop /q/(ق). Pour la seconde classe, nous notons deux plosives, /dˤ/(ض) et /tˤ/(ط) et deux fricatives, /sˤ/(ص) et /ðˤ/(ظ).

2.2 Les dialectes arabes

Les dialectes arabes sont généralement subdivisés en deux grandes zones dialectales : la zone orientale qui regroupe les pays du Moyen-Orient et la zone occidentale ou maghrébine [Cohe (73), Emba (08)]. L'intercompréhension entre les différent dialectes varie en fonction de l'éloignement des zones dialectales, et les locuteurs (lettrés généralement) de pays différents peuvent recourir à l'arabe standard pour communiquer entre eux. L'arabe standard est considéré comme le style formel de la langue. C'est une langue qu'apprennent les enfants à l'école comme une langue étrangère dans sa structure et souvent aussi dans son vocabulaire. Il résulte de cela que les productions en arabe standard varient en fonction de l'origine dialectale des locuteurs, et il n'est pas rare que les auditeurs natifs affirment pouvoir identifier le dialecte ou du moins la zone dialectale de l'origine du locuteur parlant en arabe standard.

Les dialectes sont souvent décrits par rapport à l'arabe standard, et c'est généralement l'arabe standard qui est employé pour en présenter les règles (comme par dérivations). Cependant, la question des dialectes est l'une de celles qui taraudent le plus l'esprit quand on s'intéresse à la langue arabe et cette question est parfois un défi pour les Arabophones eux-mêmes.

La distinction de parole que l'on opère entre arabe standard d'une part et arabe dialectal d'autre part, bien que commode pour l'analyse, risque toutefois d'être singulièrement réductrice si on ne l'associe pas à cette situation de continuum linguistique. Cette distinction en effet ne peut en aucun cas conduire à considérer que ces deux pôles de variétés constituent des langues distinctes et autonomes. Partout où cette langue est pratiquée, en réalité on trouve entre eux des éléments constitutifs de façon indissociable.

Il existe plusieurs dialectes arabes en nombre mesuré, mais tout même non-négligeable. On pourrait simplifier leurs répartitions en assignant à chacun une zone géographique. Cela donnerait un début d'image de la réalité, mais une image peu trop simpliste tout de même. Dans les faits, il n'y a pas de frontières nettes entre les dialectes, mais un passage progressif entre eux. Les dialectes peuvent même varier légèrement d'une ville à l'autre. L'existence de cette variation est progressive et simple à comprendre : elle correspond à une réalité humaine et sociale, elle correspond au contact, amitiés et affinités nouées dans l'environnement plus ou moins immédiat de chacun(e). Chacun(e) s'imprègne de la réalité et de la langue apportée par l'autre, et il en naît des fluctuations progressives à mesure que l'on se déplace d'un espace géographique à l'autre.

2.3 Les dialectes Maghrébins

L'introduction de l'arabe au Maghreb date à partir du VII^{e} siècle avec les troupes des conquérants arabes, Oqba Ibn Nafi (l'an 640) puis Moussa Ibn Nusayr (l'an 711). La densité de la présence de la langue arabe fut vraisemblablement négligeable. Son renforcement s'est fait par des vagues successives de tribus arabes venant de la péninsule arabique, les Banû Hilal, Banû Ma'qîl et Banû Sulaym. Enfin, l'expulsion massive des Andalous par les espagnole vers les rives d'Afrique du Nord au XV^{e} siècle consolide la présence de l'arabe dans les centres urbains des villes du Maghreb. L'immigration des Andalous a permis d'accentuer de manière durable le processus d'arabisation.

Bien qu'il existe un dialecte Marocain, un dialecte Algérien et un dialecte Tunisien, on parle indistinctement du dialecte Maghrébin (même si on devrait plutôt parler de dialectes Maghrébins, au pluriel), ce qui rappelle encore une fois que le dialecte est une notion très souple et adaptative et qu'il ne s'agit donc pas d'une langue caractérisée au sens formel du terme.

Le dialecte parlé au Maghreb est connu sous le nom de *Darija* qui signifie dialecte pour la distinguer de l'arabe standard. *Darija* est mutuellement parlé et compris dans les pays du Maghreb, spécialement au Maroc, l'Algérie et la Tunisie et non compréhensible dans le moyen orient. Pour la communication entre ces pays, l'arabe standard est utilisé.

Au Maghreb, il y a 75% des habitants parlent *Darija* et 25 % ont une deuxième langue appelé *Tamazight*, qui est une langue berbère.

2.3.1 La relation entre les dialectes du Maghreb et l'Arabe standard

Les dialectes du Maghreb sont tous issus de l'Arabe standard d'une certaine manière. Mais on pourrait, d'un point de vue linguistique, considérer les choses dans la réciproque, et y voir que l'arabe standard a repris des choses d'un peu tous les dialectes. Aussi surprenant que cela puisse paraître, ces deux visions diamétralement opposées sont toutes deux presque aussi vraies.

Les dialectes du Maghreb sont une version simplifiée de l'Arabe standard, mais ces versions simplifiées diffèrent elles-mêmes l'une de l'autre. Les simplifications et variations opérées par ces dialectes se trouvent surtout dans la grammaire, la conjugaison et les déclinaisons. Mais, les variations se trouvent aussi dans le vocabulaire. Les dialectes du Maghreb ont beaucoup de vocabulaire en commun avec l'Arabe standard. Les mots de dialectes du Maghreb qui diffèrent de l'Arabe standard sont les mots les plus souvent utilisés et sont surtout des mots fonctionnels: mots interrogatifs, mots superlatifs, mots comparatifs, pronoms, noms des chiffres et nombres, nom des heures, etc.

Une autre variation, qui peut surprendre au début, est la troncation. C'est un phénomène fréquent dans les dialectes du Maghreb surtout : les mots sont raccourcis.

Les mots peu fréquents sont les mêmes dans les dialectes du Maghreb et en Arabe standard. Ceci correspond à une règle d'usure et d'ergonomie tout à fait prévisible.

2.3.2 Le vocabulaire emprunté des dialectes du Maghreb

Les dialectes du Maghreb ne sont pas seulement que des versions simplifiées de l'Arabe standard: ils intègrent des mots étrangers à l'arabe tels que des mots provenant d'anciennes tribus non-arabophones ou berbères et des mots provenant de certain langues de colonisateurs qui ont conquis le Maghreb tels que les Espagnoles, les Turks, les Français.

Exemple de vocabulaire emprunté dans le dialecte du Maghreb

Lalla : madam, *Tamssumant*: effort, *chlaghem*: moustache, *Fertas*: chauve, *rambwa*: rondpoint, *bartma*: appartement, *blassa*: place, *Kar*: bus, *moutou*r: moteur, *courda*: cuerda (la corde), *Sabato*: chaussure.

2.3.3 Les différences de prononciation au Maghreb

La prononciation de l'Arabe dialectale du Maghreb peut changer beaucoup pour certaines lettres. La *fatha* se transforme souvent en *kasra*, ou encore parfois la *fatha* et la *kasra* se confondent. La longueur des voyelles peut également varier, et certaines voyelles longues peuvent devenir brèves. Ceci soulignera l'importance de bien apprendre les racines car, dans de tels cas, les racines permettent de reconnaître des mots même lorsqu'ils ont été transformés par un dialecte.

2.3.4 Les dialectes du Maghreb et l'écrit

On associe facilement les dialectes à la seule langue parlée, et il est parfois même prétendu que les dialectes ne s'écrivent normalement pas. Même si les dialectes ne sont pas autant formalisés que l'Arabe courant, ils peuvent tout de même s'écrire ne serait-ce que sur la base de leur phonétique. L'écriture des dialectes est même tout à fait courante dans la poésie, dans la chanson et même dans le roman (mais surtout dans la poésie et la chanson). On comprend bien qu'une chanson en dialecte algérien ne se transmet pas qu'oralement, et qu'elle est écrite de la même façon que toutes les autres chansons.

2.4 Conclusion

A partir de l'analyse précédente, il n'est pas évident d'identifier les paramètres acoustiques qui jouent un rôle décisif sur la caractérisation de l'identité vocale. Cette dernière est un amalgame entre divers paramètres acoustiques dont le degré et l'ordre d'importance diffèrent d'un individu à l'autre [Kawa (95)].

Les caractéristiques du signal de parole sont analysées à l'échelle milli-seconde. Les descripteurs acoustiques incluent les formants et leur largeur de bande, la pente spectrale, la fréquence fondamentale et l'énergie. Ces derniers dépendent principalement des propriétés physiologiques et physiques des organes de la parole et également de l'état émotionnel du locuteur [Klat (90)].

Les dialectes du Maghreb ne se distinguent pas de manière tranchée, et bien que les dialectes éloignés soient assez différents, il existe des continuums passant de l'un à l'autre. Ainsi, le dialecte Marocain, Algérien et Tunisien sont trois langues assez proches, mais vis-à-vis des autres dialectes que regroupent les pays arabes ces derniers sont très différents. On distingue deux grandes familles de dialectes: ceux de l'est et ceux de l'ouest (le *Machrek* et le *Maghreb*).

Les dialectes diffèrent de l'arabe standard en ce qu'ils présentent une grammaire souvent simplifiée et parfois des troncatures lexicales ainsi que des variations ou assimilations dans la prononciation. Ce sont les mots les plus fréquents et en nombre limité qui diffèrent le plus, tandis que le vocabulaire exceptionnel est le même en arabe dialectal et standard. Mêmes s'ils sont fréquemment associés à la seule langue parlée, les dialectes s'écrivent tout à fait naturellement.

Malgré cela, il est probablement préférable d'aborder l'arabe standard d'abord, et les Arabes dialectaux ensuite, car l'apprentissage des différents dialectes n'en sera que facilité. L'Arabe standard étant comme une langue pivot vis-à-vis des dialectes si on excepte les expressions et mots étranger(ère)s à l'arabe emprunté(e)s par les dialectes, sa connaissance ouvrira la porte à la découverte ultérieure de tous les dialectes dans leur ensemble.

Chapitre 2

Les Machines à Vecteurs Supports

2.1 Introduction

La complexité de certains phénomènes ne permet pas l'accès à une description débouchant sur une modélisation déterministe. La statistique a alors développé un ensemble d'approches basées sur la prévision d'une grandeur à partir d'une série d'observations.

Les méthodes d'apprentissage ont été utilisées dans de nombreuses disciplines. Citons la reconnaissance de caractères manuscrits, l'imagerie médicale ou satellitaire, la prévision d'une grandeur climatique ou économique, etc.

Parmi les méthodes d'apprentissage, les Machines à Vecteurs Supports représentent une forme optimisée de celles-ci.

Les Machines à Vecteurs Supports sont de puissants classifieurs inspirés par le principe de minimisation du risque structurel (**S**tructurel **R**isk **M**inimisation - **SRM**) qui ont prouvé leur efficacité pour diverses tâches de classification, parmi lesquelles: l'identification/vérification du locuteur et des langages, la catégorisation de textes, la reconnaissance des visages. Elles présentent l'avantage d'être *discriminatives* par opposition aux approches génératives qui présupposent une structure particulière (souvent mal justifiée) des formes de densité des données et exhibent en pratique une très bonne capacité de généralisation.

2.2 La Minimisation du Risque Structurel

Pour tout problème de classification qui rentre dans le cadre de l'apprentissage statistique supervisée, le but est de prévoir la classe y d'un vecteur d-dimensionnel x en se basant sur les mesures des variables qui l'expliquent avec pour seule information celle contenue dans l'échantillon d'apprentissage $\mathcal{D}$.

Nous supposons que les données (ou exemples) sont des couples $(x_i, y_i)_{1\leq i\leq l} \in \mathcal{X} \times \mathcal{Y}$ où $\mathcal{X}$ désigne l'espace des variables explicatives souvent pris dans $\mathbb{R}^d$ $\mathcal{Y} = \{-1, +1\}$ et l est la taille de l'échantillon. L'échantillon d'apprentissage $\mathcal{D}$ est ainsi une collection de couple aléatoire (x, y) dont la distribution $\mathcal{P}$ est fixe mais inconnue.

$$\mathcal{D} = \{(x_1, y_1), \dots, (x_l, y_l)\}; \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, +1\} \tag{2.1}$$

Les x_i sont ici des vecteurs d'attributs pouvant appartenir à deux classes possibles: une classe positive notée $+1$ et une classe négative notée -1. Les y_i représentent donc les étiquettes ou les valeurs cibles associées à x_i.

La tâche *d'apprentissage à partir des exemples (ou des données)* consiste à trouver parmi un ensemble $\mathcal{F}$ de fonctions de classification permettant de prédire l'étiquette y_i de x_i :

$$\mathcal{F} = \{f_\vartheta; \vartheta \in \Lambda\};\ f_\vartheta : \mathbb{R}^d \mapsto \{-1, +1\} \quad \text{avec } \Lambda \text{ un ensemble d'indices} \tag{2.2}$$

la fonction f_{ϑ^*} qui minimise le risque fonctionnel

$$\mathcal{R}(\vartheta) = \int \frac{1}{2} |f_\vartheta(x) - y| d\mathcal{P}(x, y) \tag{2.3}$$

autrement dit, la fonction qui minimise *la probabilité de mal prédire l'étiquette y de x*. La difficulté rencontrée vient du fait que $\mathcal{R}(\vartheta)$ est inconnue puisque $\mathcal{P}(x, y)$ est inconnue. Il est donc nécessaire de faire appel à un principe d'induction en se basant sur les données d'apprentissage.

L'approche la plus directe consiste à adopter une stratégie visant à minimiser l'erreur de classification sur l'ensemble d'apprentissage qu'on appelle *risque empirique* R_{emp} défini par:

$$\mathcal{R}_{emp}(\vartheta) = \frac{1}{l} \sum_{i=1}^{l} \frac{1}{2} |f_\vartheta(x_i) - y_i| \tag{2.4}$$

L'approche de *Minimisation du Risque Empirique* (**E**mpirical **R**isk **M**inimisation - **ERM**) s'appuie sur le fait que $\mathcal{R}_{emp}(\vartheta)$ tend vers $\mathcal{R}(\vartheta)$ lorsque l tend vers l'infini (en vertu de la loi des grands nombres).

Lorsque le nombre d'exemples d'apprentissage l est petit, il s'avère que minimiser le risque $\mathcal{R}_{emp}(\vartheta)$ n'implique pas forcément un risque $\mathcal{R}(\vartheta)$ minimal. En minimisant le risque empirique, il est possible d'obtenir un modèle efficace sur les exemples (ou les données) de l'ensemble d'apprentissage mais ce dernier ne garantit pas des performances satisfaisantes *en généralisation*, c'est-à-dire sur de nouveaux exemples. Ce phénomène est connu sous le terme de *sur-apprentissage* (*overfitting*).

Le principe de *Minimisation du Risque Structurel* dû à Vapnik & Chervonenkis permet de pallier cette difficulté [Vapn (95)]. Il repose sur le concept de *dimension VC* (**V**apnik **C**hervonenkis) d'un ensemble de fonctions $\{f_\vartheta\}$ qui permettent d'obtenir la borne suivante sur le risque. On obtient avec une probabilité $1 - \eta$:

$$\mathcal{R}(\vartheta) \leq \mathcal{R}_{emp}(\vartheta) + \sqrt{\frac{h\left(\log\frac{2l}{h}+1\right)-\log\left(\frac{\eta}{4}\right)}{l}} \tag{2.5}$$

La dimension VC décrit la *capacité* de séparation d'un ensemble de fonctions considérées par un algorithme d'apprentissage. Pour un problème bi-classes, h est le nombre maximum de points k qui peuvent être séparés par le biais de ces fonctions en deux classes de toutes les façons possibles (2^k façons). Ce concept est illustré à la figure 2.1.

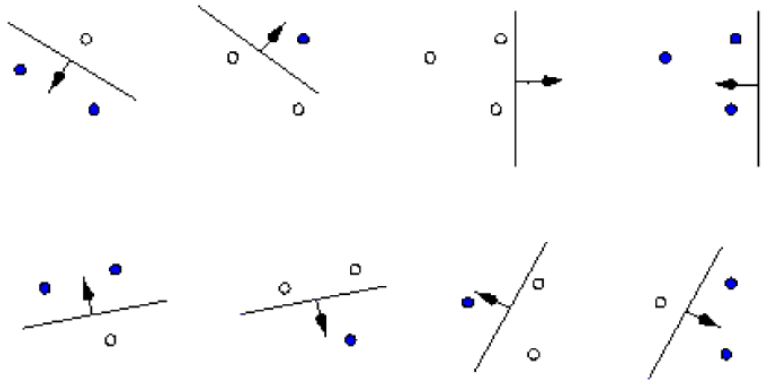


Fig. 2.1: Illustration du concept de dimension VC, d'après [Burg (98)].

Dans $\mathbb{R}^2$, en considérant un ensemble de fonctions $\{f_\vartheta\}$ représentant des droites orientées de telle manière que tous les points d'un côté de la droite soient étiquetés par $+1$ et tous ceux de l'autre côté de la droite étiquetés par -1, il n'est pas possible de trouver plus de trois points séparables de toutes les façons possibles. Par la dimension VC de l'ensemble des droites orientées dans $\mathbb{R}^2$, est trois.

L'inégalité (2.5) indique que l'erreur en généralisation $\mathcal{R}(\vartheta)$ peut être maîtrisée en contrôlant d'une part le risque empirique et, d'autre part, une quantité qui dépend du rapport $\frac{l}{h}$ appelée *intervalle de confiance* (c'est la divergence entre le risque fonctionnel et le risque empirique). Si ce rapport est suffisamment grand, le *risque garanti* (c'est ainsi que l'on désigne le membre droit de l'inégalité (2.5)) est dominé par le risque empirique, et il est suffisant de minimiser R_{emp} pour garantir un risque fonctionnel minimum. Sinon, l'approche de *Minimisation du Risque Empirique* n'est pas satisfaisante.

L'approche de *Minimisation du Risque Structurel* adopte la stratégie qui consiste à minimiser le risque en contrôlant la dimension VC. Cela est réalisé en exploitant une structuration de $\mathcal{F}$ en sous-ensembles emboités $\mathcal{F}_m = \{f_\vartheta^m; \vartheta \in \Lambda_m, \Lambda_m \subset \Lambda\}$ tels que

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \cdots \subset \mathcal{F}_m \subset \cdots \tag{2.6}$$

Les dimensions VC correspondantes vérifient alors

$$h_1 \leq h_2 \leq \cdots \leq h_m \leq \cdots \tag{2.7}$$

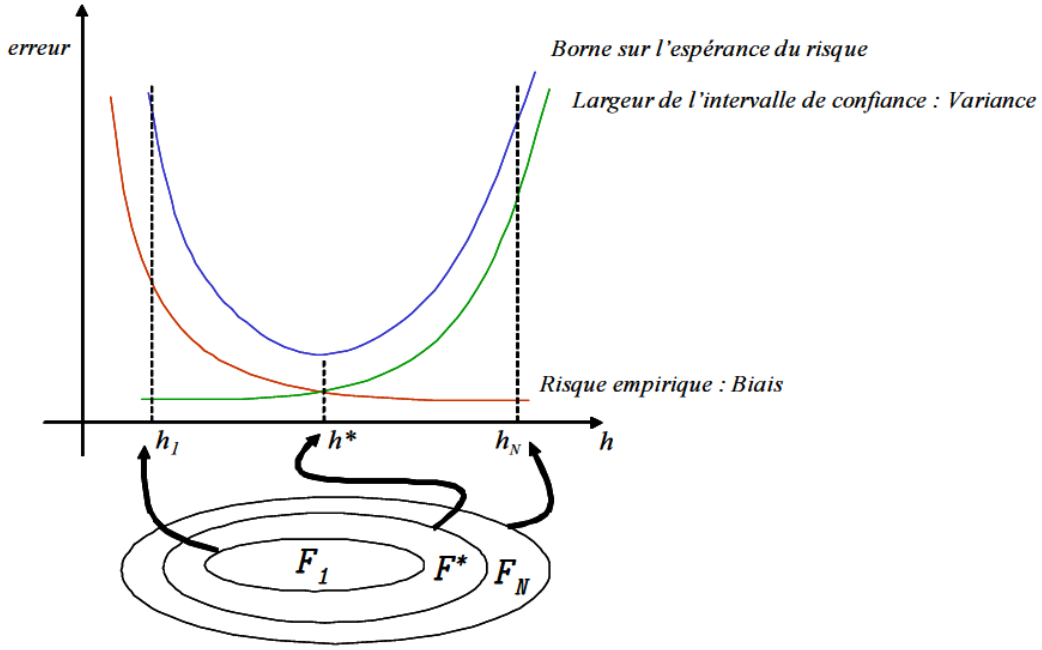


Fig. 2.2: Variation de la borne sur le risque espéré.

Il s'agit maintenant de choisir la fonction f_α^m dans l'ensemble $\mathcal{F}_m$ qui réalise la plus petite valeur du risque garanti. Cependant, il ne suffit pas de retenir le sous-ensemble associé à la plus petite des valeurs h_m puisqu'en pratique les plus petites dimensions VC correspondent à des valeurs élevées du risque empirique et vice-versa, d'où la nécessité de trouver une valeur de compromis. Ainsi, *il est possible de produire des algorithmes de classification dont l'efficacité statistique peut être contrôlée en se donnant une classe de fonctions dont la capacité peut être mesurée*.

2.3 Les Machines à Vecteurs Supports linéairement séparables

Les Machines à Vecteurs Supports sont par essence des classifieurs bi-classes qui visent à séparer les exemples de chaque classe $\mathcal{C}_m$ $1 \leq m \leq 2$ au moyen d'un hyperplan $\mathcal{H}_{w_0,b_0}$ choisi

de manière à garder un maximum de marges de séparation entre n'importe quel exemple d'apprentissage et $\mathcal{H}_{w_0,b_0}$.

De façon plus formelle, en se donnant les exemples $\{(x_1, y_1), \ldots, (x_l, y_l)\}$ dans $\mathbb{R}^d \times \{+1, -1\}$, il s'agit de déterminer *l'hyperplan optimal*

$$\mathcal{H}_{w_0,b_0}: w_0^T x + b_0 = 0; \qquad w_0 \in \mathbb{R}^d, b_0 \in \mathbb{R} \tag{2.8}$$

En utilisant une mise à l'échelle appropriée de w et b et en supposant dans un premier temps que les données sont linéairement séparables, il est possible de contraindre les exemples de chaque classe à satisfaire les conditions :

$$\begin{aligned} w^T x_i + b \geq +1 \ \ pour\ y_i = +1 \\ w^T x_i + b \leq -1 \ \ pour\ y_i = -1 \end{aligned} \tag{2.9}$$

Qui peuvent être combinées en une même inégalité:

$$y_i(w^T x_i + b) - 1 \geq 0, \ \ i = 1, \ldots, l \tag{2.10}$$

De là, on peut avoir deux hyperplans suivants permettant de définir la marge:

$$\begin{aligned} \mathcal{H}_1: w^T x_i + b = +1 \\ \mathcal{H}_2: w^T x_i + b = -1 \end{aligned} \tag{2.11}$$

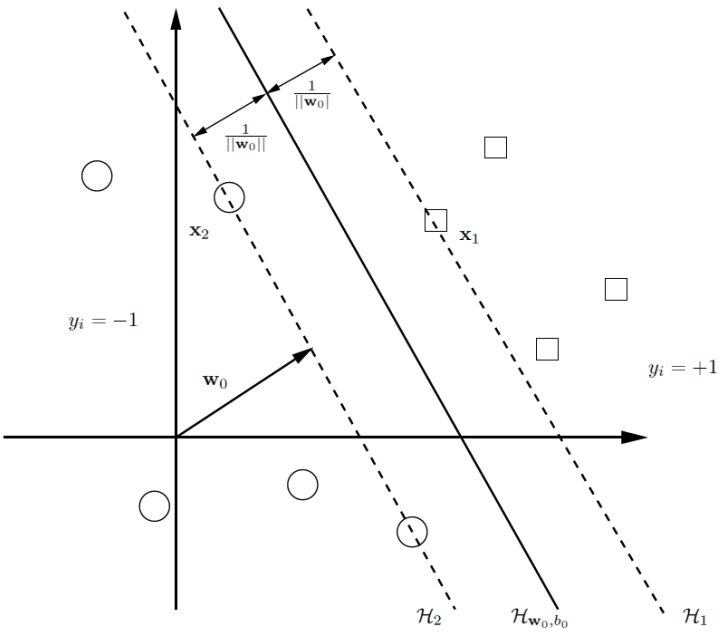


Fig. 2.3: Hyperplan optimal et marge d'un classifieur SVM.

Dans la figure 2.3, on peut remarquer que $\mathcal{H}_1$ et $\mathcal{H}_2$ sont parallèles (ils ont la même normale w), et qu'il n'existe aucun point entre les deux grâce à (2.9). Par conséquent, la marge n'est autre que la distance entre $\mathcal{H}_1$ et $\mathcal{H}_2$qui vaut $\frac{2}{\|w\|}$.

Les "*ronds*" représentent des exemples de la classe -1 et, les "*carrés*" des exemples de la classe +1, $w_0^T x_1 + b_0 = 1,\ w_0^T x_2 + b_0 = -1 \implies w_0^T(x_1 - x_2) = 2 \implies \frac{w_0}{\|w_0\|}.(x_1 - x_2) = \frac{2}{\|w_0\|}$.

Les points qui se trouvent sur les hyperplans $\mathcal{H}_1$ et $\mathcal{H}_2$ sont appelés les vecteurs supports (**S**upport **V**ectors - **SV**). Le problème posé ne dépend en fait que de ces points particuliers, en ce sens que si tous les autres points sont éliminés, la solution du problème reste la même.

Ainsi, l'hyperplan optimal est la solution du problème d'optimisation :

$$\begin{aligned} &min\,\tau(w) = \tfrac{1}{2}\|w\|^2; \quad w \in \mathbb{R}^d, \quad b \in \mathbb{R} \\ &sous\ la\ contrainte \quad y_i(w^T x_i + b) - 1 \geq 0, \quad i = 1, \dots, l \end{aligned} \tag{2.12}$$

2.3.1 Calcul des Machines à Vecteurs Supports

Le problème (2.12) est un problème *d'optimisation sous contraintes* qui est résolu en introduisant des *multiplicateurs de Lagrange* α_i. L'idée principale est d'introduire un multiplicateur de Lagrange pour chaque contrainte qui satisfait les conditions de Karush–Kuhn–Tucker (KKT) à une solution optimale:

$$\begin{aligned} &\alpha_i[y_i(w^T x_i + b) - 1] = 0 \\ &avec \quad \alpha_i \geq 0, \quad i = 1,2, \dots, l \end{aligned} \tag{2.13}$$

Ainsi, la nouvelle fonction objective primale définissant l'hyperplan optimal du problème d'optimisation devient :

$$\mathcal{L}(w, b, \alpha) = \tfrac{1}{2}\|w\|^2 - \textstyle\sum_{i=1}^{l} \alpha_i[y_i(w^T x_i + b) - 1] \tag{2.14}$$

où $\alpha = [\alpha_1 \dots \alpha_l]^T$.

Le Lagrangien $\mathcal{L}$ doit être minimisé par rapport aux variables dites primales w et b, et maximisé par rapport aux variables duales α_i. Ce sont les conditions de Karush-Kuhn-Tucker (KKT) [Schol (02)].

Dans le cas où la *fonction objective* (ici $\tau(w)$) et les contraintes (ici $c_i(x_i) = y_i(w.x_i + b) - 1$) sont convexes, les conditions de Karush-Kuhn-Tucker (KKT) sont nécessaires et suffisantes, et la solution du problème est telle que :

$$\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial b} = \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.15}$$

$$\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{l} \alpha_i y_i x_i = 0 \tag{2.16}$$

$$y_i(w^T x_i + b) - 1 \geq 0 \tag{2.17}$$

$$\alpha_i[y_i(w^T x_i + b) - 1] = 0 \tag{2.18}$$

$$\alpha_i \geq 0 \tag{2.19}$$

Les conditions (2.15) et (2.16) donnent respectivement

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.20}$$

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i \tag{2.21}$$

De plus, (2.18) implique que tous les points x_i qui ne sont pas vecteurs supports, c'est à dire ceux qui ne vérifient pas l'égalité $y_i(w^T x_i + b) - 1 = 0$ sont associés à des α_i nuls. Ainsi, on retrouve que l'hyperplan optimal ne dépend que des n_s vecteurs supports du problème $(n_s \leq l)$:

$$w = \sum_{i=1}^{n_s} \alpha_i y_i x_i \tag{2.22}$$

et la fonction de décision est définie par le signe de :

$$f(x) = w^T x + b = \left(\sum_{i=1}^{n_s} \alpha_i y_i (x^T x_i)\right) + b \tag{2.23}$$

Le paramètre b peut être déterminé au travers de la condition (2.18) en choisissant un indice i tel que $\alpha_i \neq 0$,

$$\begin{aligned} &\alpha_i[y_i(w^T x_i + b) - 1] = 0 \\ &y_i(w^T x_i + b) = 1 \\ &b_i = \frac{1}{y_i} - w^T x_i = y_i - w^T x_i \end{aligned} \tag{2.24}$$

où b est la moyenne de tous les valeurs obtenues en utilisant tous les points x_i associés à des α_i non nuls.

$$b = avg_{\alpha_i > 0}\{b_i\} \tag{2.25}$$

Reportons (2.20) et (2.21) sur la fonction objective primale (2.14), on obtient la formulation duale du problème :

$$\begin{aligned}\mathcal{L}(w,b,\alpha) &= \frac{1}{2}w^T w - w^T \underbrace{\left(\sum_{i=1}^{l}\alpha_i y_i x_i\right)}_{w} - b\underbrace{\sum_{i=1}^{l}\alpha_i y_i}_{0} + \sum_{i=1}^{l}\alpha_i \\ &= -\frac{1}{2}w^T w + \sum_{i=1}^{l}\alpha_i \\ &= \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j x_i^T x_j\end{aligned} \tag{2.26}$$

Donc, la fonction objective duale est donnée par:

$$\begin{aligned}&\max \mathcal{L}(w,b,\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j x_i^T x_j \\ &sous\ les\ contraintes \quad \sum_{i=1}^{l}\alpha_i y_i = 0, \quad \alpha_i \geq 0\end{aligned} \tag{2.27}$$

Remarquons que w et b ont été éliminés et qu'il s'agit désormais de déterminer les α_i.

2.4 Les Machines à Vecteurs Supports linéairement non séparables

Dans ce cas, il faut rendre moins rigides les contraintes (2.9) en introduisant des variables d'écart positives ξ_i pour que les contraintes deviennent :

$$\begin{aligned} w^T x_i + b &\geq +1 - \xi_i, \quad pour \ \ y_i = +1 \\ w^T x_i + b &\leq -1 - \xi_i, \quad pour \ \ y_i = -1 \\ \xi_i &\geq 0, \qquad\quad i = 1, \dots, l \end{aligned} \tag{2.28}$$

Qui peuvent être combinées en une même inégalité:

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \ \ i = 1, \dots, l \tag{2.29}$$

Les variables d'écart indiquent trois points:

- si $\xi_i = 0$, alors le point x_i correspondant est au moins loin de l'hyperplan, et il est bien classifié.
- si $0 < \xi_i < 1$, alors le point x_i correspondant est à l'intérieur des hyperplans canoniques, et il est encore bien classifié.
- si $\xi_i > 1$, alors le point x_i correspondant est mal classé, et il est dans le coté faux de l'hyperplan

La nouvelle fonction objective primale est donnée par:

$$\begin{aligned} &\min_{w,b,\xi_i} \tau(w,\xi_i) = \min_{w,b,\xi_i} \tfrac{1}{2}\|w\|^2 + C\left(\textstyle\sum_{i=1}^{l}(\xi_i)^p\right) \\ &sous\ la\ contrainte \qquad y_i(w^T x_i + b) \geq 1 - \xi_i \\ &\xi_i \geq 0, \qquad i = 1,2,\dots,l \end{aligned} \tag{2.30}$$

Où $\boldsymbol{\xi} = \{\xi_1, \dots, \xi_l\}^T$ et $C > 0$ est un paramètre permettant de contrôler le compromis entre le fait de maximiser la marge et minimiser les erreurs de classification commises sur l'ensemble d'apprentissage. On parle alors de classifieur à marge souple [Shol (02)].

C et p sont des constantes qui incorporent le coût de la dysclassification. Le terme $\sum_{i=1}^{l}(\xi_i)^p$ donne la perte qui est une estimation de la déviation à partir du cas séparable. Le scalaire C qui est choisi empiriquement est une constante de régularisation qui contrôle le compromis de maximisation de la marge (correspondant à minimiser $\frac{1}{2}\|w\|^2$) ou de minimisation de la perte (correspondant à minimiser les variables d'écart $\sum_{i=1}^{l}(\xi_i)^p$). Par exemple, si $C \longrightarrow 0$ alors le composant de perte essentiel disparaît, et on aura une maximisation de la marge. Seulement, si $C \longrightarrow \infty$ alors la marge cesse d'avoir plus d'effet, et la fonction objective essaye de minimiser la perte. La constante p gouverne la forme de la perte. Typiquement p est initialisé à 1 ou à 2. Dans le cas où on a $p = 1$, on parle de la norme L1-SVM, et $p = 2$ on parle de la norme L2-SVM.

2.4.1 La norme L1-SVM

En supposant $p = 1$, on peut calculer le Lagrangien pour le problème d'optimisation dans (2.30) en introduisant les multiplicateurs de Lagrange α_i et β_i qui satisfont les conditions KKT sur la solution optimale:

$$\begin{aligned} &\alpha_i(y_i(w^T x_i + b) - 1 + \xi_i) = 0 \\ &\beta_i(\xi_i - 0) = 0 \\ &avec \quad \alpha_i \geq 0, \ \ \beta_i \geq 0 \end{aligned} \tag{2.31}$$

Le Lagrangien sur la fonction objective primale est donné par:

$$\mathcal{L}(w,\alpha,\beta,\xi) = \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i(y_i(w^T x_i + b) - 1 + \xi_i) - \sum_{i=1}^{l}\beta_i\xi_i \tag{2.32}$$

où C est un paramètre de régularisation qui doit être fixé a priori et qui pondère le coût des erreurs sur les contraintes.

Transformons la fonction objective primale (2.32) à la forme Lagrangien duale par leur dérivée partielle sur w,b, et ξ_i en les mettant à zero:

$$\begin{aligned}
&\frac{\partial \mathcal{L}(w,\alpha,\beta,\xi)}{\partial w} = w - \sum_{i=1}^{l} \alpha_i y_i x_i = 0 \Longrightarrow w = \sum_{i=1}^{l} \alpha_i y_i x_i \\
&\frac{\partial \mathcal{L}(w,\alpha,\beta,\xi)}{\partial b} = \sum_{i=1}^{l} \alpha_i y_i = 0 \\
&\frac{\partial \mathcal{L}(w,\alpha,\beta,\xi)}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \Longrightarrow \beta_i = C - \alpha_i
\end{aligned} \tag{2.33}$$

En reportant les valeurs de (2.33) dans la fonction objective primale en (2.32), nous obtenons la forme duale de cette fonction:

$$\mathcal{L}(w,\alpha,\beta,\xi) = \frac{1}{2} w^T w - w^T \underbrace{\left(\sum_{i=1}^{l} \alpha_i y_i x_i\right)}_{w} - b \underbrace{\sum_{i=1}^{l} \alpha_i y_i}_{0} + \sum_{i=1}^{l} \alpha_i + \sum_{i=1}^{l} \underbrace{(C - \alpha_i + \beta_i)}_{0} \xi_i$$

$$\mathcal{L}(w,\alpha,\beta,\xi) = \sum_{i=1}^{l} \alpha_i + \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{2.34}$$

Pour avoir une solution à notre problème d'optimisation, la fonction objective duale doit être maximisée sur les coefficients Lagrangien α:

$$\begin{aligned}
&\max_{\alpha} \mathcal{L}(w,\alpha,\beta,\xi) = \sum_{i=1}^{l} \alpha_i + \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j x_i^T x_j \\
&avec \quad 0 \le \alpha_i \le C, \quad \sum_{i=1}^{l} \alpha_i y_i = 0, \quad i = 1, \ldots, l
\end{aligned} \tag{2.35}$$

Géométriquement, la grandeur $\frac{\xi_i}{\|w\|}$ correspond à la distance entre l'hyperplan canonique correspondant à la classe y_i (cf. Fig. 2.4) et l'exemple x_i mal classé. Pour les exemples correctement classés, ξ_i est nul.

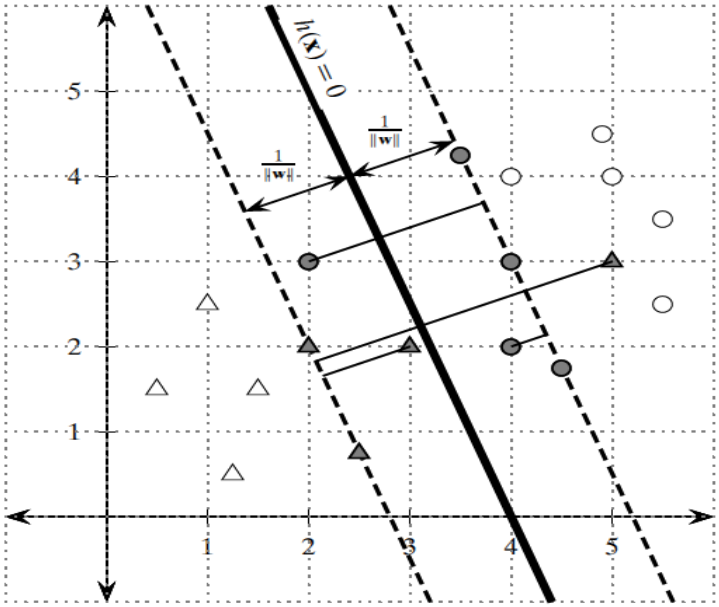


Fig. 2.4: Situation correspondant à des exemples mal classés.

On distingue deux types de vecteurs supports: Les vecteurs supports de *première catégorie* pour lesquels $0 < \alpha_i < C$ (ce sont des exemples x_i bien classés mais qui se trouvent dans la marge entre l'hyperplan $\mathcal{H}_0$ et les hyperplans canoniques $\mathcal{H}_{-1}$ et $\mathcal{H}_{+1}$) et les vecteurs supports de *seconde catégorie* pour lesquels $\alpha_i + \beta_i = C$

On note que la fonction objective est la même que pour la fonction dual Lagrangien du cas linéairement séparable. Cependant, les contraintes sur α_i sont différentes parce que on doit imposer que $\alpha_i + \beta_i = C$ avec $\alpha_i \geq 0$ et $\beta_i \geq 0$ qui implique que $0 \leq \alpha_i \leq C$.

Ainsi, on peut dire que $\alpha_i = 0$ pour les points qui ne sont pas des vecteurs supports et $\alpha_i \geq 0$ seulement pour les vecteurs supports comprenant tous les points x_i qui ont:

$$y_i(w^T x_i + b) = 1 - \xi_i \tag{2.36}$$

Dans ce cas, les vecteurs supports incluent tous les points qui sont sur la marge et qui ont $\xi_i \geq 0$. En utilisant ces vecteurs supports qui sont sur la marge et qui ont $0 \leq \alpha_i \leq C$ et $\xi_i = 0$, on peut calculer b_i:

$$\begin{aligned} &\alpha_i[y_i(w^T x_i + b_i) - 1] = 0 \\ &y_i(w^T x_i + b_i) = 1 \\ &b_i = \frac{1}{y_i} - w^T x_i = y_i - w^T x_i \end{aligned} \tag{2.37}$$

où b est la moyenne de toutes les valeurs obtenues en utilisant tous les points x_i associés

$$b = avg_{0 \leq \alpha_i \leq C}\{b_i\} \tag{2.38}$$

2.4.2 La norme L2-SVM

Dans ce cas, la somme des termes $\sum_{i=1}^{l}(\xi_i)^2$ est toujours positive et la possibilité d'avoir une variable d'écart $\xi_i < 0$ est écartée durant l'optimisation parce que un choix de $\xi_i = 0$ mène à une valeur plus petite de la fonction objective primale, et il satisfait encore la contrainte $y_i(w.x_i + b) \geq 1 - \xi_i$ même si on a $\xi_i < 0$. En d'autres termes, le processus d'optimisation remplacera toutes les variables d'écart négatives par zéro. Ainsi, la fonction objective primale pour le L2-SVM est donnée par:

$$\begin{aligned} &\min_{w,b,\xi_i} \tau(w, \xi_i) = \min_{w,b,\xi_i} \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{l}(\xi_i)^2\right) \\ &sous\ la\ contrainte \qquad y_i(w^T x_i + b) \geq 1 - \xi_i \\ &\xi_i \geq 0, \qquad i = 1,2,\dots,l \end{aligned} \tag{2.39}$$

On calcule le *Lagrangien* pour le problème d'optimisation dans (2.39) en introduisant les multiplicateurs de Lagrange α_i qui satisfont les conditions KKT sur la solution optimale:

$$\begin{aligned} &\alpha_i[y_i(w^T x_i + b) - 1 + \xi_i] = 0 \\ &avec \quad \alpha_i \geq 0 \end{aligned} \tag{2.40}$$

Dans ce cas, le *Lagrangien* primal de la fonction objective est:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i)^2 - \sum_{i=1}^{l}\alpha_i(y_i(w^T x_i + b) - 1 + \xi_i) \tag{2.41}$$

Transformons la fonction objective primale (2.41) sous la forme de Lagrangien duale par leur dérivée partielle sur w,b, et ξ_i en les mettant à zéro:

$$\begin{aligned} &\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{N}\alpha_i y_i x_i = 0 \Longrightarrow w = \sum_{i=1}^{l}\alpha_i y_i x_i \\ &\frac{\partial L(w,b,\alpha)}{\partial w_0} = \sum_{i=1}^{N}\alpha_i y_i = 0 \\ &\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial \xi_i} = 2C\xi_i - \alpha_i = 0 \Longrightarrow \xi_i = \frac{1}{2C}\alpha_i, \qquad i = 1, \dots, l \end{aligned} \tag{2.42}$$

Reportons les conditions (2.42) ci-dessus sur la fonction objective primale (2.41), on obtient la formulation duale de la fonction objective :

$$\begin{aligned} \mathcal{L}(w, b, \alpha) &= \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j x_i^T x_j - \frac{1}{4C}\sum_{i=1}^{l}\alpha_i^2 \\ &= \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \left(x_i^T x_j - \frac{1}{2C}\delta_{ij}\right) \end{aligned} \tag{2.43}$$

où δ_{ij}est la fonction Kronecker delta définie par $\delta_{ij} = 1$ si $= j$, et $\delta_{ij} = 0$ si $i \neq j$.

La résolution du problème d'optimisation est réalisée par la maximisation de (2.43) par rapport à α_i, qui est équivalente à [Crist (00)]:

$$\begin{aligned} &\max_{\alpha} \mathcal{L}\,(w, b, \alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \left(x_i^T x_j - \frac{1}{2C}\delta_{ij}\right) \\ &sous\ les\ contraintes \quad \alpha_i \geq 0, \quad \sum_{i=1}^{l}\alpha_i y_i = 0, \quad i = 1,2, \dots, l \end{aligned} \tag{2.44}$$

Une fois que nous résolvons les α_i en utilisant la méthode dans la section 2.4.1, nous trouvons le vecteur poids et le biais par :

$$\begin{aligned} w &= \sum_{i=1}^{l}\alpha_i y_i x_i \\ b &= \text{avg}_{0<\alpha_i<c}\{y_i - w^T x_i\} \end{aligned} \tag{2.45}$$

2.5 Généralisation du cas linéaire des Machines à Vecteurs Supports.

Dans le cas général, les données d'entrée appartiennent à un espace d'entrée $\mathbb{X}$ quelconque (*input space*). Pour généraliser les Machines à Vecteurs Supports dans un tel contexte, il suffit d'appliquer l'algorithme linéaire dans un espace vectoriel de dimension D où les données seront projetées. Cet espace est appelé "*Feature Space*" (espace caractéristique). On suppose donc une expansion non linaire sous-jacente ("*map*" ou "*embedding*" dans la littérature anglophone) :

$$\begin{aligned} \mathbb{X} &\longrightarrow \mathbb{R}^D \\ x &\longrightarrow \Phi(x) \end{aligned} \tag{2.46}$$

Si l'on considère le cas d'entrées vectorielles de dimension d ($\mathbb{X} = \mathbb{R}^d$) avec $d < D$, construire une fonction discriminante linéaire à partir d'une telle expansion revient à chercher une frontière non linéaire dans l'espace d'entrée $\mathbb{X}$, comme illustré pour un classifieur polynomial de degré 2 dans la figure 2.5. L'expansion Φ dans un espace de plus grande dimension $\mathbb{R}^D$ sert alors à augmenter la séparabilité des données. Accroître les caractéristiques via l'expansion Φ revient généralement à augmenter la complexité de la modélisation, la VC-dimension des classifieurs linéaires dans le *Feature Space* étant $(D + 1)$. Le critères de régularisation des Machines à Vecteurs Supports (*marge*) permet alors d'éviter la dimensionnalité dans l'espace caractéristique (*Feature Space*), au même titre qu'il permet d'éviter le sur-apprentissage des Machines à Vecteurs Supports linéaires lorsque le nombre de paramètres d'entrée est élevé.

De manière analogue au cas linéaire, la phase d'apprentissage consiste à rechercher les solutions f_θ qui minimisent une fonction objective τ, avec les formes :

$$f_\theta(x) = \sum_{u=1}^{D} \theta_u \phi_u(x) + \beta_0 \tag{2.47}$$

$$\tau(\theta) = \frac{1}{2}\sum_u \theta_u^2 + C \sum_{i=1}^{N} \left(1 - \ell_i f_\theta(a_i)\right)_+ \tag{2.48}$$

Intéressons-nous maintenant à la fonction K de deux variables définie par:

$$K(x, y) = \Phi(x)^T \Phi(y) = \sum_{u=1}^{D} \phi_u(x)\phi(y) \tag{2.49}$$

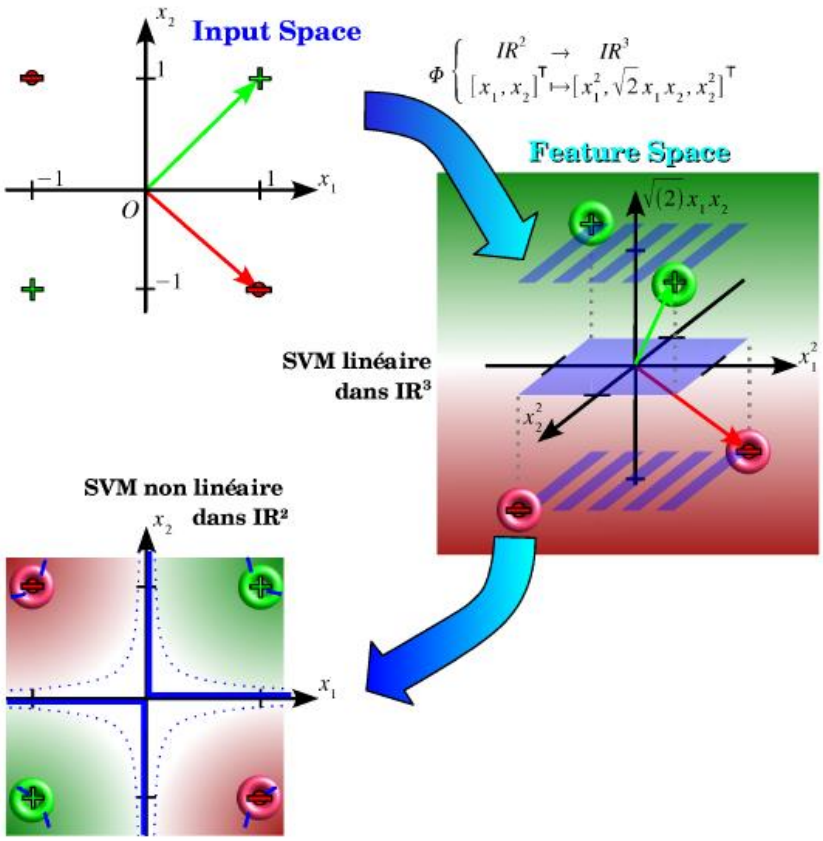


Fig. 2.5: Machines à Vecteurs Supports non linéaire - Illustration du principe.

Cette fonction symétrique et définie positive est appelée « *noyau* » (***définition 1***). Elle joue un rôle fondamental dans la généralisation des Machines à Vecteurs Supports. Comme nous allons le voir, la connaissance des valeurs $K(.,.)$ permet de faire abstraction des *expansions* $\Phi(.)$ à la fois pour résoudre le problème d'optimisation des Machines à Vecteurs Supports et pour appliquer un classifieur SVM sur de nouvelles données. Autrement dit, la fonction noyau permet d'éviter le calcul de l'*expansion* dans l'espace caractéristique. Avant d'en venir aux intérêt de telle astuce, nous présentons les notions fondamentales qui permettent de comprendre comment l'astuce du noyau permet de rendre le calcul des *expansions* implicite.

Définition 1 **(Espace de Hilbert à Noyaux Reproduisant)**

Soit $k: \mathbb{X} \times \mathbb{X} \longrightarrow \mathbb{R}$ *une fonction noyau symétrique et définie positive. On appelle Espace de Hilbert à Noyaux Reproduisant (RKHS) associé au noyau* K*, l'espace* $\mathcal{S}$ *des fonctions de* $\mathbb{X}$ *à valeur dans* $\mathbb{R}$ *engendré par les fonctions du type* $K(x_i, .)$*:*

$$f \in \mathcal{S} \Leftrightarrow \left[\exists l \leq +\infty, \exists \{x_i\}_{i=1,\dots,l} \in \mathbb{X}^l \; tq \; \forall x \in \mathbb{X}, f(x) = \sum_i \beta_i K(x_i, x)\right]$$

La produit scalaire dans cet espace $\mathcal{S}$, défini par $\langle f, K(x_i, .)\rangle_{\mathcal{S}} = f(x_i)$, est aussi symétrique et défini positif. Le caractère «reproduisant» fait référence à la propriété $\langle K(x_i, .), K(x_j, .)\rangle_{\mathcal{S}} = K(x_i, x_j)$.

Cette définition permet d'introduire le «théorème des représentants» qui jouent un rôle central dans les méthodes à noyaux. Il a été énoncé pour des fonction de coût quadratique par [Kime (1971)] et, plus tard, généralisé à des fonctions de coût quelconque par [Cox (90)]. Ce théorème appliqué à un RKHS $\mathcal{S}$ peut être formulé comme suit.

Théorème 1 **(Théorème des représentants)**

Soient

- *Une fonction strictement monotone $\Omega: \mathbb{R}^+ \longrightarrow \mathbb{R}$*
- *Un ensemble $\mathcal{A} = \{a_i, \ell_i\}_{i=1,\dots,N}$ d'élément étiquetés $(\mathbb{X} \times \mathbb{R})^N$*
- *Une fonction de coût $\mathcal{L}: \mathcal{A} \times \mathbb{R}^2 \longrightarrow \mathbb{R}^+$,*
- *et $\mathcal{S}$ le RKHS généré par le noyau reproduisant K.*

La fonction f^* dans $\mathcal{S}$ qui minimise le risque régularisé

$$f^* = \sum_{i=1}^{N} \beta_i K(a_i, x) + \beta_0, \qquad \{\beta_i\}_{i=0,\dots,N} \in \mathbb{R}^{N+1} \tag{2.50}$$

Si de plus $|\ell_i - y_i| \longmapsto \mathcal{L}(a_i, \ell_i, y_i)$ est croissante, alors elle peut s'écrire en fonction des données d'apprentissage et coefficients positifs («coefficients de Lagrange»):

$$f^*(x) = \sum_{i=1}^{N} \alpha_i \ell_i K(a_i, x) + \beta_0, \qquad \{\alpha_i\}_{i=1,\dots,N} \in (\mathbb{R}^+)^N \tag{2.51}$$

Le théorème des représentants garantit que les fonctions discriminantes $f(x)$ qui minimisent le risque régularisé peuvent se mettre sous la forme (2.51). Cela permet de reformuler le critère d'apprentissage « *primal* » τ (2.48) en un critère « *dual* » :

$$\tau_D(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \ell_i \ell_j K(a_i, a_j) \tag{2.52}$$

Ce critère est à maximiser sous les contraintes suivantes :

$$\begin{gathered} 0 \leq \alpha_1, \dots, \alpha_N \leq C \\ \sum_{i=1}^{N} \alpha_i \ell_i = 0 \end{gathered} \tag{2.53}$$

En pratique, le problème dual est résolu par des méthodes de programmation quadratique. La nouvelle forme (2.51) ne fait pas intervenir explicitement le calcul de l'*expansion* $\Phi(x)$.

Ceci représente l'astuce du noyau qui présente en fait deux intérêts fondamentaux:

1. Son expression peut souvent être simplifiée, afin de ne pas à calculer explicitement l'*expansion* dans l'espace caractéristique, dont la dimension peut être très grande

(*entrées vectorielles*), voire infinie. Dans le premier cas, l'intérêt du noyau est simplement de diminuer la complexité du problème. Dans le second cas, son intérêt est plus fondamental : certaines fonctions noyaux symétriques et définies positives correspondent à des espaces caractéristiques (*Feature Spaces)* de dimensions infinies. Ils génèrent donc des classifieurs qui sont basés sur les Machines à Vecteurs Supports à VC-dimension infinie, c'est-à-dire pouvant séparer un quelconque jeu de données selon toutes les partitions binaires possibles.

2. Le concept de noyau peut être élargi à d'autres types d'objets que les entrées vectorielles. Il permet ainsi de généraliser l'algorithme des Machines à Vecteur Supports à tout type d'objets (symboles, séquences,…etc). La fonction $K(x, y)$ manipule ces objets au même titre que le produit scalaire manipule des vecteurs.

En revanche, la complexité calculatoire de (2.51) dépend du nombre de données d'apprentissage. C'est là qu'intervient la notion de vecteurs supports.

2.6 Généralités sur les noyaux

2.6.1 L'astuce du noyau

De manière générale, l'astuce du noyau permet d'adopter des modélisations complexes capables dans le cas vectoriel de capturer une infinité de corrélations non linéaires entre les paramètres d'entrée. D'un autre côté, le critère de marge permet de garantir une certaine capacité de généralisation. Ceci explique pourquoi il a été observé dans plusieurs cas pratiques où le corpus d'apprentissage est limité en nombre, que les Machines à Vecteurs Supports donnent de meilleures performances que les approches génératives dans le traitement de la parole [Zhou (03), Aria (05)] comme dans d'autres applications de reconnaissance de formes [Just (04), Mash (05)]. En effet, les méthodes génératives donnent de bonnes performances à condition que :

1. La distribution réelle de chaque classe de vecteurs puisse être correctement capturée par la famille de fonctions choisie ;

2. Le corpus d'apprentissage soit suffisamment important et représentatif pour estimer correctement les paramètres de cette modélisation. Le nombre d'exemples nécessaires à l'apprentissage robuste d'un classifieur croît de façon polynomial avec le nombre de paramètre libres du modèle.

En pratique, l'astuce du noyau consiste à réécrire un algorithme où toutes les relations entre données d'entrée peuvent s'écrire sous forme de produits scalaires en remplaçant ce produit scalaire par une fonction scalaire de deux variables (« *noyau* »). L'astuce du noyau permet ainsi de généraliser un algorithme linéaire manipulant des vecteurs pour:

1. Traiter les vecteurs de façon non linéaire (parce que les données présentent des non linéarités qu'il est utile d'exploiter pour le problème visé) ; ou
2. Manipuler d'autres types d'objets que les vecteurs.

D'un point de vue qualitatif, le noyau peut être vu une mesure de similarité qui permet de comparer deux objets d'un même type. Pour appliquer les méthodes à noyau sur un ensemble de données, il suffit en pratique de connaître les valeurs de noyaux pour tous les couples de cet ensemble. Par exemple, pour dérouler l'algorithme d'apprentissage des Machines à Vecteurs Supports, il suffit de connaître les valeurs de noyaux estimées sur le corpus d'apprentissage. Ces valeurs sont habituellement mémorisées dans une matrice carrée qu'on appelle « matrice de Gram ».

Définition 2 **– Matrice de Gram**

Soient une fonction noyau $K: \mathbb{X}^2 \longrightarrow \mathbb{R}$ *et un ensemble de données* $\mathcal{X} = \{x_i\}_{i=1}^{l}$ *de taille* l*. La «matrice de Gram»* $\mathcal{G}_\mathcal{X}$ *est définie par la matrice carrée* $l \times l$ *contenant les valeurs du noyau sur les couples:* $(\mathcal{G}_\mathcal{X})_{i,j} = K(x_i, x_j)$

Dans la suite, nous notons K le noyau et $\mathbb{X}$ l'espace d'entrée.

2.6.2 Propriétés mathématiques

Tout comme le produit scalaire, on peut attendre de la fonction noyau qu'elle soit finie positive (***définition 3***). Avant d'en venir aux avantages d'une telle propriété mathématique, nous rappelons sa définition.

Définition 3 **– ((semi)défini-positivité)**

Une fonction scalaire $K: \mathbb{X}^2 \longrightarrow \mathbb{R}$ *est «(semi)-définie positive» si et seulement si*

$$\forall(\psi: \mathbb{X} \longrightarrow \mathbb{R}) \neq 0, \quad \iint_{\mathbb{X}^2} K(x,y)\psi(x)\psi(y)dxdy \geq 0$$

Elle est «définie positive» (propriété plus forte) si et seulement si elle vérifie la même propriété avec une inégalité stricte ('$>$' au lieu de '$\geq$').

Une matrice carrée K de taille $l \times l$ est «semi-définie positive» (respectivement «définie positive») si et seulement si pour tout vecteur colonne $\psi \in \mathbb{R}^l$ non nul, $\psi^T K \psi \geq 0$ (respectivement $\psi^T K \psi > 0$).

Les fonctions scalaires symétriques et définies positives que l'on désigne souvent simplement par « *noyaux* » sont plus précisément des « *noyaux de Mercer* » (On trouve aussi le terme de «*covariance kernels*»). Cette expression vient de ce que l'on appelle « *Théoreme de Mercer* » [Shaw (04)].

Théoreme 2 **- Théoreme de Mercer**

Si un noyau $K: \mathbb{X}^2 \longrightarrow \mathbb{R}$ est symétrique et semi-défini positif alors il admet un développement de la forme

$$\forall (x, y) \in \mathbb{X}^2, K(x, y) = \sum_{u=1}^{D} \Phi_u(x) \Phi_u(y)$$

où $D \leq +\infty$ et les D fonctions scalaires $\Phi_u: \mathbb{X} \longrightarrow \mathbb{R}$ peuvent être choisies parmi une famille orthonormée. Les conditions de symétrie et de semi-défini-positivité sont nécessaires et suffisantes.

Autrement dit, si $K: \mathbb{X}^2 \longrightarrow \mathbb{R}$ est défini positif alors il peut s'exprimer comme un produit scalaire dans un espace vectoriel où sont projetées les données, appelé «*Feature Space*» ou espace caractéristique. Inversement, si l'on définit une correspondance entre des données d'entrées et un espace vectoriel alors le produit scalaire dans cet espace vectoriel sera un noyau défini positif. L'*expansion* correspondant à un noyau est une fonction $\Phi: \mathbb{X} \longrightarrow \mathbb{R}^D$ telle que :

$$\forall (x, y) \in \mathbb{X}^2, K(x, y) = \Phi(x)^T \Phi(y) \tag{2.54}$$

Notons que les données d'entrée projetées dans l'espace caractéristique $\mathbb{R}^D$ gisent sur un sous-ensemble ouvert de $\mathbb{R}^D$ qui est une variété de Riemann. A contrario, les éléments de l'espace caractéristique $\mathbb{R}^D$ n'admettent pas forcément de pré-image dans l'espace d'entrée [Mika (99)]. La figure 2.6 représente la variété correspondant à des données d'entrée bidimensionnelles et à un noyau polynomial de degré deux : $K(x, y) = (x^T y)^2$.

La VC-dimension des fonctions discriminantes formées à partir d'un noyau correspondant à un espace caractéristique de dimension D est généralement $(D + 1)$. Pour certaines fonctions noyau symétriques et définies positives, D peut-être infini et Φ l'*expansion* n'a pas toujours une approximation analytique connue. En fait, le théorème de Mercer est un théorème d'existence. Il ne fournit par contre aucun moyen de déduire l'*expansion* Φ sous-jacente à un noyau K donné. Ainsi, il existe des noyaux pour lesquels aucune approximation analytique n'a encore été trouvé comme le cas du noyau Gaussien.

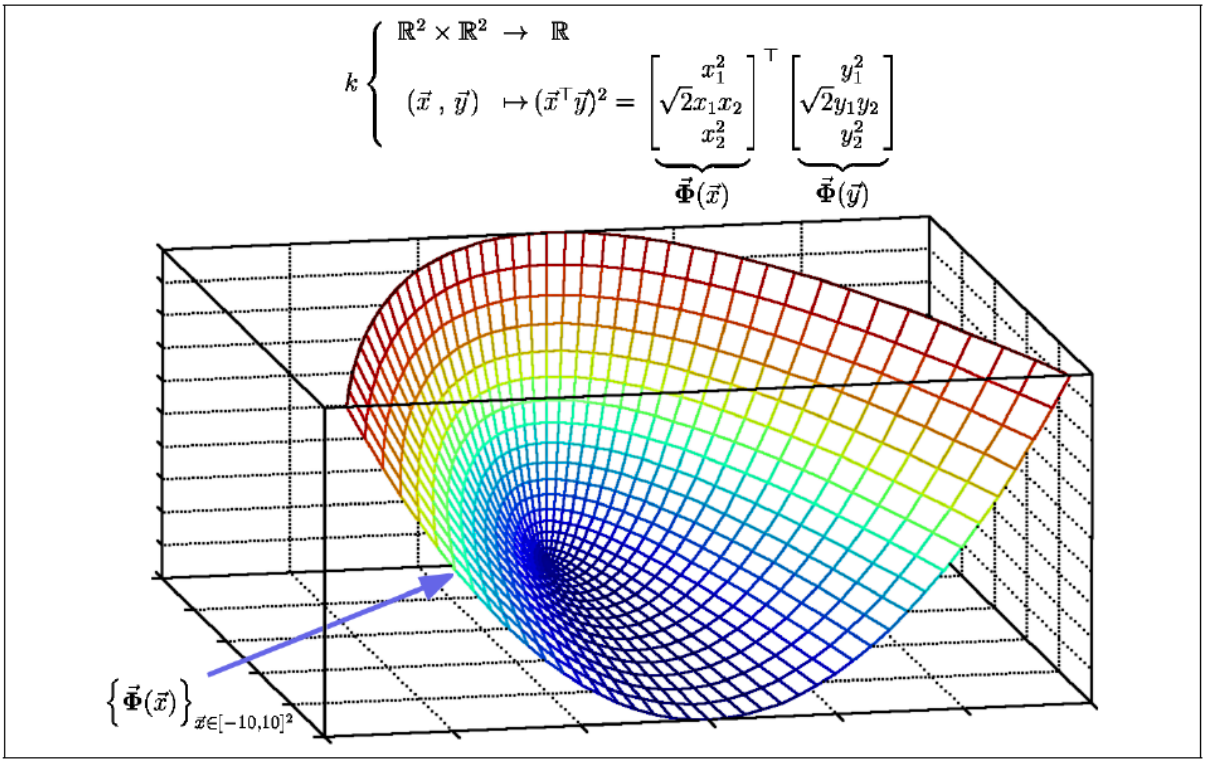


Fig. 2.6: Variété de Riemann correspondant à un noyau polynomial.

2.6.3 Combinaison de noyaux

La combinaison de noyaux peut désigner deux choses :

- *Combinaison de valeurs* (*de noyaux*) - Cela peut servir à combiner plusieurs types d'informations (numérique/symbolique) à l'intérieur de la modélisation. Pour les Machines à Vecteurs Supports, par exemple, il est en générale préférable de regrouper toutes les caractéristiques mesurées au sein d'un seul et même critère d'apprentissage, plutôt que de combiner les sorties de plusieurs Machines à Vecteurs Support.

- *Combinaison de fonctions* (*noyaux*) - Cela sert en général à augmenter la complexité de la modélisation sur un type de donnée.

Dans tous les cas, il est préférable que la combinaison de noyaux conserve les conditions de Mercer.

Combinaison de valeurs de noyaux

Concernant la combinaison de valeurs de noyaux, on dispose des résultats suivants [Scho (02)] :

Théorème 3

La combinaison linéaire positive de plusieurs noyaux de Mercer est un noyau de Mercer

$$K(x,y) = \sum_i \alpha_i K_i(x,y), \qquad \alpha_i > 0$$

Le produit de noyaux de Mercer est un noyau de Mercer

$$K(x,y) = \prod_i K_i(x,y)$$

Le choix des paramètres de combinaison optimaux (par exemple les coefficients α_i d'une combinaison linéaire) peut se faire selon plusieurs méthodes, comme le boosting [Cram (03)] ou l'optimisation de critères comme des mesures d'alignement [Crist (02), Kand (02), Poth (05)] ou encore le « critère de séparabilité de classe » [Wang (02)].

Combinaison de fonction noyaux

Pour la combinaison des fonctions noyaux, on dispose du résultat suivant :

Théorème 4

$$Si \begin{cases} K(x,y)\ est\ noyau\ de\ Mercer\ et \\ K'(x,y) = l[x^T x, x^T y, y^T y]\ est\ noyau\ de\ Mercer\ vectoriel \end{cases}$$

$$alors\ \ l[K(x,x), K(x,y), K(y,y)]\ est\ noyau\ de\ Mercer$$

Le théorème 4 implique aussi que l'on puisse créer à partir d'un noyau une infinité de noyaux résultant de combinaisons polynomiales, résultat qui se généralise dans la formulation suivante:

Théorème 5

La combinaison linéaire des puissances d'un noyau de Mercer est un noyau de Mercer.

$$K(x,y) = \sum_{i=0}^{\infty} \alpha_i K_l(x,y)^i, \qquad \alpha_i > 0$$

Dans la littérature, différentes formes de noyaux ont été proposées, dont :

- le noyau linéaire:
$$k(x,y) = x^T y \tag{2.55}$$
- le noyau polynomial de degré δ
$$k(x,y) = (x^T y)^{\delta} \tag{2.56}$$
- le noyau radial (RBF – *Radial Basis Function*) exponentiel:
$$k(x,y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}} \tag{2.57}$$

2.7 Les Machines à Vecteurs Supports et l'astuce noyau

Il s'agit de doter les Machines à Vecteurs Supports d'un mécanisme permettant de produire des surfaces de décisions non-planes. L'idée est de transformer les données de l'espace initial $\mathbb{R}^d$ en un espace de Hilbert $\mathbb{E}$ de dimension supérieure (possiblement infinie) dans lequel les données transformées deviennent linéairement séparables par l'application Φ.

$$\begin{aligned} \Phi: \mathbb{R}^d &\longrightarrow \mathbb{E} \\ x_i &\longrightarrow \Phi(x_i) \end{aligned} \tag{2.58}$$

L'algorithme de la Machine à Vecteur Support linéaire appliqué aux données $\Phi(x_i)$ dans l'espace $\mathbb{E}$ produit des surfaces de décisions non-planes dans l'espace $\mathbb{R}^D$ (mieux appropriées aux données initial pour un choix judicieux de Φ). Cette procédure peut être rendue très efficace en utilisant une astuce permettant d'effectuer les calculs nécessaires à l'algorithme dans l'espace initial $\mathbb{R}^d$ sans passer explicitement par $\mathbb{E}$.

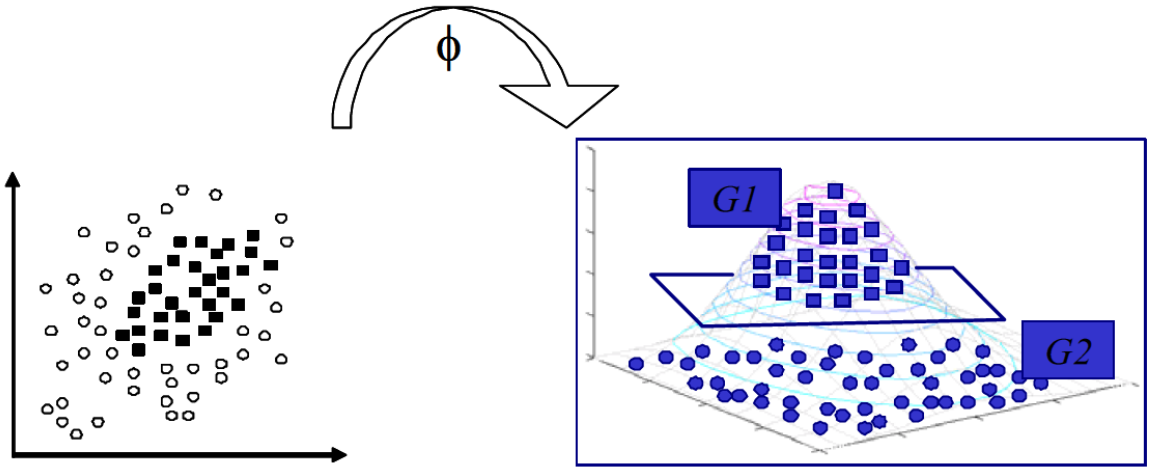


Fig. 2.7: Exemple de prolongement non-linéaire.

Du fait que les données apparaissent dans tous les calculs uniquement sous forme de produits scalaires $x_i^T x_j$, il suffit de trouver une façon efficace de calculer $\Phi(x_i)^T \Phi(x_j)$. Cela est réalisé en faisant appel à une fonction *noyau* $k(x_i, x_j)$ définie par :

$$K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j) \tag{2.59}$$

Tout le développement présenté dans les sections précédentes reste valable en remplaçant simplement les termes $\boldsymbol{x_i^T x_j}$ *par* $\boldsymbol{K(x_i, x_j)}$.

La nouvelle fonction de décision est définie par le signe de :

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b \tag{2.60}$$

2.7.1 La norme L2-SVM et l'astuce noyau

Soit l'échantillon d'apprentissage $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{l}$, appliquons le fonction Φ à chaque point x_i de $\mathcal{D}$ nous obtenons un nouveau échantillon d'apprentissage $\mathcal{D}_\Phi = \{(\Phi(x_i), y_i)\}_{i=1}^{l}$.

La fonction objective primale (2.39) pour le L2-SVM est donnée par:

$$\begin{aligned} &\min_{w,b,\xi_i} \tau(w, \xi_i) = \min_{w,b,\xi_i} \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{l} (\xi_i)^2\right) \\ &sous\ la\ contrainte \qquad y_i(w.x_i + b) \geq 1 - \xi_i \\ &\xi_i \geq 0, \qquad i = 1,2,\dots,l \end{aligned} \tag{2.39}$$

et la fonction objective duale *Lagrangien* (2.43) est donnée par:

$$\max_{\alpha} \mathcal{L}(w,b,\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i\alpha_j y_i y_j \left(x_i^T x_j - \frac{1}{2C}\delta_{ij}\right)$$
$$sous\ les\ contraintes \quad \alpha_i \geq 0, \quad \sum_{i=1}^{l} \alpha_i y_i = 0, \quad i = 1,2,\dots,l \qquad (2.43)$$

On extrait le noyau de cette fonction suivant les théorème et définitions précédentes. Le noyau sera:

$$\begin{aligned} K_2(x_i,x_j) &= \Phi(x_i)^T\Phi(x_j) - \frac{1}{2C}\delta_{ij} \\ &= K(x_i,x_j) - \frac{1}{2C}\delta_{ij} \end{aligned} \qquad (2.61)$$

et la fonction objective duale est représenté par:

$$\begin{aligned} &\max_{\alpha} \mathcal{L}(w,b,\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i\alpha_j y_i y_j\, K_2(x_i,x_j) \\ &sous\ les\ contraintes \quad \alpha_i \geq 0, \quad \sum_{i=1}^{l} \alpha_i y_i = 0, \quad i = 1,2,\dots,l \end{aligned} \qquad (2.62)$$

Ainsi, le vecteur poids a pour valeur

$$w = \sum_{i=1}^{l} \alpha_i y_i \Phi(x_i) \qquad (2.63)$$

et le biais a pour valeur

$$\begin{aligned} b_i &= y_i - w^T\Phi(x_i) \\ &= y_i - \sum_{i=1}^{l} \alpha_i y_i \Phi(x_i)^T\,\Phi(x_i) \\ &= y_i - \sum_{i=1}^{l} \alpha_i y_i\, K(x_i,x_j) \end{aligned} \qquad (2.64)$$

où

$$b = avg_{0<\alpha_i<c}\{y_i - \sum_{i=1}^{l} \alpha_i y_i\, K(x_i,x_j)\} \qquad (2.65)$$

L'avantage d'une telle approche réside dans le fait qu'il n'est pas nécessaire de connaitre Φ explicitement. Il suffit d'obtenir des noyaux convenables.

2.8 L'apprentissage d'une Machine à Vecteurs Supports

Nous avons vu qu'une Machine à vecteurs Supports (SVM) peut être un classifieur linéaire ou non linéaire, et fournit une fonction mathématique qui peut être utilisé pour faire la différence entre différents types d'objets qui deviennent des classes.

Pour cela, on adopte une méthode d'optimisation séquentielle minimale (SMO-Sequentiel Minimal Optimization) pour apprendre un SVM. La *ligne 4* de *l'algorithme 1* que nous présentons dans cette section indique une étape d'optimisation. Le détail complet de cet algorithme est expliqué dans [Plat (99)]. Au cœur de l'algorithme SMO, il y a une paire de

multiplicateurs de Lagrange α_i et α_j qu'il faut optimiser. Un facteur important est le choix du noyau K qui est une fonction mathématique qui permet de changer l'espace des données de telle sorte qu'ils soient linéairement séparables. La partie complexe d'utilisation des Machines à Vecteurs Supports est de choisir le bon noyau pour le problème à traiter.

L'optimisation est basée sur des contraintes imposées qui dépendent des conditions de Karush-Kuhn-Tucker (KKT) et du paramètre de marge souple C. Les conditions de Karush-Kuhn-Tucker sont un ensemble de contraintes selon lesquelles le modèle SVM est défini à l'intérieur d'une tolérance d'erreur donnée quand l'apprentissage mène à une optimisation [Pede (06)].

Algorithm 1 Apprentissage d'un SVM
on charge L'échantillon d'apprentissage $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{l}$
1: initialization: $0 \longrightarrow \alpha$ ou *valeur partille* $\longrightarrow \alpha$, et C une petite Valeur (exemple 10)
2: **Repeter**
3: ***Pour*** *tout* x_i, y_i, x_j , y_j ***faire***
4: *Optimiser* α_i *et* α_j
5: ***fin pour***
6: ***Jusqu'à*** *aucun changement sur* α_i
Remarque: *Retenir que les vecteur support* $(\alpha_i > 0)$

2.9 Les Machines à Vecteurs Supports Multi-classes

Les Machines à Vecteurs Supports multi-classes sont des modèles de l'apprentissage de conception relativement récents dont l'étude est actuellement en plein essor. Ceci résulte en premier lieu du fait que la communauté des théoriciens, qui avait jusqu'à un passé récent consacré l'essentiel de ses forces au développement de la théorie statistique du calcul des dichotomies, exprime à présent un intérêt de plus en plus marqué pour le cas multi-classe dont elle perçoit mieux les spécifités. Cette situation nouvelle fait naître un besoin celui de disposer d'une étude synthétique sur les Machines à Vecteurs Supports multi-classes ou plus généralement l'utilisation des Machines à Vecteurs Support pour la discrimination à

catégories multiples. Pour cela, on annonce deux approches que nous détaillerons dans ce qui suit.

2.9.1 L'approche multi-classe « *une-contre-toute* »

Cette approche est la plus simple et la plus ancienne des méthodes de décomposition. Dans ce cas, le $m^{éme}$ classifieur est destiné à distinguer la catégorie d'indice m de toutes les autres. Pour affecter un exemple, on le présente donc à $\mathcal{M}$ classifieur, et la décision s'obtient par application du principe « *winner-takes-all* »: l'étiquette retenue est celle associée au classifieur ayant renvoyé la valeur la plus élevé.

L'extension au cas multi-classe par les Machines à Vecteurs Supports a été proposée par [Vapn (95)]. A toute classe m est associé un hyperplan $\mathcal{H}(w_m, b_m)$ défini par la fonction de décision $f_m(x) = w_m^T \Phi(x) + b_m$ dont le rôle est de discriminer entre les observations de la classe m et de l'ensemble des autres classes.

Une observation $\Phi(x)$ sera donc affectée à la classe m^* selon la règle de décision discrète,

$$\begin{aligned} m^* &= arg \max_{1 \le m \le \mathcal{M}} h_m(x) \\ avec \quad h_m(x) &= sign\big(f_m(x)\big) \end{aligned} \tag{2.66}$$

Afin de bien comprendre cette généralisation, considérons le cas binaire où $\mathcal{Y} = \{-1, +1\}$. A chaque classe est associé un hyperplan défini par les fonctions de décision $f_m(x) = w_m^T \Phi(x) + b_m$ avec $m = -1, +1$.

$$\begin{aligned} \mathcal{H}_{+1} &= \{\Phi(x) \in \mathbb{R}^D ; f_{+1}(x) = 0\} \\ \mathcal{H}_{-1} &= \{\Phi(x) \in \mathbb{R}^D ; f_{-1}(x) = 0\} \end{aligned} \tag{2.67}$$

$\mathcal{H}_{+1}$ est associé à la classe (+1) et $\mathcal{H}_{-1}$ est associé à la classe (-1). Géométriquement, les deux hyperplans sont confondus, en revanche $f_{+1}(x) = -f_{-1}(x)$. Ainsi, si nous posons $w = w_{+1} - w_{-1}$ et $b = b_{+1} - b_{-1}$, nous pouvons réduire le problème à la recherche d'un seul hyperplan et c'est exactement ce qu'on fait dans le cas binaire.

Donc, le but est de construire $\mathcal{M}$ classifieurs binaires à vecteurs supports où $\mathcal{M}$ est le nombre total des classes. L'apprentissage du $m^{ème}$ classifieur à vecteurs supports s'effectue en considérant tous les exemples (données) de la $m^{ème}$ classe dans la région positive et tous les autres exemples dans la région négative. Le $m^{ème}$ classifieur à vecteurs supports s'obtient en résolvant le problème de Crammer [Cram (00)] où une approche a été proposée par la formulation suivante:

$$\begin{gathered}\min_{w_1,\dots,w_{\mathcal{M}},\xi} \frac{1}{2}\sum_{m=1}^{\mathcal{M}} w_m^T w_m + C\sum_{i=1}^{l}\xi_i \\ sous\ la\ contrainte \quad w_{y_i}^T\Phi(x_i) - w_m^T\Phi(x_i) \geq e_i^m - \xi_i \\ i = 1,2,\dots,l \qquad m = 1,2,\dots,\mathcal{M}\end{gathered} \tag{2.68}$$

où

$$e_i^m = \begin{cases}0 & if\ y_i = m\\ 1 & if\ y_i \neq m\end{cases} \tag{2.69}$$

On note que si $y_i = m$, la contrainte est la même que $\xi_i \geq 0$.

La fonction de décision pour prédire l'étiquette d'une instance $\Phi(x)$ est :

$$\arg\max_{m=1,\dots,\mathcal{M}} w_m^T\Phi(x) \tag{2.70}$$

La fonction objective duale du problème (2.68) est:

$$\begin{gathered}\alpha \qquad \min\frac{1}{2}\sum_m^{\mathcal{M}}\sum_i^l\sum_j^l \Phi(x_i)^T\Phi(x_j)\alpha_i^m\alpha_j^m + \sum_{i=1}^l\sum_{m=1}^{\mathcal{M}}\alpha_i^m e_i^m \\ sous\ la\ contrainte \quad \sum_{m=1}^{\mathcal{M}}\alpha_i^m = 0, \qquad i = 1,2,\dots,l \\ \alpha_i^m \leq C_{y_i}^m, \quad o\grave{u} \quad C_{y_i}^m = \begin{cases}0 & si\ \ y_i \neq m\\ C & si\ \ y_i = m\end{cases}\end{gathered} \tag{2.71}$$

où $\left[\alpha_1^1,\dots,\alpha_1^{\mathcal{M}},\dots,\alpha_l^1,\dots,\alpha_l^{\mathcal{M}}\right]^T$.

Après résolution du problème (2.71) par dérivation partielle sur w, on peut calculer:

$$w_m = \sum_{i=1}^l \alpha_i^m\Phi(x_i), \qquad m = 1,2,\dots,\mathcal{M} \tag{2.72}$$

On vient de traiter le problème multi-classes de la norme L1-SVM, et on peut l'étendre au problème multi-classes de la norme L2-SVM en changeant le terme ξ_i par ${\xi_i}^2$ dans (2.68). Ainsi, le problème primal devient:.

$$\begin{gathered}\min_{w_1,\dots,w_{\mathcal{M}},\xi} \frac{1}{2}\sum_{m=1}^{\mathcal{M}} w_m^T w_m + C\sum_{i=1}^{l}{\xi_i}^2 \\ sous\ la\ contrainte \quad w_{y_i}^T\Phi(x_i) - w_m^T\Phi(x_i) \geq e_i^m - \xi_i \\ i = 1,2,\dots,l \qquad m = 1,2,\dots,\mathcal{M}\end{gathered} \tag{2.73}$$

La contrainte $w_{y_i}^T\Phi(x_i) - w_m^T\Phi(x_i) \geq e_i^m - \xi_i$ quand $m = y_i$ peut-être éliminé car pour le cas de la norme L2-SVM, $\xi_i \geq 0$ est maintenu optimal sans cette contrainte.

Nous dérivons le forme Lagrangien du problème primal (2.73) par:

$$\begin{gathered}\alpha \quad min\frac{1}{2}\sum_m^{\mathcal{M}}\sum_i^l\sum_j^l \Phi(x_i)^T\Phi(x_j)\alpha_i^m\alpha_j^m + \sum_{i=1}^l\sum_{m=1}^{\mathcal{M}}\alpha_i^m e_i^m + \sum_{i=1}^l\frac{\left(\alpha_i^{y_i}\right)^2}{4C} \\ sous\ la\ contraite \quad \sum_{m=1}^{\mathcal{M}}\alpha_i^m = 0, \qquad i = 1,2,\dots,l \\ \alpha_i^m \leq 0, \ \ i = 1,\dots,l \quad m = 1,\dots,\mathcal{M} \qquad m \neq y_i\end{gathered} \tag{2.74}$$

La fonction objective duale de Lagrange de (2.73) est:

$$\mathcal{L}(w_1,\dots,w_{\mathcal{M}},\xi,\hat{\alpha}) = \frac{1}{2}\sum_{m=1}^{\mathcal{M}} w_m^T w_m + C\sum_{i=1}^{l}\xi_i^2 - \sum_{i=1}^{l}\sum_{m=1}^{k}\hat{\alpha}_i^m\left(w_{y_i}^T\Phi(x_i) - w_m^T\Phi(x_i) - e_i^m + \xi_i\right) \tag{2.75}$$

où $\hat{\alpha}_i^m \geq 0, m = 1,\dots,\mathcal{M}, i = 1,\dots,l,$ sont les multiplicateurs de Lagrange

Le problème dual de Lagrange de la fonction objective est :

$$\max_{\hat{\alpha}:\hat{\alpha}_i^m\geq 0,\forall i,m}\left(\min_{w_1,\dots,w_{\mathcal{M}},\xi}\mathcal{L}(w_1,\dots,w_{\mathcal{M}},\xi,\hat{\alpha})\right) \tag{2.76}$$

Pour minimiser $\mathcal{L}$ sous $\hat{\alpha}$ fixe, on réécrit le terme en fonction de Lagrange

$$\begin{aligned}&\sum_{i=1}^{l}\sum_{m=1}^{\mathcal{M}}\hat{\alpha}_i^m w_{y_i}^T\Phi(x_i)\\ &= \sum_{m=1}^{\mathcal{M}}\sum_{i:y_i=m}\sum_{s=1}^{\mathcal{M}}\hat{\alpha}_i^s w_{y_i}^T\Phi(x_i)\\ &= \sum_{m=1}^{\mathcal{M}} w_m^T\sum_{i=1}^{l}(1-e_i^m)\sum_{s=1}^{\mathcal{M}}\hat{\alpha}_i^s\Phi(x_i)\end{aligned} \tag{2.77}$$

et nous obtenons :

$$\begin{aligned}&\frac{\partial\mathcal{L}(w_1,\dots,w_{\mathcal{M}},\xi,\hat{\alpha})}{\partial w_m} = 0\\ \Longrightarrow\ & w_m^* - \sum_{i=1}^{l}\left((1-e_i^m)\sum_{s=1}^{\mathcal{M}}\hat{\alpha}_i^s - \hat{\alpha}_i^m\right)\Phi(x_i) = 0\\ &m = 1,2,\dots,\mathcal{M}\end{aligned} \tag{2.78}$$

$$\begin{aligned}&\frac{\partial\mathcal{L}(w_1,\dots,w_{\mathcal{M}},\xi,\hat{\alpha})}{\partial\xi_i} = 2C\xi_i - \sum_{m=1}^{\mathcal{M}}\hat{\alpha}_i^m = 0\\ &\Rightarrow \xi_i^* = \frac{\sum_{m=1}^{\mathcal{M}}\hat{\alpha}_i^m}{2C}, \quad i = 1,\dots,l\end{aligned} \tag{2.79}$$

Nous simplifions (2.78) par :

$$\alpha_i^m \equiv (1-e_i^m)\sum_{s=1}^{\mathcal{M}}\hat{\alpha}_i^s - \hat{\alpha}_i^m,\ i = 1,\dots,l \tag{2.80}$$

Cette définition est équivalente à:

$$\begin{aligned}&\alpha_i^m = -\hat{\alpha}_i^m,\ \ \forall m \neq y_i\\ &\alpha_i^{y_i} = \sum_{m:m\neq y_i}\hat{\alpha}_i^m = -\sum_{m:m\neq y_i}\alpha_i^m\end{aligned} \tag{2.81}$$

En effet, en utilisant $\sum_{m=1}^{\mathcal{M}}\alpha_i^m = 0$ et $\alpha_i^m \leq 0, \forall m \neq y_i$, nous avons $\alpha_i^{y_i} \geq 0$ pour les deux normes des problèmes duaux L1-SVM et L2-SVM.

Ainsi, nous pouvons réécrire la solution de minimisation de $\mathcal{L}$ sous $\hat{\alpha}$ fixe pour avoir :

$$w_m^* = \sum_{i=1}^{l}\alpha_i^m\Phi(x),\quad m = 1,\dots,\mathcal{M} \tag{2.82}$$

$$\xi_i^* = \frac{\hat{\alpha}_i^{y_i}+\alpha_i^{y_i}}{2C},\qquad i = 1,\dots,l \tag{2.83}$$

En conclusion, la résolution du problème (2.73) pour chaque valeur de $m \in \{1,2,\dots,\mathcal{M}\}$ donne lieu à $\mathcal{M}$ fonctions de décisions:

$$f_m(x) = {w_m}^T \Phi(x) + b_m,\ m \in \{1,2,\dots,\mathcal{M}\} \tag{2.84}$$

Pratiquement, nous résolvons le problème dual correspondant au problème (2.84) ayant exactement l variables duales. En total, nous aurons à résoudre $\mathcal{M}$ problèmes quadratiques chacun à l variables. Ainsi, le temps d'apprentissage de cette méthode croît linéairement en fonction de $\mathcal{M}$.

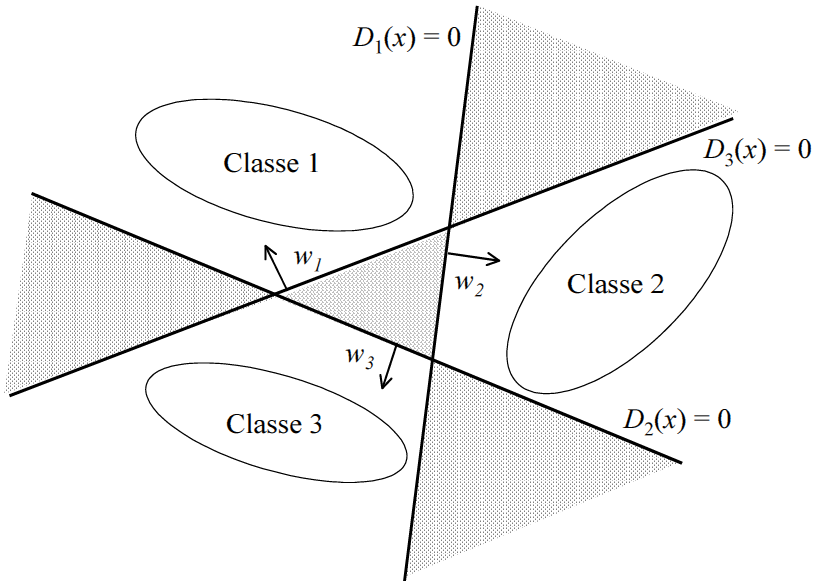


Fig. 2.8: L'espace hachuré représente la région d'ambiguïté pour l'approche multi-classe *une-contre-toute* suite à la prise de décision discrète.

Une nouvelle observation $\Phi(x)$ sera donc affectée à la classe m^* selon la règle de décision discrète (2.68) pour le cas de la norme L1-SVM ou la règle de décision discrète (2.73) pour le cas de la norme L2-SVM. Dans le cas multi-classes ($\mathcal{M} > 2$), cette égalité peut être satisfaite pour plusieurs classes. Dans ce cas, l'observation $\Phi(x)$ est dite *non-classifiable*. Toutes les observations $\Phi(x)$ non-classifiables forment *la région d'ambiguïté* dite aussi **région non-classifiable**. Cette région est schématisée dans la figure 2.8.

Afin de pouvoir classer une observation $\Phi(x)$ qui appartient à la région d'ambiguïté, la règle de décision continue a été utilisée. Cette règle est donnée par :

$$m^* = \arg \max_{1 \le m \le \mathcal{M}} f_m(x) \tag{2.85}$$

Cette approche est nommée « *le gagnant emporte le tout* ». L'inconvénient majeur de cette heuristique est qu'elle ne conserve pas les $\mathcal{M}$ frontières de séparation. Il est clair que cette heuristique a amélioré la règle de décision discrète. En revanche, elle perd totalement les capacités de généralisation des $\mathcal{M}$ hyperplans construits. Malheureusement, on ne dispose pas de borne pour l'erreur de généralisation de l'approche multi-classe *une-contre-toute*.

2.9.2 L'approche multi-classe « *une-contre-une* »

Ce schéma de décomposition a été utilisé pour la première fois dans le contexte des Machines à Vecteurs Supports par [Kreß (99)]. Il consiste à utiliser un classifieur par couple de catégories. Le classifieur indicé par le couple (u, v) (avec $1 < u < v < \mathcal{M}$) est destiné à distinguer la catégorie d'indice u de celle d'indice v. Pour affecter un exemple, on le présente donc à $C_{\mathcal{M}}^2 = \frac{\mathcal{M}(\mathcal{M}-1)}{2}$ classifieurs et la décision s'obtient habituellement en effectuant un vote majoritaire « *max win voting* ». La voix de chaque classifieur peut être pondérée par une fonction de la valeur de la sortie calculée.

L'hyperplan séparateur des classes u et v est la solution du problème d'optimisation suivant :

$$\begin{gathered} \min_{w^{uv}, \xi^{ks}, b} \frac{\|w^{uv}\|^2}{2} + C \sum_{t=1}^{l_{uv}} \xi_l^{uv}, \\ sous \ \langle w^{uv}.\Phi(x_t)\rangle + b^{uv} \ge +1 - \xi_l^{uv}, \quad si\ y_t = u \\ \langle w^{uv}.\Phi(x_t)\rangle + b^{uv} \le -1 + \xi_l^{uv}, \quad si\ y_t = v \\ \xi_l^{uv} \ge 0, \qquad \forall t \in \{1, \ldots., l_{uv}\} \end{gathered} \tag{2.86}$$

où l_{uv} est le nombre des observations issues des classes u et v.

Pratiquement, nous résolvons le problème dual correspondant au problème (2.86) ayant l_{uv} variables duales. Si chaque classe contient en moyenne $\frac{l}{\mathcal{M}}$ exemples, nous aurons à résoudre dans la phase d'apprentissage $\frac{\mathcal{M}(\mathcal{M}-1)}{2}$ problèmes quadratiques, chacun dépendant à peu près de $\frac{2l}{\mathcal{M}}$ variables.

L'approche *une-contre-une* consiste donc à construire un classifieur pour chaque paire de classes (u, v) définissant ainsi des fonctions de décisions binaires

$$\begin{gathered} h_{uv}: \Phi(\mathcal{X}) \subseteq \mathbb{R}^D \longrightarrow \{-1, +1\} \\ h_{uv}(x) = sign\big(f_{uv}(x)\big) = \begin{cases} +1 & si\ \Phi(x) \in \text{ à } la\ classe\ u \\ -1 & si\ \Phi(x) \in \text{ à } la\ classe\ v \end{cases} \end{gathered} \tag{2.87}$$

Pour des raisons de symétrie $h_{uv} \equiv -h_{uv}$ et on convient que $h_{uu} \equiv 0$ pour tout $u, v \in \{1,2,\dots,\mathcal{M}\}$. Sur la base des $\frac{\mathcal{M}(\mathcal{M}-1)}{2}$ fonctions de décisions binaires h_{uv}, nous définissons $\mathcal{M}$ autres fonctions de décisions de la façon suivante:

$$h_u(x) = \sum_{v=1}^{\mathcal{M}} h_{uv}(x), \quad u = 1,2,\dots,\mathcal{M} \tag{2.88}$$

et la règle de classification d'une nouvelle observation $\Phi(x)$ est donnée par :

$$u^* = arg \max_{1 \le u \le \mathcal{M}} h_u(x) \tag{2.89}$$

Cette règle proposée par [Frie (96)] est connue sous le nom de *vote majoritaire*, et elle a été appliquée pour la première fois avec les Machines à Vecteurs Supports par [Kreß (99)].

Il peut arriver que la règle (2.89) soit satisfaite par plus d'une classe. Ainsi, une nouvelle observation $\Phi(x)$ est dite non-classifiable et elle appartient à la région d'ambiguïté. Cette région est présentée par la figure 2.9. Toute observation située dans la région d'ambiguïté est classée arbitrairement dans l'une des classes vérifiant la règle (2.89).

Les avantages majeurs de cette combinaison sont :

- La conservation de bonnes parties des $\frac{\mathcal{M}(\mathcal{M}-1)}{2}$ hyperplans préalablement construits ;
- La diminution de la région d'ambiguïté relativement à l'approche *une-contre-toute*.

En revanche, son erreur de généralisation n'a pas encore de majorant.

Plusieurs méthodes ont été proposées pour combiner les différents classifieurs issus de toutes les paires de classes. Chaque architecture vise à réduire le *temps d'apprentissage* et le *temps de classification* d'une nouvelle observation tout en améliorant les capacités de généralisation de la machine. Dans ce qui suit, nous présentons une méthode très utilisée qui donne une bonne combinaison au niveau de la prise de décision au niveau de la région d'ambiguïté. Nous l'avons adopté pour notre système.

Méthode du Graphe de Décision Acyclique

[Plat (00] ont proposé une structure d'arbre de décision pour combiner les $\frac{\mathcal{M}(\mathcal{M}-1)}{2}$ classifications binaires construites selon la décomposition *une-contre-une*. La phase d'apprentissage de la méthode du Graphe de Décision Acyclique (**D**ecision **A**cyclic **G**raph-**DAG**) est exactement la même que celle pour le *vote majoritaire*. Elle consiste à construire toutes les $\frac{\mathcal{M}(\mathcal{M}-1)}{2}$ classifications binaires. Par contre, l'étape test utilise un graphe binaire,

enraciné , orienté et acyclique ayant $\frac{\mathcal{M}(\mathcal{M}-1)}{2}$ nœuds intérieurs répartis sur $(\mathcal{M}-1)$ couches et $\mathcal{M}$ feuilles formant la dernière couche. Chaque nœud correspond à une classification binaire de la $u^{ème}$ et la $v^{ème}$ classes et chaque feuille désigne une classe.

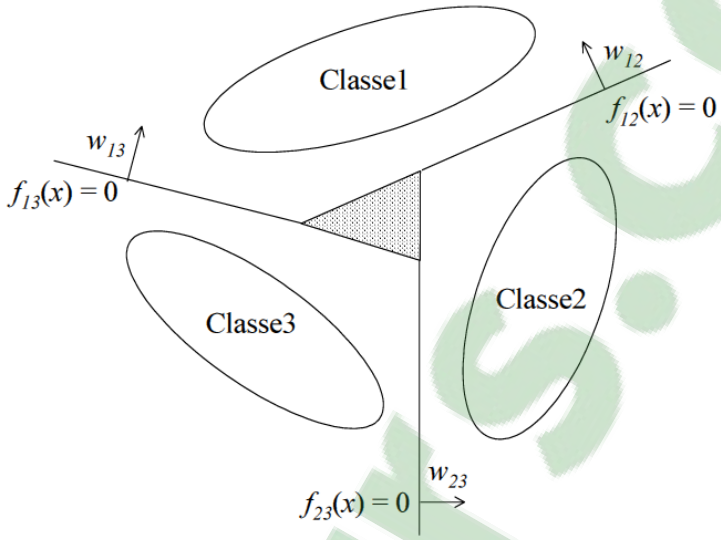


Fig. 2.9: La région d'ambiguïté hachurée est réduite pour l'approche multi-classe *une-contre-une*.

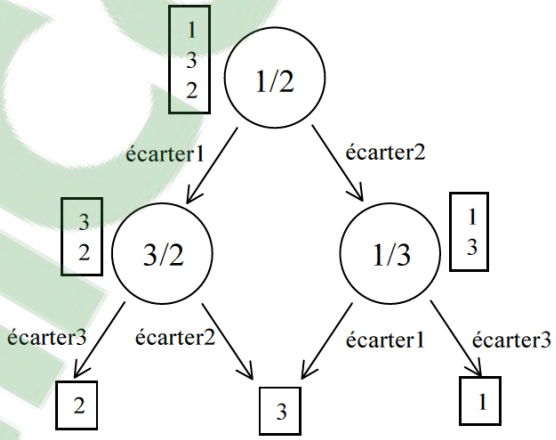


Fig. 2.10: Graphe de Décision Acyclique à trois classes.

Une nouvelle observation $\Phi(x)$, partant du nœud racine, circule d'un nœud à un autre jusqu'à atteindre une feuille qui indiquera sa classe d'appartenance. Au niveau de chaque nœud, l'observation $\Phi(x)$ se retrouve devant un choix binaire : passer à gauche ou à droite. Ce choix

dépend de la décision de classification binaire prise au niveau de ce nœud. Une illustration graphique du Graphe de Décision Acyclique pour $\mathcal{M}$ = 3 est donnée par la figure 2.10.

Cette architecture peut être vue sous forme d'une liste à $\mathcal{M}$ classes de laquelle chaque nœud élimine une classe. Cette liste est initialisée avec toutes les classes. Une nouvelle observation $\Phi(x)$ sera évaluée par le nœud de décision binaire correspondant au premier et au dernier éléments de la liste. Lorsque ce nœud préfère l'une des deux classes confrontées, l'autre sera éliminée de la liste et l'algorithme se poursuit pour la nouvelle liste. Cet algorithme s'arrête quand la liste est réduite à une seule classe, celle-ci sera attribuée à $\Phi(x)$. Ainsi, pour un problème à $\mathcal{M}$ classes, $(\mathcal{M} - 1)$ noeuds de décision binaire sont évalués dans le but de classer toute nouvelle observation.

L'avantage de la méthode du Graphe de Décision Acyclique par rapport aux autres approches multi-classes est que, grâce à sa structure particulière, son erreur de généralisation est bornée. En outre, son temps de classification est réduit comparativement au *vote majoritaire*. En revanche les capacités de généralisation de la méthode du Graphe de Décision Acyclique dépendent de l'ordre de la liste initiale sur laquelle il agit. Pour un même problème à $\mathcal{M}$ classes, il y a en tout $\frac{\mathcal{M}!}{2}$ structures différentes issues de la méthode du Graphe de Décision Acyclique. L'ordre de la liste initiale du haut vers le bas est le même que celui qu'on retrouve sur les feuilles de droite à gauche. Ainsi, pour chaque Graphe de Décision Acyclique, la région d'ambiguïté est partagée sur les feuilles internes.

Une illustration graphique du cas $\mathcal{M}$ = 3 est donnée par la figure 2.11.

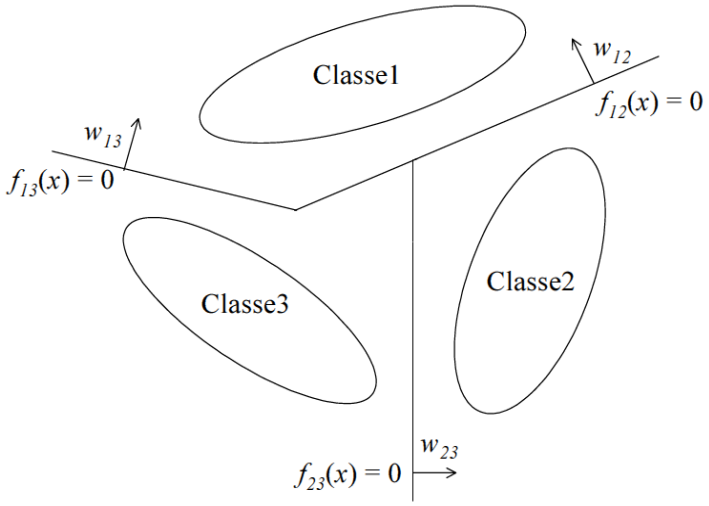


Fig. 2.11: La méthode du Graphe de Décision Acyclique qui favorise la feuille du milieu en y affectant la région d'ambiguïté.

2.10 Conclusion

Les Machines à Vecteurs Supports ont connu beaucoup de succès sur des applications provenant de domaines très variés, surtout dans les cas où le nombre de variables explicatives est largement supérieur à la taille de l'échantillon d'apprentissage.

Les Machines à Vecteurs Supports ont été introduites comme des extensions non linéaires d'un séparateur linéaire: l'hyperplan de marge maximale. Ils réalisent des séparations non-linéaires dans l'espace des données d'apprentissage à partir de séparations linéaires dans un espace transformé de dimension potentiellement grande, et ce grâce à l'idée des noyaux de *Mercer*.

L'élégance de la construction des Machines à Vecteurs Supports ne masque pas les difficultés de leur mise en œuvre. La minimisation quadratique est une tâche délicate lorsqu'il s'agit de traiter des problèmes de grande taille. En plus, le problème de réglage des paramètres C et ceux des noyaux semble lourd à résoudre. Pour ce faire, on fait souvent appel à des méthodes numériques minimisant l'une des bornes de généralisation par rapport à ces paramètres.

Aussi, dans ce chapitre nous avons décrit les méthodes usuelles pour effectuer des tâches de discrimination à catégories multiples avec les Machines à Vecteurs Supports. Elles s'appuient soit sur une machine unique (« *approches une-contre-toute* »), soit sur un ensemble de machines binaires (« *approche une-contre-une* »), utilisées dans le cadre d'une méthode de décomposition.

Il faut noter, suivant la littérature, que les performances relatives aux différentes approches peuvent fortement dépendre du problème particulier auquel elles sont appliquées. Comme tous les autres aspects de la méthodologie de l'apprentissage, aucune approche (raisonnable) ne domine toutes les autres dans toutes les situations (raisonnables). Dans ces conditions, le meilleur argument en faveur des Machines à Vecteurs Supports multi-classes est leur capacité à traiter les problèmes dans lesquels l'ensemble résultat est un ensemble d'éléments structurés.

Chapitre 3

Identification Automatique des Langues (IAL)

3.1 Introduction

L'identification automatique des langues consiste à reconnaitre automatiquement la langue qui est parlée dans un tour de parole. Les humains et machines peuvent utiliser de nombreuse sources d'informations pour distinguer une langue d'une autre. En ce sens, des expériences perceptuelles ont été menées pour déterminer l'aptitude des humains à discriminer les langues [Muth (93)].

L'identification des langues est un domaine de recherche où il devrait tirer parti d'un plus grand nombre possible de domaine d'expertise. Ces recherches s'appuient sur l'ingénierie du traitement automatique de la parole, mais elle est maintenant rejointe par la psycholinguistique, et bien d'autres disciplines devraient suivre.

La première motivation de l'identification des langues est d'utiliser des instruments qui puissent traiter le signal de parole le plus directement possible sans utiliser des ressources linguistiques trop importantes et complexes. Les discours énoncés par des locuteurs doivent être analysés dans leur ensemble afin d'en extraire une signature de chaque langue. Cette tâche est ardue puisque le signal de parole contient de multiples informations comme le sexe, l'âge, les accents et le statut émotionnel qui s'ajoutent à la langue parlée. En outre, le signal acoustique est souvent altéré par des bruits dus à l'environnement ou au canal de communication.

L'architecture des systèmes d'Identifications Automatiques de Langues (IAL) se déduit directement des sources d'informations retenues pour la discrimination accessibles d'un point de vue automatique.

3.1.1 Les enjeux en IAL

Nous sommes à une époque de communication multilingue que ce soit entre êtres humains (au sein de grandes mégapoles ou par téléphones interposés). A cela, s'ajoute une demande croissante pour des applications de traitement automatique de la parole qui intervient de plus en plus dans notre quotidien (dictée automatique, lecture de ses messages à distance, translation de la parole en texte, …).

Ce constat implique le développement des applications capables de gérer plusieurs langues et/ou d'identifier une langue parmi d'autre. De tels systèmes peuvent être envisagés dans des tâches d'assistance au dialogue humain (exemple la réservation dans les hôtels), au sein des

interfaces hommes machines. Un système d'identification des langues prendra également sa place en amont d'un système automatique de traduction où le nombre de langues à traduire est très élevé.

L'objectif d'un logiciel d'Identification Automatique des Langues est de pouvoir diriger un locuteur inconnu avec un système de traitement du langage (humain ou automatique) adapté à sa langue. Un tel système prendrait sa place dans le domaine militaire en identifiant la langue d'un sujet étranger. Aussi, une application prévue pour l'identification des langues pourrait facilement être adaptée à l'enseignement des langues.

3.1.2 Les premières études de l'IAL

Les premiers systèmes d'Identification Automatique des Langues ont été développés par la défense américaine [Duta (00)]. Beaucoup d'idées avaient été examinées mais elles ont un peu d'impact sur les recherches actuelles. Les premières études étaient basées sur les différences spectrales entre les langues. Le spectre pouvait être représenté de plusieurs manières: par des vecteurs caractéristiques à partir des formants ou encore par l'utilisation des coefficients scepstraux issues de l'analyse spectrale du signal de parole [Ziss (01)]. La similarité spectrale est ensuite calculée par une distance euclidienne, de Mahalanobis ou par le biais de l'algorithme de regroupement des *k*-moyens. Elle est calculée entre le vecteur des tests et tous ceux de la base d'apprentissage ; la distance minimum indique la langue identifiée.

Un système typique d'identification consiste d'une section d'extraction des paramètres et d'un classifieur. Suite à ce dernier, et avant de développer la conception d'un système d'Identification Automatique des Langues, développons la tâche de classification supervisée qui est à la base de tout système d'identification des langues.

3.2 Classification supervisée des données numériques

Dans cette section, nous introduisons les méthodes statistiques pour la classification automatique avec un apprentissage supervisé. Nous commençons par présenter les différences entre deux catégories d'approches pour traiter un problème de classification avec apprentissage supervisé :

- les approches dites « ***génératives*** » ou « ***informatives*** » qui incluent l'Analyse Discriminante Linéaire, les Modèles de Mélanges de lois Gaussiennes (GMM), les Modèles de Markov Cachés et les Réseaux Bayésiens.
- les approches dites « ***discriminantes*** », qui incluent la méthode des *k*-plus proches voisins, la Régressions Logistique, les Réseaux de Neurones et les Machines à Vecteurs Supports.

3.2.1 Interprétation probabiliste

De façon formelle, un classifieur assigne à une observation x une étiquette $m \in \{1, \dots, \mathcal{M}\}$ correspondant à une classe ($\mathcal{M}$ désigne le nombre de classes). Pour mesurer les performances d'un classifieur dans un contexte applicatif donné, il faut établir une mesure de coût des erreurs liée à l'application. Cette mesure $\tau: (m_s, m_r) \in \{1, \dots, \mathcal{M}\}^2 \longrightarrow \mathbb{R}^+$, peut se représenter sous forme d'une matrice liant les sorties m_s du classifieur et les étiquettes réelles m_r. Un cas particulier est le coût binaire **0**/**1**, qui vaut **0** si $m_s = m_r$ et **1** sinon. Cette fonction associe la même gravité à tous les types d'erreurs, et correspond à une matrice de coût dont les valeurs sont nulles sur la diagonale.

Probabilités et classification

Selon la théorie des probabilités, chaque observation x est générée par une variable aléatoire dont la distribution peut se décomposer d'après les règles de Bayes comme suit:

$$P(x, m) = P(m|x)P(x) = P(x|m)P(m) \tag{3.1}$$

Le but d'un classifieur est de minimiser l'espérance du coût des erreurs désignée par le «*risque global*». Le classifieur de Bayes idéal est alors celui qui renvoie les valeurs :

$$\begin{aligned} m_r^*(x) &= arg \min_{m_s} \sum_{c=1}^{\mathcal{M}} \tau(c, m_s) P(m = c|x) \\ &= arg \min_{m_s} \sum_{c=1}^{\mathcal{M}} \tau(c, m_s) P(x|m = c) \end{aligned} \tag{3.2}$$

Pour la fonction de coût **0**/**1**, cela revient à choisir la classe c qui maximise la probabilité $a\ posteriori\ P(c|x)$. En pratique, les véritables densités sont inconnues et l'on dispose d'observations d'apprentissage $\{c_i, m_i\}$ desquelles on instancie des modèles paramétriques.

3.2.2 Les approches génératives

Les approches génératives regroupent des méthodes qui utilisent les données d'apprentissage pour modéliser les densités de probabilité $P(x|m)$ de chaque classe par une famille de fonctions paramétriques. Lorsque le protocole expérimental le permet, ces méthodes peuvent aussi facilement tenir compte des probabilités $a\ priori\ P(m)$ d'apparition de chaque classe. En cas d'ignorance, il est d'usage de considérer les classes comme équiprobables $a\ priori$. Le terme "***génératif***" désigne le fait que la règle de décision déduite d'après les relations de Bayes (3.1), soit basée sur une modélisation de probabilité $P(x|m)$ qui « *génère* » les observations x pour une classe m donnée.

3.2.3 Les approches discriminantes

Les approches discriminantes regroupent une variété de méthodes statistiques qui utilisent les données d'apprentissage pour construire directement une correspondance entre les entrées $\mathcal{X}$ et les sorties $\mathcal{Y}$. Il est souvent dit que ces méthodes modélisent directement la probabilité $a\ posteriori\ P(m|x)$. Mais en fait, rares sont les méthodes discriminantes (au sens de « *non génératives* ») fondées sur une théorie probabiliste. La plupart du temps, l'accès à la probabilité $a\ posteriori$ n'est pas trivial et nécessite des hypothèses supplémentaires à celles faites lors de l'apprentissage du classifieur discriminatif. De manière générales, les approches discriminantes sont désignées par les méthodes qui ne s'intéressent qu'aux relations d'entrées-sorties $c_i \overset{?}{\rightarrow} m_i$ lors de l'apprentissage.

3.2.4 La combinaison des approches

La combinaison d'approches génératives et discriminantes pour la classification a récemment fait l'objet de nombreuses recherches en apprentissage automatique (*machine learning)* [Rain (03)]. Des classifieurs hybrides génératifs / discriminatifs ont été appliqués avec succès en bio-informatique [Jaak (98)], vision par ordinateur [Vasc (04), Frit (05)] et traitement de l'audio [More (03)]. L'idée est de tirer parti des avantages des deux types d'approches, en particulier :

- Pour les modèles génératifs, la possibilité de traiter naturellement des séquences de taille variable (avec une robustesse aux observations aberrantes).

- Pour les modèles discriminatifs, l'adéquation du critère d'apprentissage au problème de classification (minimisation d'un terme erreur adéquat, avec garanti de généralisation).

Il y a plusieurs manières de combiner une modélisation générative (G) et une modélisation discriminante (D) :

1. Deux classifieurs G et D agissent en parallèle et leurs résultat sont combinés. [Hou (03), Sche (06)].
2. G est imbriqué dans D. [Jaak (98), Fine (01), Liu (06)].
3. D est imbriqué dans G. [Gana (00), Camp (03), Scha (06)].

Il est difficile de prévoir quelle stratégie est a priori la meilleure. Généralement, combiner une démarche générative avec une démarche discriminante permet d'améliorer la robustesse par rapport à un classifieur n'utilisant qu'un des deux paradigmes. Mais la complexité calculatoire impliquée par le cumul des deux approches peut être fortement accrue. Elle dépend bien sûr de la formulation choisie pour combiner les deux approches.

3.3 Les informations de la parole pour l'IAL

3.3.1 Les indices différenciant les langues

Il y a une variété d'information que l'être humain et les machines peuvent utiliser pour distinguer un langage parmi d'autres. Généralement l'information de la parole peut être divisée au niveau ***locution*** ou au niveau ***mot***. Au niveau locution, les caractéristiques de la parole telles que l'information acoustique, phonétique et prosodique sont largement utilisées dans les systèmes d'Identification Automatiques des Langues [Ziss (01)]. Au niveau mot, la différence entre les langages peut être exploitée sur la base morphologique et syntaxique.

On définit les points cités concernant les deux niveaux comme suit:

L'**acoustique** de la parole est une représentation simple des sons brutes et peut être modélisée par les caractéristiques cepstraux tels que les coefficients MFCC (Mel Frequency Cepstral Coefficient) ou les PLP (Perceptual Linear Prediction) [Huan (01), Rabi (93)].

La **phonologie** étudie les phonèmes. Un phonème est une unité phonologique caractéristique d'une ou plusieurs langues. Seulement, ces phonèmes peuvent être traduits de plusieurs

façons acoustiques suivant la langue, l'accent et le locuteur lui-même. Une langue se distingue par un répertoire des phonèmes qu'elle emploie, mais aussi par la façon dont ces unités s'enchainent au sein du discours (phonotactique). Cependant, il arrive souvent que des phonèmes ou leurs combinaisons se retrouvent dans plusieurs langues. Dans ce cas, la fréquence d'occurrence de ces motifs permet de distinguer ces langues. Ainsi, les règles d'enchainement (phonotactiques) de ces unités varient d'une langue à l'autre. Le modèle langage *N-gramme* peut être utilisé pour modéliser les caractéristiques phonotactiques.

La **prosodie** a des données acoustiques différentes pour chaque langue (durée, tonalité, accents, etc…) tels que:

- Le **Timbre** de la voix humaine est défini comme l'ensemble des caractéristiques qui permettent de l'identifier. Généralement la qualité de la voix (grave ou aigu) rappelle celle d'un de leur proche étranger [Muth (94)-1, Stoc (96), Lorc (95)].
- L'**intonation** donne une musicalité de la langue due à la hauteur et à l'intensité des voyelles. Elle est une caractéristique paralinguistique non distinctive qui concerne l'ensemble du discours.
- Le **rythme** définit la vitesse de locution. Les auditeurs se plaignent que la vitesse est trop rapide pour certaines langues étrangères [Stoc (96), Lorc (95)].

La **morphologie** étudie la manière dont sont formés les mots et le lexique qui permettent de caractériser une langue. Chaque langue a son propre vocabulaire et sa propre manière de former les mots.

La **syntaxe** définit l'organisation des mots entre eux où les phrases sont structurées différemment selon les langues.

Ainsi, on peut classer les points énoncés en niveaux de complexité caractéristique de bas en haut. Comparée aux caractéristiques de haut niveau (cf Fig. 2.1), les caractéristiques acoustiques de bas niveau sont très faciles à obtenir, mais volatile, parce que les variations au niveau locuteur peuvent être présentes. Les caractéristiques de haut niveau tels que les caractéristiques syntaxiques sont connues pour véhiculer plus d'information de discrimination [Hier (96,97), Mend (96)], mais elles se basent sur l'utilisation d'une grande variété de codeurs vocabulaires de la parole.

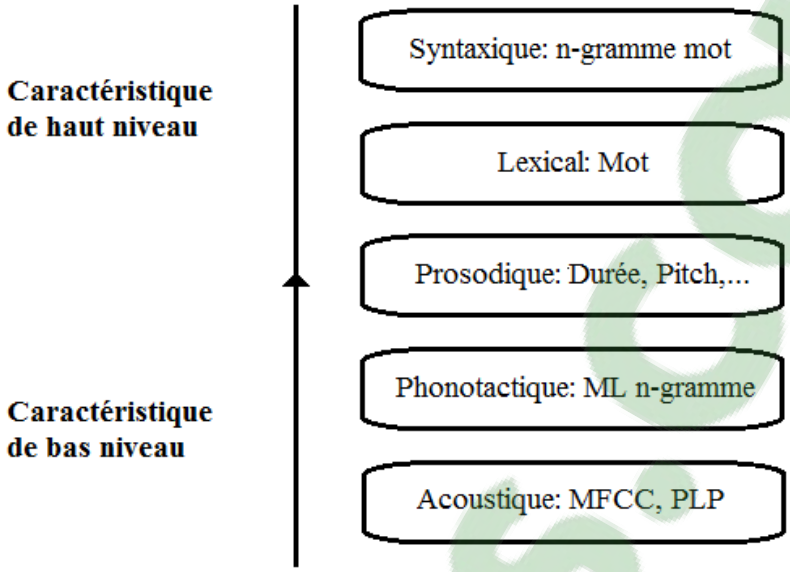


Fig. 3.1: Les niveaux de caractéristiques pour un système IAL.

Dans le cas idéal, toutes ces caractéristiques énoncées issues de la parole doivent être utilisées dans un système d'identification de langue. Suivant l'orientation de notre thèse, nous développons par la suite que les informations de la parole au niveau des locutions.

3.3.2 Les informations de niveau locution

L'information acoustique

L'information de niveau acoustique de la parole est considérée comme le niveau initial d'analyse de production de la parole, et est plus fermée sur la physique de la parole originale [Lave (94), Spen (95)]. La parole est une onde à pression longitudinale constituée de deux événements qui peuvent être discriminés à un niveau acoustique suivant leur amplitude ou leurs composants de fréquences des ondes [Lave (94)]. Deux prononciations identiques d'un même locuteur saisies par le même matériel sont plus proches mais donnent des signaux plus ou moins différents. Aussi, les mêmes prononciations énoncées par deux locuteurs différents ont des signaux différents dans la plus part des cas. Par conséquent, l'information du niveau acoustique peut être discriminatif pour extraire les différences de deux séquences de paroles, et elle est largement utilisée dans les applications de l'identification automatique de la parole et de la reconnaissance du locuteur [Huan (01), Rabi (93)].

L'information acoustique est l'une des formes la plus primitive qui peut être obtenue durant le processus de paramétrisation à partir d'un signal [Schu (06), Huan (01)]. Aussi, l'information phonotactique ***mot*** de haut niveau (*Word N-gramme*) et l'information phonotactique (*N-gramme Language Model*) peuvent être extraites à partir de l'information acoustique. Les techniques de paramétrisation les plus utilisées sont le coefficient LPC (Linear Prediction Coefficient), le coefficient MFCC (Mel Frequency Cepstral Coefficient), le coefficient PLP (Perceptual Linear Prediction) et le coefficient LPCC (Linear Prediction Cepstral Coefficient) [Jura (08), Rajm (07)].

L'information phonétique

Il est très important de distinguer entre les phonèmes et les phones. Dans la linguistique, il y a un ensemble fini de sons significatifs qui sont produits physiquement par les humains. Ces sons peuvent ne pas apparaître dans différents langages donnés. Là, on peut dire que chaque langage a son propre sous-ensemble de sons significatifs.

Un phonème est une unité de son abstrait d'un système phonétique dans un langage, et il peut véhiculer une distinction dans le sens d'un mot prononcé. En plus, les sons qui sont différents mais acceptés en tant que même phonème dans un langage sont appelés allophones (ou une phone en terme de production de son physiquement) [Yan (95), Hard (99)].

La phonétique est l'étude des sons et de la voix humaine [Lee (07), Hard (99)]. Parmi ces catégories, on énonce la phonétique articulatoire et la phonétique acoustique où cette dernière a été très utilisée dans les tâches d'identification des langages. La phonétique acoustique est l'étude de l'aspect acoustique des sons de parole tels la moyenne carrée de l'amplitude d'un signal, sa durée, sa fréquence fondamental, le spectre, etc….

L'information phonotactique

Comme la phonétique est liée à la production physique et perceptive des sons de parole, la phonologie est liée à l'étude des systèmes de son d'un langage spécifique ou d'un ensemble de langage [Lee (07)]. La phonotactique est une branche de la phonologie qui mène à des modèles de son dans un langage spécifique ; c'est-à-dire elle donne la combinaison permissible de phonèmes incluant des groupes de consonnes et des séquences de voyelles par les moyens de contraintes phonétiques [Schu (06), Lee (07)]. Il y a une grande variance dans les contraintes phonétiques à travers les langages.

L'information prosodique

L'information prosodique se réfère généralement au caractéristique de la durée d'un phonème, les modèles d'accent et l'intonation (la variation du pitch contour qui concerne la fréquence fondamentale et l'énergie qui sont issus du signal de parole par la transformée de Fourier) [Wang (06), Muth (94)-2].

Quelques phonèmes sont partagés à travers différents langages et leur durée caractéristique dépend des contraintes phonétiques du langage. Tous les langages utilisent le pitch pour véhiculer une émotion de surprise, de peur ou d'exclamation. Aussi, les variations de pitch sont utilisées pour identifier différents langages [Catf (88)]. Dans certains langages, un modèle d'accent peut déterminer le sens d'un mot, par exemple en anglais, un nom peut devenir un verbe si on place un accent sur différentes syllabes.

3.4 Description des systèmes d'IAL

3.4.1 Structure général d'un système IAL

Le système général s'apparente à un système statistique classique de reconnaissance des formes utilisant un apprentissage supervisé.

La fonction d'identification se décompose en deux phases (cf. Fig. 3.2) :

- **Apprentissage** : des paramètres sont extraits pour les signaux de parole de chaque langue. Pour chaque source d'information prise en compte, un modèle spécifique à chaque langue est appris à partir de ces paramètres.
- **Reconnaissance** : les paramètres sont extraits pour un signal de parole d'une langue inconnue. La langue la plus vraisemblable est déterminée en fonction des modèles issus de la phase d'apprentissage. Se greffe un problème de fusion si plusieurs sources d'information sont modélisées engendrant plusieurs scores.

Un comparatif des différents approches est disponible dans [Ziss (96)].

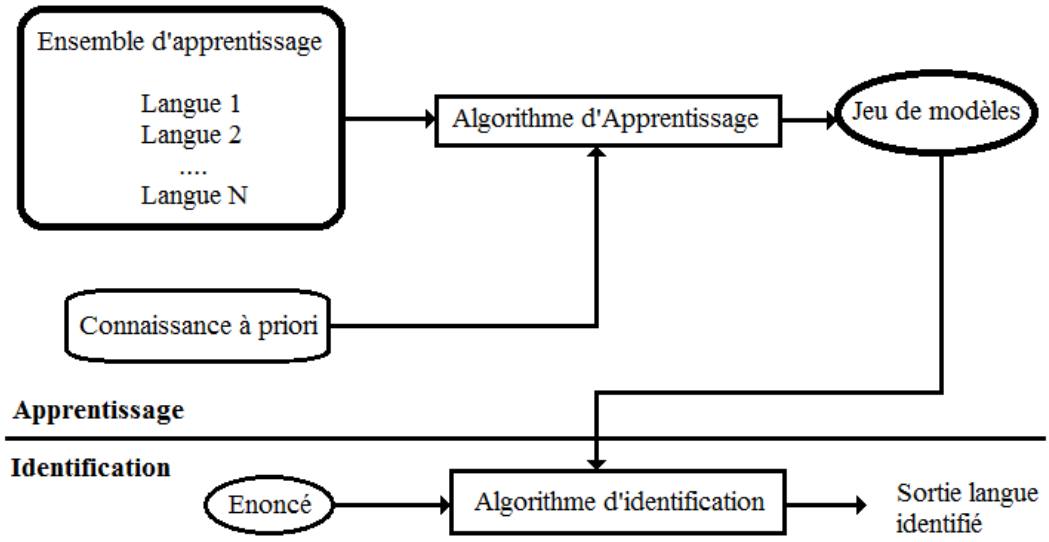


Fig. 3.2: Aspect générale d'un système IAL.

3.4.2 Modèle mathématique d'un système IAL

Suivant toute tâche d'identification, un classifieur basé sur le maximum de vraisemblance peut être employé pour traiter la tâche d'identification d'un langage (cf. Fig. 3.3).

Dans l'étape d'apprentissage, la parole est échantillonnée puis convertie en une séquence de caractéristiques représentées par des observations $X = \{x_1, x_2, \dots, x_l\}$ où l dénote le nombre de séquences ou frames correspondants. Un modèle λ est alors créé utilisant une technique de modélisation pour récupérer les caractéristiques de chaque langage par une séquence de parole. Durant la procédure de test, le même type de caractéristique de parole est extrait à partir d'une séquence de parole inconnue. La caractéristique est comparée à l'ensemble des modèles λ_m avec $m = 1,2, \dots, \mathcal{M}$ où $\mathcal{M}$ est le nombre possible des langages que le système est capable d'identifier.

La sélection finale du modèle le plus proche est:

$$\hat{l} = \arg \max_{1 \le l \le \mathcal{M}} P(\lambda_l | X) \tag{3.3}$$

En utilisant la règle de Bayes, l'équation (3.1) peut être exprimée comme:

$$\hat{l} = \arg \max_{1 \le l \le \mathcal{M}} \frac{P(X|\lambda_l)P(\lambda_l)}{P(X)} \tag{3.4}$$

Supposons que pour chaque modèle d'un langage, $P(X)$ est la même. Donc, la tâche d'identification du langage est équivalente à trouver

$$\hat{l} = \arg \max_{1 \le l \le \mathcal{M}} P(X|\lambda_l) \tag{3.5}$$

Ceci signifie que trouver le langage le plus proche pour une séquence de parole donnée est équivalent à trouver le modèle de langage dans lequel cette séquence donne la meilleure probabilité.

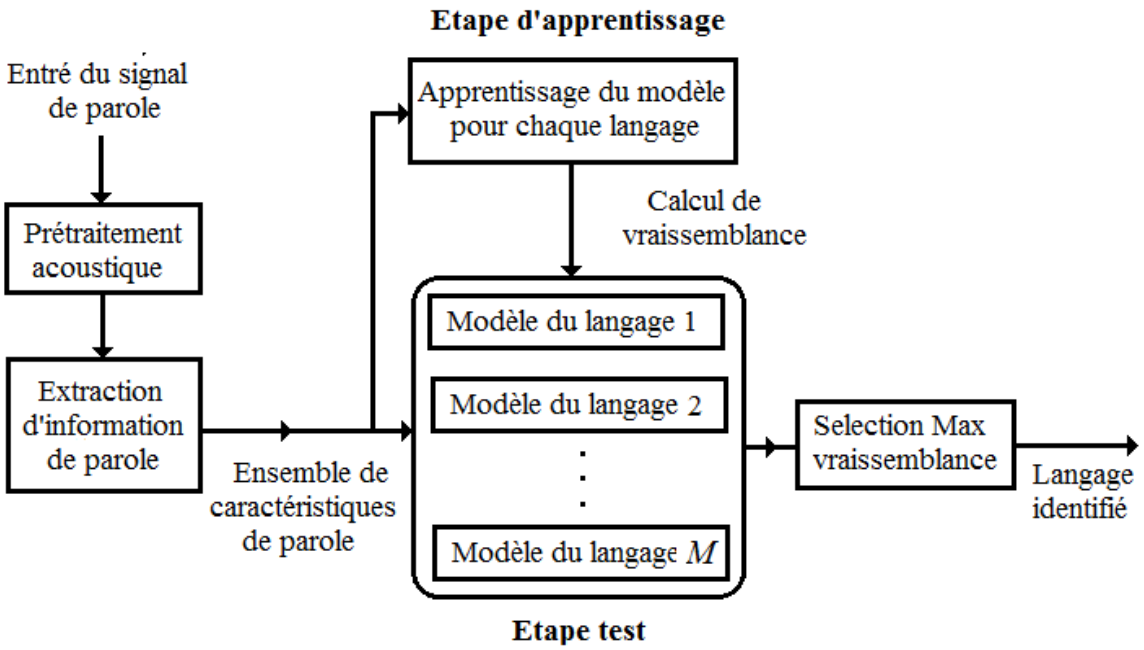


Fig. 3.3: Modèle de base d'un système IAL.

Suivant la figure 3.3, un aspect important de la tâche d'identification réside dans la façon dont les caractéristiques de cette séquence de parole sont extraites.

Dans l'identification des langages deux types de systèmes sont les plus traités dans la littérature : les systèmes acoustiques basés sur les caractéristiques du signal de parole et les systèmes basés sur la reconnaissance phonétique qu'on peut appeler aussi des systèmes phonotactiques.

3.4.3 Les systèmes acoustiques

Système basé sur les Modèles de Mélanges de lois Gaussiennes

Les premiers systèmes incluent beaucoup de système basé sur les Modèles de Mélanges de lois Gaussiennes (**G**aussian **M**ixture **M**odel-**GMM**) [Wong (02)]. Dans ce type de système, on considère que chaque vecteur suit hypothétiquement une loi de distribution dont la densité de probabilité est une somme pondérée de lois gaussiennes multidimensionnelles. Plusieurs variantes existent au niveau de l'apprentissage de modèles: le cas le plus classique consiste à apprendre un modèle par langue en utilisant les données d'apprentissages spécifiques à chaque langue.

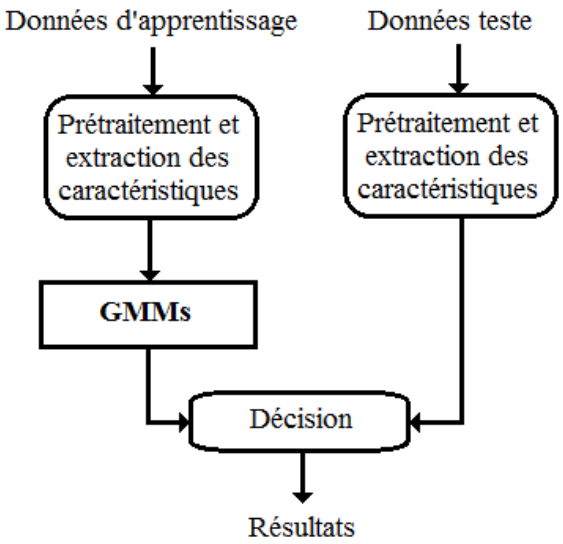


Fig. 3.4: Système IAL basé sur les Modèles de Mélanges de lois Gaussiennes.

La figure 3.4 illustre le plus simple système IAL basé sur les Modèles de Mélanges de lois Gaussiennes qui donne une approche directe pour modéliser chaque langage.

Système basé sur le Modèle du monde et les Modèles de Mélanges de lois Gaussiennes(MMG)

Ce système se base sur un apprentissage de modèle dit « ***du monde***» (**U**niversal **B**ackground **M**odel - **UBM**) comprenant toutes les données de toutes les langues. Il s'agit par la suite d'adapter les paramètres de ce modèle afin de créer des modèles spécifiques à chaque langue [Wong (04)]. Une telle méthode d'apprentissage permet d'obtenir ce que l'on appelle

couramment des modèles MMG-modèle du monde (**G**aussian **M**ixture **M**odel - **U**niversal **B**ackground **M**odel- GMM-UBM) (cf. Fig. 3.5).

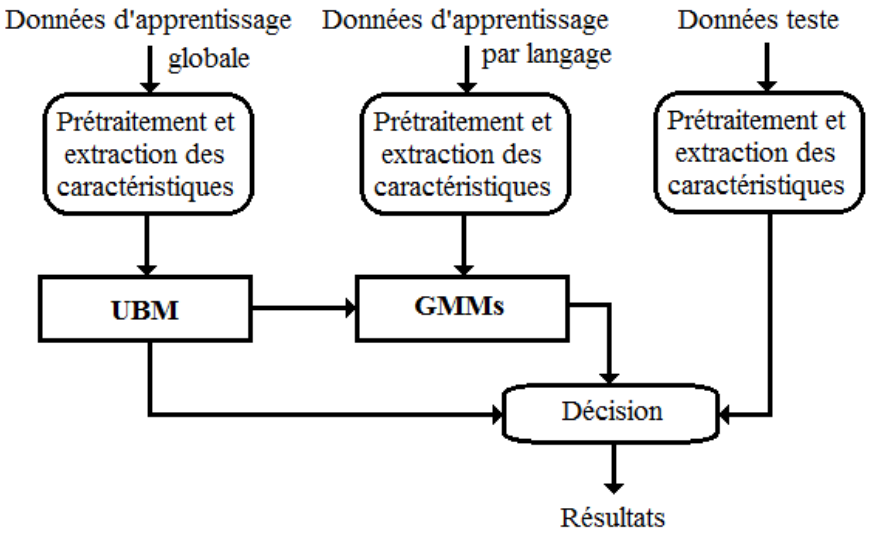


Fig. 3.5: Système IAL basé sur les Modèles de Mélanges de lois Gaussiennes par adaptation d'un modèle du monde.

Système basé sur les Modèles de Mélanges de lois Gaussiennes et les Machines à Vecteurs Supports

Dans la tâche de classification, la limite de la classe est plus critique que son centre. Les scores de vraisemblance produits par les Modèles de Mélanges de lois Gaussiennes ne décrivent pas exactement la distance à la limite de la classe mais à son centre.

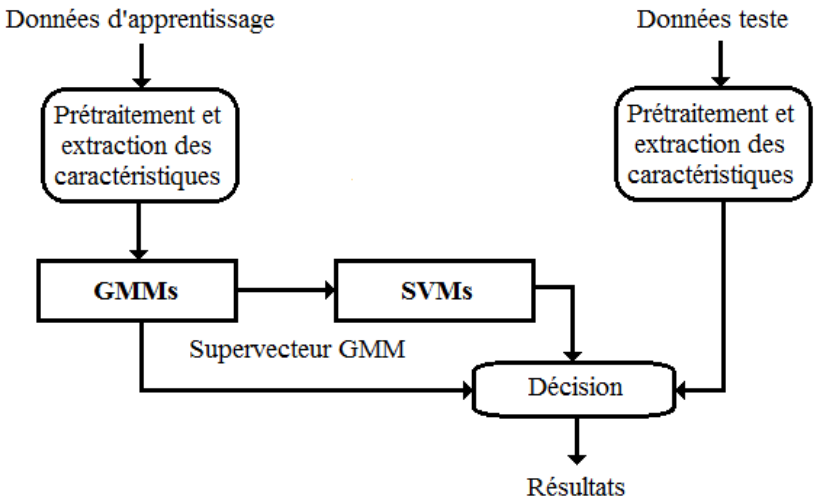


Fig. 3.6: Système IAL basé sur les Modèles de Mélanges de lois Gaussiennes et les Machines à Vecteurs Supports.

Le classifieur basé sur les Machines à Vecteurs Supports (**S**upport **V**ector **M**achines - **SVM**) est par conséquent déployé après des Modèles de Mélanges de lois Gaussiennes du langage spécifique. Dans ce système, l'information est un ensemble de super-vecteurs obtenus suite à une paramétrisation du signal de parole consistant en des valeurs moyennes issues des Modèles de Mélanges de lois Gaussiennes qui sont utilisées comme une entrée caractéristique du classifieur SVM, comme il est illustré dans la figure 3.6 [Camp (04)].

3.4.4 Les systèmes phonotactiques

Les systèmes donnant les meilleurs résultats sont ceux qui se basent principalement sur l'aspect phonologique et phonotactique. Ce sont les systèmes les plus présents dans la littérature.

En général, ces systèmes possèdent un ou plusieurs systèmes de reconnaissance de phonèmes (souvent appelés décodeurs acoustico-phonétique) constitués de Modèles de Markov Cachés (**H**idden **M**arkov **M**odels - **HMM**). Ces décodeurs acoustico-phonétiques sont le plus souvent spécifiques à l'inventaire phonétique d'une langue spécifique ; c'est pourquoi il faut en employer plusieurs en parallèle afin d'obtenir la meilleure couverture possible de l'ensemble des sons. Un décodeur acoustico-phonétique unique capable de reconnaitre des phonèmes de plusieurs langues peut également être employé sur différents systèmes d'identifications [Bena (01), Case (98)].

Le module de reconnaissance de phonèmes transforme le signal de parole en une suite d'éléments discrets formant des séquences de phonèmes. Ces séquences sont ensuite modélisées à l'aide de modèles probabilistes *N-grammes*. Ainsi, les enchaînement (de 2 ou 3 phonèmes) les plus caractéristiques des langues sont retrouvés.

Ces méthodes d'apprentissage permettent de modéliser les phonèmes en fonction de leur contexte. Cet apprentissage s'effectue sur un ensemble de signaux de paroles pré-segmentés et étiquetés en phonèmes par des expert phonéticiens. Un modèle de la langue est alors établi par un traitement statistique de leur distribution.

Ces systèmes incluent donc des connaissances a priori sur le signal de parole. Ils sont donc coûteux puisqu'ils nécessitent un traitement humain qui permet de localiser les phonèmes dans le signal de parole. Cette étape est nécessaire pour effectuer l'apprentissage des systèmes reconnaissant les phonèmes, même si ce traitement humain n'est pas utilisé lors de la reconnaissance de langue proprement dite.

Les deux systèmes les plus utilisés dans la littérature durant les dernières années sont PRLM (**P**hone **R**ecognition **L**anguage **M**odel) et PRLM Parallèle [Ziss (96)].

Système PRLM

L'idée de base dans ce système d'identification PRLM (cf. Fig. 3.7) est de modéliser une séquence de phonèmes en utilisant un modèle du langage. La séquence de phonèmes est produite par un reconnaisseur de phonèmes constitué d'un ensemble de phonèmes unifiés qui couvre la majorité des phonèmes de tous les langages cibles et est appris à partir d'un corpus conçu d'un mélange de langages. Occasionnellement, un reconnaisseur de phonèmes d'une langue quelconque peut faire l'objet d'un décodeur unifié. Seulement dans ce cas-là, nous ne serons pas sûres que tous les phones des autres langages puissent être correctement reconnus. Les HMMs (**H**idden **M**arkov **M**odels) ont été utilisés dans la plus parts des reconnaisseurs de phonèmes, cependant, on peut trouver d'autre approches tels que les reconnaisseurs de phonèmes basés sur les réseaux de neurones [Mate (05)]. La séquence de phonème est modélisée par les modèles de langage ***N-gramme*** pris comme une séquence symbolique. Durant l'évaluation, le score de confidence produit par chaque modèle de langage est utilisé pour prendre une décision finale [Ziss (96)].

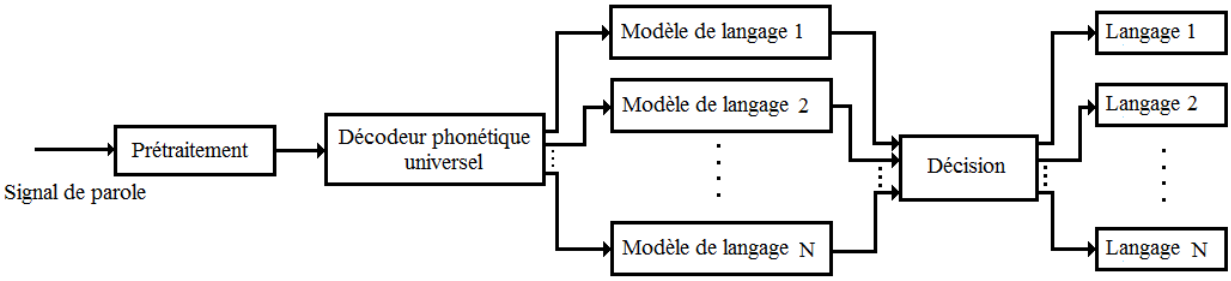


Fig. 3.7: Système IAL basé PRLM.

Système PRLM parallèle

L'une des issues clés dans le system PRLM est que certains phonèmes ne peuvent être reconnus quand il n'y a pas une part de modèles de phonèmes dans un reconnaisseur de phonème. Il est vrai que la séquence peut encore être modélisée, mais l'information clé directement associé avec ceux des phones spéciaux est inexistante. De là, les systèmes PPRLM (**P**arallel **P**hone **R**ecognition **L**anguage **M**odel) ont été introduits pour palier à ces

problèmes. Comme le montre la figure 3.8, de multiples reconnaisseurs de phonèmes sont déployés et chacun d'eux est suivi par un ensemble complet de modèles de langages [Ziss (96)]. La mesure de confidence la plus haute est choisie comme sortie qui vérifie explicitement le reconnaisseur de phonème le plus approprié.

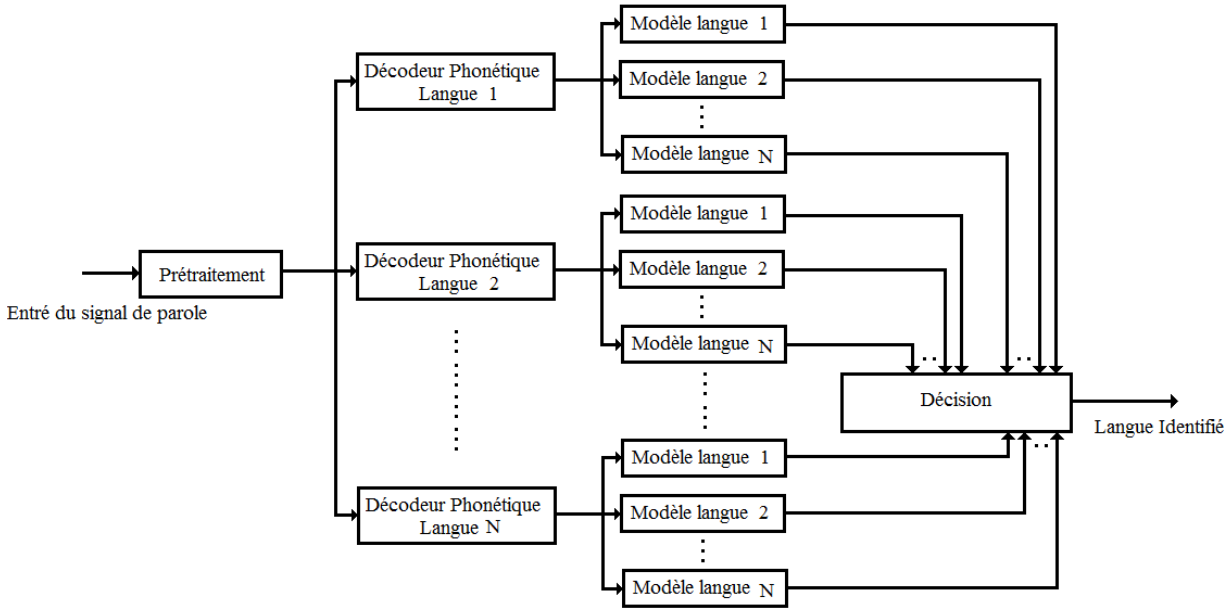


Fig. 3.8: Système IAL basé PPRLM

Malgré la haute précision donnée par ces systèmes basés sur les reconnaisseurs de phonèmes, l'apprentissage de tels systèmes a été un processus non adéquat dans plusieurs aspects. Les corpus annotés sont nécessaires à cause de ce point, malheureusement, quelque fois ces annotations ne peuvent être accomplies due au manque de ressources.

Pour prendre avantage de la haute précision des systèmes basés sur les reconnaisseurs de phonèmes et la simplicité des systèmes acoustiques, un système intéressant basé sur GMM-tokenization a été proposé. Dans ce dernier système, un reconnaisseur de phonème peut être vu comme un producteur ou un générateur de symboles qui extrait une séquence de symboles ou unités pour être représentés par des modèles de langages ou plus généralement dans les modèles de séquences. Donc, un générateur de symboles remplace potentiellement un reconnaisseur de phonèmes. Dans [Torr (02)-1], un générateur de symboles basé sur les Modèles de Mélanges de lois Gaussiennes a été conçu en ce sens.

Système basé sur la reconnaissance de parole parallèle

Un autre système de reconnaissance de parole parallèle basé sur plusieurs reconnaisseurs de parole correspondant à chaque langage cible du système [Ziss (96)]. Ce type de systèmes d'Identification Automatique des Langues atteint des résultats positifs à toutes les étapes (cf. Fig. 3.9). Cependant, il a été délaissé suite au coût important des calculs et des demandes de ressources.

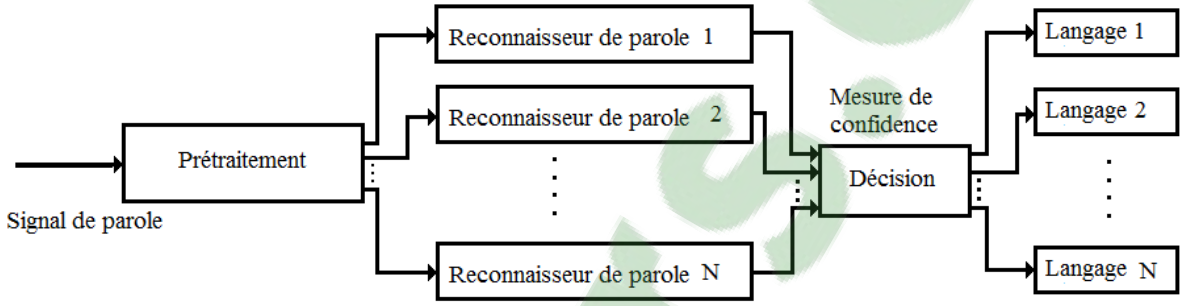


Fig. 3.9: Système IAL basé sur des reconnaisseurs de parole parallèle.

3.5 Les composants des systèmes IAL

Les systèmes d'Identifications Automatiques des Langues sont composés de quatre modules principaux interdépendants :

3.5.1 Le prétraitement

Il s'agit d'extraire du signal de parole «*pur*» (valeurs de l'amplitude échantillonnée à des fréquences de l'ordre de 16000 Hertz) des caractéristique (paramètres numériques et/ou symboliques). Un bon prétraitement fournit des paramètres dépendant des variations interlocuteurs et peu sensibles aux variations extrinsèques à l'identité du locuteur (conditions d'enregistrement, variabilité intra-locuteurs, etc.).

Le module de prétraitement doit être identique qu'il soit en amont d'une phase d'apprentissage ou de test. Il doit être robuste aux paramètres extrinsèques au locuteur. Les sous-modules typiques du prétraitement sont dans l'ordre habituel d'intervention :

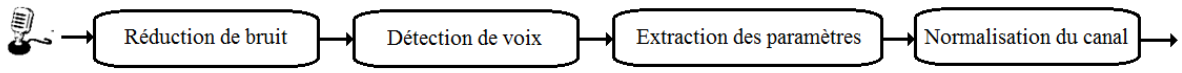


Fig. 3.10: Paramétrisation d'un système IAL

Réduction du bruit

Il s'agit de modifier le signal d'entré par l'application d'un filtre de manière à rehausser une partie de l'information pertinente amoindrie par les conditions d'enregistrement (typiquement les fréquences aigues).

Détection de voix

Cette étape est couplée à l'extraction des paramètres et sert à exclure des phénomènes extérieurs au discours du locuteur. La plupart du temps, il s'agit d'exclure des informations de certaines données correspondantes à des temps morts en terme de parole à partir des mesures d'énergies (amplitude du signal). Toutefois, étant donné la sensibilité de l'énergie aux conditions d'enregistrement, il convient de ne pas fixer de seuil a priori, mais de le choisir en tenant compte de toute une séquence (supposée contenir parole et silence). Parmi les techniques robustes, on peut citer deux lois Gaussiennes sur l'énergie proposées par [Magr (01)]. Le lecteur peut se référer à [Li (02), Zilc (04)] pour d'autres approches robustes.

Extraction de paramètres

La plupart du temps, cette extraction se fait sur des fenêtres d'échantillonnage de taille fixe (trames). Cela permet de prendre en compte l'aspect dynamique (au moins à court terme) inhérent à la parole.

Normalisation des paramètres

Cette étape intervient sur les données sélectionnées après retrait des silences. La normalisation des paramètres acoustiques en aval du prétraitement peut être utilisée pour augmenter la robustesse du système au canal utilisé (téléphone ou microphone) et réduire l'écart (mismatch) entre les conditions d'observations en phase apprentissage et celles en phase de test. Notons que si les données sont enregistrées dans les mêmes conditions alors ne pas

normaliser conduit en général à de meilleures performances, étant donné que la normalisation entraîne typiquement une perte d'information.

Les techniques de normalisation des vecteurs acoustiques couramment utilisées pour l'identification sont :

- La «**standardisation**», est une normalisation par soustraction de moyenne (**C**epstral **M**ean **S**ubstraction - **CMS**) suivie éventuellement d'une division par l'écart-type. Ces statistiques sont estimées sur l'intégralité de la séquence ou sur une fenêtre glissante de taille fixe.
- Le «**feature warping**» est une procédure qui consiste à «faire épouser» (warp) localement une distribution normale à chaque paramètre observé. Les valeurs des vecteurs paramètres normalisés sont déterminés selon le rang de chaque caractéristique (coordonnée vectorielle) sur une fenêtre centrée, glissante et de taille fixe [Pele (01)].
- Le «**short-time Gaussianization**» utilise le même principe que le feature warping, avec en amont une transformation linéaire [Xian (02)].

Parfois, les paramètres sont remodelés par un autre espace basé sur la projection d'attribut de nuisance (**N**uisance **A**ttribute **P**rojection - **NAP**) [Cam (08)] pour aider à transformer les limites d'une classe originale et améliorer les caractéristiques d'un classifieur [Wade (08), Alle (06)].

3.5.2 L'apprentissage

Il s'agit d'instancier des modèles à partir des paramètres extraits des locuteurs étiquetés ou non. L'apprentissage se fait souvent par des méthodes d'entrainement itératives.

L'apprentissage peut faire appel à une technique de modélisation quelconque. Les critères de choix portent sur :

- Les performances fournies qui reflètent la capacité de modélisation à saisir le caractère discriminant des paramètres extrait ;
- La taille des modèles (capacité mémoire requise);
- La complexité d'apprentissage et de test.

Ces deux derniers critères, d'ordre pratique, dépendent du protocole considéré pour l'application de vérification (longueur des séquences, nombre de séquences d'apprentissage, etc.).

Les algorithmes classiques d'apprentissage des modèles génératifs utilisés en traitement de la parole, les Modèles de Mélanges de lois Gaussiennes (GMMs) et les Modèles de Markov Cachés (HMMs), sont récapitulés par [Bilm (97), Brug (98), Toma (05)]. D'autres modèles stationnaires ont été appliqués avec succès à cette application, comme les Modèles de Mélanges de lois Gaussiennes hiérarchiques [Liu (02)], les Modèles de Mélanges de lois Gaussiennes structurées phonétiquement [Falt (01)] basés sur une transcription en phonèmes, les Text-Constrained GMMs [Stur (02)] basés sur une transcription en mots ou encore les réseaux bayésiens liant les coefficients cepstraux à des paramètres prosodiques [Arci (03)].

3.5.3 L'attribution de scores

Ce module est étroitement lié à la façon dont les informations ont été conçues et appris les modèles dans le module d'apprentissage. Alors que ce dernier s'applique à des séquences d'**apprentissage**, l'attribution de scores s'applique à des séquences de ***test***. Notons que l'apprentissage peut tenir compte de plusieurs séquences d'entrainement pour l'élaboration d'un modèle alors que le module de scores traite les séquence test indépendamment les unes des autres.

La stratégie d'attribution de scores est un module qui prend en entrée un modèle de langage cible et une séquence de données observées à un extrait à tester. Elle est donc intimement liée à la façon dont ont été construits les modèles: chaque type de modélisation a sa propre procédure d'attribution de scores.

3.5.4 La prise de décision

C'est un petit module qui vient directement après l'attribution de scores. Typiquement, il s'agit de comparer les scores à un seuil (fixé lors de la phase développement) pour renvoyer une décision. Notons que dans des conditions réalistes d'utilisation (conversations téléphoniques), on ne peut pas espérer 100% de réussite. Le lecteur peut se référer à [Boe (01), Bona (03)] pour des réflexions sur la limitation des systèmes biométriques vocaux.

3.6 Conclusion

Actuellement, l'approche statistique est privilégiée et un système d'identification des langues est un processus de reconnaissance qui se base sur les modèles statistiques préalablement appris pour chaque langue (cf. Fig. 3.3). Pour apprendre ces modèles, des signaux de parole de différentes langues sont présentés au système. Chaque signal de parole est traduit en une séquence de vecteurs de paramètres. Ces vecteurs sont en général calculés sur de courts extraits de parole (16 ms par exemple) afin de pouvoir faire l'hypothèse que le signal est stationnaire sur cet intervalle. Les algorithmes d'apprentissage analysent les séquences de ces vecteurs, produisent un ou plusieurs modèles de chaque langue et son destinés à être utilisés dans la phase suivante. Pendant la reconnaissance d'un signal de paroles, des vecteurs de paramètres issus de ce signal sont calculés et évalués à l'aide des modèles de chaque langue. Un score pour chaque langue est calculé et l'on estime que la langue du signal de parole présentée est celle du modèle permettant d'atteindre «le meilleur score» (plus faible ou plus fort selon le cadre).

La clef du problème consiste à construire les modèles représentant les langues. La connaissance des propriétés linguistiques des langues permet d'orienter la recherche de paramètres et de modélisation. La quantité et qualité des données disponibles pour chaque langue sont également déterminantes: un système très complexe peut nécessiter des ressources coûteuses pour chaque langue ; ce qui rend leur généralisation à un grand nombre de langues difficiles. Par exemple, un système utilisant des modélisations acoustiques peut nécessiter un étiquetage phonétique manuel pour pouvoir estimer ses modèles, ou bien un système faisant appel à des données lexicales doit disposer de corpus de très gros volumes pour pouvoir disposer d'une représentation exhaustive de la langue. En fonction des besoins, les contraintes sont plus ou moins fortes sur les données disponibles. L'idéal est de réaliser un système qui ne nécessite que quelques heures d'enregistrements non étiquetés d'une langue pour constituer les modèles.

Partie 2

Vers un système d'identification des Dialectes

L'un des défis majeurs dans le traitement de la parole dialectale est de trouver une différence parmi les dialectes existants pour leur identification vu que la majorité des mots utilisés sont issus de l'arabe standard. Dans la littérature, les travaux portent toujours sur le traitement de la parole arabe standard. On peut trouver quelques travaux sur l'arabe dialectal du moyen orient [Kirc (04), Verg (05)] mais aucune étude n'a été faite sur les dialectes du Maghreb notamment sur les dialectes Marocains, Algériens et Tunisiens. Dans la partie 2 du chapitre 1, nous avons énoncé certaines différences entre l'Arabe standard et les dialectes du Maghreb et nous avons vu que ces différences portent sur plusieurs dimensions morphologique, linguistique et phonologique. Si nous donnons pour différents locuteurs du Maghreb une même phrase à énoncer, nous trouverons que la prononciation est différente et certains phones ou phonèmes sont altérés. Les maghrébins en général parlent de l'arabe, et ces altérations au niveau des prononciations qui fassent leurs différences.

Développer un système qui permet d'identifier un dialecte parmi les autres où tous les mots utilisés par les différents dialectes ont une même racine est l'objectif de recherche menée dans cette thèse.

Chapitre 4

Systèmes de réduction de données

4.1 Introduction

Dans ce chapitre nous définirons un système de réduction de données en présentant deux approches dont le but est d'éliminer les caractéristiques de paroles non adéquates aux dialectes spécifiques. Ce système de réduction sera greffé à deux systèmes d'Identification Automatique des Langues baseline qui fera naitre une version amélioré des deux systèmes dont le détail fera l'objet du chapitre suivant.

Notre étude est basée sur une équivalence entre la formulation de la plus petite boule englobante (**M**inimal **E**nclosing **B**all - **MEB**) et la formulation de la norme L2-SVM avec un développement spécial où deux algorithmes issus de deux approches ont été développés. L'idée de base de ces deux approches est d'adopter la formulation des Machines à Vecteurs Supports Multi-classes et la formulation de la plus petite boule englobante pour réduire les données par élimination de toutes celles qui sont en dehors de cette boule.

4.2 La plus petite boule englobante

Dans le chapitre 2, nous avons vu que les Machines à Vecteurs Supports présentent en pratique de très bonnes performances en généralisation (c'est-`a-dire sur la classification de nouveaux exemples de test). Intuitivement, on sent que la marge joue en cela un rôle important. Il est en effet raisonnable de penser que si l'on parvient à séparer les exemples d'apprentissage (supposés significatifs aux classes auxquelles ils appartiennent) avec une grande marge, il y a de fortes chances pour que de nouveaux exemples soient bien classés ; ces derniers se situant dans les cas les plus défavorables à l'intérieur de la marge (ceux se retrouvant loin de la marge et du bon côté de l'hyperplan ne posent pas de problèmes).

Une autre caractéristique intrinsèque des Machines à Vecteurs Supports est qu'ils sont connus pour défier ce que l'on appelle la malédiction de la dimensionnalité («*the curse of dimensionality*») puisqu'ils sont capables de fournir de bonnes performances de classification à partir d'un nombre réduit d'exemples d'apprentissage tout en agissant dans des espaces de dimensions très élevées. Cela s'explique en partie par le fait que les Machines à Vecteurs Supports peuvent être considérés comme une réalisation du principe de minimisation du risque structurel.

On suppose dans un premier temps qu'on peut trouver la plus petite boule $\mathcal{B}(c,R)$ de centre c et de rayon R :

$$\mathcal{B}(c,R) = \{\Phi(x) \in \mathbb{R}^d;\ \|\Phi(x) - c\| < R\} \tag{4.1}$$

contenant les points $x_1, x_2, \dots, x_l$. En considérant les fonctions de décision $f_{w,b}$ telles que :

$$\begin{aligned} & f_{w,b}: \mathcal{B}(c,R) \longrightarrow \{-1,+1\} \\ & x \longrightarrow f_{w,b}(x) = signe(w.\Phi(x) + b) \\ & avec\ la\ contrainte\ \ \|w\| \leq A \end{aligned} \tag{4.2}$$

$A \in \mathbb{R}_+$, une structure sur les hyperplans est introduite. Il est montré dans [Vapn (95)] que les fonctions de décisions ainsi construites ont des dimensions VC, h vérifiant:

$$h \leq R^2 A^2 \tag{4.3}$$

La contrainte $\|w\| \leq A$ permet ainsi de contrôler la dimension VC des classieurs obtenus et de déterminer une borne sur le risque fonctionnel. Pour cela h est estimée par [Scho (02)]:

$$h \approx R^2 \|w\|^2 \tag{4.4}$$

en supposant que les bornes (4.2) et (4.3) sont atteintes.

Dans certain cas, on peut ne pas avoir d'étiquettes y_i sur les données d'apprentissage défini par un ensemble d'observations $x_i,\ i = 1,2,\dots,l$. Le nouveau problème de détection consiste à décider si une nouvelle observation x est proche ou non de cet ensemble.

L'idée est d'enfermer les données dans une boule et de les prendre en tant que données normales et de considérer les autres comme des données anormales. Une manière d'aborder ce problème consiste à rechercher une frontière de décision sous la forme de la plus petite boule englobante (***M**inimum **E**nclosing **B**all-**MEB***) définie par son centre c et son rayon R.

Cet algorithme consiste à construire une sphère dans l'espace de dimension d contenant les points d'apprentissage d'une même classe et centré en un point fixe c. Il suffit alors de distinguer les points se trouvant en dehors de la sphère ainsi construite. On définit ensuite une fonction qui prend la valeur **-1** pour chaque point en dehors de la sphère sinon **+1**.

La méthode peut-être vue comme une méthode de détection plutôt que de classification. Néanmoins, il est possible de reprendre les données à l'extérieur de l'hyper-sphère et de les étiqueter comme étant une classe et ainsi utiliser les Machines à Vecteurs Supports pour une classification.

Le choix du rayon se fait en supposant que ce dernier soit inférieur à la distance entre les points d'apprentissage et le centre $\forall\, \Phi(x_i)$ afin d'inclure tous les points de l'ensemble d'entré.

4.2.1 La plus petite boule englobante dure

Il est, cependant, possible de trouver la plus petite boule englobante. Il suffit d'optimiser la position du centre c pour obtenir une sphère de rayon R plus petite en résolvant le problème suivant :

$$\begin{aligned} &\min R^2 \\ &sous\ la\ contrainte \quad \|\Phi(x_i) - c\|^2 \leq R^2, \qquad i = 1, \dots, l \end{aligned} \tag{4.5}$$

Appliquons les multiplicateurs lagrangiens $\alpha_i \geq 0$ sur le problème d'optimisation, on obtiendra :

$$\begin{aligned} \mathcal{L}(c, R, \alpha) &= R^2 - \textstyle\sum_{i=1}^{l} \alpha_i \{R^2 - \|\Phi(x_i) - c\|^2\} \\ &= R^2 - \textstyle\sum_{i=1}^{l} \alpha_i \{R^2 - (\Phi(x_i) - c)^T (\Phi(x_i) - c)\} \\ &= R^2 - \textstyle\sum_{i=1}^{l} \alpha_i \{R^2 - (\Phi(x_i)^2 - 2\Phi(x_i)c + c^2)\} \end{aligned} \tag{4.6}$$

Calculons les derivées partielles primales en c et R, on obtient :

$$\begin{aligned} \tfrac{\partial \mathcal{L}(c,R,\alpha)}{\partial c} &= 2 \textstyle\sum_{i=1}^{l} \alpha_i (\Phi(x_i) - c) = 0 \\ \tfrac{\partial \mathcal{L}(c,R,\alpha)}{\partial R} &= 2R\left(1 - \textstyle\sum_{i=1}^{l} \alpha_i\right) = 0 \end{aligned} \tag{4.7}$$

Ainsi on obtient les équations suivantes :

$$\begin{aligned} &\textstyle\sum_{i=1}^{l} \alpha_i = 1 \\ &c = \textstyle\sum_{i=1}^{l} \alpha_i \Phi(x_i) \end{aligned} \tag{4.8}$$

Par des termes en (4.8) résulte la forme duale :

$$\begin{aligned} \mathcal{L}(c, R, \alpha) &= R^2 - \textstyle\sum_{i=1}^{l} \alpha_i \{R^2 - \|\Phi(x_i) - c\|^2\} \\ &= \textstyle\sum_{i=1}^{l} \alpha_i \, \|\Phi(x_i) - c\|^2 \\ &= \textstyle\sum_{i=1}^{l} \alpha_i K(x_i, x_i) - \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j K\left(x_i, x_j\right) \end{aligned} \tag{4.9}$$

Pour trouver α dans la forme duale, on doit résoudre le problème d'optimisation suivant:

$$\begin{aligned} &\max_{\alpha} \textstyle\sum_{i=1}^{l} \alpha_i K(x_i, x_i) - \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j K\left(x_i, x_j\right) \\ &sous\ les\ contraintes \quad \textstyle\sum_{i=1}^{l} \alpha_i = 1, \quad et \quad \alpha_i \geq 0, \quad i = 1, \dots, l \end{aligned} \tag{4.10}$$

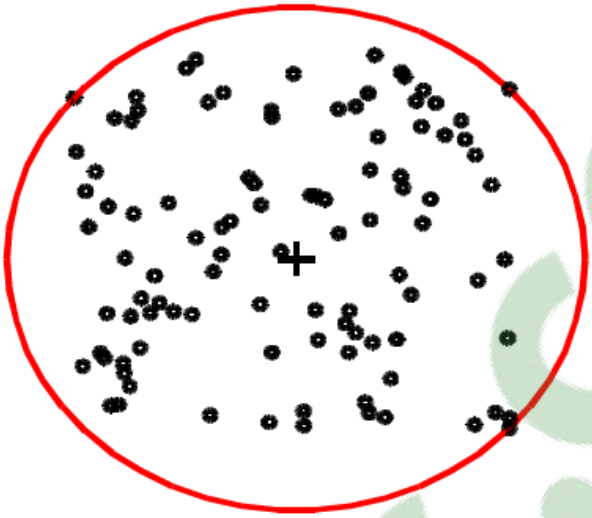


Fig. 4.1: Représentation de la plus petite boule englobante dure

Les conditions de KKT sont satisfaits par des solution optimales de α, c, R

$$\alpha_i\{\|\Phi(x_i) - c\|^2 - R^2\} = 0, \quad i = 1, \dots, l \tag{4.11}$$

Ceci implique que les données d'apprentissages $\Phi(x_i)$ liées à la surface optimale de la boule ont leurs $\alpha_i \geq 0$.

La fonction de décision est définie comme :

$$\begin{aligned} f(x) &= sign(R^2 - \|\Phi(x) - c\|^2) \\ &= sign\left(R^2 - \begin{Bmatrix} (\Phi(x)\Phi(x)) - 2\sum_{i=1}^{l}\alpha_i(\Phi(x)\Phi(x_i)) \\ +\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j\left(\Phi(x_i)\Phi(x_j)\right) \end{Bmatrix}\right) \\ &= sign\left(R^2 - \begin{Bmatrix} K(x,x) - 2\sum_{i=1}^{l}\alpha_i K(x,x_i) \\ +\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j K(x_i,x_j) \end{Bmatrix}\right) \end{aligned} \tag{4.12}$$

4.2.2 La plus petite boule englobante souple

Dans le cas où on a quelques impuretés sur notre espace d'apprentissage, la boule dure qui en résulte doit avoir un rayon plus large, ce qui conduit à une altération sur la robustesse de cet espace d'apprentissage. Donc, il faut trouver une boule englobante minimale qui contient (presque) toutes les données d'apprentissage, en évitant de prendre les données extrême.

Pour cela, on doit introduire les variables d'écart ξ, $\xi_i \geq 0$, $i = 1, \dots, l$, et un terme de tolérance C qui contrôle la taille du rayon R. Ce terme autorise la construction d'une sphère

ne contenant pas tous les points. Ceci peut permettre la résolution de certains problèmes en augmentant le terme de tolérance.

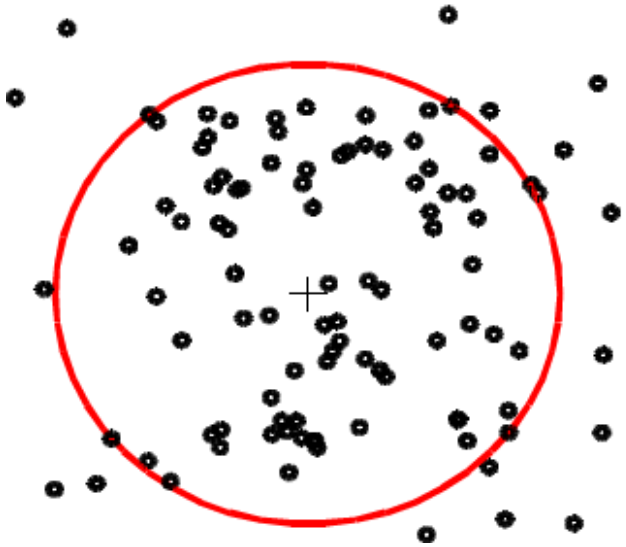


Fig 4.2: Représentation de la plus petite boule englobante souple

Pour trouver la plus petite boule englobante (MEB), on doit résoudre le problème d'optimisation suivant:

$$\begin{aligned} & min\, R^2 + C\sum_{i=1}^{l}\xi_i \\ avec \qquad & \|\Phi(x_i) - c\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, l \end{aligned} \tag{4.13}$$

Le paramètre $C = \frac{1}{vl}$ permet de régler un nombre de v points que l'on désire maintenir en dehors de la boule (*outliers*). Le dual de ce problème est le programme quadratique analogue à celui des Machines à Vecteurs Supports.

Ajoutons les multiplicateurs Lagrangien $\alpha_i, \beta_i \geq 0$ pour contrainte, on aura la formulation primale de l'optimisation

$$\mathcal{L}(c, R, \alpha, \beta) = R^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i\{R^2 + \xi_i - \|\Phi(x_i) - c\|^2\} - \sum_{i=1}^{l}\alpha_i\beta_i \tag{4.14}$$

Calculons les dérivées partielles primales en c et R et mettons les solutions à zéro, on obtient :

$$\begin{aligned} \frac{\partial\mathcal{L}(c,R,\alpha)}{\partial c} &= 2\sum_{i=1}^{l}\alpha_i(\Phi(x_i) - c) = 0 \\ \frac{\partial\mathcal{L}(c,R,\alpha)}{\partial R} &= 2R\left(1 - \sum_{i=1}^{l}\alpha_i\right) = 0 \\ \frac{\partial\mathcal{L}(c,R,\alpha)}{\partial \xi_i} &= C - \alpha_i - \beta_i = 0 \end{aligned} \tag{4.15}$$

Suite à la dernière équation en (4.15), on a $\alpha_i = C - \beta_i$. Puisque $\alpha_i \geq 0$, $\beta_i \geq 0$, on peut éliminer les multiplicateurs de Lagrange β_i en supposant que $0 \leq \alpha_i \leq C$ [Tax (99)].

Ainsi, on obtient les équations suivantes :

$$\begin{aligned} &\textstyle\sum_{i=1}^{l} \alpha_i = 1 \\ &c = \textstyle\sum_{i=1}^{l} \alpha_i \Phi(x_i) \end{aligned} \tag{4.16}$$

Pour trouver α dans la forme duale, on doit résoudre le problème d'optimisation suivant:

$$\begin{aligned} \alpha \quad & \max - \textstyle\sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j K(x_i, x_j) + \sum_{i=1}^{l} \alpha_i K(x_i, x_i) \\ & avec \quad 0 \leq \alpha_i \leq C, \quad \textstyle\sum_{i=1}^{l} \alpha_i = 1 \end{aligned} \tag{4.17}$$

La formulation duale de cette optimisation est le problème quadratique analogue à celui des Machines à Vecteurs Supports.

Les forme matricielle de (4.17) s'écrit:

$$\begin{aligned} &\max_{\alpha} -\alpha^T \boldsymbol{K} \alpha + \alpha^T diag(\boldsymbol{K}) \\ &\quad avec \quad e^T \alpha = 1 \\ &0 \leq \alpha_i \leq C \qquad i = 1, \dots, l \end{aligned} \tag{4.18}$$

où $\boldsymbol{K}$ est la matrice noyau (kernel) $l \times l$ de terme général $K(x_i, x_j)$.

On a alors $0 \leq \alpha \leq C$. Cette approche rend l'algorithme plus flexible et adapté à un ensemble d'apprentissage divergent.

4.2.3 Core-set

Un Core-set est une approximation de la plus petite boule englobante qui a été définie par Bãdoiu et Clarkson [Bãdo (08)].

Définition

Soit un ensemble de points donné $P \subset \mathbb{R}^d$ et une valeur $\epsilon > 0$, un ϵ-Core-set $S \subset P$ a la propriété que la plus petite boule contenant S de rayon R à l'interieur de la plus petite boule de rayon $(1+\epsilon)R$ contenant P: le centre de la plus petite boule contenant S est à l'intérieur de $(1+\epsilon)R$ distance de tout point de P, où R est le rayon de la plus petite boule contenant S.

Donc, on considère $P \subset \mathbb{R}^d$ où est définie la plus petite boule englobante $MEB(P) = \mathcal{B}(c_p, (1+\epsilon)R)$, il existe un sous-ensemble $S \subset P$ appelé ϵ-Core-set de $MEB(P)$ où est définie la plus petite englobante $MEB(S) = \mathcal{B}(c_S, R)$ dont le centre c_S satisfait la condition $d(x, c_S) \leq (1+\epsilon)R$, $\forall x \in S$ (d: est une fonction calculant un distance).

On donne ci-après l'algorithme de Bãdoiu-Clarson [Bãdo (08)] qui définit plusieurs ϵ-Core-sets à l'intérieur de la plus petite boule englobante d'un ensemble donné.

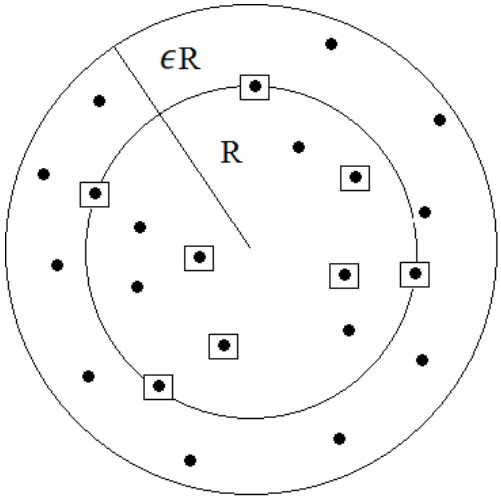


Fig. 4.3: Le cercle d'intérieur définit le Core-set et le cercle externe définit la plus petite boule englobante qui couvre tous les points de l'ensemble des données.

Algorithm 1 Bãdoiu-Clarson Algorithm

1: Initialize the Core-set S_ϵ.

2: Compute the minimal-enclosing-ball $\mathcal{B}(c_S, R)$ of the Core-set S_ϵ.

3: **while** A point $x \in S$ out of the ball $\mathcal{B}(c_P, (1+\epsilon)R)$ exist **do**

4: Include x in S_ϵ.

5: Compute the minimal-enclosing-ball $\mathcal{B}(c_S, R)$ of the Core-set S_ϵ.

6: **end while**

4.3 Classification basé L2-SVM – Nouvelle formulation

On considère un ensemble d'apprentissage $\{z_i = (x_i, y_i)\}_{i=1}^{l}$, où $y_i \in \{-1, +1\}$ pour le cas d'une classification binaire et $y_i \in \{1,2, \ldots \mathcal{M}\}$ pour le cas d'une classification multi-classes; $\mathcal{M}$ désigne le nombre de classes.

4.3.1 Classification binaire

Dans le cas d'une classification bi-classes, on définit la fonction objective primal de la formulation L2-SVM [Tsan (05)] par:

$$\begin{aligned} &\min_{w,b,\rho,\xi_i} \frac{1}{2}\left(\|w\|^2 + b^2 - 2\rho + C\sum_{i=1}^{l}\xi_i^2\right) \\ &sous\ la\ contrainte \quad y_i(w^T\Phi(x_i) + b) \geq \rho - \xi_i, \quad i = 1, \dots, l \end{aligned} \tag{4.19}$$

En appliquant les multiplicateurs de Lagrange sur le problème d'optimisation primale, on obtient:

$$\mathcal{L} = \frac{1}{2}\left(\|w\|^2 + b^2 - 2\rho + C\sum_{i=1}^{l}\xi_i^2\right) - \sum_{i=1}^{l}\left(\alpha_i(y_i(w^T\Phi(x_i) + b) - \rho + \xi_i)\right) \tag{4.20}$$

$$\mathcal{L} = \frac{1}{2}\left(\|w\|^2 + b^2 - 2\rho + C\sum_{i=1}^{l}\xi_i^2\right) - \sum_{i=1}^{l}\alpha_i y_i w^T\Phi(x_i) - \sum_{i=1}^{l}\alpha_i y_i b + \rho\sum_{i=1}^{l}\alpha_i - \sum_{i=1}^{l}\alpha_i\xi_i \tag{4.21}$$

Après calcul des dérivés partielles sur w, b, ξ_i, et ρ et leur mise à zéro, on obtient:

$$\frac{\delta\mathcal{L}}{\delta w} = 0 \Longrightarrow w = \sum_{i=1}^{l}\alpha_i y_i\Phi(x_i) \tag{4.22}$$

$$\frac{\delta\mathcal{L}}{\delta b} = 0 \Rightarrow b = \sum_{i=1}^{l}\alpha_i y_i \tag{4.23}$$

$$\frac{\delta\mathcal{L}}{\delta\xi_i} = 0 \Rightarrow \xi_i = \frac{\alpha_i}{C} \tag{4.24}$$

$$\frac{\delta\mathcal{L}}{\delta\rho} = 0 \Rightarrow \sum_{i=1}^{l}\alpha_i = 1 \tag{4.25}$$

En remplaçant les équations (4.24) et (4.25) dans (4.21) et après simplification, on obtient :

$$\mathcal{L} = \frac{1}{2}\|w\|^2 + \frac{1}{2}b^2 - \sum_{i=1}^{l}\alpha_i y_i w^T\Phi(x_i) - \sum_{i=1}^{l}\alpha_i y_i b - \frac{1}{C}\left(\sum_{i=1}^{l}\alpha_i\sum_{j=1}^{l}\alpha_j\right) \tag{4.26}$$

En remplaçant les équations (4.22) et (4.23) dans (4.26), on obtient la forme de la fonction duale finale du problème d'optimisation :

$$\mathcal{L} = -\frac{1}{2}\left(\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j\Phi(x_i)^T\Phi(x_i) + \sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j + \frac{1}{C}\left(\sum_{i=1}^{l}\alpha_i\sum_{j=1}^{l}\alpha_j\right)\right) \tag{4.27}$$

A partir de tout vecteur support, on a α optimale et $\rho = y_i(w^T\Phi(x_i) + b) + \frac{\alpha_i}{C}$ suite au contrainte imposée dans (4.19). En égalisant la fonction objective primale (4.19) et la fonction duale (4.27), on obtient:

$$\|w\|^2 + b^2 - 2\rho + C\sum_{i=1}^{l}\xi_i^2 = \sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j\left(y_i y_j K(x_i, x_j) + y_i y_j + \frac{\delta_{ij}}{C}\right) \tag{4.28}$$

La résolution de ce problème nous impose de calculer :

$$\begin{aligned} &\max_{\alpha} -\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j\left(y_iy_jK(x_i,x_j)+y_iy_j+\frac{\delta_{ij}}{C}\right) \\ &avec\ les\ contraites \quad \sum_{i=1}^{l}\alpha_i=1 \quad \alpha_i\geq 0 \quad i=1,\dots,l \end{aligned} \tag{4.29}$$

où δ_{ij} est la fonction delta Kronecker ($\delta_{ij}=1$ si $i=j$ et $\delta_{ij}=0$ si $i\neq j$).

On peut aussi énoncer (4.29) sous forme matricielle [Flet (00)] :

$$\begin{aligned} &\max_{\alpha} -\alpha^T\left(K\odot y^Ty+y^Ty+\frac{1}{C}\boldsymbol{I}\right)\alpha \\ &avec \quad \alpha\geq\mathbf{0}, \quad \alpha^T\mathbf{1}=1, \quad i=1,\dots,l \end{aligned} \tag{4.30}$$

où $\odot$ dénote le produit d'Hadamard et $y=[y_1,\dots,y_l]^T$.

Donc, on peut réécrire (4.29) comme:

$$\begin{aligned} &_{\alpha}\ max -\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j\left(y_iy_jK(x_i,x_j)+y_iy_j+\frac{\delta_{ij}}{C}\right)=\max_{\alpha} -\alpha\widetilde{\boldsymbol{K}}\alpha \\ &avec \quad \alpha\geq\mathbf{0}, \quad \alpha^T\mathbf{1}=1, \quad i=1,\dots,l \end{aligned} \tag{4.31}$$

Où $\widetilde{\boldsymbol{K}}=\left[\tilde{k}(z_i,z_j)\right]$ avec

$$\tilde{k}(z_i,z_j)=y_iy_jK(x_i,x_j)+y_iy_j+\frac{\delta_{ij}}{C} \tag{4.32}$$

4.3.2 Formulation Multi-classe

La formulation Multi-classe des Machine à Vecteurs Supports considère un vecteur de sortie tout en prenant compte les mêmes types de calculs énoncés à la section 4.3.1. Cette idée est issue d'une simple réinterprétation d'un vecteur normal de l'hyperplan séparé [Lach (10), Szed (05)]. Ce vecteur peut être vu comme un opérateur de projection du vecteur caractéristique sur un sous-espace à une dimension.

Nous définissons la fonction objective primal pour le problème d'apprentissage comme:

$$\begin{aligned} &\min_{W,b,\rho,\xi_i} trace(W^TW)+\|b\|^2-2\rho+\frac{1}{C}\sum_{i=1}^{l}\xi_i^2 \\ &sous\ la\ contrainte \quad y_i^T(W\Phi(x_i)+b)\geq\rho-\xi_i \end{aligned} \tag{4.33}$$

A partir des conditions de Karush-Kuhn-Tucker(KKT) [Flet (00)] sur l'équation (4.33), nous obtiendrons :

$$\begin{aligned} W&=\sum_{i=1}^{l}\alpha_iy_i\Phi(x_i)^T \\ b&=\sum_{i=1}^{l}\alpha_iy_i \end{aligned} \tag{4.34}$$

La formulation duale de l'optimisation primale correspond à:

$$\begin{matrix} \max_{\alpha} -\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j\left(y_i{}^T y_j K(x_i,x_j)+y_i{}^T y_j+\frac{\delta_{ij}}{C}\right) \\ avec\ les\ contraites \quad \sum_{i=1}^{l}\alpha_i=1 \quad \alpha_i\geq 0 \quad i=1,\dots,l \end{matrix} \tag{4.35}$$

Ainsi, la fonction de décision prédisant une des étiquettes parmi $1,2,\dots,\mathcal{M}$ pour tout test du modele x_j peut être exprimée comme:

$$\begin{matrix} \arg\max_{m=1,\dots,\mathcal{M}} y_m^T(W\Phi(x_i)+b) \\ =\arg\max_{m=1,\dots,\mathcal{M}}\left(\sum_{i=1}^{l}\left(\alpha_i y_i^T y_m\left(K(x_i,x_j)+1\right)\right)\right) \end{matrix} \tag{4.36}$$

Soit $y_i(m)$ qui dénote le $m^{ème}$ élément du vecteur label y_i correspondant au modèle x_i. Un choix judicieux [Szed (05)] pourrait être:

$$y_i(m)=\left\{\begin{matrix} \sqrt{\frac{(\mathcal{M}-1)}{\mathcal{M}}} & si\ l'élément\ \ i\ \ \ appartient\ \text{à}\ la\ categorie\ \ m \\ \sqrt{\frac{1}{\mathcal{M}(\mathcal{M}-1)}} & autrement \end{matrix}\right\}$$

Le produit scalaire entre les vecteurs sera alors

$$y_i^T y_j=\left\{\begin{matrix} 1 & si\ \ i\ et\ j\ sont\ de\ la\ même\ classe \\ \frac{(3\mathcal{M}-4)}{\mathcal{M}(\mathcal{M}-1)} & autrement \end{matrix}\right\}$$

Ainsi, on peut réécrire (4.35) comme:

$$\begin{matrix} \max_{\alpha} -\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j\left(y_i{}^T y_j K(x_i,x_j)+y_i{}^T y_j+\frac{\delta_{ij}}{C}\right)=\max_{\alpha}-\alpha\tilde{\boldsymbol{K}}\alpha \\ avec \quad \alpha\geq\mathbf{0}, \quad \alpha^T\mathbf{1}=1, \quad i=1,\dots,l \end{matrix} \tag{4.37}$$

Où $\tilde{\boldsymbol{K}}=\left[\tilde{k}(z_i,z_j)\right]$ avec

$$\tilde{k}(z_i,z_j)=y_i{}^T y_j K(x_i,x_j)+y_i{}^T y_j+\frac{\delta_{ij}}{C} \tag{4.36}$$

4.4 Equivalence L2-SVM / MEB

Dans cette section, nous montrons l'équivalence entre une classification L2-SVM et la formulation de la plus petite boule englobante (MEB) [Lach (12)-1] que nous exploiterons dans la réduction des données suivant deux approches (cf section 4.5).

On considère

$$k(x,x) = \kappa \tag{4.37}$$

une constante, et est satisfaite seulement pour :

1- Le noyau isotropique $k(x,y) = K(\|x-y\|)$ (Noyau Gaussien); ou
2- Le noyau à produit scalaire $k(x,y) = K(x^T y)$ (Noyau polynomial) avec des entrées normalisées; ou
3- Tout noyau normalisé $k(x,y) = \frac{K(x,y)}{\sqrt{K(x,x)}\sqrt{K(y,y)}}$

Ces trois cas couvrent la majorité des fonctions « noyaux » qui ont été utilisées dans la littérature.

Dans la formulation de la plus petite boule englobante (dure (4.10) et souple (4.17)) énoncée dans la section 4.2, la résolution de la fonction objective duale est donnée par:

$$\begin{gathered} \max_{\alpha} -\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j K(x_i,x_j) + \sum_{i=1}^{l}\alpha_i K(x_i,x_i) \\ avec \quad 0 \le \alpha_i \le C, \quad \sum_{i=1}^{l}\alpha_i = 1 \end{gathered} \tag{4.38}$$

et sa forme matricielle

$$\begin{gathered} \max_{\alpha} -\alpha^T \boldsymbol{K}\alpha + \alpha^T diag(\boldsymbol{K}) \\ avec \quad \alpha^T \mathbf{1} = 1 \\ 0 \le \alpha_i \le C \quad i = 1,\dots,l \end{gathered} \tag{4.39}$$

où $\boldsymbol{K}$ est la matrice kernel $l \times l$ de terme général $K(x_i,x_j)$.

En combinant (4.37) avec la condition $\sum_{i=1}^{l}\alpha_i = 1$ dans (4.39), nous obtiendrons $\alpha^T diag(\boldsymbol{K}) = \kappa$, qui est une constante. Ainsi l'optimisation devient :

$$\begin{gathered} \max_{\alpha} -\alpha^T \boldsymbol{K}\alpha + \alpha^T diag(\boldsymbol{K}) = \max_{\alpha} -\alpha^T \boldsymbol{K}\alpha + \kappa \cong \max_{\alpha} -\alpha^T \boldsymbol{K}\alpha \\ avec \quad \alpha^T \mathbf{1} = 1 \\ 0 \le \alpha_i \le C \quad i = 1,\dots,l \end{gathered} \tag{4.40}$$

Cependant, on remarque que la formulation de la fonction objective duale obtenue est la même que celles de la classification binaire (4.31) et multi-classes (4.37) de la norme L2-SVM qui sont données respectivement par:

$$\begin{gathered} \max_{\alpha} -\alpha \widetilde{\boldsymbol{K}}\alpha \\ avec \quad \alpha \ge \mathbf{0}, \quad \alpha^T \mathbf{1} = 1, \quad i = 1,\dots,l \end{gathered} \tag{4.41}$$

où $\widetilde{\boldsymbol{K}} = \left[\tilde{k}(z_i, z_j)\right]$ avec $\tilde{k}(z_i, z_j) = y_i y_j K(x_i, x_j) + y_i y_j + \frac{\delta_{ij}}{C}$ pour le cas binaire et $\widetilde{\boldsymbol{K}} = \left[\tilde{k}(z_i, z_j)\right]$ avec $\tilde{k}(z_i, z_j) = {y_i}^T y_j K(x_i, x_j) + {y_i}^T y_j + \frac{\delta_{ij}}{C}$ pour le cas multi-classe.

4.4.1 Affinement de l'équivalence

Le calcul avec les multiplicateurs de Lagrange conduit à un problème quadratique plus simple de l'équation (4.41) avec des contraintes positives à condition unique. Pour affiner l'optimisation MEB en (4.40), nous dérivons un algorithme basé sur l'entropie par le moyen de la dualité Lagrangienne et le principe d'entropie maximum de Jaynes. L'idée est d'utiliser l'information d'entropie et le formalisme d'entropie maximum dans la solution des problèmes de programmation non-linéaire [Lach (14)-1].

On considère le problème quadratique de l'équation (4.41) formulé comme suit:

$$\begin{aligned} &\min_{\alpha} \mathcal{L}(\alpha) = \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j K(x_i, x_j) \\ &sous\ contrainte\ \ \alpha_i \geq 0, \quad \sum_{i=1}^{l} \alpha_i = 1 \end{aligned} \tag{4.42}$$

A partir des contraintes du problème d'optimisation (4.42), on sait que les variables duales sont définies dans [0,1] et leur somme est égale à 1, ceci se rencontre avec la définition de probabilité. Notre approche à la résolution de l'équation (4.42) est basée sur une interprétation probabiliste qui montre que le centre d'une boule représente la moyenne des vecteurs de l'ensemble des données et les multiplicateurs de Lagrange α_i représentent la probabilité pour que x_i soit un Vecteur Support (SV). Cependant, on peut considérer la plus petite boule englobante (MEB) ou la classification L2-SVM comme une procédure d'assignement de probabilités qui peut suivre le principe de l'entropie maximum de Jaynes [Temp (87)]. Ainsi, à la place du problème quadratique de (4.42), nous construisons le problème de minimisation composite suivant:

$$\begin{aligned} &\min \mathcal{L}_{\mathcal{P}}(\alpha) = \mathcal{L}(\alpha) + \frac{H(\alpha)}{\mathcal{P}} \\ &sous\ la\ contrainte\ \alpha_j \geq 0, \quad \sum_{j=1}^{l} \alpha_j = 1 \end{aligned} \tag{4.43}$$

où $\mathcal{P}$ est un paramètre positif et

$$H(\alpha) = \sum_{j=1}^{l} \alpha_j \ln \alpha_j \tag{4.44}$$

A partir des perspectives de la théorie d'information, $H(\alpha)$ represente l'entropie d'information des multiplicateur de Lagrange α_j. Le terme additionnel $\frac{H(\alpha)}{\mathcal{P}}$ est proportionné

avec l'application d'un extra-critère de minimisation de l'entropie des multiplicateurs au problème d'optimisation quadratique de l'équation (4.42). Il est intuitivement évident que le terme d'entropie de la solution de l'équation (4.43) devra diminuer l'infinité à $\mathcal{P}$ approches.

Pour résoudre ce problème nous introduisons le Lagrangien

$$\mathcal{L}_{\mathcal{P}}(\alpha,\beta) = \mathcal{L}(\alpha) + \frac{H(\alpha)}{\mathcal{P}} + \beta\left(\sum_{j=1}^{l}\alpha_j - 1\right) \tag{4.45}$$

Où β est un multiplicateur de Lagrange. La mise à zèro de la derivée de $\mathcal{L}_{\mathcal{P}}(\alpha,\beta)$ sur α et β respectivement conduit à:

$$\frac{\partial \mathcal{L}_{\mathcal{P}}(\alpha,\beta)}{\partial \alpha_j} - \frac{1}{\mathcal{P}}\left(1 + \ln \alpha_j\right) + \beta = 0 \tag{4.46}$$

avec

$$\sum_{j=1}^{l}\alpha_j = 1 \tag{4.47}$$

Résolvons l'équation (4.46) pour α_j, $i = 1,2,\dots,l$

$$\alpha_j = e^{\left(\mathcal{P}\left(\frac{\partial \mathcal{L}}{\alpha_j}+\beta\right)-1\right)} \tag{4.48}$$

Remplaçons α_j de l'équation (4.48) dans (4.47), nous obtenons

$$e^{(\mathcal{P}\beta-1)}\sum_{j=1}^{l} e^{\left(\mathcal{P}\frac{\partial \mathcal{L}}{\partial \alpha_j}\right)} = 1 \tag{4.49}$$

Entre (4.48) et (4.47), on élimine le terme $e^{(\mathcal{P}\beta-1)}$ pour donner

$$\alpha_j = \frac{e^{\left(\mathcal{P}\frac{\partial \mathcal{L}}{\partial \alpha_j}\right)}}{\sum_{j=1}^{l} e^{\left(\mathcal{P}\frac{\partial \mathcal{L}}{\partial \alpha_j}\right)}} \tag{4.50}$$

Par optimisation du problème en (4.42), nous obtenons :

$$\mathcal{L}_{\alpha_i(\alpha)} \equiv \frac{\partial \mathcal{L}}{\partial \alpha_j} = 2\sum_{i=1}^{l}\alpha_i K(x_i, x_j) \tag{4.51}$$

Ainsi, on obtient la formule itérative

$$\alpha_j^{(k+1)} = \frac{e^{\left(\mathcal{P}^{(k)}\mathcal{L}_{\alpha_j}\left(\alpha^{(k)}\right)\right)}}{\sum_{j=1}^{l} e^{\left(\mathcal{P}^{(k)}\mathcal{L}_{\alpha_j}\left(\alpha^{(k)}\right)\right)}} \tag{4.52}$$

On se basant sur les formules des équations (4.43)-(4.46), nous obtenons l'algorithme itératif suivant basé sur l'entropie pour solutionner le problème d'optimisation de l'équation (4.42) :

Algorithm 2 Entropy-based iterative algorithm

1: Let $\mathcal{P}^{(0)} = 0$; from Eq. (4.50) we get $\alpha_j^{(0)} = 1/F$; $j = 1,2,\ldots,l$; let $\Delta\mathcal{P} \in (0,+\infty)$ and set $k = 0$

2: Based on formulas Eq. (4.51) and Eq. (4.52), compute $\alpha_j^{(k+1)}$, $j = 1,2,\ldots,l$; let $\mathcal{P}^{(k+1)} = \mathcal{P}^{(k)} + \Delta\mathcal{P}$

3: if Stop criteria satisfied, the stop; otherwise, we set $k = k + 1$, then return to step **2**

Cet algorithme nous permet de commencer par les multiplicateurs de Lagrange estimés lors du développement des machines à vecteurs support où de la plus petite boule englobante, et d'estimer une version améliorée par la formule itérative de l'équation (4.52).

Là, aussi on note une déduction importante qu'en améliorant les multiplicateurs de Lagrange, l'algorithme de Bãdoiu et Clarkson [Bãdo (08)] est aussi amélioré

4.5 Partitionnement des données (Clustering)

4.5.1 Algorithme du *k*-plus proche-voisins

Cette algorithme (*k*-Nearest Neighbors (*k*-NN)) [Holl (00)] présente un cas spécial de la quantification vectoriel dans lequel la partition est déterminée par un dictionnaire et une mesure de distorsion. Le vecteur quantifié définit la valeur moyenne des données d'une cellule appelée centroïde.

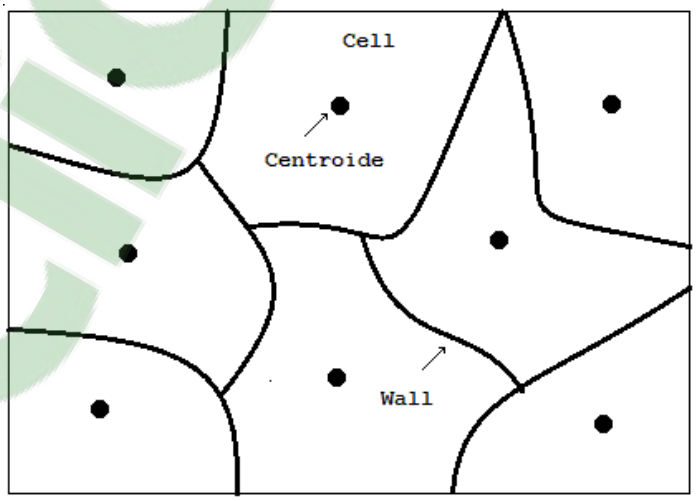


Fig. 4.4: Cellules issues de la quantification vectorielle.

Le vecteur quantifié du plus proche voisin est en fait le type de vecteur le plus commun aux données. Toute cette quantification se base sur une mesure de distorsion et celle utilisée dans cet algorithme est la distance Euclidienne entre deux vecteurs définie comme suit:

$$d(x,y) = \sum_p (x_p - y_p)^2 \tag{4.53}$$

La partition en cellules de l'espace d'entrée pour le vecteur à quantifier pour le plus proche voisin est définie par:

$$S_p = \{x: d(x,y_p) \leq d(x,y_{p'}), \forall p' \in P\} \tag{4.54}$$

Suivant l'équation (4.54), on remarque que dans l'algorithme du plus proche voisin chaque cellule S_p est constituée de tous les points x qui ont une distorsion inferieure relativement à la production du vecteur centroïde y_p. Ceci montre que $\bigcup_p S_p \in \mathbb{R}^N$ et $S_p \bigcap S_{p'} = 0$ pour $p \neq p'$.

Algorithm 3 Nearest Neighbor

1: initialization: $d = d_0,\ p = 1$

2: **compute** $D_p = d(x,y_p)$

3: **if** $D_j < d$ ***then*** set $D_p \rightarrow d$ and set $p \rightarrow 1$

4: **if** $p < P$ **then** set $p + 1 \rightarrow p$ and **goto** ***2***

5: **if** $p = P$ ***then*** stop

La valeur initialisée d_0 doit être plus grande que la distorsion attendue et P définit le nombre de partitions ou de cellules S_p. Le codage de l'algorithme du plus proche voisin exécute une recherche exhaustive dans le dictionnaire de tous les vecteurs quantifiés dans le cas où on a à déterminer la partition à laquelle doit appartenir un vecteur d'entré. Pour notre étude, on utilise cet algorithme juste pour faire un partitionnement de nos données.

4.5.2 Algorithme des C-Moyennes Floues (Fuzzy C-Mean)

L'algorithme des C-Moyennes Floues effectue une optimisation itérative en évaluant de façon approximative les minimums d'une fonction d'erreur. Il existe toute une famille de fonctions d'erreur associées à cet algorithme qui se distinguent par des valeurs différentes prises par un paramètre réglable m, appelé indice flou (*fuzzy index)* et qui détermine le degré flou de la partition obtenue [Al-Zo (07)]. Les C-Moyennes Floues sont un cas particulier d'algorithmes basés sur la minimisation d'un critère ou d'une fonction objective.

Soit $U \in M_{fc}$ une c-partition floue de X et soit v le c-uplet :

$$v = (v_1, v_2, \dots, v_c) \tag{4.55}$$

où $\forall i, v_i \in \mathbb{R}^p$. La fonction objective associée aux C-Moyennes Floues J_m est définie par:

$$J_m(U, v) = \sum_{i=1}^{l} \sum_{j=1}^{c} \left(u_{ji}\right)^m \left(d_{ij}\right)^2 \tag{4.56}$$

Pour tout i $(1 \leq j \leq c)$, v_j est un vecteur à p composantes qui représente le centroïde de la $j^{ème}$ classe, et pour tout j et tout i $(1 \leq i \leq l)$, $\left(d_{ji}\right)^2 = \left\|x_i - v_j\right\|^2$ où $\|.\|$ est une norme associée à un produit scalaire définie dans $\mathbb{R}^p$. On peut aussi écrire :

$$d_{ji}^2 = d^2\left(x_i - v_j\right) = \left(x_i - v_j\right)^T A \left(x_i - v_j\right) \tag{4.57}$$

où A est une matrice $p \times p$ définie positive. Enfin, l'indice flou m doit être strictement supérieur à 1 : $m \in]1, +\infty[$. Le carré de la distance $\left(d_{ji}\right)^2$séparant un vecteur x_i d'un centre v_j, $\left(d_{ji}\right)^2$, est pondéré par la puissance $m^{ème}$ du degré d'appartenance de la donnée x_i à la classe j : $\left(u_{ji}\right)^m$. J_m est donc une erreur quadratique généralisée et sa minimisation conduit théoriquement à la partition optimale. Les C-Moyennes Floues produisent une C-partition floue qui est une approximation de la partition optimale de l'ensemble de données $X = \{x_1, x_2, \dots, x_l\}$. L'algorithme dont la convergence a été étudiée par Bezdek [Bezd (80,81)] peut être décrit par le schéma suivant :

Algorithme 4 :

1) Fixer le nombre de classes c tel que $2 \leq c \leq l$, l étant le nombre de données. Fixer une valeur pour m telle que $m > 1$. Choisir une norme $\|.\|$ dans $\mathbb{R}^p$: $\|x - v\|^2 = (x - v)^T A (x - v)$ où A est une matrice à p lignes et p colonnes définie et positive. Le plus souvent, A est la matrice identité qui correspond à la distance euclidienne.

2) Initialiser la c-partition floue en donnant des valeurs quelconques (éventuellement les deviner dans la mesure du possible) aux éléments de la matrice initiale correspondante $U^{(0)}$ vérifiant :

$$\forall i \in \{1,2, \dots, l\}, \ \ \sum_{j=1}^{c} u_{ji}^{(0)} = 1 \tag{4.58}$$

3) initialiser le compteur de boucle : $b = 0$.

4) Calculer les c centroïdes de classe $\left\{v_j^{(b)}\right\}$, $1 \leq j \leq c$, en utilisant $U^{(b)}$ à l'aide de la formule suivante :

$$v_j^{(b)} = \frac{\sum_{i=1}^{l}\left(u_{ji}^{(b)}\right)^m . x_i}{\sum_{i=1}^{l}\left(u_{ji}^{(b)}\right)^m} \tag{4.59}$$

5) Mettre à jour la matrice U : calculer la nouvelle matrice de degrés d'appartenance $U^{(b+1)}$ comme suit :

Pour $i = 1$ allant jusqu'à l,

a) chercher I_i et $\tilde{I}_i$:

$$I_i = \left\{j | 1 \leq j \leq c \;\; et \;\; d_{ji} = \left\|x_i - v_j\right\| = 0\right\} \tag{4.60}$$

$$\tilde{I}_i = \{1,2, \dots, c\} - I_i \tag{4.61}$$

b) pour la $i^{ème}$ donnée x_i, calculer les nouveaux degrés d'appartenance selon que l'on est dans l'un ou l'autre des deux cas suivants :

i) si $I_i = \emptyset$:

$$u_{ji} = \frac{\frac{1}{\left(d\left(x_i, v_j\right)\right)^{\frac{2}{(m-1)}}}}{\sum_{j=1}^{c}\frac{1}{\left(d\left(x_i, v_j\right)\right)^{\frac{2}{(m-1)}}}} \tag{4.62}$$

ii) Sinon, pour tout $j \in \tilde{I}_i, u_{ji} = 0$ et tout $j \in I_i$, fixer une valeur pour u_{ji} de telle sorte que :

$$\sum_{j \in I_i} u_{ji} = 1 \tag{4.63}$$

c) Incrémenter i et aller à l'étape a).

6) Comparer $U^{(b)}$ et $U^{(b+1)}$ à l'aide d'une norme matricielle : si $\left\|U^{(b+1)} - U^{(b)}\right\| < \varepsilon$ arrêter l'algorithme, sinon, incrémenter b et aller à l'étape **4)**.

ε est un nombre réel prédéfini par l'utilisateur et qui sert dans le critère d'arrêt 6) basé sur la distance séparant la matrice calculée à l'itération au rang $(b + 1)$ de celle calculée au rang (b).

L'étape **5)** fait intervenir deux cas : Dans le premier (i), aucun centroïde de classe ne correspond avec la donnée x_i ; dans le second cas (ii), certains centroïdes de classe sont

identiques à x_i. La formule générale utilisée dans (i) n'est alors pas applicable et x_i est attribué à la classe (aux classes) dont le(s) centroïde(s) lui est (sont) identique(s).

Dans le cas simple de données monodimensionnelles, on voit bien que pour minimiser le critère défini dans (4.56), la dérivée partielle par rapport aux centres doit s'annuler :

$$\frac{\partial J_m}{\partial v_j} = 0 \text{ pour tout } j = 1,2,\dots,c \tag{4.64}$$

Soit dans le cas de la distance euclidienne:

$$\sum_{i=1}^{l} 2u_{ji}^{m}(v_j - x_i) = 0 \quad \text{pour tout} \quad j = 1,2,\dots,c \tag{4.65}$$

ou encore:

$$\sum_{i=1}^{l} u_{ji}^{m} v_j = \sum_{i=1}^{l} u_{ji}^{m} x_i \quad \text{pour tout} \quad j = 1,2,\dots,c \tag{4.66}$$

On tire aisément la relation (4.59) à partir de l'égalité (4.66). Si l'espace des données est de dimension supérieure à 1, on fait en général appel aux multiplicateurs de Lagrange (cf. par exemple [Bezd (80)] ou [Gust (79)]). Il est bien sûr possible de choisir les valeurs des centres au début de l'algorithme et de calculer les matrices U^b à partir des valeurs de ces centres à l'itération de rang b.

Lorsque m varie parmi tous les critères J_m, seul J_1 est sujet à une interprétation physique et géométrique simple. Minimiser J_1 revient, en effet, à minimiser la variance intra-groupe de l'ensemble X. Bezdek [Bezd (76)] a proposé une interprétation physique qui établit une analogie entre la minimisation de la somme des erreurs quadratiques dans J_1 ou J_2 et la minimisation de la résistance totale d'un certain circuit électrique. Cette analogie n'est pas valable pour des valeurs de *m* autres que 1 et 2.

4.6 Les approches de réduction des données

4.6.1 Formulation

L'idée clé de cette méthode est de prendre le problème de la classification L2-SVM comme si on a une plus petite boule engloblante assimilée à un Core-set en utilisant un espace caractéristique $\tilde{Z} = \Phi(X)$ où les données d'apprentissage sont intégrés dans un expansion Φ.

Cependant, nous formulons un algorithme pour trouver les plus petites boules englobantes (MEBs) des images $\tilde{S}$ de $S \subset \tilde{Z}$, quand S est décomposé en un ensemble de partitions S_p où

on procède à une réduction en une plus petite boule englobante (MEB) pour chaque partition (cf. Fig. 4.5).

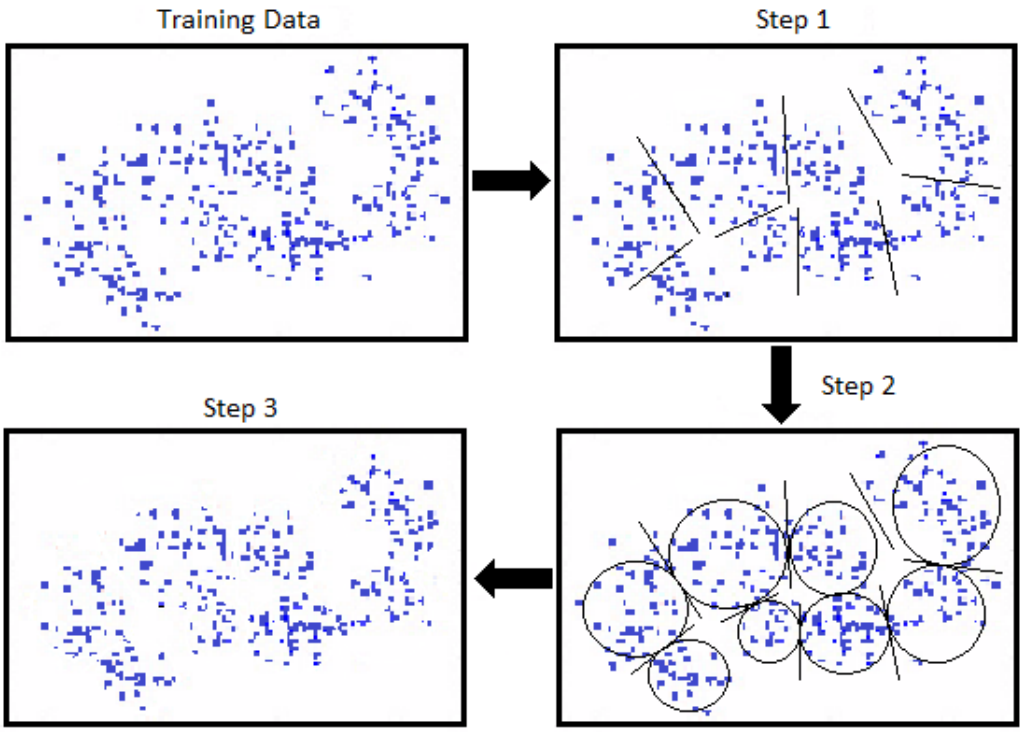


Fig. 4.5: Visualisation du processus d'apprentissage. Réduction globale des partitions en une plus petite boule englobante (MEB).

L'algorithme est basé sur le calcul des Core-sets $\mathcal{C}_k$ pour chaque partition $\tilde{S}_p = \Phi(S_p)$, ensuite on prend l'union des Core-sets $\mathcal{C} = \cup_p \mathcal{C}_p$ comme un approximation au Core-set pour $\tilde{S} = \cup_p \tilde{S}_p$. L'*algorithme 5* dépeint la procédure générique. Dans une première étape, l'algorithme extrait un Core-set pour chaque partition S_p tandis que dans la seconde étape, la plus petite boule englobante des Core-sets est calculée.

La décomposition de S en partitions S_p est réalisée par un algorithme de partitionnement. Pour notre étude, on a utilisé les algorithmes du *k*-plus proches voisins et C-Moyennes Floues développés dans la section précédente.

Algorithm 5 Computation of the MEB of $\tilde{S} = \phi(S)$

Require: A partition of the set S based a clustering algorithm in a collection of subsets S_p

***1*:** **for** Each subset S_p, $p = 1, \ldots, P$ **do**

***2*:** Compute a ϵ-Core-set C_p for one of the two instantiation

3: **end for**
4: Join the core-sets $C = C_1 \cup \ldots \cup C_P$
5: Compute the minimal enclosing ball of C. This is the Minimal enclosing ball of $\tilde{S}$ that define the reduced datasets.

Pour le calcul des Core-sets on utilise l'algorithme de Bãdoiu and Clarkson (Algorithme 1) décrit dans la section 4.3.2.

4.6.2 Instanciation pour l'approche multi-classe *une-contre-une*

Précédemment, nous avons vu que l'apprentissage par la méthode L2-SVM binaire sur un ensemble de donné S est équivalent à construire la plus petite boule englobante (MEB) de S, si $\Phi(x)^T\Phi(x)$ est implémenté en utilisant le noyau $\tilde{k}(x_i, x_j) = y_i y_j k(x_i, x_j) + y_i y_j + \frac{\delta_{ij}}{C}$. La procédure *une-contre-une* nous permet d'obtenir l'extension multi-classe en combinant des Machines à Vecteurs Support pour chaque paire de classes. Une instanciation de *l'algorithme 5* consiste à calculer les Core-sets pour toutes les partitions des données appartenant à chaque paire de classes en supposant qu'on a L classes.

Algorithm 6 Computation of the MEB using OAO (One against One) Multiclass L2-SVMs

1: **for** Each subset S_p , $p = 1, \ldots, P$ **do**
2: **for** Each Class $l = 1, \ldots, L-1$ **do**
3: **for** Each Class $l' = k+1, \ldots, L$ **do**
4: Let $S_p^{ll'}$ the subset of S_p corresponding to class l And l'.
5: Label $S_p^{ll'}$ using the standard binary codes $+1$ and -1 for class l and l' respectively
6: Compute a core-set $C_p^{ll'}$ of $S_p^{ll'}$ Using the kernel $\tilde{k}(x_i, x_j) = y_i y_j k(x_i, x_j) + y_i y_j + \frac{\delta_{ij}}{C}$
7: **end for**
8: **end for**
9: Take the union of the core-set inferred for each pair of classes $C_p = C_p^{ll'} \cup \ldots \cup C_p^{ll'}$
10: **end for**
11: Join core-set $C_S = C_1 \cup \ldots \cup C_P$.
12: Compute the minimal enclosing ball of C_S using the same kernel $\tilde{k}$

4.6.3 Instanciation pour l'approche multi-classe *une-contre-toute*

La procédure multi-classes une-contre-toute est définie par une simple optimisation de l'approche précédente qui coïncide avec la formulation de la plus petite boule englobante en utilisant le noyau $\tilde{k}(x_i, x_j) = y_i^T y_j k(x_i, x_j) + y_i^T y_j + \frac{\delta_{ij}}{C}$. En se basant sur l'*algorithme 5*, l'instanciation pour l'approche une-contre toute est définie comme suit:

Algorithm 7 Computation of the MEB using OAA (One Against All) Multiclass L2-SVM

1: **for** Each subset S_p , $p = 1, \dots, p$ **do**
2: Label each example $x_i \in S_p$ with the code y_{ip} assigned to the class of x_i and let y_i such label
3: Compute a core-set C_p of S_p using the kernel $\tilde{k}(x_i, x_j) = y_i^T y_j k(x_i, x_j) + y_i^T y_j + \frac{\delta_{ij}}{C}$
4: **end for**
5: Join the core-sets $C_S = C_1 \cup \dots \cup C_P$.
6: Compute the minimal enclosing ball of C_S using the same kernel $\tilde{k}$

4.7 Conclusion

Dans ce chapitre, nous avons développé une équivalence entre la formulation d'apprentissage supervisé basé sur le L2-SVM et la formulation d'apprentissage non supervisé basé sur la plus petite boule englobante (MEB). Cette équivalence nous a permis de concevoir un algorithme de réduction de données selon deux approches multi-classes de Machines à Vecteurs Supports.

Chapitre 5

Développement de systèmes d'identification de dialectes basé sur les Modèles de Mélanges de lois Gaussiennes

5.1 Introduction

L'utilisation des Modèles de Mélanges de lois Gaussiennes (**G**aussian **M**ixture **M**odel-**GMM**) ne demande pas une sélection d'une topologie quelconque. Bien que l'utilisation d'un GMM conduise à une perte d'information temporelle des caractéristiques acoustiques, cette dernière n'influe pas sur l'identification des dialectes qui sont considérés comme des langages.

Dans cette section, nous présentons deux systèmes d'identification de dialecte basés sur les Modèles de Mélanges de lois Gaussiennes pour modéliser les caractéristiques de parole acoustiques. Les systèmes d'identification de langages donnant de bonnes performances ont été étudiés par [Riek (91)] et [Ziss (93)]. Suite à ces études, la recherche sur l'identification des langages est focalisée sur l'information phonétique.

Pour notre cas, nous avons conçus deux systèmes d'identification des dialectes, l'un est basé sur les Modèles de Mélanges de lois Gaussiennes, et l'autre basé sur les Modèles de Mélanges de lois Gaussiennes et la technique Modèle du monde (**U**niversal **B**ackground **M**odel - **UBM**). Cette dernière (Modèle du monde) a été utilisée avec succès dans la tâche de reconnaissance du locuteur.

Les composants des deux systèmes sont créés sur la base d'information d'un espace caractéristique acoustique qui est ensuite réduit par notre système de réduction de données étudié en *chapitre 4* englobant tous les dialectes. Avec une perte minime de précision due à une certaine approximation, les techniques utilisées dans les deux systèmes montrent que l'efficacité des Modèles de Mélanges de lois Gaussiennes a été améliorée aux étapes d'apprentissage et de test, suite à de bons résultats obtenus dans la phase d'expérimentation. Evidemment, cette réduction de données conduit à une réduction en complexité de calcul qui nous a permis d'employer plus de composants pour accomplir le besoin d'un Modèle de Mélanges de lois Gaussiennes plus large pour modéliser les caractéristiques d'un dialecte quelconque.

Dans la section 5.2.2, nous détaillons un système d'identification de langages basé sur les Modèles de Mélanges de lois Gaussiennes et sa version améliorée pour l'identification des dialectes. Ensuite, en section 5.4.2, nous décrivons un système d'identification de langage basé sur les Modèles de Mélanges de lois Gaussiennes utilisant la description de la technique du modèle du monde et sa version améliorée basée sur le modèle du monde pour l'identification des dialectes.

5.2 Système d'identification de dialecte basé sur les Modèles de Mélanges de lois Gaussiennes

Un système d'identification de langages est un classifieur où chaque classe est modélisée par un Modèle de Mélanges de lois Gaussiennes (GMM). La classification d'un langage est réalisée suivant un score de vraisemblance calculé par les Modèles de Mélanges de lois Gaussiennes des langages sur un vecteur caractéristique. Pour déterminer le langage pendant l'étape de test du système, nous utilisons de multiples vecteurs caractéristiques. Ceci faisant, les scores de vraisemblance sont accumulés pour chaque langage et la décision est prise après que tous les vecteurs caractéristiques soient traités.

La structure du système d'identification basé sur les Modèles de Mélanges de lois Gaussiennes est très simple. Aussi, les demandes en calcul pour le traitement est faible. Cet intérêt s'étend aussi à la phase de développement de tels systèmes. L'avantage important d'utilisation des Modèles de Mélanges de lois Gaussiennes est qu'aucune transcription phonétique n'est nécessaire pendant la phase d'apprentissage.

5.2.1 Modèle de Mélange de lois Gaussiennes

Un Modèle de Mélange de lois Gaussiennes est une somme pondérée d'un mélange de composants de lois Gaussiennes multi-varié (cf. Fig. 5.1) qui modélise la fonction de densité de probabilité pour un ensemble donné de vecteurs caractéristiques. Elle est écrite sous la forme d'une combinaison linéaire positive de lois Gaussiennes représentant un mélange de G composants [Jeff (98)]:

$$\begin{gathered} \forall x \in \mathbb{R}^d, \quad p(x|\lambda_d) = \sum_{g=1}^{G} \omega_g \mathcal{N}\left(x; \mu_g, \Sigma_g\right) \\ avec\ \omega_g \in \mathbb{R}^+, \mu_g \in \mathbb{R}^d,\ \Sigma_g \in \mathbb{R}^{d^2}\ et\ \sum_{g=1}^{G} \omega_g = 1, \quad \forall g\ \ \omega_g \geq 0 \end{gathered} \tag{5.1}$$

La densité d'une distribution normale à d dimensions est exprimée par

$$\mathcal{N}(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d det\Sigma}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{5.2}$$

ω est le vecteur de poids de la mixture, $\mathcal{N}(x; \mu, \Sigma)$ est la loi gaussienne de moyenne μ et de variance Σ.

Le Modèle de Mélange de lois Gaussiennes (GMM) est défini par la mixture de tous les composants qui représentent le vecteur moyen, la matrice de covariance et le poids pour chacun des composants décrit par :

$$\lambda_d = \{\lambda_g\}_{g=1}^{G} = \{w_g, \mu_g, \sum_g\}_{g=1}^{G} \tag{5.3}$$

Dans un système d'identification de dialecte, chaque dialecte identifié est modélisé par $g^{ème}$ ordre des paramètres GMM.

L'estimation des paramètres d'un Modèle de Mélange de lois Gaussiennes λ_d est réalisée par un algorithme nommé EM (Expectation-Maximisation) [Demp (77)]. Le principe de cet algorithme est d'estimer (*Expectation*) et d'optimiser (*Maximization*) itérativement la vraisemblance des données d'apprentissage aux modèles jusqu'à atteindre une pseudo-stationnarité (approche d'un maximum local). L'algorithme est décrit dans l'annexe B. Soulignons que l'apprentissage par procédure EM ne garantit pas la convergence vers un optimum global. Plusieurs stratégies sont envisageables pour la phase d'initialisation. Par exemple, à partir du dictionnaire fourni par une Quantification Vectorielle [Lind (80)] et l'algorithme décrit dans l'annexe A, nous pouvons procéder à une classification non supervisée (clustering) des données d'apprentissage et estimer les poids, moyennes et covariances de chacun des *clusters*.

L'algorithme EM commence par des paramètres initiales λ_d. Ensuite, Nous procédons à l'estimation des nouveaux paramètres tels que $P(X|\hat{\lambda}_d) \geq P(X|\lambda_d)$, où $X = \{x_1, x_2, \ldots, x_l\}$ est un ensemble de vecteurs caractéristique d'apprentissage extraits à partir d'une collection de prononciations de parole énoncées par un dialecte d et $\hat{\lambda}_d$ représente les nouveaux paramètres.

Sur chaque EM-itération, les paramètres sont mis à jour en utilisant les équations suivantes:

Les poids du composant de la mixture:

$$\widehat{\omega}_j = \frac{1}{l}\sum_{i=1}^{l} p(x_i|\lambda_d)_j \tag{5.4}$$

Les moyennes du composant de la mixture:

$$\hat{\mu}_j = \frac{\sum_{i=1}^{l} p(x_i|\lambda_d)_j x_i}{\sum_{i=1}^{l} p(x_i|\lambda_d)_j} \tag{5.5}$$

Les covariances du composant de la mixture:

$$\hat{\Sigma}_j = \frac{\sum_{i=1}^{l} p(x_i|\lambda_d)_j x_t x_{t'}}{\sum_{i=1}^{l} p(x_i|\lambda_d)_j} - \hat{\mu}_j \hat{\mu}_j^T \tag{5.6}$$

où $\widehat{\omega}_j$, $\hat{\mu}_j$ et $\hat{\Sigma}_j$ sont les paramètres mis à jour.

La probabilité *a posteriori* pour le composant de mélange gaussien *j* est donné par

$$p(x_i|\lambda_d)_j = \frac{\omega_j \mathcal{N}(x_i;\mu_j,\Sigma_j)}{\sum_{g=1}^{G} \omega_g \mathcal{N}(x_i;\mu_g,\Sigma_g)} \quad (5.7)$$

La modélisation à l'aide de Modèle de Mélange de lois Gaussiennes peut être utilisée pour deux objectifs :

- Pour une estimation de n'importe quelle distribution ; le mélange de distributions gaussiennes étant reconnue comme un approximateur universel d'un large éventail de distributions (cf. Fig. 5.1).
- Pour du *soft clustering* où chaque lois gaussienne est considérée comme un représentant (ou un état) (cf. Fig. 5.2). Contrairement à la méthode des K-*moyennes* où l'appartenance à une classe est quantifiée de façon binaire (1 ou 0) pour un modèle de mixture gaussien, l'appartenance étant la probabilité *a posteriori* de la classe connaissant les données. C'est par exemple le cas d'un détecteur parole/non-parole où l'objectif est de séparer les trames de silence de celles de parole.

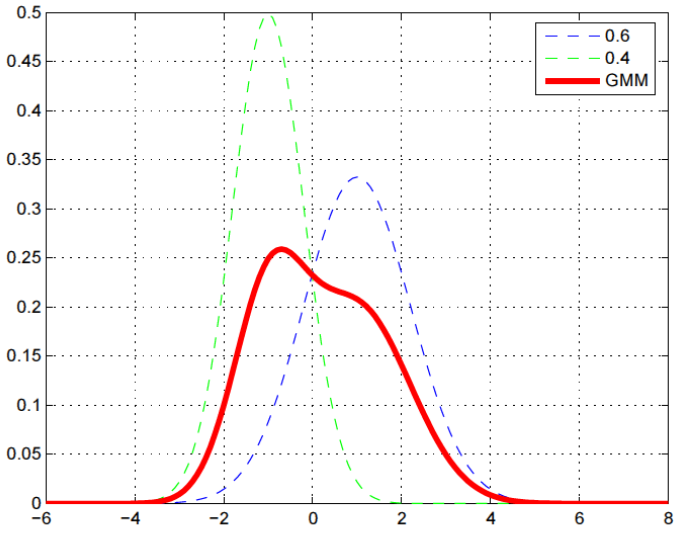


Fig. 5.1: Le Modèle de Mélange de lois Gaussiennes comme un estimateur de densité de probabilité (courbe pleine : combinaison linéaire des deux lois gaussiennes).

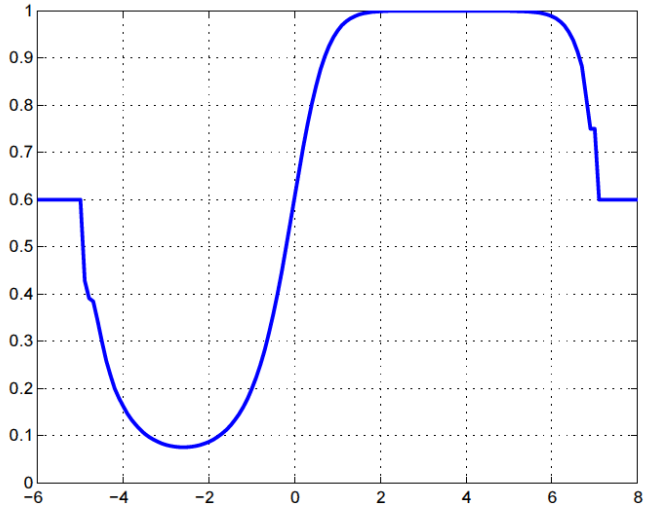


Fig. 5.2: Le Modèles de Mélange Gaussien comme un classifieur souple, la courbe représentant la probabilité de l'appartenance à la classe de droite grâce à une décision bayésienne.

Une propriété importante des Modèles de Mélanges de lois Gaussiennes est la capacité de la combinaison linéaire des lois gaussiennes pour fournir une approximation de lissage pour toute forme de distribution arbitraire [Reyn (95)]. Puisque toutes les données du monde réel ont des distributions multimodales, le Modèle de Mélange de lois Gaussiennes fournit un outil important pour modéliser les caractéristiques des données. Une autre propriété extrêmement utile des Modèles de Mélanges de lois Gaussiennes est la possibilité d'employer la matrice de covariance diagonale au lieu de la matrice de covariance complète. La modélisation d'une distribution avec une matrice de covariance diagonale seule permet de fournir le domaine pour utiliser les composants qui ont leurs vecteurs propres alignés avec les axes de l'espace caractéristique. Ceci signifie que la matrice diagonale du Modèle de Mélange lois Gaussiennes est capable de donner une modélisation approximative des corrélations entre les éléments du vecteur caractéristique.

Dans le cas de notre étude sur l'identification des dialectes, les matrices de covariances sont généralement estimées sous forme diagonale. Dans le cas d'un problème de séparation bi-classe, chaque classe étant représentée par une loi gaussienne de l'équation (5.2), le type de matrices de covariances Σ associées à la loi gaussienne permet de changer la forme de la fonction de décision résultante. *En particulier, si* $\Sigma = \sigma^2.I$ *alors, géométriquement, les distributions sont des hyper-sphères de dimension d et de même rayon. La frontière de décision peut alors se résumer à un hyperplan qui sépare les régions suite à une erreur de*

classification minimale. De plus, si les probabilités *a priori* des deux classes sont identiques alors la frontière se situe à mi-chemin entre les moyennes.

Pendant l'apprentissage, les paramètres du Modèle de Mél20ange de lois Gaussiennes sont définis en utilisant l'estimation de vraisemblance maximale tels que

$$\lambda_d = arg \max_{\lambda_g} \{\prod_{i=1}^{l} p(x_i|\lambda_g)\} \tag{5.8}$$

Pour calculer la vraisemblance d'une séquence de trame $X = \{x_1, x_2, \dots, x_l\}$, pour un modèle paramétré par λ_d, le logarithme est généralement utilisé en considérant l'indépendance des réalisations de la séquence d'apprentissage. Posons la notation $\log(p(.)) = \ell(.)$,

$$\log p(X|\lambda_d) = \ell(X|\lambda_d) = \sum_{i=1}^{l} \log \sum_{g=1}^{G} \omega_g \mathcal{N}(x_i; \mu_g, \Sigma_g) \tag{5.9}$$

La maximisation de $p(X|\lambda_d)$ nécessite l'introduction de variables cachées dont la connaissance permet de trouver une forme analytique au problème. Ces variables cachées sont représentées dans le vecteur $\omega = \{\omega_g\}_{g=1,\dots,G}$.

5.2.2 L'identification de dialectes basée sur les Modèles de Mélanges de lois Gaussiennes

Soit une prononciation d'une dialecte d de séquence de parole $X = \{x_1, x_2, \dots, x_l\}$, le dialecte parlé dans cet énoncé est classifié par

$$\widehat{D} = \arg \max_{1 \le d \le D} p(\lambda_d|X) \tag{5.10}$$

où $\lambda_d$est le Modèle de Mélange lois Gaussiennes du dialecte d et D est le nombre des dialectes que le système peut identifier. Par application de la règle de Bayes, l'équation (5.10) devient :

$$\hat{d} = \arg \max_{1 \le d \le D} \frac{p(X|\lambda_d)p(\lambda_d)}{p(X)} \tag{5.11}$$

Pour la tâche d'identification du dialecte, la chance pour qu'un dialecte soit énoncé dans une prononciation de test est habituellement supposée être équiprobable. De plus, $p(X)$ est la même à travers tous les dialectes. Ainsi, la règle de décision finale du système d'identification peut être réduite à

$$\hat{d} = \arg \max_{1 \le d \le D} p(X|\lambda_d) \tag{5.12}$$

ou alternativement basée sur le score de la moyenne log de vraisemblance :

$$\hat{d} = \arg \max_{1 \le d \le D} \frac{1}{l} \sum_{i=1}^{l} \log p(x_i|\lambda_d) \qquad (5.13)$$

ou $p(x_i|\lambda_d)$ est donnée dans l'équation (5.1). La figure 5.3 montre un système d'identification de dialecte basé sur les Modèles de Mélanges de lois Gaussiennes désignant les étapes d'apprentissage et de test.

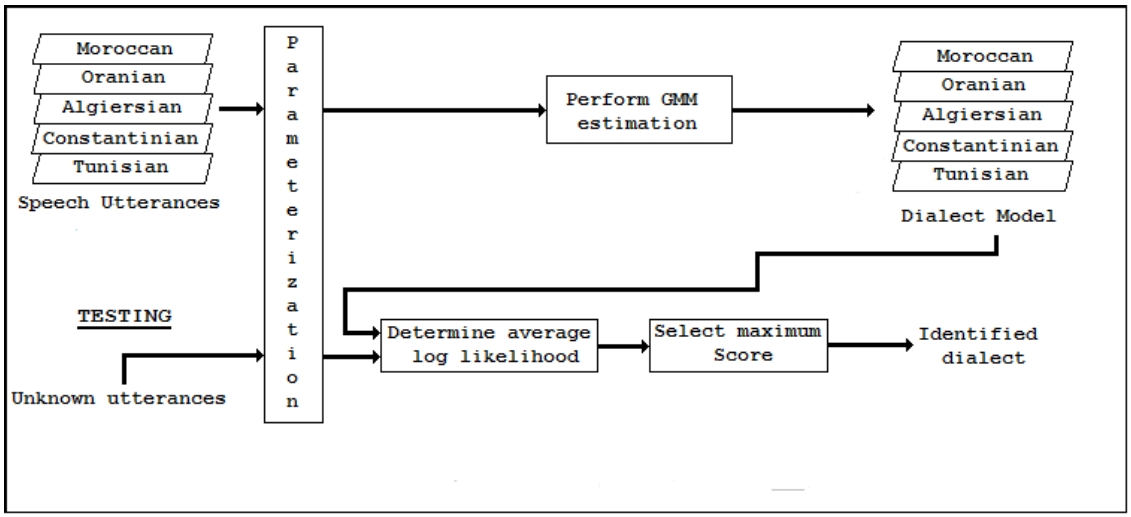


Fig. 5.3: Système d'identification de dialecte basé les Modèles de Mélanges de lois Gaussiennes (baseline).

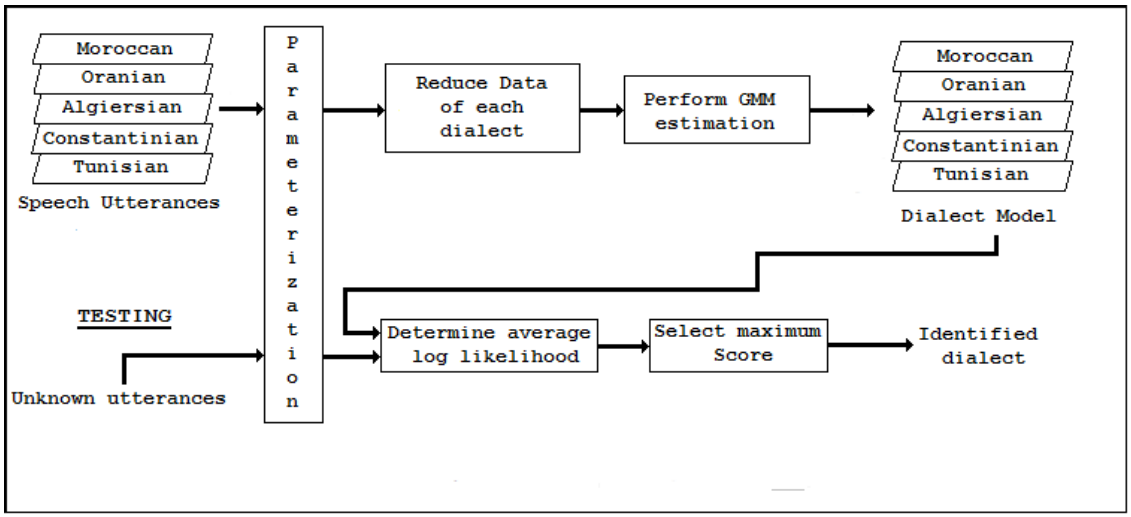


Fig. 5.4: Système d'identification de dialecte basé sur les Modèles de Mélanges de lois Gaussiennes (version amélioré).

La figure 5.4 montre notre système d'identification de dialecte amélioré par l'introduction du système de réduction des données des caractéristiques de paroles en appliquant le problème de

l'équivalence entre la formulation multi-classe sur la norme L2-SVM et la plus petite boule englobante (MEB) traité dans le ***chapitre 4***.

Dans ce qui suit, nous définissons les composants utilisées pour notre deuxième système d'identification de dialectes.

5.3 Adaptation MAP – Maximum à Posteriori

Le but de l'adaptation est de combler l'information manquante dans l'ensemble des observations disponibles pour un dialecte cible (phonèmes peu ou pas présents dans le contenu prononcé qui représenté par son vecteur caractéristique) tout en utilisant des modèles suffisamment complexes pour capter la distribution fortement multimodale des paramètres acoustiques extraits de la parole prononcée. Elle consiste en pratique à initialiser l'apprentissage des Modèles des dialectes avec le Modèle du Monde (UBM supposé appris de manière robuste) et à limiter l'écart entre le modèle appris et l'UBM en intervenant dans les itérations d'apprentissage.

La technique la plus répandue en traitement de la parole est l'adaptation MAP que l'on doit à [Gauv (94)] et qui est devenue la technique de référence dans la vérification du locuteur [Reyn (00)]. Dans notre système, seules les moyennes des lois Gaussiennes de l'UBM sont adaptées en appliquant l'équation (5.5) qui remplace la seconde étape de l'itération EM classique (Maximization). Le paramètre d'adaptation α dépend de l'occupation de chaque loi gaussienne par les données d'apprentissage.

L'adaptation MAP permet de modifier le modèle initial grâce à un jeu de données spécifiques, généralement plus proches des données de test que celles qui ont permis l'apprentissage des modèles acoustiques.

Cette méthode peut être perçue comme étant à mi-chemin entre l'adaptation et l'apprentissage. Aussi, l'adaptation MAP peut être vue comme une seconde phase d'apprentissage contrainte par des connaissances a priori (la valeur initiale des paramètres et/ou leurs distributions). L'objectif de cette adaptation est de rapprocher les paramètres du modèle initial vers des données spécifiques (données d'adaptation). Comme illustré par la figure 5.5 [Bell (06)], l'idée est de déplacer une partie des paramètres par des transformations spécifiques appliquées indépendamment à chaque paramètre.

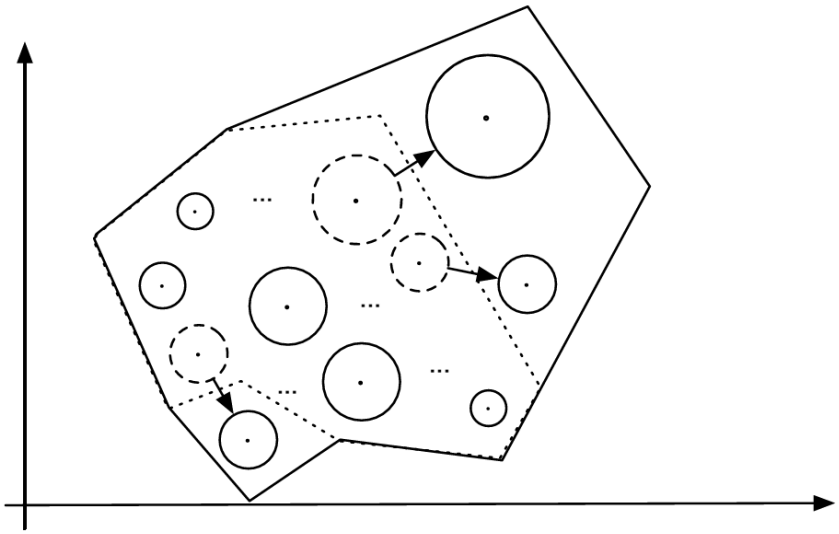


Fig. 5.5: Méthode d'adaptation MAP.

L'inconvénient majeur de cette méthode d'adaptation réside dans la nécessité de disposer suffisamment de données d'adaptation car les paramètres de chaque composante du Modèle de Mélanges de lois Gaussiennes sont considérés comme indépendant les uns des autres. Ceci implique aussi de disposer de données pour toutes les composantes des Modèles de Mélanges de lois Gaussiennes que l'on souhaite adapter. MAP peut donc être considérée comme une méthode d'adaptation locale, du fait de l'adaptation indépendante de chaque composante du Modèle de Mélange de lois Gaussiennes.

Le coût de calcul requis par MAP peut également être considéré comme un inconvénient non négligeable dans le cadre de la reconnaissance/adaptation embarquée dans un système d'identification. Une phase d'apprentissage avec les nouvelles données s'avère nécessaire avant de pouvoir effectuer l'adaptation des paramètres initiaux, et ceci pour chaque modèle acoustique.

La définition de [Gauv (94)] est couramment utilisée en reconnaissance de la parole et celle de [Reyn (00)] en reconnaissance du locuteur ; la différence entre les deux est faible. Le contexte adaptatif utilisé dans notre système étant plus proche de la reconnaissance du locuteur (Identification de dialecte par GMM-UBM) où nous utiliserons la définition de [Reyn (00)].

L'algorithme EM estime le modèle du monde ou le modèle du dialecte de la même façon. Cependant, pour réduire le temps de calcul et améliorer la performance, on extrait qu'un nombre limité de vecteurs caractéristique d'apprentissage extrait à partir d'une collection de

prononciation de paroles parlé par un dialecte, l'adaptation MAP (Bayesian Maximum A Posteriori) est proposée.

Dans MAP, nous supposons que les paramètres λ_d de la distribution $p(X|\lambda_d)$ tels une variable aléatoire qui a une distribution à priori $p(\lambda_d)$. Le principe de MAP relate qu'on doit sélectionner $\widehat{\lambda}_d$ où la densité de probabilité a posteriori $p(X|\lambda_d)$ est maximisée.

$$\begin{aligned}\hat{\lambda}_d &= \arg\max_{\lambda_d} p(\lambda_d|X) \\ &= \arg\max_{\lambda_d} p(X|\lambda_d)\, p(\lambda_d)\end{aligned} \tag{5.14}$$

Nous utilisons généralement l'adaptation MAP des moyennes pour modéliser un dialecte quelconque. De plus, nous pouvons avoir une simplification sans perte de performance en utilisant les paramètres globaux pour s'accorder à la distribution à priori. En se basant sur la densité de probabilité *à posteriori* d e la loi gaussienne g, nous calculons $\widehat{w}_g$, $\widehat{\mu}_g$ et $\widehat{\sum}_g$ qui sont les nouveaux poids, moyennes et matrices de covariance diagonales correspondant au poids, moyennes et matrices de covariance diagonales du model du monde.

La probabilité *a posteriori* pour le composant de mélange de lois gaussiennes g est donné par

$$p(x_i|\lambda_d) = \frac{\omega_g \mathcal{N}(x_i;\mu_g,\Sigma_g)}{\sum_{g=1}^{G} \omega_g \mathcal{N}(x_i;\mu_g,\Sigma_g)} \tag{5.15}$$

L'adaptation pour tous les paramètres de lois gaussiennes g sont défini par:

$$\widehat{\omega}_g = \alpha_g^p \frac{\sum_{i=1}^{l} p(x_i|\lambda_d)}{T} + (1-\alpha)\omega_g \tag{5.16}$$

$$\hat{\mu}_g = \alpha_g^m \frac{\sum_{i=1}^{l} p(x_i|\lambda_d)\, x_i}{\sum_{i=1}^{l} p(x_i|\lambda_d)} + (1-\alpha)\mu_g \tag{5.17}$$

$$\widehat{\sum}_g^2 = \alpha_g^v \frac{\sum_{i=1}^{l} p(x_i|\lambda_d)\, x_i^2}{\sum_{i=1}^{l} p(x_i|\lambda_d)} + (1-\alpha)\left(\sum_g^2 + \mu_g^2\right) - \hat{\mu}_g^2 \tag{5.18}$$

Pour chaque mixture et chaque paramètre, une donnée adaptée dépends d'un coefficient α défini par :

$$\alpha_g^\rho = \frac{\sum_{i=1}^{l} p(x_i|\lambda_d)}{(\sum_{i=1}^{l} p(x_i|\lambda_d)) + r^\rho} \tag{5.19}$$

où r est un facteur de relevance fixe, r^ρ correspond à un facteur de régulation et $\rho \in \{p, m, v\}$. Les coefficients d'adaptation contrôlant la pondération entre les anciennes et les nouvelles estimations sont $\{\alpha_g^p, \alpha_g^m, \alpha_g^v\}$ pour les poids, les moyennes et les variances respectivement. Généralement dans les systèmes GMM-UBM, nous utilisons une seule

adaptation pour tous les paramètres $\alpha_g^p = \alpha_g^m = \alpha_g^v = \frac{\sum_{i=1}^{l} p(x_i|\lambda_d)}{(\sum_{i=1}^{l} p(x_i|\lambda_d)\,)+r}$ avec un facteur de relevance $r = 16$ suite à des expérimentations faites dans [Vure (99)] et il a été montré que le gain était mineur si nous utilisons des coefficients d'adaptation dépendante.

Les poids et la matrice de covariance restent quant à eux fixés aux valeurs de Modèle du Monde. En effet, adapter les poids à une séquence prononcée dans langage cible, selon le taux d'occupation de chaque loi gaussienne, rendrait le modèle trop dépendant du contenu textuel (phonèmes utilisés). Ensuite, adapter les covariances risquerait d'induire un sur-apprentissage (variances trop faibles) dans le cas où le nombre d'observations est limité.

D'autres techniques d'adaptation ont été développées [Meng (03)]. Le lecteur peut se référer aux travaux de [Wood (99), Kunn (00)] pour une revue des techniques utilisées pour l'identification du locuteur ainsi que leur étude théorique. [Mari (02)] compare les performances des diverses techniques d'adaptation pour la tâche de vérification. Enfin, [Sioh (99)] développe une technique d'adaptation analogue à l'adaptation MAP des Modèles de Mélanges de lois Gaussiennes pour les Modèles de Markov Cachés.

5.4 La technique du modèle du monde

Dans le but d'avoir un score de vraisemblance plus robuste et atteindre une performance à l'identification, un ensemble de modèles de tous les dialectes ont été conçus et sont utilisés pour normaliser le score de vraisemblance tels que le dialecte identifié obtiendra un score élevé. Un désavantage de cette approche est que la variation des caractéristiques des dialectes est limitée par le nombre des modèles disponibles durant le calcul du score. Pour résoudre ce problème, nous proposons d'utiliser un seul Modèle de Mélange de lois Gaussiennes qui agit comme un modèle de dialectes de base nommé modèle du monde. Comme son nom l'indique, un modèle du monde représente les caractéristiques de tous les dialectes. C'est un simple Modèle de Mélange de lois Gaussiennes qui comprend un grand nombre de mixtures (typiquement >128) afin de modéliser un nombre conséquent de dialectes.

Le système d'identification GMM-UBM est basé sur une modélisation qui nécessite l'utilisation d'un modèle générique qui a pour objectif l'aboutissement à une modélisation générative où l'estimation de la distribution se base sur les vecteurs cepstraux du signal généré pendant l'apprentissage (modèle de production). En termes statistiques, l'apprentissage

consiste à estimer les paramètres du Modèle de Mélange de lois Gaussiennes maximisant la vraisemblance des données d'apprentissage.

5.4.1 Le modèle du monde

La technique du modèle du monde (UBM) a été appliquée avec succès pour des tâches de vérification et d'identification. L'intérêt de cette technique est qu'elle fournit une réduction de calculs dans les étapes d'apprentissage et de test. De plus, cette technique permet un nombre de mixtures élevé pour chaque Modèle de Mélange de lois Gaussiennes, comparé à l'approche standard des Modèles de Mélanges de lois Gaussiennes.

Pour implémenter la technique du modèle du monde (UBM), il faut avoir une grande quantité de données sur les dialectes afin d'estimer de façon efficace tous les paramètres des Modèles de Mélanges de lois Gaussiennes d'ordre élevé. Pour apprendre un modèle du monde UBM, les données de parole doivent présenter des variations suffisantes pour couvrir toutes les caractéristiques acoustiques de plusieurs locuteurs dans différents dialectes.

Apprentissage des modèles de dialectes

Un modèle du monde représente les caractéristiques d'un grand nombre de locuteurs. Ainsi, l'UBM peut être employé comme une base de modèles pour créer le Modèle de Mélange de lois Gaussiennes pour chaque dialecte à travers une adaptation bayésienne. L'avantage de créer un modèle par adaptation est que la quantité de calcul est beaucoup moindre que le calcul par l'algorithme EM standard. Le nombre des composants de la mixture employée dans un Modèle de Mélange lois Gaussiennes standard basé sur l'identification est généralement limité (typiquement autour de 64 lois gaussiennes) comparé à la technique UBM (autour de 128) due à la demande de calcul sur l'algorithme EM. Cependant, ce problème est résolu par l'approche d'adaptation qui permet de capter des détails fins sur les caractéristiques acoustiques. De plus, l'utilisation de l'adaptation au lieu de l'apprentissage par EM a l'avantage que l'UBM utilise plusieurs types de données différentes. Par conséquent, un espace acoustique plus large est créé et beaucoup de données non visuelles (évènement acoustique) ne résulteront pas de plusieurs modèles particuliers, étant donné que chaque modèle est adapté à partir d'UBM.

Ceci produit un modèle de dialecte plus robuste. L'approche du Modèle de Mélange de lois Gaussien standard ne peut pas traiter ou manipuler ce cas car la donnée utilisée pour apprendre le modèle est limitée à un dialecte particulier.

L'adaptation bayésienne est traitée en utilisant un facteur de relevance fixe r pour adapter les informations du modèle du monde pendant la phase d'apprentissage [Reyn (97)].

Test sur les modèles de dialectes

Le bénéfice en temps de calcul sur la technique du modèle du monde (UBM) n'est pas seulement utile pour l'apprentissage des modèles de dialectes, mais aussi pendant l'étape de test en employant les deux propriétés suivantes:

- Seulement peu de composants de la mixture du Modèle de Mélange de lois Gaussiennes contribue au score de vraisemblance pour un vecteur caractéristique.
- Les composants de la mixture du modèle de dialecte adapté partage une certaine correspondance avec le modèle du monde, du moment que le modèle de dialecte en est adapté.

Par conséquent, le score de vraisemblance du modèle de dialecte contre une entrée de vecteurs caractéristiques peut être calculé en marquant seulement les composants de la mixture les plus significatifs. Suivant la correspondance des composants de la mixture entre le modèle du monde et le modèle de dialecte, les composants de la mixture les plus significatifs peuvent être obtenus en sélectionnant les composants de la mixture dont le score est élevé.

5.4.2 Application de la technique du modèle du monde dans l'identification des dialectes

La discussion sur la technique du modèle du monde citée en haut est dans un contexte de tâche de vérification du locuteur et certaines modifications doivent être faites pour une tâche d'identification des dialectes. Dans la tâche d'identification, aucun modèle du monde n'est nécessaire ; ceci est dû au fait que la nature de sélection est fermée sur l'ensemble de tous les dialectes. Par conséquent, le sens du modèle du monde susmentionné est légèrement modifié. Dans cette tâche, le modèle du monde doit avoir le nom de Modèle de Dialecte Universel pour la tâche d'identification des dialectes dans lequel le modèle représente essentiellement les caractéristiques de différentes classes de la tâche d'identification. Mais pour éviter toute confusion, le même terme modèle du monde est employé.

De là, pour implémenter la technique du modèle du monde pour l'identification des dialectes, nous devons apprendre la modèle du monde par la mise en commun des données de tous les dialectes pour obtenir une description complète de l'espace acoustique caractéristique où aucune normalisation de données n'est nécessaire.

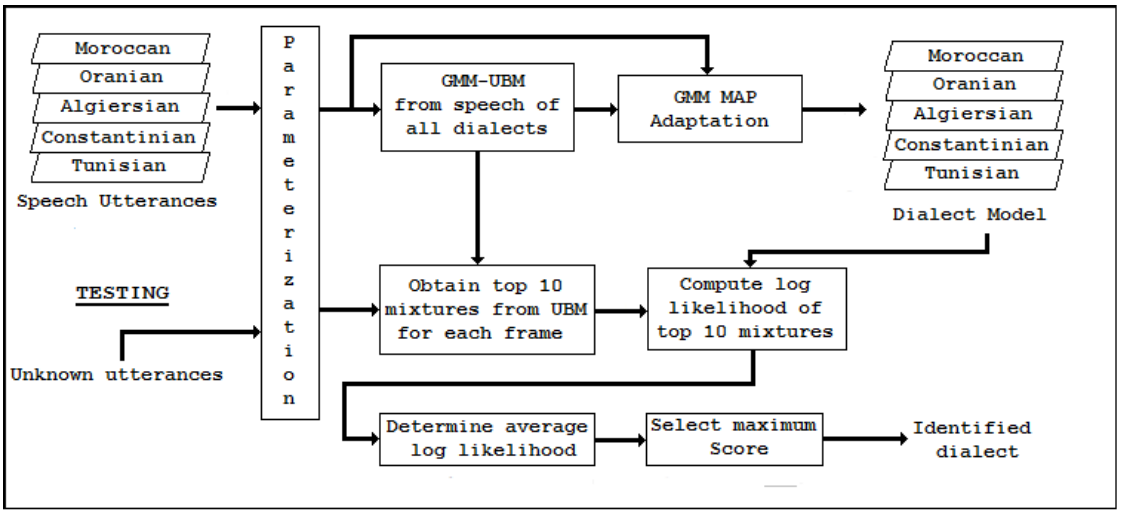


Fig. 5.6: Système d'identification de dialecte basé sur les Modèles de Mélanges de lois Gaussiennes et modèles du monde (GMM-UBM) (baseline).

Contrairement à l'identification du locuteur, un ensemble de base de données du locuteur, séparé de celui des données des imposteurs, est nécessaire pour préparer l'apprentissage de du modèle du monde. Par conséquent, l'implémentation de la technique du modèle du monde pour la tâche d'identification des dialectes est plus simplifiée en termes de préparation des données. Une fois un modèle du monde est obtenu, le Modèle de Mélange de lois Gaussiennes du dialecte est créé par le traitement d'adaptation à partir du modèle du monde en utilisant les données spécifiques du dialecte.

Le processus du système d'identification de dialecte baseline est montré en figure 5.6 où la technique de test est réalisée par les composants de la mixture les plus significatifs. En employant cette stratégie de mixture, le calcul demandé pour le test est réduit. L'inconvénient de cette technique est qu'il peut y avoir une dégradation possible de la précision. Cependant [McLa (99)] a montré que le sacrifice en précision est minime.

Pour ce type de système, la motivation la plus importante d'appliquer la technique de modèle du monde est la réduction en temps de calcul, parce que la plus part des applications

d'identifications des langages nécessitent une opération plus rapide que le temps réelle pour traiter cette tâche d'identification.

En employant cette approche dans un système d'identification de dialecte, la vitesse de calcul est rapidement améliorée comme il est montré dans ce qui suit :

Soit N composant de la mixture pour le Modèle de Mélange Gaussien et pour le modèle du monde, nous choisissons pour le test les C meilleurs composants pour les L dialectes utilisant la technique du modèle du monde. Le nombre de composants de la mixture demandé R pour le test est :

$$R = N_{GMM} + C_{UBM} \times L \tag{5.20}$$

Alternativement, pour le système GMM standard utilisant N composants de la mixture, le nombre de composants demandé pour le test sera:

$$R = N_{GMM} \times L \tag{5.21}$$

Pour le cas de 5 dialectes utilisant 512 composants donne $R = (512 \times 5) = 2560$ composants de test pour un système de Modèles de Mélanges de lois Gaussiennes standard et dans le cas où on prend que les 10 composants top à partir des modèles adapté on aura seulement $R = (512 + 5 \times 10) = 562$ composants de test. Ceci nous donne 450% d'amélioration en temps de calcul.

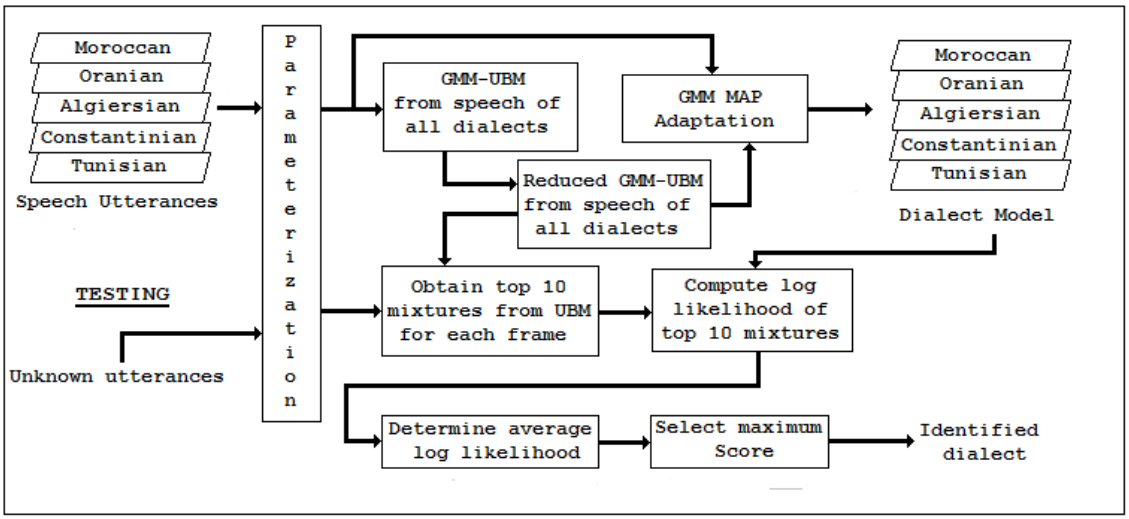


Fig. 5.7: Système d'identification de dialecte basé sur la réduction des Modèles de Mélanges de lois Gaussiennes et modèle du monde.

La figure 5.7 présente notre système d'identification par réduction des Modèles de Mélanges de lois Gaussiennes du modèle du monde. La réduction des Modèles de Mélanges de lois

Gaussiennes conduit à l'élimination des séquences de parole qui ne sont pas adéquates aux modèles du monde des dialectes ou à des séquences bruit intrus pendant l'enregistrement dû aux variations dynamique du canal. Dans ce cas-là, nous pouvons dire que cette réduction peut être assimilée à un filtre de signaux qui ne laisse passer que les séquences de parole par le procédé de classification issue de l'équivalence entre L2-SVM et la plus petite boule englobante (MEB).

5.4.3 Le rapport de vraisemblance

Le score habituellement utilisé dans les systèmes basés sur les Modèles de Mélanges de lois Gaussiennes et modèle du monde est le "log-rapport de vraisemblance" moyen (average log-likelihood ratio) [Reyn (95)] donné par la formule :

$$score_{dial}(X) = \frac{1}{T_X}(\log p(X|\theta_{dial}) - \log p(X|\theta_{UBM})) \tag{5.22}$$

où $p(.|\theta_{UBM})$ et $p(.|\theta_{loc})$ représentent les vraisemblances au Modèle du monde (hypothèse de refus) et au modèle du dialecte cible (hypothèse d'acceptation). L'échelle logarithmique permet d'éviter les problèmes de précision numérique (la densité de probabilité d'une séquence étant une multiplication de densités de probabilités très faibles). La normalisation par la longueur T_X de la séquence sert à assurer la cohérence du seuil de décision. Le terme impliquant θ_{UBM} peut être vu comme une normalisation de score (vraisemblance) par rapport à un corpus global (background). Nous pouvons alors jouer sur la sélection de ce corpus global pour améliorer les performances. Le lecteur peut se référer à [Reyn (97)] pour une comparaison des diverses méthodes de calcul du rapport de vraisemblance.

Pour des soucis d'efficacité mais aussi pour améliorer la robustesse, les vraisemblances des vecteurs d'observations sont couramment estimées par "*N-best scoring*" (où typiquement $N = 10$). Il s'agit d'estimer pour chaque vecteur x_t d'une séquence test :

$$\begin{array}{c} p_{N\ best}\left(x_t|\left\{\omega_g^{UBM}, \mu_g, \Sigma_g^{UBM}\right\}\right) = \sum_{i=1}^{N} \omega_{best(i)}^{UBM} \mathcal{N}\left(x_t|\mu_{best(i)}, \Sigma_{best(i)}^{UBM}\right) \\ au\ lieu\ de \qquad \sum_{g=1}^{G} \omega_g^{UBM} \mathcal{N}\left(x_t|\mu_g, \Sigma_g^{UBM}\right) \end{array} \tag{5.23}$$

où les lois gaussiennes sont rangées par ordre décroissant de vraisemblance du vecteur observé au modèle du monde :

$\omega_{best(1)}^{UBM} \mathcal{N}\left(x_t|\mu_{best(1)}, \Sigma_{best(1)}^{UBM}\right) > \cdots > \omega_{best(N)}^{UBM} \mathcal{N}\left(x_t|\mu_{best(N)}, \Sigma_{best(N)}^{UBM}\right) > \cdots >$
$\omega_{best(G)}^{UBM} \mathcal{N}\left(x_t|\mu_{best(G)}, \Sigma_{best(G)}^{UBM}\right)$.

Etant donné un Modèle de Mélange de lois Gaussiennes adapté du modèle du monde (tel le Modèle de Mélange de lois Gaussiennes du dialecte cible), le calcul des vraisemblances se fait de manière analogue en ne considérant que les N lois gaussiennes dérivées des N "meilleures" lois gaussiennes du modèle du monde. Cette astuce permet alors, une fois la vraisemblance au modèle du monde calculée, d'accélérer l'estimation de la vraisemblance aux autres modèles dérivés du modèle du monde. Ceci allège considérablement les calculs.

5.5 Les expérimentations

5.5.1 Le corpus

Pour cette expérimentation, nous avons utilisé notre propre corpus que nous avons conçu nous-même contenant de la parole spontanée issue des films et des séries télévisées où les conditions acoustiques d'enregistrement ne sont pas similaires pour tous les locuteurs.

Pour la conception de ce corpus, nous avons divisé le Maghreb Arabe en 5 régions sur trois pays le Maroc, l'Algérie et la Tunisie. Le résultat était le corpus Marocain, Oranais, Algérois, Constantinois et Tunisien. Le Tableau suivant définit notre corpus

Table 5.1: Corpus des dialectes du Maghreb

Corpus	Durée (apprentissage.)	Durée (Test)
Marocain	44.19 h	10.53 h
Oranais	40.73 h	9.15 h
Algérois	41.32 h	10.25 h
Constantinois	38.18	8.40 h
Tunisien	42.73	10.35 h

5.5.2 La paramétrisation du signal vocal

L'objectif de cette partie du système est d'extraire des coefficients représentatifs du signal de la parole. Ces coefficients sont calculés à intervalles réguliers. En simplifiant les choses, le

signal de la parole est transformé en une série de vecteurs de coefficients, ces coefficients doivent représenter au mieux ce qu'ils sont censés modéliser et doivent extraire le maximum d'informations utiles pour l'identification.

Prétraitement - Accentuation

Le signal acoustique est numérisé. Il est échantillonné à 16 kHz avec 16 bits de résolution en amplitude.

Une accentuation des aigus est réalisée car les composantes de fréquence élevée sont plus atténuées pendant la transmission du son dans l'air. Nous faisons un filtrage du type passe-haut avec la fonction de transfert :

$$H(z) = 1 - 0.98z^{-1} \tag{5.24}$$

La représentation paramétrique du signal est faite sur des trames de 10 à 50 ms, durées où ce signal est considéré stationnaire.

Analyse cepstrale (MFCC, Mel Frequency Cepstral Coefficients)

L'utilisation des MFCC est motivée par les deux propriétés suivantes :

- Déconvolution : les MFCC découplent les caractéristiques du conduit vocal (qui véhicule la plus grande partie de l'information disponible sur les traits distinctifs de la parole) des caractéristiques générées par l'excitation (information prosodique et information dépendante du locuteur).
- Décorrélation : La transformée en cosinus discrète possède un effet de décorrélation entre les éléments du vecteur de traits. Les MFCC sont une représentation définie comme étant la transformée cosinus inverse du logarithme du spectre de l'énergie du segment de la parole. L'énergie spectrale est calculée en appliquant un banc de filtres uniformément espacés sur une échelle fréquentielle modifiée, appelée échelle Mel. L'échelle Mel redistribue les fréquences selon une échelle non linéaire qui simule la perception humaine des sons [Atta (08)].

Les étapes nécessaires pour l'obtention d'un vecteur caractéristique tiré des coefficients MFCC est illustré par la Figure 5.8.

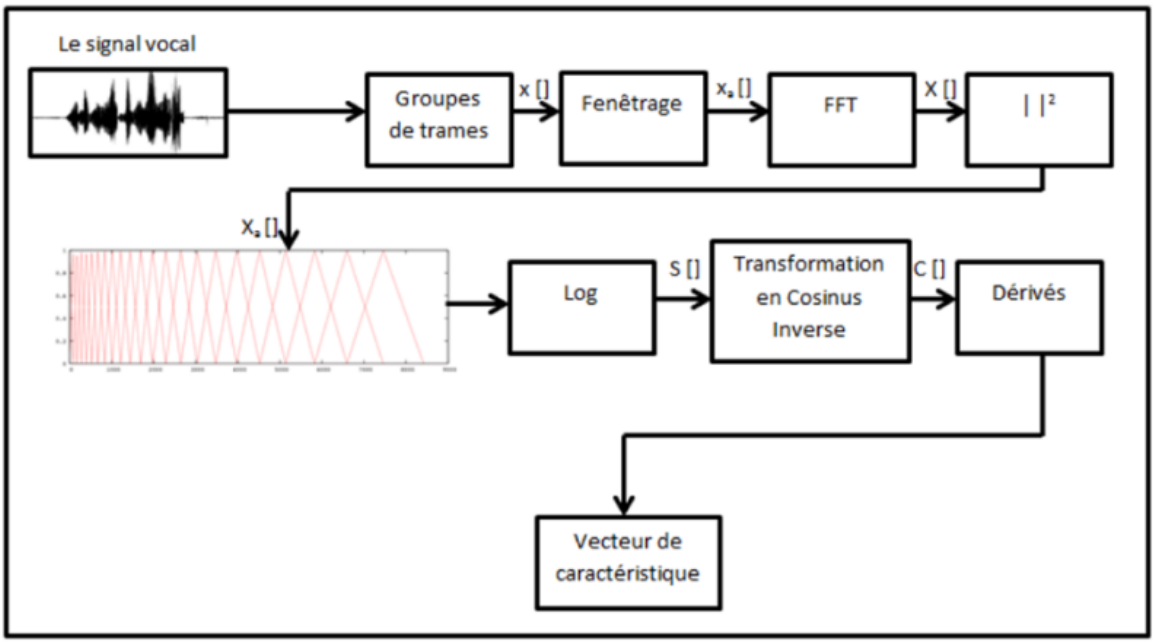


Fig. 5.8: Étapes de calcul d'un vecteur caractéristique de type MFCC.

Après l'accentuation des aigus et le fenêtrage de Hamming, une transformée de Fourier (FFT) est calculée sur la trame d'analyse.

Un filtrage de type Mel s'effectue sur la transformée : des filtres triangulaires sont centrés sur les fréquences 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1150, 1300, 1500, 1700, 2000, 2350, 2700, 3100, 3550, 4000, 4500, 5050, 5600, 6200 et 6850 Hz. Ils sont appliqués après chaque FFT.

Ensuite, nous calculons le logarithme du module de ce spectre puis la FFT inverse pour extraire l'énergie et les 12 premiers coefficients cepstraux. A l'aide des quatre trames adjacentes, nous calculons aussi les dérivées de chaque coefficient et de l'énergie.

Pour normaliser l'effet du canal de transmission, chaque coefficient cepstral est diminué de la valeur moyenne des coefficients prise sur plusieurs secondes.

Shifted Delta Cepstral coefficients (SDC)

Notre utilisation des coefficients SDC pour l'identification du langage est motivée par l'idée d'incorporer une information temporelle relative au signal de parole en vecteurs caractéristiques. Comme il est montré dans [Torr (02)-2], la performance des systèmes d'identification des langages basés sur les Modèles de Mélanges de lois Gaussiennes (GMM)

qui utilisent les SDC sont plus efficaces, et ils sont devenus très populaires parmi les systèmes d'identification des langages.

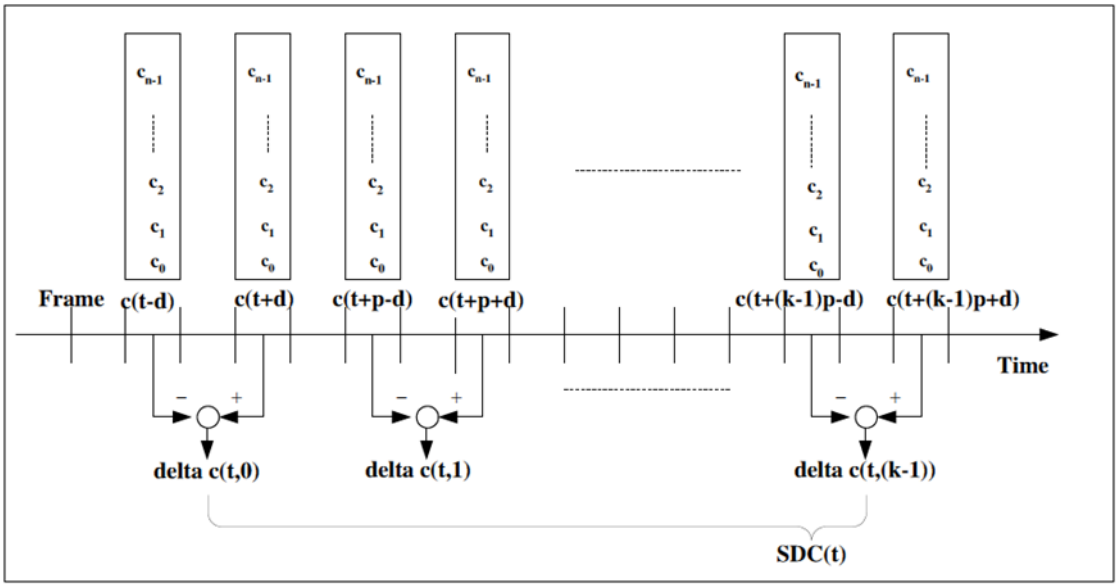


Fig. 5.9: Calcul d'un vecteur caractéristique SDC pour un temps t donné.

Les coefficients SDC sont obtenus par l'empilement des cepstres delta calculés à travers des frames de parole multiples. Les caractéristiques SDC sont spécifiés par 4 paramètres N,d,P,k (habituellement noté par $N-d-P-k$), où :

- N est le nombre des coefficients cepstraux calculés pour chaque frame
- d est le temps d'avance et de retard pour le calcul de delta
- k est le nombre de blocs où les coefficients sont concaténés pour former le vecteur caractéristique final, et
- P est le temps de décalage entre les blocs consécutifs.

Pour chaque frame de données, les MFCCs sont calculés et sont basés sur N incluant c_0 (i.e. les coefficients $c_0, c_1, \dots, c_{N-1}$). Les composants du vecteur SDC au temps t sont calculés comme suit :

$$\Delta c(t,i) = c(t+iP+d) - c(t+iP-d) \tag{5.25}$$

Où $i = 0,1,\dots,k-1$. Le calcul des caractéristiques SDC est illustré dans la figure 5.9.

5.5.3 Expérimentation sur le système d'identification basé sur les Modèles de Mélanges de lois Gaussiennes

La paramétrisation : Nous avons utilisé des trames de 10 secondes pour l'apprentissage et le test. Ensuite, nous avons extrait 12 coefficients MFCC de dimension 39 dérivé de 20 bancs de filtre. Chaque caractéristique est extraite sur des intervalles de 10 millisecondes utilisant un fenêtrage de Hamming de 30 millisecondes sur une bande de parole limitée entre 300 Hz et 3400 Hz. En se basant sur les coefficients MFCCs, nous avons calculé 12 coefficient SDC contrôlés par 4 paramètres $N = 10$, $d = 1$, $P = 3$, et $k = 3$.

La réduction des données: nous avons utilisé une réduction des données selon les deux approches développées dans le ***chapitre 4*** suivant l'*algorithme 6* (approche multi-classe *une-contre-une*) et l'*algorithme 7* (approche multi-classe *une-contre-toute*) utilisant le partitionnement C-Moyenne Flou (Fuzzy C-Mean Clustering) et celui du plus proche voisin (k-NN -- k-Nearest Neighbor).

L'apprentissage: L'apprentissage est basé sur 512 Modèles de Mélanges de lois Gaussiennes (GMMs) avec des matrices de covariance diagonales pour chacun des cinq dialectes en utilisant l'algorithme EM (Expectation Maximisation). Le noyau utilisé pour les deux approches est le RBF (Radial Basis Function) gaussien avec une valeur fixe du $\sigma = 0.50$.

Le test: Le but du test est de trouver le score maximal pour l'identification d'un dialecte. Pour chaque échantillon test, les coefficients SDC sont calculés et leurs Modèles de Mélange de lois Gaussiennes sont comparés à chacun des cinq dialectes sur un ordre de mixture allant de 2 à 512. L'échantillon test appartient au dialecte qui a un score élevé. La précision est calculée pour chaque dialecte en utilisant la formule $Precision = (Correct/Total) \times 100$, où $Correct$ définit le nombre d'échantillons correctement classifiés et $Total$ est le nombre total des échantillons donné pour le test.

Les expériences: Nous avons réalisé une série de cinq expériences, la première a été faite sur le système d'identification de dialecte Baseline (cf. Fig. 5.3), la deuxième a été faite sur le système d'identification de dialecte utilisant la réduction des données (cf. Fig. 5.4) selon l'approche multi-classe *une-contre-une* avec un partitionnement k-NN, la troisième a été faite

sur le système d'identification de dialecte utilisant la réduction des données (cf. Fig. 5.4) selon l'approche multi-classe *une-contre-toute* avec un partitionnement k-NN, la quatrième a été faite sur le système d'identification de dialecte utilisant la réduction des données (cf. Fig.5.4) selon l'approche multi-classe *une-contre-une* avec un partitionnement C-Moyenne Flou, et la cinquième a été faite sur le système d'identification de dialecte utilisant la réduction des données (cf. Fig. 5.4) selon l'approche multi-class une-contre-toute avec un partitionnement C-Moyenne Flou. Finalement, nous comparons la précision de l'identification de dialecte pour les deux systèmes. Suivant les tables Table 5.2, Table 5.3, Table 5.4, Table 5.5, et Table 5.6, nous montrons la précision en pourcentage pour les cinq dialectes selon différentes mixtures.

Table 5.2: Précision des cinq dialectes sur le système d'identification de dialecte Baseline.

Mixture	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**
Marocain	51.36	58.07	62.21	64.01	65.56	66.93	67.33	68.01	69.33
Oranais	60.93	55.37	54.48	61.95	62.85	63.22	67.45	67.33	68.13
Algérois	55.43	58.03	61.02	61.97	62.48	64.07	64.57	65.13	65.63
Constantinois	56.27	58.43	65.23	65.53	67.83	67.73	69.68	70.96	70.11
Tunisien	60.84	66.34	70.07	70.67	73.84	75.13	75.76	77.05	77.85

Table 5.3: Précision des cinq dialectes sur le système d'identification de dialecte basé sur l'approche mult-classe *une-contre-toute* utilisant le partitionnement k-NN.

Mixture	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**
Marocain	55.75	62.46	64.93	65.97	68.72	69.33	69.66	71.04	71.77
Oranais	55.71	55.97	59.08	63.33	65.57	67.11	69.03	69.57	70.07
Algérois	58.23	60.13	62.91	64.84	65.04	66.73	67.02	67.67	68.73
Constantinois	58.14	59.71	66.93	67.04	68.13	69.76	70.89	72.06	72.58
Tunisien	65.73	69.07	71.96	73.09	75.91	77.31	78.02	79.21	80.11

Table 5.4: Précision des cinq dialectes sur le système d'identification de dialecte basé sur l'approche multi-classe *une-contre-une* utilisant le partitionnement *k*-NN.

Mixture	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**
Marocain	59.01	66.82	68.48	71.73	72.11	72.38	73.92	74.15	75.52
Oranais	61.78	58.93	60.24	64.13	65.72	69.48	70.16	71.07	71.96
Algérois	58.17	62.42	63.83	64.35	64.93	66.97	68.08	69.52	70.61
Constantinois	60.72	65.23	67.02	67.79	70.83	70.93	72.03	73.26	74.76
Tunisien	64.73	68.81	73.48	74.08	75.51	76.83	79.74	80.94	81.53

Table 5.5: Précision des cinq dialectes sur le système d'identification de dialecte basé sur l'approche multi-classe *une-contre-toute* utilisant le partitionnement C-Moyenne Flou.

Mixture	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**
Marocain	55.53	66.79	68.34	71.89	72.19	73.02	73.92	74.73	76.18
Oranais	58.08	58.89	61.12	64.72	67.83	68.11	69.98	71.37	73.07
Algérois	56.87	58.93	63.79	66.03	67.11	68.03	68.36	70.07	70.62
Constantinois	61.96	63.88	67.57	68.91	69.91	71.04	73.27	75.06	75.19
Tunisien	66.92	69.14	74.62	75.27	76.86	79.97	82.37	83.03	83.47

Table 5.6: Précision des cinq dialectes sur le système d'identification de dialecte GMM basé sur l'approche multi-classe *une-contre-une* utilisant le partitionnement C-Moyenne Flou.

Mixture	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**
Marocain	59.27	70.11	70.89	74.42	75.33	76.03	76.57	77.84	78.12
Oranais	59.53	60.23	62.73	66.28	68.02	72.04	72.94	73.91	74.63
Algérois	58.37	63.27	66.64	67.13	68.83	69.72	70.61	71.09	71.81
Constantinois	57.34	65.15	71.43	72.16	72.84	73.63	74.19	76.21	77.13
Tunisien	69.57	71.97	74.06	76.16	79.23	81.17	82.84	84.61	85.67

Nous notons que le système d'identification de dialecte basé sur les Modèles de Mélanges de lois Gaussiennes par réduction des données selon l'approche multi-classe *une-contre-une* dépasse le système d'identification de dialecte basé sur les Modèles de Mélanges de lois Gaussiennes selon l'approche multi-classe *une-contre-toute* avec une précision de 73.71 % contre une précision de 71.58% pour un partitionnement k-NN et une précision de 76.09 % contre une précision de 74,33 % pour un partitionnement C-Moyenne Flou. Ces résultats comparés à ceux issus du système d'identification de dialecte basé sur les Modèles de Mélange de lois Gaussiennes Baseline, prouve que notre système donne des résultats meilleurs.

5.5.4 Expérimentation sur le système d'identification basé sur les Modèles de Mélanges de lois Gaussiennes et le modèle du monde

La paramétrisation: Nous avons utilisé des trames de 10 secondes pour l'apprentissage et le test. Ensuite, nous avons extrait 12 coefficients MFCC de dimension 39 dérivés de 20 bancs de filtre. Chaque caractéristique est extraite sur des intervalles de 10 millisecondes utilisant un fenêtrage de Hamming de 30 millisecondes sur une bande de parole limitée entre 300 Hz et 3400 Hz. En se basant sur les coefficients MFCC, nous avons calculé 12 coefficient SDC contrôlés par 4 paramètres $N = 10$, $d = 1$, $P = 3$, et $k = 3$.

La réduction des données: nous avons réduit les données selon les deux approches développées dans le ***chapitre 4*** suivant l'*algorithme 6* (approche multi-classe *une-contre-une*) et l'*algorithme 7* (approche multi-classe *une-contre-toute*) utilisant le partitionnement C-Moyenne Flou (Fuzzy C-Mean Clustering).

L'apprentissage: Pour un modèle du monde (UBM), tous le corpus a été utilisé (les signaux des cinq dialectes ont été mis en ensemble). Ainsi, le modèle du monde est appris sur un grand nombre de mixtures (512) par rapport à l'apprentissage des dialectes. L'apprentissage a été réalisé en se basant sur une matrice de covariance diagonale. Le noyau utilisé pour les deux algorithmes est le RBF Gaussien (Radial Basis Function) avec une valeur $\sigma = 0.50$. L'adaptation MAP est faite seulement sur la moyenne avec un facteur de relevance $r = 16$.

Le test: Le but du test est de trouver le score maximal pour l'identification d'un dialecte. Dans ce processus, les modèles des cinq dialectes sont créés avec un ordre de mixture allant de 2 à 512 pour chaque dialecte. Ainsi, pour chaque échantillon de test, les coefficients SDC sont calculés et leurs Modèles de Mélange de lois Gaussiennes sont comparés à chacun des modèles des cinq différents dialectes. L'échantillon test appartient au dialecte qui a un score élevé. La précision est calculée pour chaque dialecte utilisant la formule $Precision = (Correct/Total) \times 100$, où $Correct$ définit le nombre d'échantillon correctement classifiés et $Total$ est le nombre total des échantillons donné pour le test.

Les expériences: Nous avons réalisé une série de trois expériences, la première a été faite sur le système d'identification de dialecte Baseline (cf. Fig. 5.6), la deuxième a été faite sur le système d'identification de dialecte basé sur les Modèles de Mélange Gaussien et le modèle du monde (GMM-UBM) utilisant la réduction des données (cf. Fig. 5.7) selon l'approche multi-classe *une-contre-une*, et la troisième a été faite sur le système d'identification de dialecte basé sur les Modèles de Mélange Gaussien et le modèle du monde (GMM-UBM) utilisant la réduction des données (cf. Fig. 5.7) selon l'approche multi-class *une-contre-toute*. Finalement, nous comparons la précision de l'identification de dialecte pour les deux systèmes. Suivant les tables Table 5.7, Table 5.8, et Table 5.9, nous montrons la précision en pourcentage pour les cinq dialectes selon différentes mixtures.

Table 5.7: La précision des cinq dialectes sur le système d'identification de dialecte basé sur les Modèles de Mélange de lois Gaussiennes et modèle du modèle du monde Baseline

Mixture	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**
Marocain	54.28	63.11	65.88	70.08	68.87	70.21	70.33	71.43	71.67
Oranais	62.77	55.23	54.93	63.17	65.33	65.15	69.53	69.93	70.54
Algerois	45.67	59.13	62.73	64.83	64.98	66.94	67.12	67.78	67.85
Constantinois	48.03	60.34	67.27	67.93	69.01	69.41	71.83	72.16	72.18
Tunisien	62.57	68.25	72.33	72.91	76.16	76.22	80.91	81.39	81.95

Table 5.8: La précision des cinq dialectes sur le système d'identification de dialecte basé sur les Modèles de Mélanges de lois Gaussiennes et modèle du modèle du monde par réduction des données selon l'approche multi-classe *une-contre-toute.*

Mixture	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**
Marocain	56.08	67.33	68.93	73.93	74.06	74.89	75.14	75.55	75.78
Oranais	63.23	58.67	59.43	64.37	65.19	70.88	71.33	71.93	72.13
Algérois	49.11	61.17	64.58	65.43	65.79	68.16	68.83	69.87	70.18
Constantinois	51.07	60.28	68.57	69.23	71.88	72.09	72.97	73.19	73.83
Tunisien	65.19	69.35	74.53	74.72	76.46	77.03	81.86	82.19	83.02

Table 5.9: La précision des cinq dialectes sur le système d'identification de dialecte basé sur les Modèles de Mélanges de lois Gaussiennes et modèle du modèle du monde par réduction des données selon l'approche multi-classe *une-contre-une*.

Mixture	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**
Marocain	63.83	68.32	71.54	76.19	78.03	78.93	82.32	82.55	83.92
Oranais	64.56	65.27	67.94	68.15	72.73	73.26	75.13	75.19	76.22
Algérois	53.34	62.41	66.17	68.73	69.37	72.11	72.19	74.27	77.67
Constantinois	55.07	63.28	70.39	72.57	73.13	74.61	76.55	77.38	78.58
Tunisien	68.14	71.73	76.23	79.19	81.66	82.95	83.65	85.85	86.07

Nous notons que le système d'identification basé sur les Modèles de Mélanges de lois Gaussien et le modèle du monde (GMM-UBM) par réduction des données selon l'approche multi-classe *une-contre-toute* dépasse le système d'identification basé sur les GMM-UBM avec une précision de 74.99 % contre une précision de 72.84%. Aussi, le système d'identification basé sur les GMM-UBM par réduction des données selon l'approche multi-classe *une-contre-une* donne la meilleure performance avec une précision de 80,49 %.

5.6 Conclusion

Dans ce chapitre, Nous avons développé deux nouveaux systèmes d'identification de Dialectes : le premier est basé sur les Modèles de Mélanges de lois Gaussiennes (GMM) et le deuxième est basé sur les Modèles de Mélanges de lois Gaussiennes utilisant la technique de modèle du monde (GMM-UBM). Des détails ont été donnés sur ces systèmes afin de montrer toute la différence entre un système Baseline et nos deux systèmes pour les deux approches. Aussi, nous avons montré que la réduction des données basée sur l'équivalence entre la formulation L2-SVM et la plus petite boule englobante (MEB) donne de meilleurs résultats, où cette dernière joue vraiment un rôle de filtre sur les données, ce qui explique les bons résultats que nous avons obtenus.

Conclusion Générale et Perspectives

Le travail mené au cours de cette thèse a permis d'obtenir un système d'identification de dialectes performant, capable de prendre en charge des données issues de séquences de paroles reflétant la diversité des conditions d'enregistrement rencontrées dans le monde réel.

Tout d'abord, une étude détaillée est présentée sur le mécanisme complexe qui repose sur une interaction entre les systèmes neurologique et physiologique de l'être humain pour permettre à la mise en action des organes phonatoires nécessaire à la production de la parole qui se base sur deux mécanisme, l'articulation et la phonation.

Aussi, la définition de la langue arabe standard parmi les langues existantes dans le monde donne un aperçu général pour comprendre les caractéristiques des dialectes du Maghreb et d'autres dialectes du monde arabe. La langue arabe appartient au groupe des langues sémitiques et c'est aussi la langue du *Qur'an* (le livre saint islamique). Les mots arabes des mêmes familles ont des racines communes. Une racine sémitique est une séquence de trois consonnes reliées à un concept. Pour créer un mot, on doit rajouter des voyelles et des consonnes qui ne sont pas des racines.

Les dialectes arabes sont généralement subdivisés en deux grandes zones dialectales : la zone orientale qui regroupe les pays du Moyen-Orient et la zone occidentale ou maghrébine. Les dialectes sont souvent décrits par rapport à l'arabe standard, et c'est généralement l'arabe standard qui est employée pour en présenter les règles (comme par dérivation). Les dialectes diffèrent de l'arabe standard en ce qu'ils présentent une grammaire souvent simplifiée et parfois des troncatures lexicales ainsi que des variations ou assimilations dans la prononciation.

Les dialectes du Maghreb ne se distinguent pas de manière tranchée, et bien que les dialectes éloignés soient assez différents, il existe des continuums passant de l'un à l'autre. Ainsi l'Arabe Marocain, Algérien et Tunisien sont trois langues assez proches, mais vis-à-vis des autres dialectes que regroupent les pays arabes, ces derniers sont très différents.

L'identification automatique des langues consiste à reconnaitre automatiquement la langue qui est parlée dans un tour de parole. La première motivation de l'identification des langues est d'utiliser des instruments qui puissent traiter le signal de parole le plus directement possible sans utiliser des ressources linguistiques trop importantes et complexes, en tenant compte des altérations du signal acoustique par des bruits dus à l'environnement ou au canal de communication.

Pour notre cas, le défi était : comment éliminer tous les paramètres des séquences de paroles qui présentent un défaut et ne suivent pas la majorité des séquences du point de vue caractéristique acoustique au niveau d'un dialecte?

Pour remédier à ce problème, nous nous sommes intéressés aux Machines à Vecteurs Support, basés sur la théorie d'apprentissage de Vapnik, qui sont réputés pour donner de bonnes performances dans de nombreux problèmes de classification et c'est un axe de recherche récent.

Une autre difficulté pour la mise en application des Machines à Vecteurs Support est liée au volume de données nécessaire pour que les machines puissent apprendre à réaliser des tâches automatiques sur le signal de parole de manière suffisamment robuste. La complexité calculatoire peut être rédhibitoire dans le cas d'un corpus trop volumineux.

Les Machines à Vecteurs Support ont été introduites comme des extensions non linéaires d'un séparateur linéaire: l'hyperplan de marge maximale. Ils réalisent des séparations non-linéaires dans l'espace des données d'apprentissage à partir de séparations linéaires dans un espace transformé de dimension potentiellement grande, et ce grâce à l'idée des noyaux de *Mercer*.

L'élégance de la construction des Machines à Vecteurs Supports ne masque pas les difficultés de leurs mise en œuvre. La minimisation quadratique est une tâche délicate lorsqu'il s'agit de traiter des problèmes de grande taille. En plus, le problème de réglage des paramètres C et ceux des noyaux semble lourd à résoudre. Pour ce faire, on a fait souvent appel à des méthodes numériques minimisant l'une des bornes de généralisation par rapport à ces paramètres.

Le développement d'une équivalence entre la formulation d'apprentissage supervisé basé sur le L2-SVM et la formulation d'apprentissage non supervisé basé sur la plus petite boule englobante (MEB), nous a permis de concevoir un algorithme de réduction de données selon deux approches multi-classe de Machines à Vecteurs Supports.

Cette étude nous a conduit à développer deux nouveaux systèmes d'identification de dialecte : l'un basé sur les Modèles de Mélange de lois Gaussiennes (GMM), l'autre est basé sur une combinaison du modèle du monde et des Modèles de Mélanges de lois Gaussiennes (UBM-GMM). Des détails ont été donnés sur ces systèmes afin de montrer toute la différence entre un système baseline et les deux systèmes que nous avons proposés basés sur les deux approches. Aussi, nous avons montré que la réduction des données basée sur l'équivalence entre la formulation L2-SVM et la plus petite boule englobante (MEB) donne de meilleurs

résultats où cette dernière joue vraiment un rôle de filtre sur les données, ce qui explique les bons résultats que nous avons obtenus.

Perspectives

En perspective, nous nous fixons comme objectif principal la recherche d'une discrimination efficace entre plusieurs langues par modélisation de leur système vocalique au niveau acoustico-phonétique. Cette problématique doit se baser sur une double réflexion sur les techniques d'ingénierie classiquement mises en œuvre au sein des systèmes d'Identification Automatique des Langues (IAL) et sur les études typologiques menées en linguistique.

L’approche doit être définie sur trois phases

La première phase consiste à extraire du signal les voyelles et consonnes qui nous permettront par la suite d'identifier le dialecte parlé. Le but est de trouver des algorithmes basés sur une localisation spectrale d'événements vocaliques, qui sera adaptée à l'objectif d’identification. Ceci peut assurer une relative indépendance vis à vis des conditions d'enregistrement et de la langue parlée tout en obtenant de bons taux de détection.

La seconde phase vise à obtenir une modélisation discriminante des systèmes vocaliques de chaque langue à partir du cadre classique de la modélisation par les Modèles de Mélanges de lois Gaussiennes (GMM), qui doivent permettre un adaptation de la topologie des modèles aux données d'apprentissage.

La troisième phase vise la prise en compte des segments vocaliques et consonantiques au sein de modèles à trouver et qui doit être très efficace pour une approche vocalique unique, même si l'apport d'une modélisation commune de toutes les consonnes reste faible.

Toute cette étude proposée doit être adaptée aux systèmes réalisés par notre approche développée dans cette Thèse.

ANNEXES

Annexe A

Algorithme VQ (Quantification Vectorielle)

Objectif

La quantification vectorielle consiste à extraire un « dictionnaire » de « prototypes » (ensemble des centroides) d'un grand ensemble représentatifs de données. Le dictionnaire doit respecter le mieux possible leur répartition dans l'espace.

La première version de l'algorithme de construction du dictionnaire pour la quantification est connue sous le nom de Lloyd et fut utilisé pour la quantification scalaire. Cet algorithme a ensuite été généralisé pour la classification automatique et la reconnaissance des formes sous le nom d'algorithme des « *K-means* » ou méthodes des « *nuées dynamiques* ».

Algorithme des K-means

$(y_n), 0 \leq n \leq N$ représente un nuage de points (observations) de $\mathbb{R}^d$, d et la distance euclidienne et la taille du dictionnaire K est fixée.

1 Initialisation

Soit un dictionnaire D_0 de taille K.

2 Construction de la partition

A la $t^{ième}$ itération, le dictionnaire est noté :

$$D_t = \{D_{i,t}\}_{i=1,\dots,K} \quad (1)$$

La partition qui minimise l'erreur de quantification associée à D_t est composée des classes :

$$C_{i,t} = \{y_n | d(y_n, D_{i,t}) \leq d(y_n, D_{j,t}),\ j \neq i\} \quad (2)$$

L'erreur de quantification vaut :

$$Dis_t = \frac{1}{N} \sum_{n=1}^{N} \left[\min_{i=1,\dots,K} d(y_n, \mu_{i,t}) \right] \quad (3)$$

où $\mu_{i,t}$ est le centroide de $C_{i,t}$

3 Test d'arrêt

Si $\frac{(Dis_{t-1}-Dis_t)}{Dis_t} < \epsilon$ alors l'algorithme est terminé. Le dictionnaire recherché est D_{t+1} composé des nouveaux centroides, soit:

$$D_{i,t+1} = \mu_{i,t} \tag{4}$$

Sinon $t = t + 1$ et l'algorithme est repris à l'étape 2.

Puisque cet algorithme n'est que localement optimal, le choix du dictionnaire de départ est important. Une variante très utilisée de l'algorithme de Lloyd est l'algorithme LBG: il procède hiérarchiquement et réalise une sorte d'initialisation itérative au cours de la construction.

Algorithme LBG (Linde, Buzo, Gray)

Le but est de construire un dictionnaire de K, où $K = 2^p$.

1 Initialisation

Le centre de gravité de l'ensemble d'apprentissage est calculé.

Soit d_0 ce vecteur. Le dictionnaire est constitué de d_0, $p = 0$.

$$D_0 = \{d_0\}, \quad |D_0| = 2^p \tag{5}$$

2 Eclatement « Splitting »

Tous les éléments d en nombre 2^k du dictionnaire sont « éclatés » en deux vecteurs.

Ceci se fait par exemple en transformant chaque d en $d + \epsilon$ et $d - \epsilon$, où ϵ est un vecteur aléatoire de variance adaptée aux points du nuage associés à d.

3 Convergence

L'algorithme de Lloyd (cf. section précédente) est appliqué sur le dictionnaire des 2^{k+1} éléments ainsi constitué. Après convergence un dictionnaire « optimal » de 2^{k+1} éléments est obtenu.

4 Arrêt

$k = k + 1$

Si $k > k_0$ fixé à l'avance, alors l'algorithme prend fin, sinon le processus est itéré (2).

Le test d'arrêt peut se faire aussi par rapport à un seuil minimal sur la distorsion des données d'apprentissage par rapport au dictionnaire, comme dans le cas de l'algorithme de Lloyd.

Annexe B

Algorithme EM (Expectation Maximisation)

Rappels

L'expression de la vraisemblance d'une observation y de l'ensemble d'apprentissage, supposée réalisation d'un modèle de mélanges gaussiennes, est donnée par :

$$\sum_{k=1}^{N} v_k \mathcal{N}(y; \mu_k, \Sigma_k) \quad (1)$$

Avec

$$\mathcal{N}(y; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_k|^{\frac{1}{2}}} e^{\left[-\frac{1}{2}(y-\mu_k)^T \Sigma_k^{-1}(y-\mu_k)\right]} \quad (2)$$

et :

N le nombre de composantes du mélange,

v_k le poids de chaque composante,

μ_k la moyenne de chaque composante

Σ_k la matrice de covariance associée.

L'algorithme EM est basé sur la vraisemblance de chaque vecteur observé par rapport à chaque composante gaussienne du modèles.

Algorithme de base

1 **Initialisation** ($t = 0$)

- Initialisation des moyennes μ_k par N points extraits aléatoirement de l'ensemble des observations X. $X = \{x_1, \dots, x_N\}$.
- Initialisation de toutes les matrices de covariance Σ_k à la matrice unité I_p.
- Initialisation équiprobable des poids des composantes : $v_k = \frac{1}{N}$.

OU

- Utilisation de l'algorithme VQ (Quantification Vectorielle) présenté dans l'annexe E pour l'initialisation.

2 Itération (t)

Pour tout $k = 1, \dots, N$

- *Phase d'estimation*

  Calcul de probabilité P_{nk} que le vecteur y_n soit généré par la loi gaussiènne k.

$$P_{nk} = \frac{\frac{v_k}{\frac{d}{(2\pi)^{\frac{d}{2}}}|\Sigma_k|^{\frac{1}{2}}} e^{\left[-\frac{1}{2}(y-\mu_k)^T \Sigma_k^{-1}(y-\mu_k)\right]}}{\sum_{k'=1}^{K} \frac{v_{k'}}{\frac{d}{(2\pi)^{\frac{d}{2}}}|\Sigma_{k'}|^{\frac{1}{2}}} e^{\left[-\frac{1}{2}(y-\mu_{k'})^T \Sigma_{k'}^{-1}(y-\mu_{k'})\right]}} \quad (3)$$

- *Phase de maximisation*

  Réestimation des paramètres à partir des probabilités P_{nk} :

$$\tilde{v}_k = \frac{1}{N}\sum_{n=1}^{N} P_{nk} \quad (4)$$

$$\tilde{\mu}_k = \frac{\sum_{n=1}^{N} P_{nk} y_n}{\sum_{n=1}^{N} P_{nk}} \quad (5)$$

$$\tilde{\Sigma}_k = \frac{\sum_{n=1}^{N} P_{nk}(y_n - \tilde{\mu}_k)(y_n - \tilde{\mu}_k)^T}{\sum_{n=1}^{N} P_{nk}} \quad (6)$$

- Incrémentation de t à $t + 1$ et retour à la phase d'estimation

3 Arrêt de l'algorithme

Calcul de la vraisemblance des observations (y_n).

Si la variation de la vraisemblance descend en dessous d'un seuil fixé, alors l'estimation est terminée, sinon l'estimation est reprise à l'étape 2.

Bibliographie

[Alle (06)] Allen F., Ambikairajah E., and Epps J. (2006) - « ***Warped Magnitude and Phase-Based Features for Language Identification*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06), Toulouse, France, pp. I-201-I204.

[Al-Zo (07)] Al-Zoubi M.B., Hudaib A. and Al-Shboul B. (2007) - « ***A fast fuzzy clustering algorithm*** ». Proceedings of the 6th WSEAS Internatinal Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, Corfu Island, Greece, pp. 28-32.

[Atta (08)] Attabi yazid (2008) - « ***Reconnaissance automatique des émotions à partir du signal acoustique*** ». Thèse PhD, École de technologie supérieure, Montreal, Canada.

[Aria (05)] Arias J., Pinquier J., et André-Obrecht R. (2005) - « ***Evaluation of classification techniques for audio indexing*** ». Proceeding of EUSIPCO.

[Arci (03)] Arcienega M. et Drygajlo A. (2003) - « ***A bayesian network approach for combining pitch and spectral envelope features for robust speaker verification*** ». AVBPA 2003, in Lecture Notes and Computer Science, vol. 2688, pp: 78-85.

[Bãdo (08)] Bãdoiu M., Clarkson K.L. (2008) - « ***Optimal core-sets for balls*** ». Computing Geometry Theory Application. vol. 1(40), pp: 14–22.

[Bell (06)] Bellot O. (2006) - « ***Adaptation au locuteur des modèles acoustiques dans le cadre de la reconnaissance automatique de la parole*** ». Thèse de doctorat, université d'Avignon, LIA.

[Bena (01)] Benarousse L. et Geoffrois E. (2001) - « ***Preliminary Experiments On Language Identification Using Broadcast News Recording*** ». Proceeding of Eurospeech, Aalborg, Denmark.

[Bezd (76)] Bezdek J. C. (1976) - « ***A physical interpretation of fuzzy Isodata*** ». IEEE Transaction Systeme Man Cyber, vol. 6, pp: 387-390.

[Bezd (80)] Bezdek J. C. (1980) - « ***A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms*** ». IEEE Transaction Pattern Analytical Machine Intelligent, vol. 1, pp: 1-8.

[Bezd (81)] Bezdek J. C. (1981) - **« Pattern Recognition with Fuzzy Objective Function Algorithms »**. Advanced Applications in Pattern Recognition, Springer edition.

[Bilm (97)] Bilmes J. (1997) – **« *A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models* ».** Technical report ICSI-TR-97-021, University of Berkeley.

[Boe (01)] Boe L.-J., Bonastre J.-F., et Bimbot F. (2001) -« ***Pourquoi la justice doit arrêter les expertises vocales*** ». Justice, vol. 169, pp:5–11.

[Bona (03)] Bonastre J.-F., Bimbot F., Boe L.-J., Campbell J., Reynolds D., et Magrin-Chagnolleau I. (2003) - « ***Person authentification by voice : A need for caution*** ». Proceeding of Eurospeech.

[Brug (98)] Brugnara F. et De Mori R. (1998) - « ***Spoken Dialogue with Computers*** ». Chapter on Training of Acoustic Models, pp: 171–196. Academic Press.

[Burg (98)] Burges Christopher J.C (1998) – « ***A tutorial on support vector machines for pattern recognition*** ». Journal of Data Mining and knowledge Discovery, vol. 2(2), pp:1–43.

[Call (89)] Calliope (1989) - « ***La parole et son traitement automatique*** ». Edition Masson.

[Camp (03)] Campbell, W. (2003) – « ***A SVM/HMM system for speaker recognition*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03).

[Camp (04)] Campbell W., Singer E., and Torres-Carrasquillo P. (2004), « ***Language Recognition with Support Vector Machines*** ». Proceeding of IEEE Odyssey - The Speaker and Language Recognition Workshop, Toledo, Spain.

[Camp (08)] Campbell W., Sturim D., Torres-Carrasquillo P., and Reynolds D. (2008), « ***A Comparison of Subspace Feature-Domain Methods for Language Recognition*** ». Proceeding of InterSpeech, Brisbane, Australia.

[Catf (88)] Catford J.C. (1988) - « ***A Practical Introduction to Phonetics*** ». Oxford University Press.

[Case (98)] Caseiro D. et Trancoso I. (1998) - « ***Spoken Language Identification Using The Speech-Data corpus*** ». Proceeding of International Conference on Spoken Language Processing ICSLP'98, Sydney, Australia, December.

[Cohe (73)] Cohen D. (1973) - « ***Pour un atlas linguistique de l'arabe*** ». Actes du 1er Congrès d'étude des cultures méditerranéennews d'influence arabo-berbère, pp. 63-69, Alger, Algérie.

[Cox (90)] Cox, D. et O'Sullivan, F. (1990) - « ***Asymptotic analysis of penalized likelihood and related estimators*** ». Annals of Statistics, vol. 18, pp:1676–1695.

[Cram (00)] Crammer K. and Singer Y. (2000) - « ***On the learnability and design of output codes for multi-class problems*** ». Computer Learning Theory, pp. 35–46.

[Cram (03)] Crammer K., Keshet J., et Singer Y. (2003) - « ***Advances in Neural Information Processing Systems*** ». Chapter Kernel design using boosting, vol.15, MIT Press.

[Crist (00)] Cristianini N., Shawe-Taylor J.(2000) – « ***An Introduction to support vector Machines and other kernel-based learning methods***». Cambridge University Press, 2000.

[Crist (02)] Cristianini N., Shawe-Taylor J., Elissee. A., et Kandola J. (2002) – « ***On kernel-target alignment*** ». Advances in Neural Information Processing Systems, vol. 14, pp:367–373.

[Demp (77)] Dempster A. P., Laird N. M., et Rubin D. B. (1977) - « ***Maximum-likelihood from incomplete data via the EM algorithm*** ». Journal of Acoustical Society of America JASA, Vol. 39, pp:1–38.

[Duta (00)] Dutat M. (2000) - « ***Caractérisation de la langue parlée par modèles de séquences d'évènements acoustiques*** ». Thèse de doctorat, Ecole Nationale Supérieure des Télécommunications, Paris, France

[Emba (08)] Embarki M. (2008) - « ***Les dialectes arabes modernes: état et nouvelles perspectives pour la classification géosociologique*** ». Arabica, vol. 55, pp: 583-604

[Falt (01)] Faltlhauser R. et Ruske G. (2001) - « ***Improving speaker recognition performance using phonetically structured gaussian mixture models*** ». Proceeding of Eurospeech.

[Fant (85)] Fant G., Lijencrant J. et Lin Q. (1985) - « ***A four parameter model og glottal fow***». Technical Report STL-QPSR.

[Fine (01)] Fine S., Navràtil J., et Gopinath R. (2001) – « ***Enhancing GMM scores using SVM hints*** ». Proceeding of Eurospeech.

[Flet (00)] Fletcher R. (2000) - « ***Practical methods of optimization*** ». Wiley-Interscience, 2nd ed, New York:.

[Frie (96)] Friedman J. H. (1996) - « ***Another approach to Polychotomous classification*** ». Technical report, Department of Statistics, Stanford University.

[Frit (05)] Fritz, M., Leibe, B., Caputo, B., et Schiele, B. (2005) – « ***Integrating representative and discriminative models for object category detection*** ». Proceeding of IEEE - International Conference in Computer Vision.

[Gana (00)] Ganapathiraju A. et Picone J. (2000) – « ***Hybrid SVM/HMM architectures for speech recognition*** ». Proceeding of Speech Transcription Workshop.

[Gauv (94)] Gauvain J.-L. et Lee C.-H.(1994) – « ***Maximum A Posteriori estimation for multivariate gaussian mixture observations of Markov chains*** ». IEEE Transactions on Speech and Audio Processing, vol. (2)2, pp: 291–298.

[Gibb (02)] Gibbon D., Moore R., and Winski R. (2002) - « ***Handbook of Standards and Resources for Spoken Language Systems*** ». Mouton de Gruyter, Berlin New York.

[Gust (79)] Gustafson D. and Kessel W. (1979) – « ***Fuzzy clustering with a fuzzy covariance matrix*** ». Proceeding of IEEE Conference on Decision Control, San Diego, CA, pp: 761-766.

[Hard (99)] Hardcastle W. and Laver J.(1999), « ***The Handbook of Phonetic Sciences*** ». Wiley-Blackwell.

[Henr (01)] Henrich N. (2001) - « ***Etude de la source glottique en voix parlée et chantée*** ». Thèse de Doctorat, Université de Paris 6, France.

[Hier (96)] Hieronymus J. and S. Kadambe (1996) – « ***Spoken Language Identification Using Large Vocabulary Speech Recognition*** ». Proceeding of International Conference on Spoken Language Processing ICSLP'96. Philadelphia, USA.

[Hier (97)] Hieronymus J. and S. Kadambe (1997) - « ***Robust Spoken Language Identification Using Large Vocabulary Speech Recognition*** ». Proceedings of the IEEE

International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97), p: 1111-1114. Munich, Germany.

[Holl (00)] Hollmén J., Trep V., and Simula O.(2000) - « ***A Learning vector quantization algorithm for probabilistic models*** ». Proceedings of EUSIPCO 2000- European Processing Conference, Vol. 2, pp: 721-724.

[Hou (03)] Hou, F. et Wang, B. (2003) – « ***Text-independent speaker recognition using probabilistic SVM with GMM adjustment*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03).

[Huan (01)] Huang X., Acero A., and Hon H. W., (2001) - « ***Spoken Language Processing: A Guide to Theory, Algorithm and System Development*** ». Prentice Hall PTR.

[Jaak (98)] Jaakkola, T. et Haussler, D. (1998) – « ***Exploiting generative models in discriminative classifiers*** ». Advances in Neural Information Processing Systems, vol. 11.

[Jeff (98)] Jeff A. Bilmes (1998) - «***A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models***». Technical report. International computer science institute.

[Junq (92)] Junqua J. C. (1992) - « ***The variability of speech produced in noise*** ». Proceedings of ESCA on Speech Processing in Adverse Conditions, pp: 43–51.

[Jura (08)] Jurafsky D. and J.H. Martin (2008) – « ***Speech and Language Processing: An Introduction to Natural Language Processing*** ». Computational Linguistics and Speech Recognition, Upper Saddle River, New Jersey: Prentice Hall.

[Just (04)] Justino E., Bortolozzi F., et Sabourin R. (2004) - « ***A comparison of SVM and HMM classifiers in the on-line signature verification*** ». Pattern Recognition Letters.

[Kand (02)] Kandola J., Shawe-Taylor J., et Cristianini N. (2002) - « ***Optimizing kernel alignement over combinations of kernels*** ». Technical report 121, University of London, Department of Computer Science.

[Kawa (95)] Kawabara H., Sagisaka Y., (1995) - « ***Acoustic characteristics of speaker individuality: Control and conversion*** ». Speech Communication Journal, vol. 16, no. 2, p: 165-173.

[Kime (71)] Kimeldorf G. et Wahba G. (1971) – « ***Some results on tchebycheffian spline functions*** ». Journal of Mathematical Analysis and Applications, vol. 33, pp:82–95.

[Kirc (03)] Kirchhoff K., Bilmes J., and Das S. (2003) - « ***Novel approaches to Arabic speech recognition*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03), Hong Kong, vol. 1, pp. 344–347.

[Kirc (04)] Kirchhoff K. and Vergyri D. (2004) – « ***Cross-Dialectal Acoustic Data Sharing for Arabic Speech Recognition*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04).

[Klat (90)] Klatt D., Klatt L., (1990) - « ***Analysis, synthesis, and perception of voice quality variations among female and male talkers*** ». Journal of Acoustical Society of America JASA, vol. 87, no. 2, p: 820-857.

[Kreß (99)] Kreßel U. H. G. (1999) - «***Pairwise classification and support vector machines***». Advances in Kernel Methods - Support Vector Learning. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, pp: 255-268, The MIT Press, Cambridge.

[Kunn (00)] Kunn R., Junqua J.-C., Nguyen P., et Niedzielski N. (2000) - « ***Rapid speaker adaptation in eigenvoice space*** ». IEEE Transaction on Speech and Audio Processing, vol. 8, pp :695–707.

[Lach (10)] Lachachi Nour-Eddine, Adla Abdelkader, (2010) - « ***Multi-class Support Vector Machines Methodology*** ». 1er Congrès International sur les modèles, Optimisation et Sécurité des Systèmes, ICMOSS 2010, Tiaret, Algérie,pp 325-329.

[Lach (11)] Lachachi Nour-Eddine, Adla Abdelkader, (2011) - «***Two Multi-Class Approaches for Reduced Massive Data Sets*** ». The International Arab Conference on Information Technology, ACIT 2011, Riyadh, Saudi Arabia.

[Lach (12)-1] Lachachi Nour-Eddine, Adla Abdelkader (2012) - « ***Reduced Large Datasets by Fuzzy C-Mean Clustering using Minimal Enclosing Ball*** ». Advances in Intelligent Systems and Computing, Volume 171, 2012, pp 305-314 , Springer-Verlag.

[Lach (12)-2] Lachachi Nour-Eddine, Adla Abdelkader (2012) - « ***Reduced universal background model for speech recognition and identification system*** ». Pattern Recognition, Lecture Notes in computer science, volume 7329, pp 303-312, Springer-Verlag.

[Lach (14)-1] Lachachi Nour-Eddine, Adla Abdelkader (2014) - « ***Reduced Data Based Improved MEB/L2-SVM Equivalence*** ». Pattern Recognition, Lecture Notes in computer science, volume 8495, pp 1-10, Springer-Verlag.

[Lach (14)-2] Lachachi Nour-Eddine, (2014) - « ***Reduced Universal Background Model for Speech Identification System based improved Minimum Enclosing Ball Algorithms*** ». International Conference on Artificial Intelligence and Information Technology ICA2IT'14.

[Lach (14)-3] Lachachi Nour-Eddine, Adla Abdelkader (à paraitre) - « ***Two approaches for dialect identification based on L2-SVMs reduced to MEB problems*** ». Special Issue on: **"Recent Advances in Signal and Image Processing".** International Journal of Computational Vision and Robotics, INDERSCIENCE PUBLISHERS

[Lach (15)] Lachachi Nour-Eddine, Adla Abdelkader (2015) - « ***GMM-Based Maghreb Dialect Identification system*** ». Journal of Information Processing System (JIPS), volume 11, n.01, pp 22-33, edition KIPS.

[Lave (94)] Laver J. (1994) – « ***Principles of Phonetics*** ». Cambridge University Press.

[Lee (07)] Lee C.h. et al. (2007) – « ***Advances in Chinese Spoken Language Processing*** ». World Scientific Publishing Company.

[Li (02)] Li Q., Zheng J., Tsai A., et Zhou Q. (2002) – « ***Robust endpoint detection and energy normalization for real-time speech and speaker recognition*** ». IEEE Transaction on Speech and Audio Processing, vol. 10(3).

[Lind (80)] Linde Y., Buzo A., et Gray R. (1980) – « ***An algorithm for vector quantization design*** ». IEEE Transaction on Communications, vol. 28(1), pp:84–95.

[Liu (02)] Liu, M., Chang, E., et Dai, B.-Q. (2002) – « ***Hierarchical gaussian mixture models for speaker verification*** ». Proceeding of International Conference on Spoken Language Processing ICSLP'02.

[Liu (06)] Liu, M., Dai, B., Xie, Y., et Yao, Z. (2006) – « ***Improved GMM-UBM/SVM for speaker verification*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06).

[Lorc (95)] Lorche M. P. et Meara P. (1995), « ***Can people discriminate language they don't know?*** ». Journal of Language Science, vol. 17(1), pp:65-71.

[Magr (01)] Magrin-Chagnolleau I., Gravier G., et Blouet R. (2001) – « ***Overview of the 2000-2001 ELISA consortium research activities*** ». Proceeding of IEEE Odyssey.

[Mari (02)] Mariéthoz J. et Bengio S. (2002) – « ***A comparative study of adaptation methods for speaker verification*** ». Proceeding of International Conference on Spoken Language Processing ICSLP'02.

[Mash (05)] Mashao, D. (2005) - « ***Comparing SVM and GMM classifiers on the parametric feature-sets*** ». Technical report. South Africa Institute of Electrical Engineers (SAIEE).

[Mate (05)] Matejka P., Schwarz P., Cernocky J., and Chytil P. (2005), « ***Phonotactic Language Identification using High Quality Phoneme Recognition*** ». Proceeding of InterSpeech, Lisbon, Portugal.

[Mend (96)] Mendoza S., et al (1996) – « ***Automatic Language Identification Using Large Vocabulary Continuous Speech Recognition*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'96).

[Meng (03)] Mengusoglu E. (2003) – « ***Confidence measure based model adaptation for speaker verification*** ». Proceeding of Communication, Internet and Information Technology.

[McLa (99)] McLaighlin J., Reynolds D. A. and Cleason T. (1999) – « ***A Study of Computation Speed-Ups of the GMM-UBM Speaker Recognition Système*** ». Proceeding of EuroSpeech, vol. 3 pp. 1215-1218.

[Mika (99)] Mika S., Scholkopf B., Smola A., Muller K., (1999) - « ***Advances in Neural Information Processing Systems*** ». Kernel PCA and denoising in feature spaces chapter, vol. 11, pp: 536–542. MIT Press.

[More (03)] Moreno, P. et Ho, P. (2003) – « ***A generative model based kernel for SVM classification in multimedia applications*** ». Advances in Neural Information Processing Systems.

[Muth (93)] Muthusamy Y. K., (1993) - « ***A Segmental Approach to Automatic Language Identification*** ». Phd Thesis, Oregon Graduate Institute of Science and Technology.

[Muth (94)-2] Muthusamy Y.K., Barnard E., and Cole R.A. (1994) - « ***Reviewing Automatic Language Identification*** ». IEEE Signal Processing Magazine. p. 33-41.

[Muth (94)-1] Muthusamy Y. K, Jain N., et Cole R. A. (1994) - « ***Perceptual Benchmarks for Automatic Language Identification*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'94), Adelaide, Australia.

[Pede (06)] Pedersen R. and Schoeberl M. (2006) - « ***An embedded support vector machine***». Proceedings of the IEEE in Intelligent Solutions in Embedded Systems, International Workshop, pp. 1–11.

[Pele (01)] Pelecanos J. et Sridharan S. (2001) - « ***Feature warping for robust speaker verification*** ». Proceeding of ISCA – A Spreaker Odysssey – The Speaker Recognition Workshop, Crete, Greece.

[Plat (99)] Platt J. C. (1999) - « ***fast training of support vector machines using sequential minimal optimization*** ». Advances in Kernel methods, pp:185-208. MIT Press Cambridge

[Plat (00] Platt J. C., Cristianini N., and Shawe-Taylor J. (2000) - « ***Large margin DAGs for multiclass classification*** ». S. A. Solla, T. K. Leen, and K. R. Müller, editors, Advances in Neural Information Processing Systems, vol. 12, pp: 547-553, The MIT Press.

[Poth (05)] Pothin J.-B. et Richard C. (2005) - « ***Kernel machines : une nouvelle méthode pour l'optimisation de l'alignement des noyaux et l'amélioration des performances*** ». 20eme Colloque sur le Traitement du Signal et des Images, pp:1196-1199.

[Rabi (93)] Rabiner, L. and Juang B.H. (1993) – « ***Fundamentals of Speech Recognition*** ». Prentice-Hall, NJ.

[Rain (03)] Raina, R., Shen, Y., Y.Ng, A., et McCallum, A. (2003) – « ***Classification with hybrid generative/discriminative models*** ». Advances in Neural Information Processing Systems Proceeding.

[Rajm (07)] Rajman M., (2007) - « ***Speech and Language Engineering*** ». EPFL Press.

[Reyn (95)] Reynolds D. A. and Rose R. C. (1995) – « ***Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'95), vol. 3, no. 1, pp: 72-83.

[Reyn (97)] Reynolds D. A. (1997) - « ***Comparaison of Background Normalization Methods for Test-Independant Speaker Verification*** ». Proceeding of EuroSpeech, vol. 2, pp: 963-963.

[Reyn (00)] Reynolds D.-A., Quatieri T.-F. et Dunn R.-B. (2000) – « ***Speaker verification using adapted gaussian mixture models*** ». Digital Signal Processing, vol. 10, pp:19–41.

[Riek (91)] Riek L., Mistreta W., and Morgan D. (1991) – « ***Experiments in Langage Identification*** ». Technical Report SPCOT-91-002, Lockheed Sanders Inc.

[Scha (06)] Schaffoner, M., Kruger, S., Andelic, E., Katz, M., et Wendemuth, A. (2006) - « ***Limited training data robust speech recognition using kernel-based acoustic models*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06).

[Sche (06)] Scheffer, N. et Bonastre, J.-F. (2006) – « ***Fusing generative and discriminative UBM-based systems for speaker verification*** ». Proceeding of the 2nd international workshop on MMUA (*MultiModal User Authentification*).

[Scho (02)] Scholkopf B. et Smola A. (2002) - « ***Learning with kernels*** », MIT Press.

[Schu (06)] Schultz T. and K. Kirchhoff (2006) - « Multilingual Speech Processing ». Elsevier Edition.

[Shaw (04)] Shawe-Taylor J. et Cristianini N. (2004) - « ***Kernel Methods for Pattern Analysis*** ». Cambridge University Press.

[Sioh (99)] Siohan O., Chesta C., et Lee C.-H. (1999) – « ***Hidden Markov Model Adaptation using Maximum A Posteriori Linear Regression*** ». Proceeding of Workshop for Robust Methods for Speech Recognition in Adverse Conditions.

[Spen (95)] Spencer A. (1995), « ***Phonology: Theory and Description (Introducing Linguistics)*** ». Wiley-Blackwell.

[Stoc (96)] Stockmal V., Muljani D., et Bond Z. S. (1996) - « ***Perceptual Features of unknown Foreign Language as revealed by Multi-dimensional Scaling*** ». Proceeding of International Conference on Spoken Language Processing ICSLP'96.

[Stur (02)] Sturim D., Reynolds D., Dunn R., et Quatieri T. (2002) - « ***Speaker verification using text-constrained gaussian mixture models*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02).

[Szed (05)] Szedmak S., & Shawe-Taylor J. (2005) - « ***Multiclass learning at one-class complexity*** ». Technical Report No: 1508, School of Electronics and Computer Science, Southampton, UK.

[Tax (99)] Tax D. M. J. and Duin R. P. W (1999) - « ***Support vector domain description*** ». Pattern Recognition Letters, vol. 20(14), pp:1191–1199.

[Temp (87)] Templeman A. B. and Li X. S. (1987) - « ***A maximum entropy approach to constrained nonlinear programming*** ». Engineering Optimization, vol. 12, pp: 191–205.

[Toma (05)] Tomasi C. (2005) – « ***Estimating gaussian mixture densities with EM - a tutorial*** ». Technical report, Duke University.

[Torr (02)-1] Torres-Carrasquillo P. A., Reynolds D. A., and Deller J. J.R. (2002), « ***Language Identification using Gaussian Mixture Model Tokenization*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02).

[Torr (02)-2] Torres-Carrasquillo P.A., Singer E., Kohler M.A., Greene R.J., Reynolds D.A., and Deller Jr J.R. (2002) – « ***Approaches to language identification using Gaussian***

mixture models and shifted delta cepstral features ». Proceeding of International Conference on Spoken Language Processing ICSLP'02, pp.89-92, Denver, USA.

[Tsan (05)] Tsang I., Kwok W. & Cheung P.-M. (2005), « ***Core Vector Machines: Fast SVM training on very large data sets*** ». Journal of Machine Learning Research, vol. 6, pp: 363-392.

[Vapn (95)] Vladimir Vapnik (1995) - « ***The nature of statistical learning theory*** ». Springer-Verlag.

[Vasc (04)] Vasconcelos, N., Ho, P., et Moreno, P. (2004) – « ***The kullback-leibler kernel as a framework for discriminant and localized representations for visual recognition*** ». Proceeding of European Conference on Computer Vision.

[Verg (05)] Vergyri D., Kirchhoff K., Gadde R., Stolcke A., and Zheng J. (2005) – «***Development of a Conversational Telephone Speech Recognizer for Levantine Arabic***». Proceeding of Interspeech.

[Vure (99)] Vuren S. (1999) - « ***Speaker Verification in a Time Feature Space*** ». Ph.D. thesis, Oregon Graduate Institute.

[Wade (08)] Wade S. and Reynolds D. (2008), « ***Improved GMM-based language recognition using constrained MLLR transforms*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'08), pp. 4149-4152.

[Wang (02)] Wang L. et Luk Chan K. (2002) - « ***Learning kernel parameters by using class separability measure*** ». Advances in Neural Information Processing Systems.

[Wang (06)] Wang L., Ambikairajah E., and Choi E.H.C (2006) - « ***Automatic Tonal and Non-Tonal Language Classification and Language Identification Using Prosodic Information*** ». Proceeding of International Symposium on Chinese Spoken Language Processing, pp: 485-496. Singapore.

[Wong (02)] Wong E. and Sridharan S. (2002) - « ***Methods to Improve Gaussian Mixture Model Based Language Identification System*** ». Proceeding of International Conference on Spoken Language Processing ICSLP'02, Denver, USA.

[Wong (04)] Wong E. (2004) - « ***Automatic Spoken Language Identification Utilizing Acoustic and Phonetic Speech Information*** ». Phd Thesis, Queensland University of Technology.

[Wood (99)] Woodland P. (1999) - « ***Speaker adaptation : Techniques and challenges*** ». Technical Report. Cambridge University.

[Xian (02)] Xiang B., Chaudhari U., Navràtil J., Ramaswamy G., et Gopinath R. (2002) - « ***Short-time gaussianization for robust speaker verification*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02).

[Yan (95)] Yan Y.(1995) - « ***Development of An Approach to Language Identification based on Language-dependent Phone Recognition*** ». PhD Thesis, Centre for Spoken Language Understanding, Oregon Graduate Institute of Science and Tecnology.

[Zilc (04)] Zilca R., Pelecanos J., Chaudhari U., et Ramaswamy G. (2004) - « ***Real Time Robust Speech Detection for Text Independent Speaker Recognition*** ». Proceeding of IEEE ODYSSEY - The Speaker and Language Recognition Workshop.

[Zhou (03)] Zhou B. et Hansen J. (2003) - « ***Discriminative acoustic model using eigenspace mapping for rapid speaker adaptation*** ». Proceedings of the IEEE International Conference on Acoustics Speech, and Signal Processing (ICASSP'03).

[Ziss (93)] Zissman M. A. (1993) - « ***Automatic Langage Identification Using Gaussian Mixture and Hidden Markov Models*** ». Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'93), vol. 2, pp. 399-402.

[Ziss (96)] Zissman M. A. (1996) - « ***Comparison of four approaches to automatic language identification of telephone speech*** ». Proceedings of the IEEE Transactions on Speech and Audio Processing, vol. 4, pp. 31-44.

[Ziss (01)] Zissman M. A. et Berkling K. M., (2001)-« ***Automatic Language Identification*** ». Speech Communication, vol. 35(1-2), pp:. 115–124