Classification automatique de flux radiophoniques par Machines à Vecteurs de Support

Mathieu Ramona

▶ **To cite this version:**

Mathieu Ramona. Classification automatique de flux radiophoniques par Machines à Vecteurs de Support. Machine Learning [stat.ML]. Télécom ParisTech, 2010. Français. NNT : . pastel-00529331

HAL Id: pastel-00529331
https://pastel.hal.science/pastel-00529331

Submitted on 25 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse

présentée pour obtenir le grade de docteur
de l'Ecole Télécom ParisTech

Spécialité : **Signal et Images**

Mathieu RAMONA

Classification automatique de flux radiophoniques par Machines à Vecteurs de Support

Soutenue le 21 juin 2010 devant le jury composé de :

Cédric Richard	Président
Régine André-Obrecht	Rapporteurs
Jean-François Bonastre	
Geoffroy Peeters	Examinateurs
Marc Brelot	
Gaël Richard	Directeurs de Thèse
Bertrand David	

Remerciements

Je tiens à remercier tout d'abord Alexis Ferrero pour m'avoir donné l'opportunité de réaliser cette thèse au sein de RTL, ainsi que Yves Grenier, pour m'avoir accueilli au département TSI de TELECOM ParisTech.

Je remercie également mes directeurs de thèse Gaël Richard et Bertrand David, avec une grande gratitude, pour leur encadrement, leur patience et leurs nombreux conseils. J'estime avoir eu beaucoup de chance de pouvoir travailler avec des directeurs de thèse sachant doser à la perfection le subtil équilibre entre encadrement et cordialité.

Merci également à Marc Brelot, pour ces trois années passées à RTL à ses côtés, pour sa bonne humeur constante et ses efforts pour suivre d'aussi près que le lui permettaient ses obligations, l'avancement de mes travaux. Je remercie également mes collègues à RTL, en particulier Benoît Pellan, Pierre Souchay et Jonathan Launay.

Merci également à tous les membres de la troupe de l'ATQL, qui m'ont apporté un cadre d'évasion unique. J'adresse en particulier une pensée émue et infiniment reconnaissante à M. Claude Coulon, qui m'a transmis l'amour du théâtre.

Merci enfin à tous ceux qui m'ont aidé et soutenu durant ces années de thèse : ma famille, mes amis, et surtout Anne, pour son soutien inconditionnel, sa patience et sa simple présence à mes côtés durant ces derniers mois. Merci à toi.

Table des matières

Remerciements **3**

Table des matières **5**

Notations **7**

1 Introduction **9**
1.1 Vers une radio numérique . . . 9
1.2 Applications de l'indexation audio pour la radio . . . 10
1.3 « Qu'est-ce que la musique ? » . . . 11
1.4 Classification par Machines à Vecteurs de Support . . . 12
1.5 Problématiques . . . 13
1.6 Résumé des contributions . . . 13
1.7 Structure du document . . . 14

2 État de l'art **16**
2.1 Applications de la classification audio . . . 16
2.2 Taxonomie audio . . . 17
2.3 Techniques de classification . . . 18
2.4 Caractérisation audio . . . 23

I Classification par Machines à Vecteurs de Support **29**

3 Présentation des Machines à Vecteurs de Support **33**
3.1 Classification supervisée . . . 33
3.2 Prélude . . . 34
3.3 Machines à Vecteurs de Support linéaires . . . 34
3.4 Principe de Minimisation du Risque Structurel . . . 36
3.5 Noyaux . . . 37
3.6 Machines à Marge souple . . . 41
3.7 Méthodes à noyaux . . . 42
3.8 Une méthode universelle d'apprentissage . . . 43

4 Sélection du noyau **45**
4.1 Illustration sur des données artificielles . . . 46
4.2 Stratégies d'affinage . . . 48
4.3 Critères d'évaluation basés sur l'erreur de généralisation . . . 50
4.4 Critères basés sur la séparation de classes . . . 53
4.5 Facteur d'erreur C . . . 59
4.6 Evaluation des critères de sélection de noyau . . . 63

5 Stratégies multi-classes **69**
5.1 Combinaisons de SVM . . . 70
5.2 Reformulation des SVM . . . 76
5.3 Discussion et Conclusion . . . 77

II Caractérisation audio **79**

6 Application sur un signal audio **83**
6.1 Architecture du système de classification . . . 83
6.2 Analyse du signal en trames . . . 84
6.3 Intégration temporelle . . . 84
6.4 Normalisation des descripteurs . . . 86
6.5 Liste des descripteurs employés . . . 87
6.6 Discussion . . . 89

7 Sélection de descripteurs **91**
7.1 Introduction . . . 91
7.2 Taxonomie des algorithmes de sélection . . . 92
7.3 Méthodes filtres classiques . . . 95
7.4 Méthodes à noyaux . . . 97
7.5 Propositions d'algorithmes efficaces de sélection . . . 100
7.6 Synthèse . . . 105
7.7 Expériences comparatives . . . 105
7.8 Commentaires . . . 112

III Approches dynamiques **115**

8 Algorithmes de post-traitement **119**
8.1 Règles heuristiques . . . 119
8.2 Filtrages . . . 120
8.3 Modèles de Markov Cachés (HMM) . . . 122
8.4 Hidden Semi-Markov Models . . . 125

9 Approche hybride par segmentation aveugle **129**
9.1 Principe . . . 129
9.2 Détection de rupture . . . 130
9.3 Méthodes classiques . . . 131
9.4 Mesures probabilistes dans les espaces RKHS . . . 134
9.5 SVM à une classe . . . 135
9.6 Recherche de maxima pour la détection de rupture . . . 140

IV Évaluation et analyse **143**

10 Évaluations **145**
10.1 Corpora audio . . . 146
10.2 Protocole d'évaluation . . . 149
10.3 Expérience 1 : comparaison des taxonomies . . . 151
10.4 Expérience 2 : post-traitements . . . 154
10.5 Résultats à ESTER 2 et sur le corpus de Scheirer . . . 157
10.6 Expérience 4 : Détection de voix chantée . . . 160

11 Implémentation **163**

12 Conclusion **165**

Annexes **170**

A Estimation du rayon R **171**
A.1 Minimisation de Vapnik 171
A.2 Approximation moyenne 171
A.3 Approximation de Joachims 172

B Bases de données pour l'évaluation **173**
B.1 Spambase & Ionosphere 173
B.2 Lymphoma 173
B.3 Parole / musique 174

C Descripteurs audio pour la classification **175**
C.1 Descripteurs spectraux 175
C.2 Descripteurs temporels 178
C.3 Descripteurs cepstraux 180
C.4 Descripteurs perceptifs 181

Publications personnelles **183**

Bibliographie **198**

Notations

Symboles, fonctions et opérations mathématiques

x	Scalaire
$\boldsymbol{x}$	Vecteur
$\boldsymbol{A}$	Matrice
$\boldsymbol{A} = [a_{ij}]_{ij}$	Composantes de la matrice $\boldsymbol{A}$
$\boldsymbol{A} \bullet \boldsymbol{B}$	Produit de Hadamard (terme à terme) sur les matrices $\boldsymbol{A}$ et $\boldsymbol{B}$
$\Sigma(\boldsymbol{A}) = \sum a_{ij}$	Somme des termes de la matrice $\boldsymbol{A}$
$\mathrm{Card}(\boldsymbol{E})$	Cardinal de l'ensemble $\boldsymbol{E}$
c	Indice de classe
i	Indice d'exemple
d	Indice de dimension (ou de composante)
$\mathcal{S}$	Ensemble des exemples
$\mathcal{S}_c$	Ensemble des exemples de la classe c
$x_{i,d}$	Composante d de l'exemple i
$\boldsymbol{\mu}_c$	Centre des exemples de la classe c
$\boldsymbol{\Sigma}_c$	Matrice de covariance des exemples de la classe c
$p_X(\boldsymbol{x})$ ou $p(\boldsymbol{x})$	Densité de probabilité de la variable aléatoire X pour une réalisation $\boldsymbol{x}$.
$p_c(\boldsymbol{x})$	Probabilité a posteriori de la classe c par rapport à l'observation $\boldsymbol{x}$.
$\boldsymbol{\alpha} = [\alpha_i]_i$	Multiplicateurs de Lagrange
$k(\boldsymbol{x}, \boldsymbol{y})$	Fonction noyau sur les exemples $\boldsymbol{x}$ et $\boldsymbol{y}$
$\Phi(\boldsymbol{x})$	Fonction de transformation associée au noyau k.
$k_{\boldsymbol{w}}$	Noyau pondéré par les facteurs $\boldsymbol{w} = [w_d]_d$.
$\boldsymbol{K}$	Matrice de Gram d'un noyau k sur l'ensemble $\mathcal{S}$
$f(\boldsymbol{x})$	Fonction de décision du classifieur appliquée sur l'exemple $\boldsymbol{x}$.
$f^*(\boldsymbol{x})$	Fonction de décision probabilisée.
$\partial_d x$, $\partial_{w_d} x$, $\dfrac{\partial x}{\partial w_d}$	Dérivée partielle de x par rapport à la composante d du vecteur $\boldsymbol{w}$.
$\partial_d \boldsymbol{K}$, $\partial_{w_d} \boldsymbol{K}$, $[\partial_d k_{ij}]_{ij}$	Matrice des dérivées partielles des composantes de $\boldsymbol{K}$ par rapport à la composante d du vecteur $\boldsymbol{w}$
$\langle \cdot, \cdot \rangle_F$	Produit de Frobenius
$\lVert \cdot \rVert_F$	Norme de Frobenius

Acronymes

GMM	*Gaussian Mixture Model*, Modèle de Mélange de Gaussiennes
HMM	*Hidden Markov Model*, Modèle de Markov Caché
HSMM	*Hidden Semi-Markov Model*, Modèle Semi-Markovien Caché
kNN	*k Nearest Neighbors*, k Plus Proches Voisins
ANN	*Artificial Neural Networks*, Réseaux de Neurones Artificiels
SVM	*Support Vector Machine*, Machine à Vecteurs de Support
RKHS	*Reproducting Kernel Hilbert Space*, Espace de Hilbert à Noyaux Reproduisants
SVM1C	Machine à Vecteurs de Support à une Classe
RBF	*Radial Basis Function*, Fonction à Base Radiale
OVA	*One-vs-All*, « un contre tous »
OVO	*One-vs-One*, « un contre un »
ECOC	*Error Correcting Output Codes*, Codes Correcteurs d'Erreurs
DAGSVM	*Direct Acyclic Graph SVM*, SVM multi-classes par Graphe Acyclique Direct
DSVM	*Dendogram SVM*, Dendogramme de SVM
LOO	Erreur *Leave One Out*
KTA	*Kernel Target Alignement*, ou Alignement du noyau
KCS	*Kernel Class Separability*, ou Séparabilité des Classes kernelisée
R2W2	Borne Rayon-Marge
IRMFSP	*Inertia Ratio Maximization using Feature Space Projection*
FSV	*Feature Selection concaVe*
AROM	*Approximation of the zeRO-norm Minimization*
SAS	*Scaled Alignement Selection*, Sélection pondérée basée sur le critère d'Alignement
SFS	*Scaled Frobenius Selection*, Sélection pondérée basée sur le critère de Frobenius
FAS	*Forward Alignement Selection*, Sélection Forward basée sur le critère d'Alignement
SCSS	*Scaled Class Separability Selection*, Sélection Pondérée sur le critère de Séparabilité des Classes
KFDS	*Kernel Fisher Discriminant Analysis*, Sélection sur le Discriminant de Fisher Kernelisé
GLR	*Generalized Likelihood Ratio*, Rapport de Vraisemblance Généralisé
BIC	*Bayesian Information Criterion*, Critère d'Information Bayésienne
LLR	*Log-Likelihood Ratio*, Rapport de Vraisemblance Logarithmique
KCD	*Kernel Change Detection*, Détection de Changement Kernelisée
ESTER	Évaluation des Systèmes de Transcription enrichie d'Émissions Radiophoniques
SES	Segmentation en Événements Sonores

Chapitre 1

Introduction

Sommaire

1.1 Vers une radio numérique 9
1.2 Applications de l'indexation audio pour la radio 10
1.3 « Qu'est-ce que la musique ? » 11
1.4 Classification par Machines à Vecteurs de Support 12
1.5 Problématiques 13
1.6 Résumé des contributions 13
1.7 Structure du document 14

1.1 Vers une radio numérique

Ce document décrit le travail de recherche exécuté durant mon doctorat en convention CIFRE dans l'entreprise RTL, en cotutelle académique avec le département TSI [1] du laboratoire de l'école TELECOM ParisTech. Ce doctorat est né de la nécessité pour RTL de moderniser ses moyens techniques pour demeurer l'un des principaux acteurs du paysage radiophonique français dans le cadre du projet national de numérisation de la radio. Aujourd'hui l'un des derniers médias encore analogiques, la radio prépare actuellement sa transition vers le numérique, dans le sillage de la Télévision Numérique Terrestre.

Pourtant le contexte est très différent. De par sa simplicité technologique et parce qu'elle peut être une occupation auxiliaire, la radio est le compagnon de notre quotidien et trouve sa place dans une multiplicité d'endroits tels que la cuisine, la salle de bain, le salon, dans un baladeur, ou surtout dans la voiture. Ainsi le pari de la radio numérique implique le renouvellement de 160 millions de postes de radio en France, et, contrairement à l'image hertzienne, la qualité de son est suffisamment satisfaisante pour que de nombreux utilisateurs demeurent sceptiques quant à l'intérêt de renouveler leurs postes pour une offre dont l'avantage n'est pas évident.

C'est ainsi que, sous l'impulsion de plusieurs acteurs, parmi lesquels RTL joue un rôle essentiel, la révolution numérique s'accompagne d'une valeur ajoutée. Le protocole de diffusion T-DMB (*Terrestrial Digital Multimedia Broadcasting*, soit Diffusion Multimédia Numérique Terrestre) permet d'adjoindre au flux audio un flux de services multimédias accessibles à partir d'un écran interactif. Afin de ne pas se dénaturer, la radio se doit de demeurer un média n'accaparant pas l'attention de son auditeur ; aussi le service ajouté n'est pas un flux vidéo qui viendrait en outre concurrencer les acteurs très compétitifs du paysage audiovisuel, mais une offre d'informations auxiliaires qui viennent agrémenter l'expérience radiophonique sans jamais s'y substituer.

Ainsi on pourra par exemple y trouver, dans le cas d'une émission musicale, le titre et l'artiste de la chanson diffusée, voire un lien vers un site de vente en ligne ; ou dans le cas d'une interview, une courte présentation textuelle ainsi qu'une photographie de l'invité. De nombreux services non synchronisés peuvent également venir compléter le flux audio, comme des prévisions météorologiques ou la grille des programmes de la station. La figure 1.1 montre un exemple d'affichage

1. Traitement du Signal et de l'Image

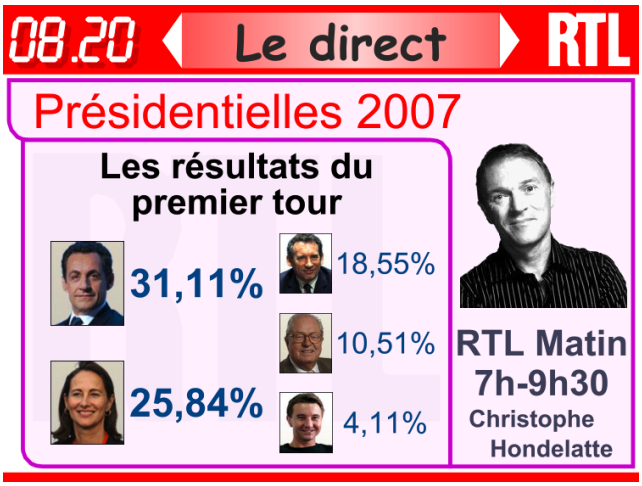


FIGURE 1.1 – Exemple d'affichage interactif accompagnant la radio numérique.

complémentaire pour une soirée électorale, qui permet ainsi à l'auditeur de consulter les résultats à tout moment.

Le projet idéal pour une radio comme RTL consisterait à pouvoir produire ce contenu automatiquement en temps réel ou en ligne [2], à partir du flux audio, ou au moins à réunir le plus possible d'informations pertinentes pour la personne en charge de ce travail. Pourtant, actuellement, la plupart des grandes radio n'ont pas de contrôle en aval sur ce qu'elles émettent. Les logiciels de diffusion exploités sont des applications propriétaires volumineuses et se contentent de réunir les informations sonores voulues, sans fournir d'informations sur ce qu'elles diffusent, qui soient exploitables par un ordinateur. De plus, dans de nombreux cas, comme par exemple une interview impliquant une personnalité et plusieurs journalistes dans un même studio, l'information qui nous intéresserait, à savoir l'identité des locuteurs et la localisation des tours de parole, est totalement inconnue du système de diffusion.

C'est pourquoi RTL, entamant sa mutation numérique, a choisi de se doter des meilleurs atouts en faisant appel aux technologies d'indexation audio, qui substituent à l'indexation manuelle classique l'extraction automatique d'informations (on parle généralement de *méta-données*) à partir du signal audio. Celles-ci ouvrent une autre perspective prometteuse dans la mise en place d'un système d'indexation automatique des archives de la station. En effet, la station conserve depuis 1997 la totalité du flux d'antenne, mais l'annotation manuelle d'un tel volume de données dépasse largement les possibilités d'une entreprise dont le cœur de métier reste avant tout la production d'informations et non l'archivage.

1.2 Applications de l'indexation audio pour la radio

On peut ainsi lister nombre d'applications qui profiteraient directement à un média radio comme RTL :

1. **Reconnaissance des titres musicaux** : l'identification des titres musicaux est un atout essentiel pour une radio puisqu'elle permet de maintenir l'auditeur informé de ce qu'il écoute en lui fournissant les informations de titre, artiste et album, en plus de fournir la pige [3] nécessaire pour les organismes de contrôle de droits d'auteur (SACEM ...). On fait généralement appel pour cela à des techniques d'identification audio qui se concentrent sur la construction d'une empreinte compacte pour chaque titre musical et sa recherche parmi une très vaste collection d'empreintes indexées.

2. Nous faisons la distinction entre le temps réel, qui désigne une réponse quasi instantanée par rapport au contenu d'un flux audio, et le traitement en ligne (*online*) qui correspond à un temps de réponse différé mais régulier et borné par une durée raisonnable (quelques secondes...).

3. Vocable de radio désignant la description détaillée de ce qui est émit par la station.

2. **Recherche de la voix chantée dans un titre musical** : il n'est pas rare que le présentateur d'une émission musicale ou de variété parle sur le début d'une chanson, grapillant ainsi quelques précieuses secondes de parole sur une introduction trop longue, ou assurant simplement la transition entre deux titres. Les présentateurs s'interrompent par convention lorsque l'artiste commence à chanter. À cette fin, il est actuellement nécessaire d'annoter manuellement les chansons afin de permettre au présentateur de savoir précisemment jusqu'à quand il peut parler ou à partir de quand, sur la fin d'une chanson. La détection automatique de voix chantée dans une chanson permettrait ainsi d'automatiser ce processus.
3. **Reconnaissance et suivi de locuteurs** : la reconnaissance de locuteurs permet de fournir à l'auditeur des informations (biographie...) sur le journaliste ou la personne interviewée. Le suivi de locuteur permet de plus d'indiquer en temps réel qui a la parole. L'application d'une telle technique sur les archives permettrait en outre une recherche par locuteur, ce qui se révèlerait un outil très utile pour le travail des journalistes ou pour la constitution des meilleurs moments de certaines émissions (par exemple les fameuses « Grosses têtes »).
4. **Transcription de la parole** : les bulletins d'informations sont généralement fournis aux journalistes à l'antenne sous forme écrite. Le texte est par la suite corrigé manuellement afin de prendre en compte les modifications éventuelles apportées par le présentateur en direct. La transcription automatique permettrait de simplifier ce processus et de le généraliser à l'ensemble des programmes d'antenne. La forme textuelle représente un avantage énorme sur l'archive audio puisqu'elle permet l'application des outils de recherche textuelle beaucoup plus puissants et moins gourmands que l'indexation audio.
5. **Détection de rires, d'applaudissements ou de foule** : les rires et les applaudissements du public ou des invités peuvent être interprétés comme des indices de moments forts de certaines émissions. De même, lors d'une retransmission sportive, la clameur de la foule est généralement révélatrice d'un événement clé du match. La détection de ce type d'événement peut ainsi aider à la constitution du résumé ou des meilleurs moments d'une émission.
6. **Recherche de sons-clés (jingles...)** : la recherche de sons-clés caractéristiques et récurrents, comme les jingles, les habillages sonores ou les publicités, permet de structurer les archives et ainsi de faciliter son exploration.

On remarque que les quatre premières applications énumérées se basent sur une hypothèse forte sur le contenu acoustique analysé. Ainsi les deux premières concernent des plages de musique tandis que les deux suivantes ne s'appliquent que sur des extraits de voix parlée. Le premier outil indispensable à RTL pour l'implémentation de ces traitements plus complexes consiste donc en l'annotation automatique des plages de parole et de musique dans un flux audio. De plus, une fois la musique détectée, la détection de voix chantée constitue une application dont le principe est très similaire. En effet, chacune de ces tâches implique la reconnaissance d'une catégorie acoustique identifiable sans ambiguïté par un être humain.

Les tâches de recherche de sons-clés et de reconnaissance de titre se basent par contre sur un formalisme différent et dépassent le cadre de cette thèse. De même la reconnaissance de locuteur et la transcription automatique sont des sujets de recherche à part entière qui impliquent, l'un la connaissance d'une vaste collection de locuteurs dont la multiplicité a un impact radical sur l'approche suivie, l'autre des notions sur le langage et la sémantique qui dépassent largement le cadre purement audio de cette étude.

Le problème de la classification audio, et particulièrement la classification parole/musique et la détection de chant, constituent donc les sujets couverts par cette thèse, et présentés dans ce document.

1.3 « Qu'est-ce que la musique ? »

Alors que j'expliquais, durant une école d'été, mes travaux de jeune doctorant sur la classification parole/musique à un chercheur expérimenté, celui-ci me posa avec amusement la question suivante, qui me laissa sans réponse :

« Mais qu'est-ce que la musique ? »

En effet définir la musique de manière formelle est problématique. Même si l'on dépasse les querelles sur la musicalité de tel ou tel genre (un éternel débat entre générations), on conviendra que celle-ci est généralement le produit d'un consensus culturel basé sur de nombreuses notions cognitives complexes difficilement formalisables. On trouve les définitions suivantes, respectivement dans le Dictionnaire de l'Académie Française et le Robert :

> Art de composer une mélodie selon une harmonie et un rythme ; théorie, science des sons considérés sous le rapport de la mélodie, de l'harmonie, du rythme.
>
> Art de combiner des sons d'après des règles (variables selon les lieux et les époques), d'organiser une durée avec des éléments sonores ; productions de cet art (sons ou œuvres).

On remarque que dans les deux cas, la musique est caractérisée par son mode de production, à savoir l'acte de composition, qui consiste en un agencement de sons dans le temps. On trouvera pour la parole des définitions qui renvoient au mode de production, ou qui sont même cycliques (« Élément(s) de langage parlé » dans le Robert), liant inévitablement le phénomène sonore à sa source.

Travaillant sur la reconnaissance de ces sources dans un signal audio, je revenais parfois sur cette question, me disant que ne pas y apporter une piste de réponse constituait une lacune. Pourtant j'ai trouvé dans mon incapacité à apporter une réponse formelle la justification de la démarche scientifique employée. Si toute personne est en effet capable d'identifier un son de production musicale ou vocale, c'est bien parce que cette action, comme la plupart des processus cognitifs, échappe à la nécessité d'une définition formelle et repose en réalité sur l'apprentissage empirique de très nombreux exemples associés à une ou plusieurs catégories, qui nous permet de reconnaître celles-ci en présence d'exemples inconnus. Le cerveau est fondamentalement une machine associative, avant d'être une machine logique.

L'apprentissage statistique, qui constitue l'outil prédominant dans le domaine de l'indexation audio, repose précisément sur ce principe, et revient à poser la question plus empirique : « Est-ce de la musique ? »
Cette dernière constitue un problème fondamentalement différent, reposant sur la classification. On peut retrouver dans les deux questions posées la dualité classique entre approches « top-down » et « bottom-up » [4], la première partant d'une définition englobant tous les exemples d'une catégorie et permettant de les reconnaître, la seconde construisant la définition de la catégorie à partir d'une collection d'exemples représentatifs.

Ce que nous appelons « classification audio » consiste en l'application de ce principe de catégorisation d'exemples parmi un ensemble prédéfini de classes, sur un signal audio.

1.4 Classification par Machines à Vecteurs de Support

Le domaine de l'apprentissage statistique est aujourd'hui riche et l'expérimentateur dispose de nombreuses méthodes de classification, généralement formalisées par les statisticiens. Parmi celles-ci, les Machines à Vecteurs de Support (SVM, *Support Vector Machines*) sont une approche récente (datant de la décennie passée) qui modernise le cadre classique de la séparation linéaire en introduisant une non-linéarité dans la surface de décision. La régularité de cette surface de décision est contrôlée par un principe de Minimisation du Risque Structurel qui garantit les bonnes propriétés de généralisation du classifieur. Les excellentes propriétés des SVM nous ont conduit à restreindre notre étude de la classification audio à cette méthode. Une étude préliminaire de l'état de l'art dans le domaine de la classification audio, au chapitre 2, suivie d'une présentation détaillée de la théorie des SVM, dans la partie I, nous permettrons d'étayer notre propos et de justifier ce choix.

L'introduction des SVM dans le domaine de la classification audio est relativement récent (le premier article que nous avons trouvé ne remonte qu'à 2001), et on compte aujourd'hui encore relativement peu d'articles tirant parti de cette méthode pour la tâche en question, par rapport aux autres méthodes plus connues de la communauté. De plus, les SVM restent souvent exploités comme une « boîte noire » de classification que l'expérimentateur n'exploite pas toujours de manière

4. littéralement « du sommet vers le bas » et « du bas vers le haut ».

optimale, en partie en raison des nombreuses *toolbox* publiques lui apportant une interface simple pour employer cette technique sans avoir a en maîtriser les détails théoriques.

Nous verrons que le point central dans la mise en place d'une machine à vecteurs de support est le choix d'une fonction noyau, qui réalise implicitement une transformation sur les données, qui les place dans un espace de dimension supérieure, où la séparation linéaire classique est appliquée. Afin de maximiser la séparabilité des données dans l'espace transformé, la transformation doit donc être directement déterminée par la structure des données dans l'espace d'origine. Un soin particulier doit ainsi être porté à la fois sur le choix de cette transformation et sur la caractérisation des données audio, qui détermine leur répartition dans l'espace d'origine.

Les contraintes propres aux machines à vecteurs de support déterminent donc un certain nombre de problématiques qui constitueront les axes de recherches de cette étude.

1.5 Problématiques

- **Comment employer efficacement les Machines à Vecteurs de Support ?**
  Bien qu'elles réduisent considérablement le nombre de paramètres de réglages par rapport à certaines méthodes classiques comme les réseaux de neurones, les SVM restent fortement dépendantes de l'ajustement de certaines variables, comme le facteur C, fixant le compromis entre régularité et minimisation de l'erreur, ou les paramètres propres au noyau. Le réglage de ces derniers est généralement mené par une procédure de validation croisée dont la complexité devient trop lourde lorsque le nombre de paramètres augmente. Nous étudierons donc les critères permettant d'évaluer de manière fiable et économique les performances d'une SVM par rapport à ses paramètres.
- **Comment appliquer les SVM sur un problème multi-classes ?**
  Cette technique, dérivée de la séparation linéaire, est fondamentalement discriminative. Or nous verrons que le problème de la classification parole/musique, s'il est correctement posé, implique en réalité plus de deux classes. Il nous faut donc déployer une stratégie efficace permettant d'adapter leur usage à ce genre de configuration.
- **Comment caractériser efficacement le signal audio ?**
  Comme la plupart des approches en apprentissage statistique, les SVM se basent sur une modélisation vectorielle des données traitées. Il nous faut donc en premier lieu déterminer les descripteurs qui permettront de décrire au mieux le signal audio, dans l'optique d'une séparabilité maximale entre les classes traitées. En outre, ces données étant groupées et mises en concurrence dans le processus de classification, leur rôle individuel et leurs interactions mutuelles sont difficilement prédictibles pour l'expérimentateur. Il nous faudra donc appliquer des techniques permettant de déterminer automatiquement le sous-ensemble des descripteurs disponibles qui optimise les performances d'un classifieur donné. La prise en compte du noyau, élément central des SVM, sera dans cette opération l'un des enjeux majeurs de cette étude.
- **Comment introduire la donnée temporelle ?**
  Comme nous le verrons par la suite, le calcul des descripteurs résulte d'un découpage du signal audio en trames successives de courte durée. L'approche par « sac de trames » fragmente ainsi le problème en supprimant tout lien ou corrélation entre trames voisines. Nous examinerons donc les techniques de post-traitement qui réintroduisent cette relation temporelle pour augmenter les performances de classification.

Le système développé doit en outre répondre aux contraintes pratiques de l'entreprise RTL. Ainsi le traitement doit être le plus rapide possible, ce qui réduit le champ des possibilités par rapport à une approche purement académique de recherche, et doit pouvoir fonctionner en temps réel, on du moins « en ligne », c'est-à-dire sur un flux audio, avec un retard éventuel mais qui reste contrôlé.

1.6 Résumé des contributions

Afin de répondre aux problématiques exposées dans la section précédente, nous avons apporté durant cette thèse les contributions suivantes.

Nous avons en premier lieu exploité les critères d'alignement du noyau et de séparabilité de classes, que nous présentons dans la section 4.4, pour l'évaluation du noyau dans le contexte de la classification audio. Nous montrerons dans la section 4.6 la pertinence de ces critères en terme de performances et de temps de calcul, par rapport aux autres méthodes plus connues de la communauté. De plus, après avoir montré l'importance du facteur d'erreur C, nous proposerons dans la section 4.5 une procédure d'ajustement de la matrice de Gram pour la prise en compte du facteur C, dans le calcul des mesures d'alignement du noyau et de séparabilité de classes. Nous montrerons dans la section expérimentale 4.6 que cet ajustement améliore sensiblement les résultats de la sélection, pour un coût additionnel minime. En outre, l'inclusion du noyau dans les algorithmes de sélection de descripteurs se révèle équivalente à la sélection de noyau, où la contribution de chaque descripteur constitue un paramètre de ce dernier. Ceci nous conduira donc à proposer, dans la section 7.5, cinq nouvelles méthodes de sélection de descripteurs basées sur les critères sus-mentionnés d'alignement et de séparabilité de classes. Une étude comparative, détaillée dans la section 7.7, viendra confirmer l'efficacité de ces méthodes dans diverses configurations, synthétiques ou réelles.

Après un rapide parcours des paradigmes multi-classes pour les SVM, nous adapterons dans la section 5.1.5.4 le principe des arbres de classification hiérarchique afin d'estimer les probabilités a posteriori par classes. Ces dernières nous permettrons de déployer les techniques de post-traitement. L'examen de diverses configurations hiérarchiques, incluant des taxonomies hybrides basées sur le paradigme *one-vs-one*, fera l'objet d'une étude expérimentale détaillée dans la section 10.3.

Nous proposerons également, dans la section 8.3.2.2, un nouveau paradigme de post-traitement basé sur l'exploitation des probabilités a posteriori estimées comme observation d'un modèle de Markov caché (HMM) dont on estime le chemin optimal. Nous introduirons en outre le modèle moins connu des HSMM (semi-markovien), pour lequel nous proposerons une méthode simple pour la modélisation probabiliste de la durée passée dans un état donné, qui permet de relacher la contrainte de distribution géométrique induite sur cette dernière par le modèle HMM. L'étude sur les post-traitements nous permettra d'introduire dans le chapitre 9 une approche hybride combinant l'approche SVM par trames à un panel de méthodes de segmentation aveugle dont le principe est de détecter automatiquement les frontières entre segments au contenu acoustique homogène. Nous détaillerons ainsi cinq méthodes de segmentation aveugle, dont les plus récentes tirent parti des apports de la théorie des noyaux. Une étude comparative sur les différentes métriques, en section 10.4, montrera l'avantage de l'approche hybride proposée sur les méthodes plus traditionnelles.

Nous aborderons enfin dans la section 10.5 notre participation durant cette thèse à la campagne d'évaluation nationale ESTER 2 qui apporte une comparaison objective aux contributions de l'état de l'art en France sur la classification parole/musique. Le problème de la détection du chant étant moins couvert par la littérature, il est difficile de trouver des corpus publics suffisamment conséquents pour l'évaluation des résultats. Nous avons donc constitué un corpus réunissant des titres libres de droit pour cette tâche, que nous décrirons dans la section 10.1.4, et sur laquelle une expérience comparative est mené pour évaluer notre approche.

On trouvera à la fin de ce document, en page 183, la liste de nos publications.

1.7 Structure du document

Ce document est structuré en trois parties théoriques, suivies d'un quatrième partie expérimentale. La figure 1.2 synthétise la structure en question.

Dans la partie I, nous commencerons par traiter les questions relatives à l'application des Machines à Vecteurs de Support pour la classification. Après avoir présenté en détail la **théorie** et ses implications en terme de **contrôle du Risque Structurel** dans le chapitre 3, nous expliquerons en quoi les SVM constituent une synthèse de nombreuses autres méthodes d'apprentissage, et nous montrerons enfin les avantages qu'implique le principe de maximisation de la marge. Par la suite nous aborderons, dans le chapitre 4, la question de la sélection ou **paramétrisation du noyau**, élément central des SVM, en présentant les différents critères existants dans la littérature, pour mettre l'accent sur le critère d'Alignement, encore très peu utilisé en indexation audio. Nous terminerons cette première partie en comparant dans le chapitre 5 les différentes approches dé-

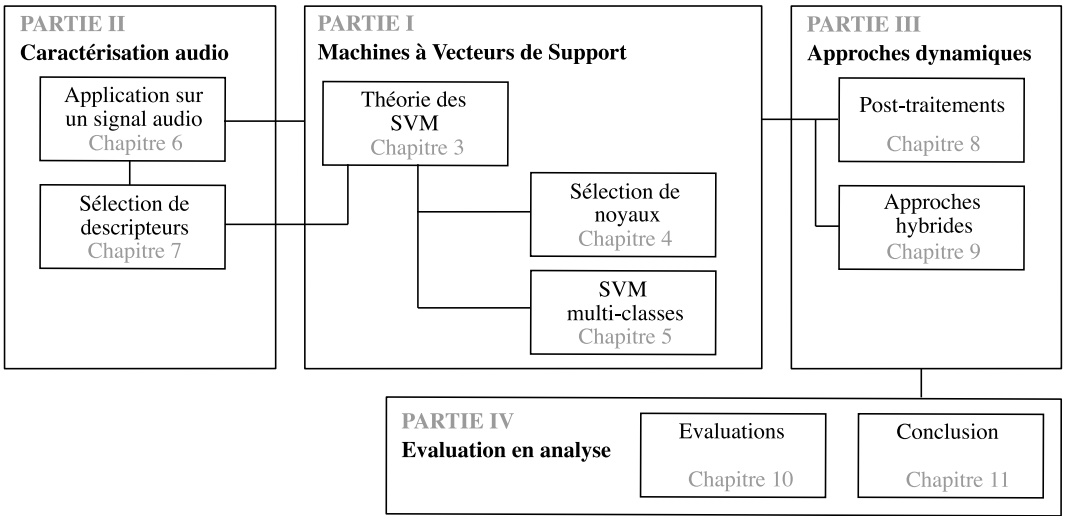


FIGURE 1.2 – Résumé de la structure du document.

diées à l'application des SVM sur un **problème multi-classes**, c'est-à-dire impliquant plus de deux classes. Ceci nous amènera à proposer une stratégie de classification hiérarchique permettant d'estimer les probabilités a posteriori, qui nous seront nécessaires par la suite.

La caractérisation numérique du signal audio sera traitée dans la partie II. Nous commencerons par détailler dans le chapitre 6 le processus de **découpage audio** en trame, puis nous présenterons l'**ensemble de descripteurs** retenus pour leurs propriétés discriminatives sur le problème posé. Cette collection sera complétée par un descripteur proposé dans ce document pour le problème particulier de la classe mixte de parole sur fond musical. Le chapitre 7 traitera des techniques de **sélection automatique de descripteurs**. Après une courte étude sur la notion de pertinence et une proposition de taxonomie des algorithmes, nous présenterons plusieurs approches de la littérature en mettant l'accent sur celles liées aux Machines à Vecteurs de Support. Nous terminerons ce chapitre par la proposition de plusieurs algorithmes exploitant entre autres le critère d'Alignement introduit précédemment.

La dernière partie théorique III examinera les moyens envisagés pour corriger autant que possible les résultats en replaçant les trames dans leur contexte temporel. Ainsi, le chapitre 8 présentera plusieurs **techniques de post-traitement** sur les probabilités a posteriori, allant du simple filtrage à l'application de modèles de Markov, dont nous examinerons une amélioration possible. Cette étude sera complétée dans le chapitre 9 par la proposition d'une **approche hybride** combinant le processus de classification au résultat d'une **segmentation aveugle**, afin de fournir un découpage en segments acoustiquement homogènes. Nous présenterons à cet effet plusieurs algorithmes de segmentation aveugle dont certains sont fortement liés à la théorie des noyaux.

Nous conclurons cette étude dans la partie IV, chapitre 10, par une **évaluation** des différents aspects du système proposé sur plusieurs corpora publics, dont l'un fut créé durant cette thèse, ainsi que dans le cadre de notre participation à la **campagne d'évaluation** nationale ESTER 2. Nous présenterons brièvement ensuite, dans le chapitre 11, l'**implémentation** en C++ du système de classification, que nous avons livré à l'entreprise RTL à la fin de notre thèse. Enfin, le chapitre 12 de **conclusion** apportera quelques commentaires sur ce travail ainsi qu'un aperçu des perspectives ouvertes par ce dernier.

Chapitre 2

État de l'art

Sommaire

2.1 Applications de la classification audio 16
2.2 Taxonomie audio 17
2.3 Techniques de classification 18
2.3.1 Méthodes génératives 19
2.3.2 Méthodes discriminatives 21
2.3.3 Approches hybrides 22
2.3.4 Discussion 23
2.3.5 Un mot sur la détection de chant 23
2.4 Caractérisation audio 23
2.4.1 Classification parole/musique 24
2.4.2 Détection de chant 26
2.4.3 Discussion 27

2.1 Applications de la classification audio

Le principe de la classification a de nombreuses applications en indexation audio. On trouve ainsi le découpage automatique de données audio pour le parcours structuré d'archives vidéo dans [261], [41] et [204]. La distinction entre parole et musique permet également, dans le domaine du codage audio, d'adopter des algorithmes de codage plus adaptés au contenu et ainsi d'accroître le taux de compression [164] ou d'opérer une allocation intelligente de la bande passante en temps réel [41].

La classification s'applique également sur d'autres classes, en se basant généralement sur la même architecture. On peut ainsi exploiter celle-ci dans le domaine de la Recherche d'Information Musicale (MIR, *Music Information Retrieval*) pour la reconnaissance de genres musicaux [229][68][151][180] ou d'instruments de musique [132][73], ou encore pour l'identification de l'artiste ou du chanteur dans un titre musical [25][124][228] (on distingue les deux dans le cas d'artistes invités pour des duos). Mandel et al. [146] étendent également le domaine d'application de la classification audio à la recherche de titres musicaux par similarité, généralement basée sur d'autres techniques comme le *fingerprint audio* (empreintes audio).

Le signal de parole est également matière à certaines classifications plus approfondies, par exemple pour découper ce dernier en tours de parole successifs [125][155] ou pour la détermination du sexe [102] ou de l'âge [32] du locuteur, qui permettrait d'apporter un complément d'information pour la tâche de reconnaissance de locuteurs. On peut également considérer la reconnaissance de parole comme un exemple de classification audio, bien que celle-ci en dépasse le cadre puisqu'elle fait intervenir des notions linguistiques et sémantiques.

Étendant considérablement le champ acoustique considéré, le domaine de le reconnaissance de scènes auditives [182][194][69] (CASR, *Computational Auditory Scene Recognition*) a pour principe l'identification de l'environnement capté par un enregistrement audio, par exemple la rue, la

Référence	Sil	Par	Tel	Mus	Par+Mus	Ch	Br	Par+Br	Aut
[207],...		X		X					
[85]		X		X					X
[41], [168]		X		X			X		
[52], [202]		X		X		X	X		
[100]		X	X	X					X
[159]		X		X					X
[86]		X		X	X				X
[169]	X	X		X	X			X	
[194]		X		X			X	X	

TABLE 2.1 – Taxonomies de classes exploitées dans la littérature pour la classification parole/musique. Par : parole – Tel : parole au téléphone – Mus : musique – Ch : chant – Br : bruit – Aut : autre – A+B : les deux classes superposées.

nature, un café, l'intérieur d'une voiture, une bibliothèque ou encore une église. Le problème posé est beaucoup plus complexe et les classes moins clairement définies, mais sa résolution, au moins partielle, aurait pléthore d'applications concrètes.

La détection du chant est le plus souvent destinée à mettre en évidence les zones à analyser pour la reconnaissance de chanteurs ou d'artistes, mentionnée plus haut. Elle sert également à d'autres applications comme la reconnaissance de la langue chantée [141], la transcription d'une mélodie [193] ou sa requête dans une base de données [131], ou encore la transcription textuelle ou la synchronisation de paroles (par rapport au texte) [135][141]. Le lecteur intéressé trouvera dans l'étude de Rocamora [199] une liste assez complète des applications possibles de la détection de chant.

2.2 Taxonomie audio

Les exemples précédents montrent un large éventail de possibilités dans le choix des classes employées. Une attention particulière doit cependant être portée à la définition de classes pour que le problème soit bien posé. Burred et Lerch [41] distinguent deux défauts courants dans ce qu'ils nomment les taxonomies audio : la *non-complétude*, qui désigne l'absence flagrante d'une classe implicite importante, par exemple l'absence d'une classe de musique classique dans un problème de reconnaissance de genres musicaux, et l'*inconsistance*, qui désigne une définition trop ambigüe des classes ou leur mauvaise partition, par exemple en présence d'une classe de musique classique et d'une autre d'opéra. Ce second défaut, plus courant dans la littérature, souligne en général l'absence de consensus sur les classes considérées dans certains domaines. Ainsi la reconnaissance de genre se heurte généralement à l'impossibilité de trouver un consensus sur une taxonomie cohérente entre les différents sous-genres musicaux [229], de même pour la reconnaisance de scènes auditives, mentionnée précédemment.

Un problème méthodologique consiste par ailleurs à distinguer des classes définies non pas par un phénomène acoustique identifiable, mais par une notion sémantique qui n'a pas de sens d'un point de vue auditif. Par exemple la distinction de la publicité [170] ou des jingles (par rapport à la musique) [179] dépasse le cadre de la classification audio puisque ces deux classes ne sont pas définies par leur contenu acoustique mais par le sens que leur accorde l'auditeur. De la même manière, la mise en concurrence de classes définies sur des niveaux incompatibles ou se chevauchant (par exemple la publicité et la violence physique [170] peuvent décrire un même signal audio) constitue généralement un obstacle pour le système de classification.

De manière générale on considérera qu'une taxonomie audio est bien définie si elle forme une partition de classes disjointes sur l'ensemble des phénomènes audio couverts par la classification.

Sur le problème le plus basique de distinction entre parole et musique, on constate déjà de nombreux points de divergences entre les taxonomies employées dans la littérature, que nous comparons dans le tableau 2.1, indiquant les classes exploitées dans quelques article.

Les points de suspension sur la première ligne indiquent que l'approche à deux classes (parole

et musique pures) est de loin la plus généralement suivie dans la littérature (on s'en convaincra par le nombre de références : [205],[42],[68],[91],[247],[121],...). Il est cependant naturel que d'autres classes interviennent dans le processus, ces deux seules ne suffisant pas à décrire un signal audio de manière exhaustive. Une solution simple consiste parfois à introduire une classe complémentaire « Autres » [85][100] qui permet d'y ranger tout ce qui ne correspond pas aux autres classes. Mais une telle classe est généralement très mal définie puisqu'elle fait cohabiter des phénomènes acoustiques très différents qu'un classifieur peinera à caractériser dans leur ensemble. On évitera donc en pratique cette solution trop simple.

Certaines classes sont parfois bien définies mais n'apportent pas grand chose au processus de classification, en raison de leur détection très aisée. Ainsi la voix téléphonique, prise en compte dans [100], est caractérisée par un filtrage passe-bande très net ; de même le silence [169] est très simple à localiser et est généralement détecté dans une phase préliminaire à la classification. De manière générale, on parle de *détection* lorsque la classification n'implique que la reconnaissance d'une classe.

On remarque également la présence des classes mixtes parole+musique et parole+bruit dans un grand nombre de publications [86][169][194]. Les approches classiques (parole et musique pures) sont en effet parcellaires puisqu'elles font l'impasse sur l'éventualité de la présence simultanée de parole et de musique dans un même extrait sonore. Cette situation est pourtant très courante, à la radio par exemple, où les titres des bulletins d'informations sont généralement accompagnés d'un fond musical destiné à agrémenter le discours d'une certaine tension, ou par exemple dans le cas d'émissions de variété où le fond musical apporte au contraire une ambiance à l'antenne.

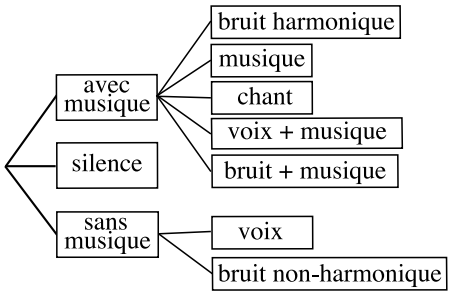
Certains auteurs contournent le problème des classes mixtes en traitant chaque classe par un problème de détection indépendant [150][261]. Ainsi la reconnaissance de parole sur fond musical sera menée implicitement par les détections conjointes de parole et de musique. Si l'approche a le mérite d'être simple et de limiter le nombre de classes, elle est cependant pénalisée par la constitution de classes fortement hétérogènes d'un point de vue acoustique, et donc difficilement caractérisables.

Une autre approche couramment exploitée pour prendre en compte la multiplicité des classes mises en jeu consiste à suivre un arbre hiérarchique de classifications successives permettant d'affiner itérativement la détermination des classes présentes. La figure 2.1 montre plusieurs exemples de taxonomies hiérarchiques employées dans la littérature. La définition de l'arbre est en général empirique et suit une logique intuitive, par exemple détecter dans un premier temps la présence de parole pour affiner par la suite la caractérisation des régions où la parole est absente [5][140][139] (exemples d et e), ou au contraire commencer par détecter la présence de musique [261] (exemple a). On trouve également des graphes de décision plus complexes ne constituant pas des arbres [112]. Certains auteurs, en présence d'un ensemble plus complexe de classes, préfèrent baser la construction de l'arbre sur des critères de séparabilité des classes, comme Essid [73] qui, pour un problème de reconnaissance d'instruments de musique, regroupe itérativement les classes par *clustering* (regroupement) hiérarchique, appliquant ainsi une stratégie *bottom-up*.

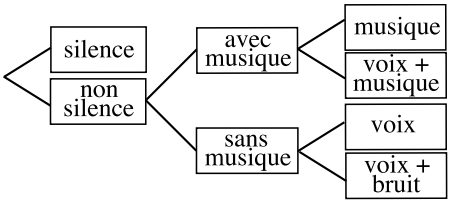
Les problèmes de classification étant posés, nous poursuivons cet état de l'art par un examen des contributions dans ce domaine. Une énorme majorité des publications concentrent leurs efforts sur l'un des deux axes suivants : l'exploitation d'un algorithme de classification original ou efficace, ou bien la proposition de descripteurs destinés à caractériser au mieux les classes pour la tâche en question. Les sections suivantes détaillent les principales propositions pour chacun de ces deux aspects.

2.3 Techniques de classification

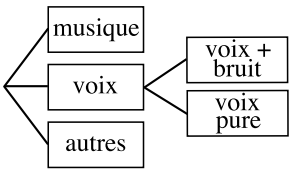
Nous avons évoqué dans la section précédente l'existence de méthodes dites *discriminatives*. Les techniques exploitées en apprentissage statistique se distinguent en effet parmi deux modalités



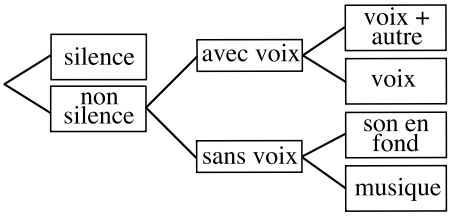
(a) Zhang & Kuo [261]



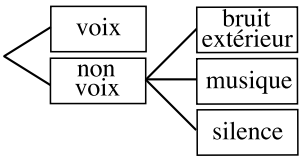
(b) Nitanda et al. [169]



(c) Gauvain et al. [85]



(d) Lu, Li & Zhang [140]



(e) Lu, Jiang & Zhang [139]

FIGURE 2.1 – Quelques exemples de taxonomies hiérarchiques de la littérature sur le problème de la classification parole/musique.

classiques :

- Les **méthodes génératives** ont pour principe de modéliser la distribution des exemples de chaque classe, et ainsi de « comprendre » implicitement la nature des différentes classes dans l'espace de description.
- Les **méthodes discriminatives** portent uniquement l'attention sur la détermination de la frontière séparant les exemples de deux classes.

Les deux approches ont leurs avantages respectifs. Les méthodes génératives apportent une meilleure compréhension de la distribution des classes et permettent de prendre en compte un nombre élevé de classes. En revanche les méthodes discriminatives simplifient généralement le problème en le limitant à la détermination d'une frontière (dont la caractérisation est nécessairement plus compacte que celle d'une distribution), mais se restreignent pour cela à deux classes ; l'application sur plus de deux classes se fera alors par une combinaison de discriminateurs, comme nous l'avons évoqué pour les arbres hiérarchiques de classification. La figure 2.2 illustre le principe des deux méthodes sur un exemple simple à 3 classes.

2.3.1 Méthodes génératives

Les méthodes génératives sont les plus couramment employées dans la littérature, en raison de l'héritage historique des techniques de traitement de la parole. La théorie de la décision de Bayes apporte aux modèles évalués le complément nécessaire pour la classification. Ainsi, si l'on suppose les exemples de chaque classe générés par un modèle aléatoire de densité de probabilité $p(\boldsymbol{x}|\omega_c)$, où ω_c représente la classe d'indice c, la formule de Bayes nous permet de déterminer la probabilité

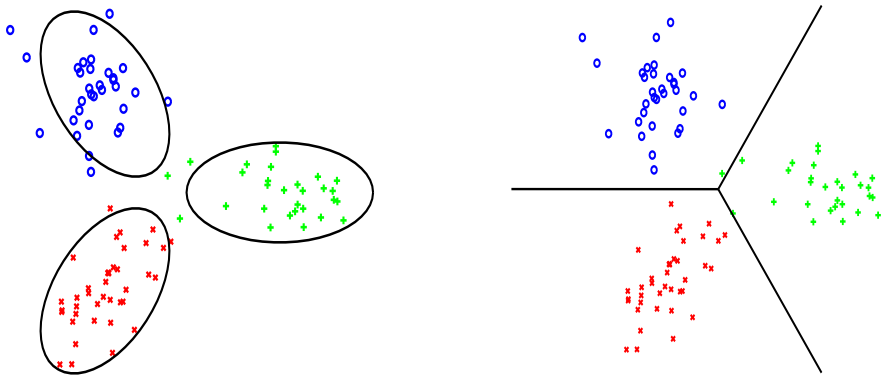


FIGURE 2.2 – Illustration des principes génératifs et discriminatifs (respectivement à gauche et à droite) sur un exemple simple à 3 classes.

a posteriori d'une classe sous l'observation d'un échantillon donné :

$$p(\omega_c|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_c)P(\omega_c)}{P(\boldsymbol{x})}.$$

Le cadre habituel des méthodes génératives consiste à appliquer la stratégie du *Maximum A Posteriori* (MAP), qui associe à l'exemple $\boldsymbol{x}$ la classe $\hat{c}$ maximisant la probabilité a posteriori (la probabilité $P(\boldsymbol{x})$ n'intervient pas dans le choix puisqu'elle est constante au regard de la variable c) :

$$\begin{aligned} \hat{c} &= \arg\max_{1\leq c\leq C} p(\omega_c|\boldsymbol{x}) \\ &= \arg\max_{1\leq c\leq C} p(\boldsymbol{x}|\omega_c)P(\omega_c). \end{aligned}$$

Les probabilités a priori $P(\omega_c)$ sont en général supposées uniformes ou bien estimées à partir de la distribution des exemples du corpus d'apprentissage. Le principe des méthodes génératives consiste ainsi à estimer les densités de probabilités $p(\boldsymbol{x}|\omega_c)$ par des modèles statistiques.

Le **modèle gaussien multi-dimensionnel** caractérise la distribution par sa moyenne $\boldsymbol{\mu}_c$ et sa matrice de covariance $\boldsymbol{\Sigma}_c$, dont l'estimation à partir des exemples du corpus est immédiate. La distribution gaussienne est définie par :

$$\mathcal{N}(\boldsymbol{x}|\omega_c) = \frac{1}{(2\pi)^{d/2}\left|\boldsymbol{\Sigma}_c\right|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_c)^T\boldsymbol{\Sigma}_c(\boldsymbol{x}-\boldsymbol{\mu}_c)\right). \tag{2.1}$$

Celui-ci est par exemple exploité par Scheirer et Slaney [207], ou encore par Saunders [205], sur le problème de classification parole/musique.

Le modèle gaussien est néanmoins généralement trop restrictif et ne permet pas de modéliser la plupart des distributions réelles. On peut montrer, cependant, que toute distribution régulière est asymptotiquement modélisable par une somme de gaussiennes pondérées (par asymptotiquement, on entend : lorsque le nombre de gaussiennes tend vers l'infini), que l'on appelle communément **Modèle de Mélange de Gaussiennes** (GMM, *Gaussian Mixture Model*). On estime donc dans le cadre du modèle GMM la distribution par la somme suivante de M composantes :

$$\hat{p}(\boldsymbol{x}|\omega_c) = \sum_{i=1}^{M} m_i\,\mathcal{N}\left(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right),$$

où chaque composante d'indice i est définie de manière similaire à l'équation 2.1 et caractérisée par la moyenne $\boldsymbol{\mu}_i$, la matrice de covariance $\boldsymbol{\Sigma}_i$ et le coefficient de pondération m_i. Ces paramètres sont estimés au moyen de l'algorithme Espérance-Maximisation [62][263] (EM, *Expectation Maximization*), guidé par la maximisation de la vraisemblance du modèle par rapport aux exemples.

Le nombre de composantes M reste sujet à une détermination manuelle de l'expérimentateur, et peut se révéler crucial pour la pertinence du modèle puisqu'il constitue un compromis entre la précision et la complexité. De plus, un nombre trop élevé de composantes peut impliquer un sur-apprentissage du classifieur et ainsi pénaliser ses capacités de généralisation sur des exemples inconnus. Le modèle GMM est l'un des modèles les plus largement employés dans la littérature [52][180][100][207][42].

Bien que cette technique soit à priori tout à fait indépendante des GMM, les **Modèles de Markov Cachés** (HMM, *Hidden Markov Models*) sont généralement couplés à ces derniers. Les HMM [190], que nous exploiterons et présenterons en détail dans la section 8.3 de la partie III, modélisent l'évolution temporelle d'un système par une séquence d'états tirés parmi un ensemble fini, où à chaque itération une observation est produite, dont la distribution est classiquement décrite par un modèle GMM. La combinaison HMM/GMM est très populaire dans la communauté pour sa simplicité d'implémentation et son interprétation aisée. On trouve ainsi de nombreux exemples de l'exploitation de ce dernier pour la classification audio [125][9][55][259][109].

2.3.2 Méthodes discriminatives

Le principe général des méthodes discriminatives est la détermination d'une frontière de séparation optimale entre deux classes. La décision sur un exemple se fait en évaluant de quel côté de la frontière ce dernier se situe. Si l'éventail des frontières possibles est infini, nous verrons que, comme le modèle GMM, celles-ci sont avant tout contraintes par une condition de régularité qui influence directement la capacité de généralisation du discriminateur.

La méthode discriminative la plus sommaire consiste à appliquer une **heuristique** consistant en une combinaison logique de seuils sur les descripteurs, empiriquement déterminés à partir des données d'apprentissage. Bien que très basique, et souvent implicitement couverte par des méthodes automatiques plus complexes, cette approche demeure relativement populaire dans de nombreux domaines, y compris la classification audio [139][261]. Elle est souvent employée pour montrer la pertinence d'un nouveau descripteur fortement discriminant pour une tâche donnée [176][111], en particulier dans le domaine de la détection de chant [196][143].

Il est possible de rationaliser l'application d'heuristiques par seuillages successifs en suivant un **Arbre de Classification** (ou **CART**, *Classification And Regression Trees*) comme dans [242].

C'est historiquement le modèle le plus simple d'un hyperplan de séparation linéaire qui a ouvert la voie dans ce domaine. Ainsi l'**Analyse Discriminante Linéaire** (LDA, *Linear Discriminant Analysis*), que nous présenterons dans la section 3.2, est l'une des premières méthodes d'apprentissage automatique, qui consiste en la détermination d'un vecteur $\boldsymbol{w}$ normal définissant l'hyperplan de séparation optimale pour la fonction de décision suivante :

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b. \tag{2.2}$$

Malgré son principe discriminatif, on classe généralement la LDA dans les méthodes génératives, car celle-ci implique une modélisation gaussienne des distributions de classes. Néanmoins, parce qu'elle est le dénominateur commun de la plupart des méthodes discriminatives ultérieures, nous préférons l'introduire dans cette section. La LDA reste encore aujourd'hui exploitée dans le domaine de la classification audio [91][71][13], pour sa simplicité, généralement dans des travaux où l'accent est porté avant tout sur la caractérisation du signal audio, et non sur la phase de classification.

Afin de relâcher l'hypothèse de séparabilité linéaire, l'**Analyse Discriminante Quadratique** (QDA, *Quadratic Discriminant Analysis*), employée dans [68] et [151], permet d'étendre le champ des surfaces de séparation à l'ensemble des sections coniques, par la recherche d'un fonction de décision de la forme $f(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T \boldsymbol{x} + c$, au prix, bien sûr, d'un apprentissage plus complexe.

L'algorithme du **Plus Proche Voisin** (NN, *Nearest Neighbor*) est un exemple d'approche discriminante d'une simplicité extrême puisqu'il consiste à assigner à un exemple de test la classe associée à l'exemple le plus proche parmi le corpus d'apprentissage. La facilité d'implémentation et l'efficacité de cet algorithme lui a offert une certaine popularité dans les dernières décennies et on

trouve plusieurs exemples de son usage sur le problème de classification audio [207][68]. Son extension aux **k Plus Proches Voisins** (kNN, *k Nearest Neighbors*) permet de renforcer l'algorithme en présence d'exemples d'apprentissage marginaux en basant la décision sur un vote majoritaire parmi les k exemples les plus proches dans l'ensemble d'apprentissage. Cette version plus robuste du plus proche voisin est beaucoup plus diffusée dans la communauté scientifique et encore assez exploitée dans notre domaine [139][202][182][13]. Toutefois, les kNN tombent aujourd'hui en désuétude, principalement parce que la recherche des exemples les plus proches se heurte à la fameuse *malédiction de la dimensionalité*, et peut ainsi impliquer, dans des espaces à grande dimension, une recherche exhaustive parmi les exemples de la base, ce qui se traduit par un coût en mémoire et en temps de calcul prohibitifs par rapport aux méthodes plus récentes.

Le **perceptron**, qui est à l'origine une émulation artificielle du comportement d'un neurone proposée par Rosenblatt [200], est en fait strictement équivalent au modèle de la LDA (équation 2.2). La contrainte de linéarité a considérablement limité le développement du perceptron durant de nombreuses années. Dans les années 70 cependant, les techniques neuronales ont connu un regain d'intérêt grâce à l'introduction du **Perceptron Multi-Couches** (MLP, *Multi-Layer Perceptron*) qui introduit la non-linéarité par le biais d'une structure de couches de perceptrons mutuellement alimentées, ce qui explique l'autre nom couramment employé pour cette technique : les **Réseaux de Neurones Artificiels** (ANN, *Artificial Neural Networks*). Cette technique a rencontré un grand engouement et l'on trouve beaucoup d'exemples de son application sur le problème de la classification audio [106][202][121][204]. Cependant l'apprentissage des réseaux de neurones est une procédure difficile qui nécessite généralement une supervision manuelle pour garantir la convergence. De plus ceux-ci constituent une sorte de boîte noire qui n'offre pas ou peu d'interprétations sur les données exploitées.

Les **Machines à Vecteurs de Support** (SVM, *Support Vector Machines*), que nous avons évoquées en introduction, sont actuellement plus populaires que les réseaux de neurones (bien que le débat soit encore vif entre les tenants des deux approches) et ont en outre montré leur équivalence implicite à certaines structures de réseaux. Pourtant on compte aujourd'hui encore peu d'articles [140][48][96][133] tirant parti de cette méthode pour la classification audio, par rapport aux autres méthodes plus connues de la communauté.

On trouve également quelques exemples dans la littérature d'application pour la tâche de classification audio [43][194] du méta-algorithme **AdaBoost** [79][80], qui renforce un discriminateur faible (tel l'algorithme C4.5) en en combinant de multiples instances, et permet ainsi de complexifier la surface de décision, tout en évitant le problème du sur-apprentissage. On montre par ailleurs [206] que celui-ci induit implicitement un principe de maximisation de marge et présente ainsi de fortes similarités théoriques avec les Machines à Vecteurs de Support, notamment concernant les bornes sur l'erreur de généralisation.

2.3.3 Approches hybrides

Sans trop en détailler le contenu, nous noterons que la plupart des propositions visant à améliorer le processus de classification pour le problème parole/musique se basent sur des approches hybrides combinant certains des algorithmes présentés précédemment. Ainsi, Ellis et Williams [247] et Ajmera et al. [9] proposent un algorithme appliquant un modèle HMM sur les sorties d'un réseau de neurones appris pour identifier les phonèmes de parole. Goodwin et Laroche [91] introduisent le facteur temporel dans une approche par LDA en y adjoignant une procédure de programmation dynamique.

Certains auteurs proposent également des solutions pour fusionner les résultats de divers classificateurs. Ainsi, outre les systèmes multi-experts classiques [102][5], on trouve des paradigmes de fusion basés sur la théorie de l'Evidence (qui se substitue à la théorie des probabilités) [150], sur les réseaux bayésiens [87] ou encore sur des combinaisons de modèles gaussiens [102].

On trouve également plusieurs propositions d'approches **hybrides SVM/GMM** censées tirer parti des avantages des deux approches (discriminative et générative), comme les supervecteurs proposés dans [32] pour la reconnaissance du sexe et du genre du locuteur, ou la combinaison de

Milgram et al. [159].

2.3.4 Discussion

On constate, au regard de cet état de l'art, que la majorité des méthodes utilisées pour la classification audio sont de nature discriminative. Ceci s'explique en partie par le fait qu'elles répondent au principe fondamental énoncé par Vapnik [236], qui se résume à ne jamais traiter un problème par la résolution d'un problème plus général, et donc plus complexe. Ainsi, tandis que les méthodes génératives emploient une grande partie de l'effort d'apprentissage dans la modélisation d'une distribution sur l'ensemble de son support, les méthodes discriminatives se limitent à la seule caractérisation de la région délimitant les classes. Le but de la classification étant en définitive d'associer une classe à chaque exemple, on comprend que la modélisation générative des classes se traduit par une recherche d'informations non-pertinentes qui implique soit un coût supplémentaire inutile, soit une pénalisation des performances à coût égal. Ce constat oriente donc notre choix vers l'usage d'une méthode discriminative.

Nous avons en outre évoqué la *malédiction de la dimension* (ou *curse of dimensionality*). Ce phénomène, décrit pour la première fois par Bellman [20], constitue l'un des problèmes majeurs en apprentissage statistique. En effet, le « volume » d'un espace augmente exponentiellement avec sa dimension, si bien qu'un espace à grande dimension peuplé par un nombre fini d'exemples peut être considéré comme quasiment vide [104], et donc difficilement caractérisable. De la répartition éparse des exemples dans l'espace résulte donc généralement un *sur-apprentissage* qui réduit toute capacité de généralisation de l'algorithme sur des exemples inconnus. Les modèles à mélanges de gaussiennes et la technique des k plus proches voisins sont tous deux sujets à ce phénomène, le premier parce que l'augmentation de la dimension oblige à accoître le nombre de gaussiennes pour caractériser correctement les distributions, le second à cause de la structure éparse des exemples dans l'espace et la complexité implicite de la métrique. Nous verrons dans la présentation des Machines à Vecteurs de Support (partie I) que celles-ci se distinguent entre autres par leur moindre sensibilité à la dimension de l'espace des descripteurs.

2.3.5 Un mot sur la détection de chant

Nous ne détaillons pas ici les algorithmes de classification employés dans la littérature pour la tâche de détection de chant parce que ceux-ci sont globalement les mêmes que ceux employés pour la classification parole/musique, à savoir les modèles à mélanges de gaussiennes (GMM) [52][228][135][105], les modèles de Markov cachés (HMM) [24][172][18], les réseaux de neurones [25][237] et les Machines à Vecteurs de Support (SVM) [131][144][237] [199], ainsi que plusieurs heuristiques par seuillages empiriques [124][143][258][129][196].

Nous verrons que les principaux efforts sur cette tâche se concentrent sur la construction de descripteurs pertinents pour identifier le chant sur un fond musical, plutôt que sur la méthode de classification.

2.4 Caractérisation audio

Nous avons vu, dans la section précédente, une collection parcellaire mais assez représentative des méthodes d'apprentissage statistique déployées sur les problèmes de classification audio. Ces dernières sont généralement le fruit des travaux de statisticiens et leur développement théorique est donc indépendant de toute application pratique. La caractérisation du signal audio, par la définition de descripteurs numériques susceptibles d'apporter l'information pertinente pour la tâche voulue, est au contraire fortement liée aux classes en présence. Nous présenterons ainsi les propositions de la littérature pour les deux problèmes posés : la classification parole/musique et la détection de chant.

Le calcul des descripteurs se base presque unanimement sur le principe du « sac de trames » (*bag of frames*), qui consiste à découper le signal audio en trames temporelles successives suffisamment courtes (de l'ordre de quelques centièmes de secondes) pour respecter la contrainte de quasi stationnarité des propriétés acoustiques. Le terme « sac de trames » décrit également le fait que

les trames sont considérées comme indépendantes, identiquement distribuées (IID) [145], et donc classifiées en dehors de toute considération d'ordre temporel, ce qui implique, comme nous le verrons dans la partie III, l'usage complémentaire de techniques de post-traitement pour réintroduire la donnée temporelle dans le processus de classification.

2.4.1 Classification parole/musique

Le problème de la discrimination entre parole et musique se focalise en général sur la caractérisation de la parole. En effet, la musique est un phénomène de nature très hétéroclite, impliquant une diversité de timbres et de dynamiques quasi infinie, et se trouve donc plus difficile à résumer par des propriétés simples.

C'est ainsi que le problème de classification parole/musique fut considéré à l'origine comme un problème annexe au traitement de la parole. Il est donc naturel que de nombreuses publications se soient dans un premier temps contentées de transposer l'usage de descripteurs reconnus dans ce domaine. Parmi ces descripteurs classiques on trouve les coefficients cepstraux sur échelle Mel (MFCC, *Mel Frequency Cepstral Coefficients*), qui constituent sans conteste le groupe de descripteurs le plus populaire dans la littérature [41][59][132][195][77][85][100][86][13], ainsi que les coefficients de prédiction linéaire (LPC, *Linear Prediction Coefficients*) [182][85] ou les coefficients de prédiction linéaire perceptifs (PLP, *Perceptual Linear Predictive analysis*) [9].

Alors que les articles précédents [106] se limitent au problème de la détection de voix parlée (généralement en présence de bruit), Saunders est le premier à publier [205] sur le problème spécifique de la classification parole/musique, suivi l'année suivante par l'article référence de Scheirer et Slaney [207]. Ces deux articles fournissent une analyse des propriétés permettant de discriminer parole et musique, dont nous retenons les points suivants :

- La voix parlée est une alternance de sons voisés (typiquement les voyelles et certaines consonnes), dont le spectre est quasi-harmonique, et de sons non-voisés (la plupart des consonnes) proches d'un bruit modulé. Cette alternance est beaucoup plus marquée que dans un signal de musique, où les parties harmoniques (notes tenues) sont généralement beaucoup plus longues que les parties non-harmoniques (percussives ou attaques transitoires).
- Cette alternance pour la parole se manifeste à une cadence relativement constante de 4 Hz que l'on nomme débit syllabique, et qui se traduit par un pic d'énergie autour de cette fréquence.
- Elle s'observe également en termes d'énergie globale puisque les consonnes non-voisées consistent généralement en attaques très fortes dont l'énergie contraste sensiblement avec les parties voisées. De plus, le signal de parole contient habituellement, si le débit n'est pas trop rapide, de nombreux interstices silencieux qui accentuent également cette alternance énergétique.
- L'alternance décrite précédemment se traduit également par des variations plus fréquentes du spectre d'un signal de parole que d'un signal de musique.
- La musique contient en général de nombreux phénomènes percussifs ou d'attaques qui se traduisent par un spectre centré sur une moyenne supérieure à celle du spectre de parole, qui lui se distingue dans les hautes fréquences par une nette décroissance spectrale d'environ 12 dB par octave.
- La voix est à priori plus localisée en fréquences, et limitée à 8 kHz, de même que la hauteur des sons, qui s'étend sur un ambitus moins large que la musique.

On peut y ajouter deux propriétés, secondaires parce qu'elles ne concernent pas directement le son lui-même :

- La musique populaire suit souvent un schéma rythmique très régulier qui se traduit par une périodicité marquée entre 40 et 200 battements par minutes.
- Les algorithmes de codage de la voix sont optimisés par rapport aux propriétés de cette dernière. Le résultat du codage d'un signal de musique par un algorithme de ce type doit donc, à débit constant, être plus bruité que sur un signal de parole. Le résiduel peut donc servir d'indice discriminant entre les deux sources.

La préoccupation principale de Saunders étant de délivrer un algorithme fonctionnant en temps réel, contrainte assez restrictive en 1996, il propose [205] une série de descripteurs exclusivement

basés sur le taux de tassage par zéro (ZCR, *Zero Crossing Rate*), dont le calcul est très rapide. Ces descripteurs consistent en une collection de processus d'intégration long-terme (moyenne, déviation standard, 3^{e} moment central, ...) appliqués sur les valeurs court-terme du ZCR. La pertinence de ce descripteur pour cette tâche est confirmée par le nombre de publications en faisant usage [201][41][42][261][202][176][169][140].

À la différence de Saunders, Scheirer et Slaney [207] ne se préoccupent pas des contraintes de temps de calcul et exploitent les caractéristiques énoncées plus haut en déployant une batterie de descripteurs beaucoup plus diversifiés, composée de la modulation d'énergie à 4 Hz, du taux de trames à basse énergie, de la fréquence du 95^{e} percentile d'énergie (appelée fréquence de coupure), du centroïde spectral (également appelé « clarté » sonore, *brightness*), du flux spectral, du ZCR, de la magnitude du résiduel après resynthèse spectrale, et enfin d'une mesure de battement rythmique. Les travaux des auteurs auront une certaine influence sur la communauté et l'on retrouvera un grand nombre de ces descripteurs dans la plupart des publications postérieures [42][202][87][182][151][139][169][13][229][140], le centroïde spectral et le flux spectral étant de loin les plus largement repris. On trouve en outre quelques descripteurs proches de ces derniers, comme le taux de hautes valeurs de ZCR (HZCRR, *High ZCR Ratio*) [139][68], le taux de trames silencieuses [71][138] ou le niveau d'activité [13], ou d'autres exemples de descripteurs spectraux mono-dimensionnels assez simples comme la largeur de bande [59][132][248][182][169], ou définis dans le standard MPEG 7 [3], comme la platitude spectrale et l'étalement spectral [41].

Bien qu'elle soit fortement liée aux conditions d'enregistrement, la mesure d'énergie instantanée (ou RMS, *Root Mean Square*), qu'elle soit mesurée sur le signal ou sur le spectre (en vertu du théorème de Plancherel), est également très populaire dans la littérature [41][132][261][87], au point de parfois constituer, de par son coût de calcul très réduit, la base principale de certaines propositions [176]. Certains auteurs préfèrent exploiter une version perceptive de ce dernier appelée *loudness*, qui prend en compte l'échelle de perception humaine quasi logarithmique [248][41][151]. L'energie instantanée apporte cependant peu d'information sur le contenu spectral, si bien qu'il est en général plus intéressant de calculer les énergies de sous-bandes fréquentielles [59][132][140][48], ou même les rapports d'énergie entre sous-bandes [182][151], qui impliquent une invariance par rapport à l'énergie globale ; la largeur de bande est parfois calibrée sur une échelle musicale comme l'octave [242][243]. Nwe et Li [170] font en outre précéder le calcul des énergies de sous-bande d'une phase d'accentuation harmonique (par le biais d'un banc de filtres triangulaires calés sur les partiels de la fréquence fondamentale estimée), destinée à atténuer les signaux non harmoniques.

Dans un article présentant une comparaison de différents descripteurs pour la tâche de classification parole/musique, Carey et al. [42] commentent les travaux de Saunders et Scheirer et Slaney en s'étonnant de ne pas y voir figurer une mesure impliquant la hauteur des sons, dont les variations dans la parole sont plus homogènes que dans la musique. Ils proposent en ce sens une mesure de fréquence fondamentale (ou *pitch*), que l'on retrouve également dans de nombreuses autres contributions [59][132][248][261][151] et dont l'apport peut également se traduire par une mesure du « rapport harmonique » [140][48][4], c'est-à-dire le taux de trames où une fréquence fondamentale peut être mesurée, qui permet ainsi de quantifier l'alternance entre trames voisées et non-voisées. Nielsen et al. [168] présentent une étude plus approfondie sur le pitch et proposent une série de descripteurs pour la classification audio, basés sur cette mesure.

Nous avons également mentionné la présence d'une structure rythmique, et en particulier d'un battement régulier, comme critère caractérisant le signal de musique. Ce point est en partie traité par la mesure de battements de Scheirer et Slaney, et sera repris et amélioré par Burred et Lerch [41] par le biais d'un histogramme d'intensités rythmiques sur lequel sont extraites diverses mesures statistiques (moyenne, déviation standard...) et une mesure de régularité par auto-corrélation. Tzanetakis et Cook ont également proposé [229] un algorithme très détaillé pour le calcul d'histogrammes d'intensités rythmiques basés sur une mesure d'auto-corrélation appliquée sur une estimation de l'enveloppe du signal. On retrouve encore l'usage de l'auto-corrélation pour la détection de rythme dans [111], cette fois appliquée sur les facteurs d'échelles du codage audio MPEG 1.

Les descripteurs spectraux présentés jusqu'ici sont tous construits sur une échelle linéaire des fréquences ou sur une échelle logarithmique. Plusieurs auteurs tentent de reproduire plus fidèlement le comportement auditif humain en appliquant des échelles perceptives pour le calcul de certaines grandeurs. Ainsi, dans [164], l'échelle Bark se substitue à l'échelle linéaire pour le calcul

du centroïde spectral. L'aspect psychoacoustique peut également être pris en compte sous d'autres formes, par exemple à travers la détection des phénomènes de rugosité (caractérisés par la modulation de l'enveloppe entre 20 et 150 Hz) ou d'enveloppes temporelles sur un banc de filtres d'échelle perceptive [151]. Des études beaucoup plus poussées visent à reproduire de manière précise le comportement du système auditif humain à travers différentes modalités, comme la représentation Taux-Echelle-Fréquence-Temps (*Rate-Scale-Frequency-Time*) introduite dans [194] ou le modèle cochléaire de [157]. Ces modèles sont toutefois d'une complexité sensiblement supérieure aux descripteurs de la littérature, et souvent inadaptés à un traitement en temps réel ou en ligne.

2.4.2 Détection de chant

Bien que la voix soit le point central dans les deux cas, le problème de la détection de chant diffère sensiblement de la classification parole/musique car les propriétés de la voix chantée ne sont pas les mêmes que celles de la voix parlée.

En premier lieu c'est le débit qui, sous la contrainte du temps musical, est profondément modifié par rapport à la voix naturelle (parlée). En effet, les notes musicales étant tenues par les chanteurs sur les voyelles, la proportion de trames voisées, qui n'est que de 60% en moyenne sur la parole, monte à 90% pour un signal de chant [53]. La plupart des descripteurs décrits précédemment basés sur l'alternance voisé/non-voisé sont donc moins pertinents dans ce cadre particulier. De plus, le débit syllabique à 4 Hz qui caractérise la parole devient ici caduque et ne permet plus d'identifier la voix chantée [52].

En définitive, on retient pour caractériser la voix chantée les critères suivants :

- Un des traits les plus souvent cités est sans doute le fameux « formant du chanteur », une résonance dans la bande de fréquences 2000-3000 Hz qui aide le chanteur à se faire entendre par dessus un accompagnement instrumental [223]. Mais l'accentuation de ce formant nécessite une technique très particulière que l'on ne retrouve guère qu'en musique lyrique, et ne permet donc pas d'identifier la voix chantée dans la musique populaire, qui constitue pourtant la cible principale de notre application.
- Le chant étant intrinsèquement de la musique, on retrouve certaines des propriétés énoncées précédemment pour différencier cette dernière de la parole. En particulier la dynamique des hauteurs musicales est beaucoup plus développée que dans la parole où la hauteur suit principalement une fonction prosodique qui se caractérise par des variations moindres et plus subtiles. À titre d'exemple on considère que la parole évolue habituellement entre 80 et 400 Hz tandis qu'une chanteuse soprano peut raisonnablement atteindre les 1400 Hz [199].
- Cette dynamique accrue se retrouve également sur les intensités, celles-ci faisant partie intégrante du langage musical (mais peut toutefois se trouver fortement réduite par les procédés de compression dynamique, couramment employés par les radios populaires).
- Comme nous l'avons mentionné plus haut, la voix chantée étire les sons voisés et peut en général être modélisée comme une séquence de hauteurs relativement constantes par morceaux (en suivant la terminologie mathématique), à la différence de la parole où la hauteur fluctue constamment pour les besoins de l'expression prosodique.
- La prédominance des sons voisés, et leur importance musicale, a pour effet de rendre le signal de chant beaucoup plus harmonique (dans le sens d'un spectre à structure de peigne régulier très marqué) que la parole.
- Enfin, l'un des traits qui caractérisent sans doute le mieux le chant est la présence quasi systématique d'un vibrato, que l'on peut plus ou moins différentier des vibratos instrumentaux [196].

La difficulté principale réside dans le fait que le signal de chant est mélangé au fond instrumental, qui peut être d'intensité comparable à la partie de voix, et dont le contenu est généralement fortement corrélé à celle-ci, en termes de schéma rythmique ou de notes jouées ; il est donc d'autant plus complexe de distinguer les deux contributions dans le signal, que la musique couvre une bande fréquentielle très large et ne peut donc être isolé du mélange qu'au prix d'une forte dégradation du signal de chant.

Pour certaines publications, la phase de détection de chant n'est qu'un pré-traitement pour

l'application d'algorithmes de reconnaissance du chanteur, aussi les solutions proposées y sont généralement assez simples et l'on retrouve quelques cas d'algorithmes de détection de la parole adaptés pour l'occasion [134][105]. La filiation évidente avec la détection de la parole, malgré les différences énoncées plus haut, se traduit également par l'exploitation de descripteurs classiques dans ce domaine, tels les PLP [1] [25][131][237], les LFPC [172], les MFCC [237][135][141], ainsi que d'autres descripteurs que nous avons présentés pour la classification parole/musique (énergie, ZCR, flux spectral...) [258].

Nous avons mentionné comme caractéristique principale de la voix chantée la présence de vibrato. Celui-ci se manifeste par une modulation conjointe du son en fréquence et en intensité (cette seconde modalité est parfois appelée *tremolo* pour la distinguer du vibrato fréquentiel), à la différence des instruments de musique qui dans leur grande majorité ne produisent qu'un seul de ces phénomènes à la fois. Ainsi dans le cas des instruments à vent c'est le tremolo qui prédomine tandis que les cordes favorisent le vibrato fréquentiel [196]. Lachambre et al. [129] proposent ainsi un critère de mesure de vibrato basé sur la recherche d'un pic fréquentiel entre 4 et 8 Hz. Afin de répondre à une contrainte de stabilité spectrale, le signal est segmenté en trames temporelles dont les frontières sont déterminées à partir de la structure des pics fréquentiels dans le spectrogramme. Le critère proposé consiste à calculer le taux de trames où le vibrato est détecté, parmi les trames d'un même segment temporel. Regnier et Peeters [196] accroissent la robustesse du critère en combinant les mesures de modulation de fréquence et d'intensité pour la détection de partiels dits « vibrants ». L'observation de plusieurs partiels vibrants simultanés détermine alors la détection de chant. Ces deux approches sont très efficaces car les critères proposés sont fortement discriminants pour la tâche considérée. Cependant elles impliquent toutes deux une phase très coûteuse de détection de partiels dans le spectrogramme. Nwe et Li [171] proposent à l'inverse une approche plus économique, mais moins efficace, basée sur le calcul de coefficients cepstraux après l'application de « filtres numériques de vibrato », dont la définition manque de clarté.

Les mêmes auteurs complètent cet apport en construisant d'autres filtres caractérisant certaines propriétés du chant. Ainsi un banc de filtres centrés sur les moyennes des formants permet également d'accentuer les résonances vocales. Un autre processus, appelé « attenuation harmonique » et initialement introduit dans [172], vient atténuer le signal de musique par rapport au chant par un filtrage harmonique triangulaire, le vibrato fréquentiel ayant un effet d'étalement des pics harmoniques spectraux qui rend donc le signal de chant moins harmonique que la musique. Kim et Whitman [124] emploient paradoxalement un traitement similaire (filtrage par peigne harmonique sur la fondamentale, après filtre passe bande entre 200 et 2500 Hz) en le justifiant par l'argument contraire, à savoir que le chant est plus harmonique et qu'un seuil sur la mesure d'harmonicité peut ainsi constituer un critère de décision satisfaisant.

On retrouve également les descripteurs introduits à l'origine par Williams et Ellis [247] pour la tâche de classification parole/musique, adaptés dans [24] pour la détection de chant. Les PPF (*Post Probability Features*) sont le résultat à 54 composantes d'un réseau de neurones de reconnaissance de phonèmes de la parole. Les auteurs comparent par la suite l'efficacité de ces descripteurs dans une approche classique du maximum de vraisemblance avec des critères d'information (entropie, dynamisme, ...) calculés sur ces derniers.

Enfin, Maddage et al. proposent une approche originale [143] qui consiste à itérer deux fois la transformée de Fourier sur une fenêtre de signal. En effet, si l'on considère que la FFT [2] d'un signal périodique est un train de pulsation périodique (les partiels), alors la FFT de cette FFT est un sinus cardinal, dont les premières composantes contiennent plus d'énergie dans le cas du chant, parce que son spectre harmonique est plus dense. Un seuil sur l'énergie cumulée des premières composantes permet ainsi de décider entre chant et musique. Cependant, pour pouvoir appliquer la FFT sur un spectre stationnaire, l'algorithme nécessite une première phase de détection du rythme assez coûteuse afin d'être appliqué sur chaque fenêtre encadrée pardes battements successifs.

1. se référer à la section 6.5 sur les descripteurs employés pour la classification parole/musique pour la signification des acronymes.
2. *Fast Fourier Transform*, désigne ici le résultat du calcul numérique de la transformée de Fourier.

2.4.3 Discussion

Cet état de l'art donne une idée de la diversité des descripteurs mis en jeu sur les problèmes posés. Bien que la plupart s'accompagnent d'une argumentation raisonnée concernant leur efficacité réelle ou supposée, il est a priori impossible pour l'expérimentateur qui voudrait tirer parti de ces contributions de déterminer lesquels choisir en premier lieu, d'un point de vue théorique, d'autant que beaucoup d'entre eux sont largement redondants.

Beaucoup d'auteurs proposent ainsi des protocoles expérimentaux comparant les résultats obtenus pour un même classifieur avec chacun des descripteurs d'un ensemble donné. C'est l'approche que suivent par exemple Scheirer et Slaney [207] sur l'ensemble des descripteurs qu'ils proposent pour la tâche de classification parole/musique, et que l'on retrouve dans une longue liste de publications : [42][195][68][202][176][87]...

Il peut bien sûr être avantageux de grouper les différents descripteurs dans la phase de classification, et l'on trouve parfois dans ces comparaisons les résultats obtenus pour différentes combinaisons, qui montrent en général l'avantage à exploiter tous les descripteurs en même temps. Ce résultat met pourtant en évidence les limites d'un tel protocole, d'abord parce qu'en définitive il ne fait que montrer que l'accumulation d'information profite d'une façon ou d'une autre au classifieur, ensuite parce qu'il exclut la prise en compte de la complexité induite par la superposition ded descripteurs. Or, pour les applications visées par notre système (l'annotation de gros volumes d'archives et la classification du flux audio en direct), le coût en temps de calcul est un aspect essentiel. De plus, nous verrons que l'accumulation d'un grand nombre de descripteurs finit par être préjudiciable au processus de classification parce qu'une partie de cette information peut se révéler non pertinente pour la tâche choisie, et le reste largement redondant.

Une alternative raisonnable consiste à choisir une combinaison de taille raisonnable parmi une collection de descripteurs disponibles. L'opération, si elle est guidée par une mesure de performances après apprentissage du classificateur, se révèle vite irréaliste car l'évaluation de tous les sous-ensembles possibles implique une explosion combinatoire.

La sélection automatique de descripteurs est en fait un sujet d'importance croissante dans le domaine de l'apprentissage statistique, dont le développement est en grande partie dû aux besoins de la bioinformatique, qui traite couramment des données contenant plusieurs milliers de composantes fortement redondantes ou bruitées. Un examen poussé de la littérature sur les deux problèmes posés nous indique pourtant que pratiquement aucun article ne met à contribution ces techniques de sélection de descripteurs. On peut néanmoins citer Peeters [179] qui exploite l'algorithme IRMFSP (que nous présenterons dans la section 7.3.2), algorithme relativement simple mais efficace qu'il avait proposé précédemment [181] pour la reconnaissance d'instruments de musique, et Rocamora [199] qui utilise un algorithme basé sur les mesures de corrélation entre descripteurs, pour la détection de chant.

Nous proposerons donc dans ce document une approche qui se démarque des publications antérieures en mettant l'accent non sur le développement de nouveaux descripteurs mais sur la mise en place d'un cadre de sélection efficace appliqué sur un grand ensemble de descripteurs collectés dans la littérature. Nous verrons en outre que la notion de pertinence n'est pas absolue et que l'efficacité d'un ensemble de descripteurs est fortement liée au classifieur mis en jeu, ce qui nous amènera à proposer de nouveaux algorithmes de sélection adaptés aux Machines à Vecteurs de Support.

Première partie

Classification par Machines à Vecteurs de Support

Introduction de la partie I

Cette partie est consacrée à la mise en application des Machines à Vecteurs de Support (SVM) sur un problème d'apprentissage multi-classes [3].

Nous commencerons par présenter dans le chapitre 3 les Machines à Vecteurs de Support d'un point de vue théorique, en mettant l'accent sur le principe fondamental de maximisation de la marge, duquel émerge un sous-ensemble restreint d'exemples, appelés Vecteurs de Support, définissant exhaustivement le classifieur.

On montre en outre que ce principe est directement lié au principe de Minimisation du Risque Structurel, qui permet de contrôler de manière conjointe le risque empirique et les propriétés de généralisation du classifieur. L'adjonction d'une fonction noyau aux SVM permet d'étendre le champ d'application à des surfaces de séparation non-linéaires par le biais d'une transformation implicite vers un espace de plus grande dimension. Elle peut d'ailleurs être interprétée comme l'adaptation du principe de Maximisation de la Marge à divers algorithmes de classification plus anciens simulés par le comportement du noyau. Ce dernier point, ainsi que le principe de Minimisation du Risque Structurel, justifie l'usage des SVM par rapport à de nombreuses autres méthodes de la littérature.

Bien qu'elles réduisent considérablement la proportion de paramètres dont l'affinage, généralement empirique, est essentiel dans certaines méthodes comme les réseaux de neurones, les Machines à Vecteurs de Support restent essentiellement déterminées par le choix du noyau utilisé. Nous abordons dans le chapitre 4 la question de l'évaluation des performances d'un noyau et de ses paramètres par rapport à un problème, en présentant divers critères, généralement basés sur l'estimation de l'erreur *Leave one out*, dont le critère d'Alignement du noyau, que nous introduisons dans le domaine de l'indexation audio.

Construites sur la séparation linéaire, les SVM sont par nature discriminatives. L'application à un problème multi-classes n'est donc pas immédiate et nous explorons dans ce sens les approches proposées dans la littérature au chapitre 5. Nous proposons une approche hybride couplant les approches *un contre un* et par dendogramme, et permettant d'estimer les probabilités a posteriori des classes impliquées, qui constitueront la base des processus de post-traitements présentés dans la partie III.

3. Par *multi-classes* nous entendons un problème impliquant plus de 2 classes.

Chapitre 3

Présentation des Machines à Vecteurs de Support

Sommaire

3.1 Classification supervisée 33
3.2 Prélude 34
3.3 Machines à Vecteurs de Support linéaires 34
3.4 Principe de Minimisation du Risque Structurel 36
3.5 Noyaux 37
3.5.1 Introduction 37
3.5.2 Théorie des Noyaux Reproduisants 38
3.5.3 Noyaux d'usage courant 40
3.5.4 Machines à Vecteurs de Support non-linéaires 41
3.6 Machines à Marge souple 41
3.7 Méthodes à noyaux 42
3.8 Une méthode universelle d'apprentissage 43

3.1 Classification supervisée

Les Machines à Vecteurs de Support (SVM) font partie d'une vaste famille d'algorithmes originellement regroupés dans le domaine de la reconnaissance de formes (*pattern recognition*). Les données sont généralement modélisées sous forme d'un vecteur aléatoire réelle $\boldsymbol{x} \in \mathbb{R}^d$ dont la génération, gouvernée par une densité de probabilité $p(\boldsymbol{x}, y)$, est dépendante de $y \in \{1, \ldots, C\}$, la classe d'appartenance de $\boldsymbol{x}$. Dans le cas particulier de la discrimination (problème à deux classes), on suppose $y \in \{+1; -1\}$, où $y_i = +1$ est associé à la classe 1, et $y_i = -1$ à la classe 2, pour alléger les notations. La densité de probabilité de $p(\boldsymbol{x}, y)$ étant généralement inconnue, on se base sur un ensemble de n réalisations $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1\ldots n}$ pour caractériser cette dernière. On distinguera par la suite les ensembles $\mathcal{S}_1 = \{(\boldsymbol{x}_i, y_i), y_i = +1\}$ et $\mathcal{S}_2 = \{(\boldsymbol{x}_i, y_i), y_i = -1\}$, avec $\mathrm{Card}(\mathcal{S}_1) = n_1$ et $\mathrm{Card}(\mathcal{S}_2) = n_2$.

Contrairement à d'autres méthodes de classification, comme les modèles à mélanges de gaussiennes (GMM), les SVM ne se basent pas sur l'estimation de la densité de probabilité, mais sur l'estimation d'une fonction de discrimination entre les exemples des deux classes. On cherche donc une fonction de décision $f : \mathbb{R}^d \to \mathbb{R}$ telle que

$$\mathrm{sign}(f(\boldsymbol{x})) = y_i.$$

Les Machines à Vecteurs de Support se distinguent en premier lieu parmi les méthodes discriminatives par le critère d'optimalité guidant le choix d'une telle fonction, mais leur émergence fait écho à de nombreuses techniques antérieures de classification supervisée.

3.2 Prélude

Fisher [76] propose en 1936 l'un des premiers algorithmes de reconnaissance de formes, qui deviendra par la suite l'*Analyse Discriminante de Fisher*. Cette dernière consiste en la détermination d'un hyperplan de séparation linéaire optimal entre les exemples des deux classes. Ainsi si l'on caractérise cet hyperplan par son vecteur normal $\boldsymbol{w}$, la fonction de décision prend la forme suivante :

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b, \tag{3.1}$$

où b est appelé le *biais*, ou *poids de seuil*. Ce problème est résolu dans le cadre de l'Analyse Discriminante de Fisher en maximisant le critère de séparation S, défini comme le rapport de la variance inter-classes sur la variance intra-classes :

$$S = \frac{\left(\boldsymbol{w}^T \boldsymbol{\mu}_1 - \boldsymbol{w}^T \boldsymbol{\mu}_2\right)^2}{\boldsymbol{w}^T \left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\right) \boldsymbol{w}}, \tag{3.2}$$

où $\boldsymbol{\mu}_i$ et $\boldsymbol{\Sigma}_i$ sont respectivement le centre et la matrice de covariance des exemples de la classe i. On montre que, sous l'hypothèse de gaussianité des densités de probabilités, l'hyperplan optimal au sens du critère de séparation est caractérisé par le vecteur suivant :

$$\boldsymbol{w} = \left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\right)^{-1} \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right). \tag{3.3}$$

Néanmoins l'hypothèse de gaussianité est très forte et rarement rencontrée dans des situations réelles.

Les Machines à Vecteurs de Support résolvent ce handicap en basant la résolution sur un critère d'optimalité différent qui offre en outre des bases théoriques sur les propriétés de généralisation de la fonction de décision. Elles constituent au début des années 90 un carrefour entre plusieurs domaines jusque-là indépendants : la reconnaissance de formes, les réseaux de neurones, les techniques de programmation mathématique et la théorie mathématique des Noyaux Reproduisants (RKHS, *Reproducing Kernel Hilbert Spaces*) introduite par Aronszajn en 1950 [15].

3.3 Machines à Vecteurs de Support linéaires

En 1964, Vapnik et Lerner introduisent le principe de maximisation de la marge dans un algorithme (*Generalized Portrait Algorithm*) qui constituera le principe fondamental des futures Machines à Vecteurs de Support [235].

Dans le cas de données séparables, il existe une infinité d'hyperplans permettant de séparer les deux classes, comme l'illustre la figure 3.1 (dont on ne retiendra pour l'instant que les hyperplans en trait plein). Néanmoins, les données étant considérées comme des réalisations d'une variable aléatoire, le choix de l'hyperplan doit être guidé par les propriétés de généralisation de la fonction de décision.

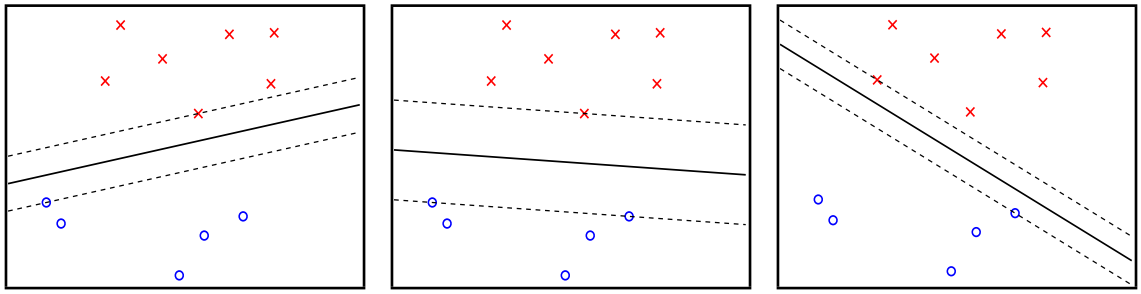


FIGURE 3.1 – Exemples d'hyperplans de séparation (en trait plein) et d'hyperplans de marge (en pointillés), entre deux classes dont les exemples sont respectivement représentés par des cercles et des croix.

La séparabilité des données implique que la contrainte $y_i f(\boldsymbol{x}_i) > 0$ est remplie pour chaque exemple. Il existe donc une valeur M, appelée *marge*, minimisant l'ensemble des distances $\frac{y_i f(\boldsymbol{x}_i)}{||\boldsymbol{w}||}$

entre les exemples et l'hyperplan :

$$\frac{y_i f(\boldsymbol{x}_i)}{\|\boldsymbol{w}\|} \geq M. \tag{3.4}$$

Le principe de l'algorithme consiste à déterminer le vecteur $\boldsymbol{w}$ maximisant la marge M. Il s'agit donc d'une optimisation de type *minimax* (ou plutôt maximin). Il est cependant nécessaire de fixer une contrainte additionnelle sur la norme de $||\boldsymbol{w}||$ afin de restreindre le champ infini de solutions ne différant que par un facteur d'échelle :

$$M \|\boldsymbol{w}\| = 1. \tag{3.5}$$

Ainsi, maximiser la marge M revient à minimiser la norme du vecteur $||\boldsymbol{w}||$. L'inéquation 3.4 devient donc la contrainte $y_i f(\boldsymbol{x}_i) \geq 1$. On appelle ainsi *hyperplans de marge* les deux hyperplans satisfaisant la condition $f(\boldsymbol{x}) = \pm 1$. La figure 3.1 montre trois exemples d'hyperplans de séparation (en trait plein) définis par des vecteurs $\boldsymbol{w}$ différents, ainsi que les hyperplans de marge correspondants (en pointillés). On peut voir que la marge, distance entre les hyperplans de marge et de séparation, diffère entre les trois cas, la figure centrale représentant la situation de marge maximale.

L'utilisation de la norme quadratique de $\boldsymbol{w}$ facilitera par la suite la résolution du problème et permettra en outre l'introduction du noyau, présentée plus bas. La recherche de l'hyperplan maximisant la marge se fait donc par la résolution du problème quadratique suivant :

$$\begin{array}{rl} \text{minimiser} & \frac{||\boldsymbol{w}||^2}{2} \\ \text{sous les contraintes} & y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right) - 1 \geq 0, \quad i = 1, \ldots, n. \end{array} \tag{3.6}$$

On exprime alors le Lagrangien par l'introduction des *multiplicateurs de Lagrange* α_i (également appelés *coefficients de Kühn-Tucker*) sur les contraintes :

$$\begin{array}{l} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{n} \alpha_i \left[y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right) - 1\right], \\ \text{avec } \alpha_i \geq 0, \quad i = 1, \ldots, n. \end{array} \tag{3.7}$$

À l'optimum, les multiplicateurs de Lagrange sont en outre soumis aux conditions suivantes :

$$\alpha_i \left(y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right) - 1\right) = 0. \tag{3.8}$$

Le problème d'optimisation 3.6 revient à minimiser le Lagrangien L par rapport à $\boldsymbol{w}$ et b et à le maximiser par rapport aux variables α_i. Ils satisfait donc, au point optimal, les conditions nécessaires suivantes :

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i = 0 \tag{3.9}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.10}$$

$$\text{soit :} \qquad \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i \tag{3.11}$$

Les exemples satisfaisant l'égalité $y_i f(\boldsymbol{x}_i) = 1$ sont appelés les *vecteurs de support*. Ils sont situés sur les hyperplans de marge et sont les exemples les plus proches de l'hyperplan de séparation. L'inégalité 3.8 montre que les α_i sont tous nuls à l'exception de ceux associés aux vecteurs de support. Ainsi si l'on définit $\mathcal{S}_{SV} = \{i, \alpha_i > 0\}$ l'ensemble des indices de ces vecteurs de support, on remarque que $\boldsymbol{w}$ s'exprime exclusivement en fonction de ces derniers :

$$\boldsymbol{w} = \sum_{i \in \mathcal{S}_{SV}} \alpha_i y_i \boldsymbol{x}_i. \tag{3.12}$$

C'est là un des avantages fondamentaux des Machines à Vecteurs de Support ; la contrainte de maximisation de la marge réduit le problème à un nombre restreint d'exemples, induisant ainsi

une robustesse aux exemples marginaux et de bonnes propriétés de généralisation, comme nous le verrons par la suite.

La fonction de décision prend donc la forme suivante (en développant l'équation 3.1) :

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \, y_i \, \boldsymbol{x}_i^T \boldsymbol{x} + b = \sum_{i \in \mathcal{S}_{SV}} \alpha_i \, y_i \, \boldsymbol{x}_i^T \boldsymbol{x} + b. \tag{3.13}$$

Toutefois, le problème précédent (équation 3.6), dit *primal*, est généralement compliqué à résoudre. Si l'on développe l'expression du Lagrangien (éq. 3.7) :

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \boldsymbol{w}^T \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i - b \sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i \,, \tag{3.14}$$

l'injection des équations 3.10 et 3.11 fait disparaître les variables $\boldsymbol{w}$ et b et permet d'obtenir la forme *duale* du problème d'optimisation, où le Lagrangien L_D ne dépend que des multiplicateurs de Lagrange :

$$\begin{aligned} \text{maximiser} \quad & L_D(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T\mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{H}\boldsymbol{\alpha} \\ \text{sous les contraintes} \quad & \alpha_i \geq 0, \quad i = 1, \ldots, n \\ \text{et} \quad & \boldsymbol{\alpha}^T\boldsymbol{y} = 0 \end{aligned} \tag{3.15}$$

où on a préféré l'écriture matricielle, plus concise, avec $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^T$, $\boldsymbol{y} = [y_1, \ldots, y_n]^T$, $\mathbf{1} = [1, \ldots, 1]^T$ et $[\boldsymbol{H}]_{ij} = y_i \, y_j \, \boldsymbol{x}_i^T \boldsymbol{x}_j$. Cette forme duale est généralement choisie pour opérer la résolution du problème d'optimisation.

3.4 Principe de Minimisation du Risque Structurel

Le principe de maximisation de la marge induit donc l'émergence d'un sous-ensemble d'exemples, appelés *vecteurs de support*, décrivant à eux seuls le comportement du classifieur. Ce principe, appliqué à la séparation linéaire dans la section précédente, peut être étendu à une multitude d'algorithmes [208], par le biais de fonctions noyaux (voir la section 3.5). L'engouement pour les Machines à Vecteurs de Support s'explique ainsi par le fait que diverses méthodes de classifications se sont trouvées fédérées dans un cadre théorique commun, mais surtout parce que leur principe est validé par la théorie de l'apprentissage statistique développée par Vapnik et Chervonenkis dans les années 70 [236].

On peut en effet reformuler le processus d'apprentissage comme la détermination d'une fonction de décision f_λ parmi un ensemble

$$\mathcal{H} = \{f_\lambda(\boldsymbol{x}), \lambda \in \Lambda\} \quad \text{avec} \quad f_\lambda : \mathbb{R}^d \to \mathbb{R}, \tag{3.16}$$

où λ est l'ensemble des paramètres ajustés durant l'apprentissage (par exemple les variables $\boldsymbol{w}$ et b dans le cas des SVM). On cherche donc une fonction f_{λ^*} qui minimise le *risque fonctionnel* suivant :

$$R(\lambda) = \int |f_\lambda(\boldsymbol{x}) - y| \, P(\boldsymbol{x}, y) d\boldsymbol{x} \, dy.$$

Néanmoins, la probabilité $P(\boldsymbol{x}, y)$ étant inconnue, il est impossible d'évaluer le risque $R(\lambda)$. N'ayant accès qu'à des réalisations de $P(\boldsymbol{x}, y)$, on calcule donc une approximation stochastique du risque, le *risque empirique* :

$$R_{emp}(\lambda) = \frac{1}{n} \sum_{i=1}^{n} |f_\lambda(\boldsymbol{x}_i) - y| \,.$$

où les $\boldsymbol{x}_i$ sont les exemples de l'ensemble d'apprentissage. La loi des grands nombres garantit la convergence du risque empirique vers le risque fonctionnel lorsque le nombre d'exemples est suffisamment conséquent. Cependant, seule une convergence uniforme peut garantir que le minimum

de R_{emp} converge vers le minimum R. Vapnik et Chervonenkis ont montré [232][233] que la condition nécessaire et suffisante pour garantir cette convergence est la finitude de la *dimension VC* (d'après les initiales des auteurs), notée h, de l'espace $\mathcal{H}$ (éq. 3.16). Celle-ci est un nombre entier naturel ou infini, défini comme le nombre maximum d'exemples pouvant être séparés de toutes les façons possibles par les fonctions de $\mathcal{H}$. Elle constitue une mesure de complexité de l'ensemble de classifieurs $\mathcal{H}$.

Un théorème de Vapnik et Chervonenkis [236] fournit une borne supérieure au risque fonctionnel, avec la probabilité $1-\eta$, décrivant l'influence de la dimension VC sur la convergence de R_{emp} vers R :

$$R(\lambda) \leq R_{emp}(\lambda) + \sqrt{\frac{1}{n}\left[h\left(\ln\frac{2n}{h}+1\right) - \ln\frac{\eta}{4}\right]}. \tag{3.17}$$

L'augmentation de la dimension VC h induit des fonctions de décision plus complexes, et par conséquent une réduction du risque empirique (du taux d'erreur). Mais ce gain, s'il n'est pas contrôlé, peut se faire au détriment de la capacité de généralisation du classifieur. Ce phénomène est connu sous le nom de *sur-apprentissage*. Il est en effet toujours possible d'obtenir un taux d'erreur nul sur tout ensemble d'apprentissage (il suffit pour cela d'en mémoriser tous les exemples), sans que le classifieur n'ait modélisé ou « *compris* » la structure des données en question. Le deuxième terme de la partie droite de l'inégalité 3.17, appelé *risque structurel*, apporte précisément l'information relative à la complexité de l'ensemble de recherche des classifieurs. Tandis que le risque empirique ($R_{emp}(\lambda)$) décroît avec l'augmentation de h, le risque structurel augmente. Il existe donc une valeur de h minimisant la borne ainsi formulée, qui traduit le compromis optimal entre le risque empirique et la complexité de la famille de classifieurs.

Le *principe de minimisation du risque structurel* est un résultat fondamental de la théorie de Vapnik et Chervonenkis mais la dimension VC reste généralement difficile à évaluer. Néanmoins, on peut montrer [236] que dans le cas où $\mathcal{H}$ décrit l'ensemble des hyperplans de séparation, si l'on impose la contrainte

$$||\boldsymbol{w}|| \leq A,$$

et si l'ensemble des exemples peuvent être contenus dans une sphère de rayon R, alors la dimension VC satisfait l'inégalité suivante

$$h \leq \min\left(\lceil R^2A^2\rceil, d\right) + 1, \tag{3.18}$$

où d est la dimension de l'espace d'entrée des fonctions de décision $f \in \mathcal{H}$. Ainsi, on utilise en général la relation suivante pour estimer la dimension VC :

$$h \approx R^2||\boldsymbol{w}||^2. \tag{3.19}$$

Le rayon R étant fixé par la répartition des exemples d'apprentissage, on voit donc que le principe de maximisation de la marge, présenté en section 3.3, qui équivaut à minimiser $||\boldsymbol{w}||$, remplit le principe de minimisation du risque structurel puisqu'il applique conjointement la minimisation du risque empirique et de la dimension VC.

On pourra trouver une introduction plus détaillée aux fondements statistiques des Machines à Vecteurs de Support dans [39] et [175], ou se référer à l'ouvrage de référence de Vapnik [236] pour une étude plus approfondie.

3.5 Noyaux

3.5.1 Introduction

Malgré une base théorique solide, les SVM restent toutefois fortement limitées par la restriction aux séparateurs linéaires. Il est en effet rare que des données réelles soient providentiellement réparties de chaque côté d'un hyperplan.

Le domaine des *classifieurs polynômiaux* étudie la mise en forme d'algorithmes basés sur des combinaisons multiplicatives de descripteurs. Ainsi, soit un exemple à d composantes $\boldsymbol{x} =$

$[x_1 \dots x_d]$, l'ensemble $\mathcal{M}^\delta$ des produits à l'ordre $\delta \in \mathbb{N}$ (appelés *monômes*) peut apporter une information non exprimée par les composantes originales :

$$\mathcal{M}^\delta = \{x_{j_1} \cdot x_{j_2} \cdot \ldots \cdot x_{j_\delta}\} \quad \text{où} \quad j_1 \leq \ldots \leq j_\delta \in \{1, \ldots, d\}\,.$$

On peut ainsi définir une transformation Φ qui porte des exemples bi-dimensionnels de l'espace originel $\mathbb{R}^2$ vers un espace de dimension supérieure contenant tous les monômes à l'ordre 2 :

$$\begin{array}{rcl} \Phi : \mathbb{R}^2 & \rightarrow & \mathbb{R}^4 \\ \boldsymbol{x} = (x_1,\, x_2) & \mapsto & (x_1^2,\, x_2^2,\, x_1x_2,\, x_2x_1). \end{array} \tag{3.20}$$

On remarque que les formulations primale et duale du problème d'optimisation des SVM (éq. 3.7 et 3.15) n'impliquent que des produits scalaires d'éléments de l'espace de classification. Ainsi, si l'on souhaite appliquer la transformation Φ comme pré-traitement pour enrichir la collection de descripteurs, il nous suffit d'évaluer la fonction k suivante :

$$k(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle = \left(x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 y_1 x_2 y_2\right) \tag{3.21}$$

$$= \langle \boldsymbol{x}, \boldsymbol{y} \rangle^2 . \tag{3.22}$$

On peut montrer [212] que ce résultat se généralise pour tout ordre δ : si Φ est la transformation associant à un exemple $\boldsymbol{x}$ l'ensemble des monômes à l'ordre δ, alors :

$$k(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^\delta .$$

Ce résultat est particulièrement intéressant puisqu'il montre qu'il est possible d'appliquer la transformation Φ vers un espace de dimension supérieure sans calculer explicitement la fonction Φ. Celle-ci se trouve exprimée implicitement au travers de la fonction k, dont l'expression est beaucoup plus simple que celle de Φ. En effet, la dimension de l'espace image de ϕ peut très facilement excéder les capacités computationnelles d'une machine puisque l'on dénombre, pour des exemples de dimension d, N_δ monômes d'ordre δ :

$$N_\delta = \left(\begin{array}{c} \delta + d - 1 \\ \delta \end{array} \right) = \frac{(\delta + d - 1)!}{\delta!(d-1)!} .$$

Soit, par exemple, pour des données regroupant $d = 100$ composantes, un espace des monômes d'ordre $\delta = 5$ de dimension proche de 10^7.

La structure de cet espace se trouve cependant synthétisée par l'expression du produit scalaire dans la fonction k. La Théorie des Noyaux Reproduisants, introduite dans le paragraphe suivant, formalise ce résultat et l'étend à d'autres types de transformations.

3.5.2 Théorie des Noyaux Reproduisants

La Théorie des Noyaux Reproduisants a été introduite par Azonszajn en 1950 [15]. Elle formalise la relation entre la fonction k introduite précédemment et la transformation Φ, par le biais de concepts d'analyse fonctionnelle sur les espaces de Hilbert. On considère que les données évoluent dans un espace $\mathcal{X}$ quelconque.

Définition : Soit une fonction $k : \mathcal{X}^2 \rightarrow \mathbb{R}$. k est un *noyau semi-défini positif* si

$$\int_{\mathcal{X}} k(\boldsymbol{x}, \boldsymbol{y}) g(\boldsymbol{x}) g(\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \geq 0 \qquad \forall g \in \mathcal{C}(\mathcal{X}).$$

Il est possible d'utiliser la définition équivalente ci-dessous, généralement plus exploitable, basée sur les échantillonnages possibles de $\mathcal{X}$.

Définition (alternative) : $k : \mathcal{X}^2 \rightarrow \mathbb{R}$ est un *noyau semi-défini positif* si

$$\sum_{i,j=1}^{n} c_i c_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0 \qquad \forall \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \qquad \forall c_1, \ldots, c_n \in \mathbb{R}.$$

Si l'on définit la *matrice de Gram* $\boldsymbol{K}$ d'un noyau k par rapport aux éléments $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ comme $[\boldsymbol{K}]_{ij} = k_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, alors cette définition est équivalente à la positivité semi-définie (au sens matriciel) de $\boldsymbol{K}$ pour tout ensemble $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.

Les résultats d'algèbre matricielle nous permettent donc d'inférer les propriétés de *symétrie* du noyau

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_j, \boldsymbol{x}_i),$$

et de *positivité de la diagonale*

$$k(\boldsymbol{x}, \boldsymbol{x}) \geq 0 \qquad \forall \boldsymbol{x} \in \mathcal{X}.$$

Introduisons maintenant la transformation Φ suivante, de l'espace $\mathcal{X}$ vers l'espace fonctionnel $\mathbb{R}^{\mathcal{X}} = \{f : \mathcal{X} \to \mathbb{R}\}$:

$$\begin{array}{rccc} \Phi : \mathcal{X} & \to & \mathbb{R}^{\mathcal{X}} \\ \boldsymbol{x} & \mapsto & k(.\,, \boldsymbol{x}). \end{array} \tag{3.23}$$

On peut montrer que pour un ensemble arbitraire d'éléments $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$, l'ensemble $\mathcal{F}$ de fonctions a une structure d'espace vectoriel :

$$\mathcal{F} = \left\{ f, \quad f = \sum_{i=1}^{n} \alpha_i \Phi(\boldsymbol{x}_i) \right\}.$$

Soit deux fonctions de cet espace $f = \sum_{i=1}^{n} \alpha_i \Phi(\boldsymbol{x}_i)$ et $g = \sum_{j=1}^{n} \beta_j \Phi(\boldsymbol{x}_j)$, on montre en outre [212] que l'opérateur défini par

$$\langle f, g \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \beta_i k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

est un produit scalaire dans l'espace $\mathcal{F}$. De plus on remarque que

$$\langle f, g \rangle = \sum_{j=1}^{n} \beta_i f(\boldsymbol{x}_j) = \sum_{i=1}^{n} \alpha_i g(\boldsymbol{x}_i). \tag{3.24}$$

On peut également exprimer la fonction noyau sur tout exemple $\boldsymbol{x}_i$ comme élément de l'espace $\mathcal{F}$: $k(.\,, \boldsymbol{x}_i) = \sum_{j=1}^{n} \alpha_j \Phi(\boldsymbol{x}_i)$, avec $\alpha_j = \delta_{ij}$ (où δ_{ij} est le symbole de Kronecker). La propriété 3.24 est particulièrement intéressante puisqu'elle implique les deux relations suivantes sur la fonction noyau :

$$\langle k(.\,, \boldsymbol{x}_i), f \rangle = f(\boldsymbol{x}_i)$$

$$\langle k(.\,, \boldsymbol{x}_i), k(.\,, \boldsymbol{x}_j) \rangle = k(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

Cette dernière observation justifie le nom de *noyaux reproduisants* donné aux noyaux définis positifs, puisque ces derniers présentent la particularité de pouvoir s'exprimer comme un produit scalaire dans un espace fonctionnel en bijection avec l'espace $\mathcal{X}$. On retrouve finalement, en remplaçant les $k(.\,, \boldsymbol{x})$ par la transformation Φ introduite, la relation qui nous avait dans un premier temps servi à introduire la fonction noyau (équation 3.21), formalisée dans le théorème ci-dessous.

Le **théorème de Mercer** [156] affirme que tout noyau défini positif peut s'exprimer comme un produit scalaire sur l'espace image d'une fonction de transformation Φ :

$$k(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle .$$

Un noyau défini positif est d'ailleurs généralement décrit comme vérifiant la *condition de Mercer*. Le terme noyau désignera implicitement par la suite un noyau semi-défini positif.

On remarque que le théorème de Mercer stipule l'existence d'une fonction Φ mais n'apporte pas de moyen de la construire analytiquement. Il n'existe pas en fait de correspondance bijective entre le noyau k et la transformation Φ, parce que l'expression de cette dernière dépend de l'espace dans lequel on décrit le noyau reproduisant. Schölkopf et Smola présentent d'ailleurs une manière alternative de construire la fonction Φ à partir du noyau [212]. Il n'est pas rare, en outre, que Φ ne soit pas exprimable analytiquement (on parle alors de fonction *implicite*), comme dans le cas

d'espaces transformés de dimension infinie, ce qui vaut pour le noyau Gaussien RBF (que nous présentons dans la section suivante).

L'exploitation de noyaux fut introduite pour la première fois dans le domaine de l'apprentissage statistique en 1964 par Aizerman et al. [8], qui en présentent en outre l'interprétation géométrique. L'usage de la fonction noyau prend généralement le nom de *kernel trick* (littéralement « l'astuce du noyau ») puisqu'il permet, avec simplicité, d'introduire la non-linéarité dans un algorithme exclusivement basé sur des produits scalaires, c'est-à-dire invariant en rotation [212]. Bien qu'il leur soit antérieur de plusieurs décennies, le *kernel trick* ne fut réellement compris et largement utilisé qu'à partir du début des années 90 avec l'apparition des Machines à Vecteurs de Support [208].

3.5.3 Noyaux d'usage courant

Il est généralement difficile de vérifier analytiquement la condition de Mercer. Néanmoins, un certain nombre de noyaux sont connus comme étant définis positifs et largement exploités par la communauté :

- **Linéaire :** $k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y}$
  celui-ci correspond au produit scalaire sans transformation. Il traduit donc la forme traditionnelle des algorithmes, sans l'usage du *kernel trick*.
- **Polynômial homogène :** $k(\boldsymbol{x}, \boldsymbol{y}) = \left(\boldsymbol{x}^T \boldsymbol{y}\right)^\delta$
  présenté dans la section 3.5.1. Celui-ci permet indirectement d'appliquer le principe de maximisation de la marge aux classifieurs polynômiaux.
- **Polynômial inhomogène :** $k(\boldsymbol{x}, \boldsymbol{y}) = \left(1 + \frac{c}{d} \boldsymbol{x}^T \boldsymbol{y}\right)^\delta$
  L'ajout d'une constante au produit scalaire permet d'inclure dans la transformation Φ tous les monômes d'ordre inférieur ou égal à δ. Le noyau polynômial inhomogène implique donc un espace transformé de dimension supérieure au noyau homogène.
- **Gaussien RBF :** $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{y}||^2}{d\,\sigma^2}\right)$
  Les fonctions à base radiale (RBF *Radial Basis Functions*) sont définies par le fait qu'elles ne dépendent que de la distance entre leurs arguments : $\phi(\boldsymbol{x}, \boldsymbol{y}) = \phi(||\boldsymbol{x} - \boldsymbol{y}||)$.
  Le noyau Gaussien RBF applique ainsi une gaussienne sur la distance entre les exemples. On peut montrer que l'espace transformé dans ce cas est de dimension infinie, puisque les exemples d'une collection arbitrairement grande y sont linéairement indépendants. Le caractère radial présente en outre la particularité de placer tous les exemples sur la sphère unité dans l'espace transformé ($||\Phi(\boldsymbol{x})||^2 = k(\boldsymbol{x}, \boldsymbol{x}) = \exp(0) = 1 \quad \forall \boldsymbol{x}$).
- **Sigmoïdal :** $k(\boldsymbol{x}, \boldsymbol{y}) = \tanh\left(\frac{c}{d}\, \boldsymbol{x}^T \boldsymbol{y} + \theta\right)$
  La fonction de décision construite avec un noyau sigmoïdal est égale à celle d'un réseau de neurones à deux couches [187][39][236]. Le noyau sigmoïdal ne respecte pas la condition de Mercer pour toutes les valeurs de c et θ. Il demeure cependant couramment utilisé et reste généralement exploitable, malgré sa possible inadéquation théorique.

On remarquera que contrairement à la plupart des publications de la littérature, nous avons introduit dans les 3 derniers noyaux un facteur de normalisation dépendant de la dimension d des exemples. Celle-ci, suggérée par Schölkopf et al. [208], permet de compenser l'influence de la dimension sur les paramètres du noyau (σ, c, ou encore θ ; la détermination de ces paramètres sera abordée dans la section 4.2).

Il est également possible de construire des noyaux à partir de noyaux existants. On peut en effet montrer [212][230] que :

- Toute combinaison linéaire positive de noyaux $(k_i)_{i=1,\ldots,N}$ est un noyau :
  $k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{N} \alpha_i k_i(\boldsymbol{x}, \boldsymbol{y}) \qquad \alpha_i > 0 \; \forall i$
- Tout produit de noyaux $(k_i)_{i=1,\ldots,N}$ est un noyau :
  $k(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{N} k_i(\boldsymbol{x}, \boldsymbol{y})$

3.5.4 Machines à Vecteurs de Support non-linéaires

L'introduction du *kernel trick* laisse le problème d'optimisation dual inchangé (voir équation 3.15) :

$$\begin{array}{rl} \text{maximiser} & L_D(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T \mathbf{1} - \dfrac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{H} \boldsymbol{\alpha} \\ \text{sous les contraintes} & \alpha_i \geq 0, \quad i = 1, \ldots, n \\ \text{et} & \boldsymbol{\alpha}^T \boldsymbol{y} = 0. \end{array} \tag{3.25}$$

Seule la matrice $\boldsymbol{H}$ est modifiée pour substituer aux produits scalaires la fonction noyau : $[\boldsymbol{H}]_{ij} = y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. L'usage du noyau n'introduit donc aucune complexification de la méthode de résolution du problème. La fonction de décision prend la forme suivante :

$$f(\boldsymbol{x}) = \sum_{i \in \mathcal{S}_{SV}} \alpha_i \, y_i \, k(\boldsymbol{x}_i, \boldsymbol{x}). \tag{3.26}$$

L'adjonction du noyau apporte une grande souplesse aux Machines à Vecteurs de Support. La transformation implicite dans un espace à haute dimension élargit considérablement le champ des surfaces de séparation applicables tout en maintenant un contrôle sur le risque structurel (les considérations sur la dimension VC d'un hyperplan présentée plus haut s'appliquant également dans l'espace transformé).

Toutefois, le formalisme présenté jusqu'ici suppose la séparabilité des données, nécessaire pour définir la marge M (équation 3.4), qui permet de déduire le problème d'optimisation sur le vecteur normal $\boldsymbol{w}$. De plus, la présence éventuelle d'un exemple mal étiqueté a un impact considérable sur l'hyperplan de séparation. Nous présentons dans la section suivante le concept des hyperplans à marge souple (*soft margin*) qui permettent de s'affranchir de la contrainte de séparabilité.

3.6 Machines à Marge souple

Afin de relâcher les contraintes du problème (équation 3.6), Vapnik et Cortes introduisent [54] les *variables d'écart* positives $\xi_i \geq 0$, introduites par Smith en 1968 [219] et reprises par Bennett et Mangasarian [23] pour la résolution du problème de séparation par programmation linéaire. Les contraintes *relâchées* deviennent donc

$$\begin{array}{l} y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b \right) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{array} \qquad i = 1, \ldots, n. \tag{3.27}$$

Un exemple est donc mal classifié si $\xi_i > 1$, puisqu'il se situe alors du mauvais côté de l'hyperplan de séparation. Ainsi, pour des valeurs arbitraitement grandes de ξ_i, les contraintes peuvent être toujours respectées. Cependant, afin de minimiser l'erreur de classification, il convient d'ajouter à la fonction objectif une pénalité sur les variables d'écart dans le problème :

$$\min \frac{1}{2} ||\boldsymbol{w}||^2 + C \left(\sum_{i=1}^{n} \xi_i \right)^k, \tag{3.28}$$

où l'on introduit le *facteur d'erreur* $C > 0$ ajustant le compromis entre les deux critères. Le choix de l'exposant k implique diverses formes de pénalisation. Par exemple $k = 0$ implique une pénalité basée sur le nombre d'exemples hors de la marge. Seuls les cas $k = 1$ et $k = 2$ évitent de rendre le problème NP-complet [54] en conservant une structure de programme quadratique. On peut montrer que dans le cas $k = 2$ ($L2$ SVM), le problème dual prend la forme :

$$\begin{array}{ll} \max\limits_{\boldsymbol{\alpha} \in \mathbb{R}^n} & \boldsymbol{\alpha}^T \mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T \left(\boldsymbol{H} + \frac{1}{C}\boldsymbol{I} \right) \boldsymbol{\alpha} \\ \text{avec} & \boldsymbol{\alpha}^T \boldsymbol{y} = 0 \;\text{ et }\; 0 \leq \alpha_i \qquad i = 1, \ldots, n \end{array} \qquad L2 \text{ SVM}, \tag{3.29}$$

ce qui revient à appliquer le même algorithme que dans le cas séparable en ajoutant la constante $\frac{1}{C}$ aux termes diagonaux de la matrice de Gram du problème.

Cependant, on précédent généralement les $L1$ SVM ($k = 1$) qui conservent la forme originale du Lagrangien en imposant une nouvelle contrainte :

$$\begin{array}{ll} \max\limits_{\boldsymbol{\alpha}\in\mathbb{R}^n} & \boldsymbol{\alpha}^T\mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{H}\boldsymbol{\alpha} \\ \text{avec} & \boldsymbol{\alpha}^T\boldsymbol{y} = 0 \;\text{ et }\; 0 \leq \alpha_i \leq C \qquad i = 1, \ldots, n \end{array} \qquad L1 \text{ SVM.} \tag{3.30}$$

Dans ce cas, la variable C vient borner les multiplicateurs de Lagrange α_i. À nouveau, seuls les vecteurs de support ($\alpha_i > 0$) interviennent dans la fonction de décision, mais ceux-ci incluent désormais des exemples se situant hors de la marge ($\alpha_i = C$), correspondant aux valeurs non nulles des variables d'écart ($\xi_i > 0$) ; on parle alors de *marge souple*. Si la résolution du problème de maximisation s'en trouve modifiée, la solution demeure identique au cas séparable (équation 3.12) :

$$\boldsymbol{w} \;=\; \sum_{i\in\mathcal{S}_{SV}} \alpha_i y_i \Phi(\boldsymbol{x}_i). \tag{3.31}$$

Le choix de la constante C reste une question ouverte, cette dernière n'offrant malheureusement pas d'interprétation intuitive. Nous traiterons ce point dans le chapitre suivant en section 4.5.

Les ν-SVM ont été proposés comme alternative [211], impliquant à la place de C une variable ν plus intuitive permettant de contrôler le nombre de vecteurs de support, mais elles ne seront pas abordés dans ce document.

3.7 Méthodes à noyaux

On peut résumer les Machines à Vecteurs de Support comme la conjonction des trois points suivants :

- **Le principe de Maximisation de la Marge**, qui est choisi pour respecter le paradigme de Minimisation du Risque Structurel, implique une « éparsification » du problème en ne faisant intervenir dans le processus de décision que les exemples (les Vecteurs de Support) les plus porteurs d'information.
- **La fonction noyau**, se substituant aux produits scalaires, permet de transformer implicitement les exemples dans un espace de dimension supérieure où la surface de décision linéaire se traduit dans l'espace d'entrée par une surface beaucoup plus complexe.
- **L'utilisation des variables d'écart** permet de relâcher les contraintes et d'autoriser la prise en compte d'exemples mal classifiés, supprimant toute contrainte sur la répartition des exemples d'apprentissage.

Le champ d'application de ces principes ne se limite pas à la classification supervisée. Ils peuvent en fait s'appliquer aux trois problèmes fondamentaux de l'apprentissage statistique, énoncés par Vapnik [236] :

- La **reconnaissance de formes**, par le biais des SVM.
- La **régression** pour l'estimation de fonction [234][236].
- L'**estimation de densité** [209]. On trouve également ce problème sous le nom de classification à une classe dans la littérature, souvent exploitée pour la détection de nouveauté [214]. Nous exploiterons cette dernière dans la section 9.5.

L'utilisation de noyaux a également permis de « kerneliser » d'autres algorithmes exclusivement formulés en termes de produits scalaires :

- L'**Analyse en Composantes Non-Linéaires** (ou *Kernel PCA*) [210]
- L'**Analyse Discriminante de Fisher Kernelisée** (*Kernel FDA*) [158][203][19], que nous aborderons dans la section 7.5.5.

On pourra consulter [163] et [22] pour une vue d'ensemble des algorithmes liés à l'usage des noyaux.

3.8 Une méthode universelle d'apprentissage

On voit, au regard des trois principes énoncés dans le paragraphe précédent, ce qui pousse Vapnik à qualifier les SVM de « méthode universelle d'apprentissage » [236]. En effet, loin d'être un énième algorithme de reconnaissance des formes, les SVM apportent la rigueur de l'approche statistique, par le biais de la minimisation de bornes sur le risque, à une multitudes d'algorithmes existants, dont le comportement est « simulé » par le choix de la fonction noyau.

Ainsi on a vu que le noyau polynômial constitue une implémentation implicite des classifieurs polynômiaux. On peut montrer également que le noyau gaussien RBF simule la classification par réseaux de fonctions RBF (on pourra trouver une comparaison des deux approches dans l'article de Schölkopf et al. [213]). Enfin, Le noyau sigmoïdal reprend la fonction de décision d'un réseau de neurones à deux couches [236]. Pour chacun de ces cas, le principe des SVM améliore l'approche originale en y adjoignant la mise en évidence d'un ensemble restreint d'exemples (les vecteurs de support) exprimant à eux seuls la complexité du problème [208]. Plusieurs auteurs ont d'ailleurs montré l'équivalence entre la résolution par SVM linéaire et l'Analyse Discriminante Linéaire opérée sur le sous-ensemble des vecteurs de support [216][98][123].

De plus, le procédé d'optimisation permet de s'affranchir des affinages empiriques, souvent employés dans les techniques traditionnelles. Ainsi le choix de la structure des réseaux de neurones (nombre de couches et de neurones) reste l'une des carences principale de cette approche, tandis qu'elle se trouve implicitement déterminée lors de la phase d'apprentissage par SVM avec noyau sigmoïdal ; de même pour la détermination des centres dans la classification par fonctions RBF (généralement évalués par des algorithmes de clustering, type K-means).

Enfin, il est important de souligner que le problème de maximisation posé par l'équation 3.30 est convexe et implique donc la convergence vers un maximum global unique [40], contrairement à beaucoup d'autres approches, comme les réseaux de neurones, qui ne garantissent que la convergence vers un optimum local.

Toutefois, les SVM présentent deux problèmes pour leur mise en application :

1. Nous avons vu que les noyaux apportent une souplesse considérable au processus d'apprentissage. Néanmoins la question du choix d'un noyau optimal est fondamentale et loin d'être évidente ; de plus, si le champ des paramètres est considérablement restreint chaque noyau comporte généralement une ou deux variables qu'il faut déterminer. Ces questions sont abordées dans le chapitre 4.
2. Dans leur construction, les SVM sont une méthode discriminative. Il nous faut donc déterminer une stratégie permettant d'étendre leur champ d'application aux cas multi-classes. Ce point est traité dans le chapitre 5.

Chapitre 4

Sélection du noyau

Sommaire

4.1 Illustration sur des données artificielles 46
4.1.1 Surface de décision 46
4.1.2 Tolérance aux *outliers* 47
4.2 Stratégies d'affinage 48
4.2.1 Recherche par maillage 48
4.2.2 Optimisation 49
4.2.3 Recherche par voisinage 50
4.3 Critères d'évaluation basés sur l'erreur de généralisation 50
4.3.1 Erreur sur un ensemble de validation 50
4.3.2 Validation croisée 50
4.3.3 Erreur *Leave-One-Out* 51
4.3.4 Nombre de Vecteurs de Support 51
4.3.5 Estimée $\xi\alpha$ 52
4.3.6 Borne Rayon-Marge 52
4.3.7 Borne sur l'étendue 52
4.4 Critères basés sur la séparation de classes 53
4.4.1 Critère d'Alignement 53
4.4.1.1 Interprétation géométrique 54
4.4.1.2 Fenêtres de Parzen 55
4.4.1.3 Dérivation 57
4.4.1.4 Critique de l'Alignement 57
4.4.2 Séparabilité dans l'espace Transformé (KCS) 58
4.5 Facteur d'erreur C 59
4.5.1 Valeur de Joachims 60
4.5.2 Inclusion du facteur C dans les critères 62
4.6 Evaluation des critères de sélection de noyau 63

La question du choix du noyau est un point essentiel de l'apprentissage par Machines à Vecteurs de Support. En effet, la fonction noyau détermine le champ des surfaces de décision possibles et, comme nous l'avons vu, elle implique l'utilisation d'une technique de classification sous-jacente. Nous donnerons dans cette section une vue d'ensemble des possibilités qu'offrent les noyaux et présenterons diverses stratégies d'affinage ou de sélection de noyau pour une tâche donnée. La plupart des noyaux, excepté le noyau linéaire, incluent un ou plusieurs paramètres dans leur expression, appelés *hyper-paramètres*. Nous traiterons en section 4.2 des stratégies de détermination des valeurs optimales de ces paramètres. Un état de l'art des critères les plus courants pour estimer l'erreur *Leave-one-out*, en section 4.3, nous permettra d'introduire les notions nécessaires et de comparer ces derniers à deux récents critères de la littérature, basés sur la maximisation de la séparabilité des classes, dont nous proposons dans la section 4.4 l'application pour la première fois dans le domaine de l'indexation audio.

Nous porterons en outre une attention au facteur d'erreur C, qui fixe le compromis entre la pénalisation des erreurs et la minimisation de l'erreur de généralisation dans les Machines à Marge Souple. Nous aborderons par la suite les aspects propres à ce paramètre en section 4.5, que nous exploiterons pour proposer des améliorations sur les critères de séparabilité des classes introduits.

4.1 Illustration sur des données artificielles

Nous allons montrer l'importance de l'affinage des hyper-paramètres à travers une courte étude sur des données artificielles à deux dimensions. Cette étude portera sur le paramètre σ du noyau RBF gaussien, parce que les noyaux sigmoïdes sont très instables (le noyau n'est pas positif pour toutes les valeurs de ses paramètres), et les noyaux polynomiaux ont intérêt limité sur deux dimensions. On s'intéressera donc à des problèmes définis sur des variables bi-dimensionnelles $\boldsymbol{x} = [x_1, x_2]$. Dans le cas de données artificielles séparables, on pourra définir une fonction de décision *idéale* $f_I(\boldsymbol{x})$ telle que $y = f_I(\boldsymbol{x})\ \forall \boldsymbol{x}$, où y est la classe associée à l'exemple $\boldsymbol{x}$.

4.1.1 Surface de décision

Le problème de l'échiquier constitue un très bon exemple de données artificielles linéairement inséparables. Si l'on fixe à N_C le nombre de case par côté, la fonction de décision idéale $f_I : \mathbb{R}^2 \rightarrow [0;1] \times [0;1]$ est définie comme suit :

$$f_I(x_1, x_2) = \text{sign}(\text{mod}(\lfloor N_C(x_1 + x_2) \rfloor, 2)),$$

où $\text{mod}(a, b)$ est le reste de la division entière de a par b, et l'opérateur $\lfloor x \rfloor$ désigne l'arrondi par défaut. La figure 4.1 montre la fonction de décision idéale pour un échiquier à $N_C = 3$ cases de largeur, ainsi que 250 exemples générés aléatoirement. Les croix claires désignent les exemples de la classe 1, symbolisée par les régions noires pour la fonction de décision, tandis que les cercles foncés désignent les exemples de la classe 2, symbolisée par les régions blanches pour la fonction de décision.

Les figures 4.2(a), (b) et (c) illustrent le résultat de l'apprentissage par SVM avec noyau gaussien RBF pour différentes valeurs de σ. La ligne pleine blanche représente la surface de séparation et les pointillés gris représentent les surfaces de marge pour chaque classe. Les Vecteurs de Support sont donc les exemples situés sur ces lignes grises.

Si l'on rappelle l'expression du noyau gaussien RBF,

$$k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{y}||^2}{d\,\sigma^2}\right),$$

celui-ci traduit bien une mesure de similarité basée sur la distance entre les exemples. Lorsque σ augmente, le terme sur lequel s'applique l'exponentielle inverse tend vers 0, ce qui accroît la similarité entre les exemples. À l'inverse, lorsque l'on réduit le terme σ, la distance entre les exemples se trouve amplifiée.

Ce phénomène apparaît clairement dans l'exemple présent. Lorsque σ est trop bas (figure 4.2(a)), les exemples sont plus distants entre eux et la fonction de décision doit donc inclure plus de Vecteurs de Support pour pouvoir couvrir tout l'espace. On voit en effet sur la figure que de

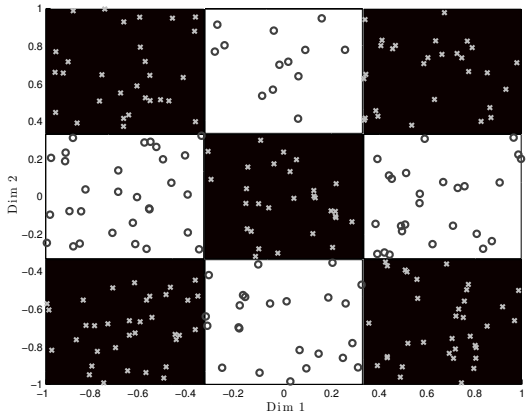
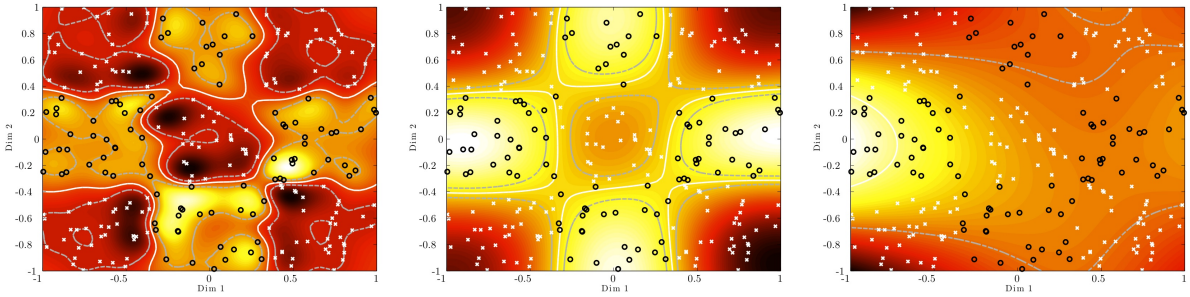


FIGURE 4.1 – Fonction de décision idéale de la distribution *Echiquier* et répartition des 100 exemples d'apprentissage.



(a) $\sigma = 0.2, C = 100$ (b) $\sigma = 1, C = 100$ (c) $\sigma = 3, C = 100$

FIGURE 4.2 – Surfaces de décision obtenues par apprentissage SVM à noyau gaussien RBF pour différentes valeurs de σ sur la distribution *Echiquier*.

nombreux exemples figurent sur la surface de marge, qui se trouve ainsi trop adaptée à l'ensemble d'apprentissage, au détriment de la capacité de généralisation du classificateur. On parle alors de sur-apprentissage (ou d'*overfitting*).

Lorsque, par contre, σ est trop élevé (figure 4.2(c)), la mesure de similarité ne permet plus de distinguer les exemples de classes opposées. Ainsi l'algorithme est incapable de déduire une surface de décision traduisant la frontière entre les classes.

La figure centrale (4.2(b)) présente dans ce cas un bon compromis entre généralisation et erreur de classification.

4.1.2 Tolérance aux *outliers*

On utilise ici une distribution plus simple, nommée *Courbe* et représentée figure 4.3, pour illustrer l'influence du paramètre C par rapport à la présence d'*outliers*. 100 exemples sont générés aléatoirement, dont 8 *outliers* sont assignés à la classe erronée. La figure 4.4 montre les résultats de l'apprentissage par SVM avec noyau gaussien RBF pour différentes combinaisons de valeurs pour σ et C.

On peut ainsi constater qu'une valeur trop basse du facteur d'erreur C (figures (a) à gauche) induit une trop grande tolérance aux erreurs de classification. Ainsi le classificateur peine à établir une surface de décision qui réponde au problème posé puisque les exemples peuvent se situer indifféremment d'un côté ou de l'autre de la surface de décision.

A l'inverse, une valeur trop élevée de C (figures (c) à droite) pousse le classificateur à prendre en compte le moindre exemple erroné (dans la mesure où le paramètre σ lui permet de complexifier la surface de décision, ce qui n'est pas le cas par exemple pour la figure (c,3) où $\sigma = 6$). La comparaison des figures (c,1) et (c,2) illustre clairement la tendance à « encercler » chaque *outlier*

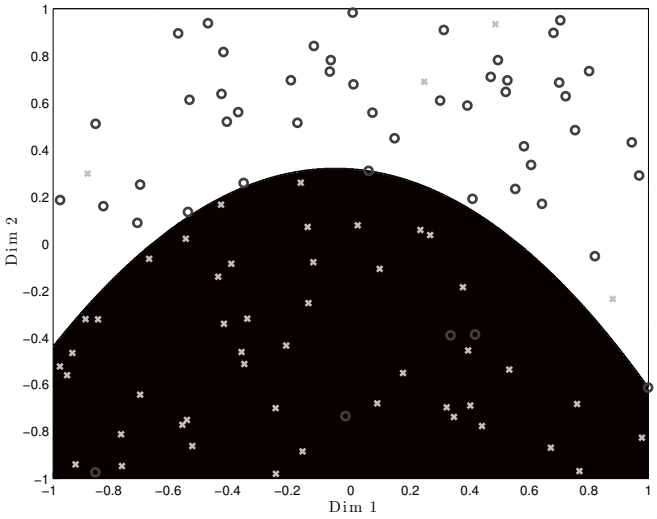


FIGURE 4.3 – Fonction de décision idéale de la distribution *Courbe* et répartition des 100 exemples d'apprentissage.

pour C trop élevé.

Dans le cas étudié, $C = 2$ (figures (b) au centre) constitue un bon compromis. Néanmoins, on voit que cette valeur n'a de sens que pour un σ adéquat. Les paramètres sont donc fortement liés entre eux, ce qui explique la complexité de l'affinage des noyaux puisque les paramètres ne peuvent être affinés indépendamment.

Nous présentons dans la section suivante les stratégies possibles pour l'affinage des paramètres, ainsi que les critères employés. On reviendra plus précisément dans la section 4.5 sur la question du facteur d'erreur C, et sa relation au paramètre σ.

4.2 Stratégies d'affinage

On aborde ici le problème de l'affinage des hyper-paramètres indépendamment de leur nature et de leur nombre. On considère donc qu'un noyau $k_{\mathbf{\Theta}}$ est paramétré par P valeurs $\mathbf{\Theta} = [\theta_1, \ldots, \theta_P]$.

4.2.1 Recherche par maillage

La recherche par maillage (*grid-search*) est la méthode la plus généralement employée. Elle consiste à évaluer les performances du classifieur SVM appris sur un ensemble fini de V valeurs $\mathcal{V} = \{\mathbf{\Theta}_i, \quad i \in [1, \ldots, V]\}$. Soit $\mathcal{P}(k_{\mathbf{\Theta}})$ la mesure de performances du noyau $k_{\mathbf{\Theta}}$, l'algorithme consiste donc à retenir la valeur $\hat{\mathbf{\Theta}}$ telle que

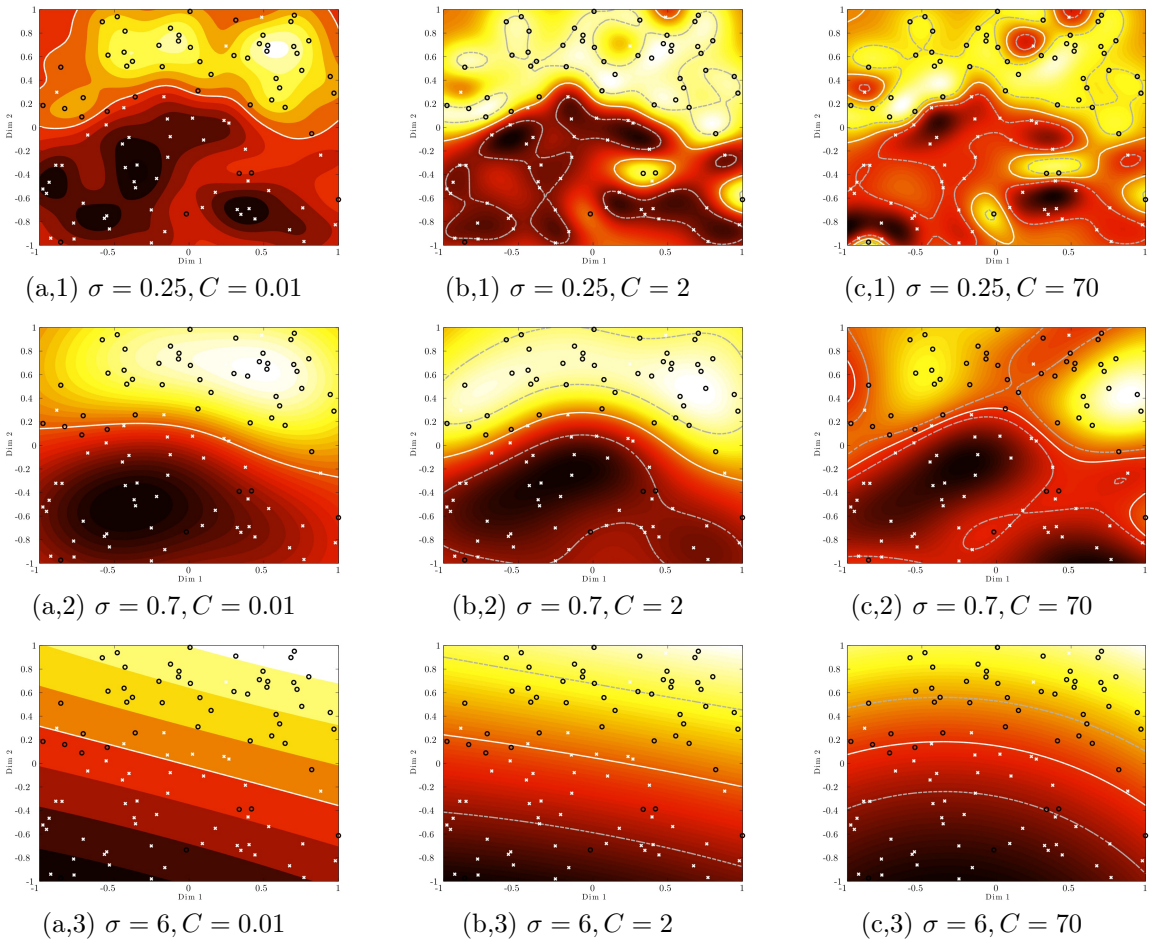
$$\hat{\mathbf{\Theta}} = \arg\max_{\mathbf{\Theta} \in \mathcal{V}} \mathcal{P}(k_{\mathbf{\Theta}}).$$

Concernant le choix des valeurs de l'ensemble $\mathcal{V}$, on utilise généralement pour chaque paramètre un ensemble de valeurs également réparties dans un intervalle donné. $\mathcal{V}$ est alors le produit cartésien de ces ensembles et constitue un *maillage* de l'espace des paramètres sur un intervalle donné. Il est également courant d'utiliser des valeurs logarithmiquement réparties.

Nous détaillerons par la suite (section 4.3) la plupart des critères permettant d'évaluer la mesure de performance d'un classifieur SVM. La solution la plus basique et la plus généralement employée consiste à mesurer le taux d'erreur sur un ensemble dit de *validation*.

La recherche par maillage souffre ainsi de deux défauts majeurs :

- La complexité algorithmique est polynomiale, en $O(n^P)$. On fait donc face à une explosion combinatoire dès que le nombre de paramètres dépasse 1 ou 2.



(a,1) $\sigma = 0.25, C = 0.01$ (b,1) $\sigma = 0.25, C = 2$ (c,1) $\sigma = 0.25, C = 70$

(a,2) $\sigma = 0.7, C = 0.01$ (b,2) $\sigma = 0.7, C = 2$ (c,2) $\sigma = 0.7, C = 70$

(a,3) $\sigma = 6, C = 0.01$ (b,3) $\sigma = 6, C = 2$ (c,3) $\sigma = 6, C = 70$

FIGURE 4.4 – Surfaces de décision obtenues par apprentissage SVM à noyau gaussien RBF pour différentes valeurs de σ et C sur la distribution *Courbe*.

- En supposant que le critère de performance est fiable, on n'a aucune garantie que le maximum global (ou qu'un maximum local) se trouve dans le champ du maillage. Un maillage suffisamment serré couplé à un critère régulier résoud ce problème, mais au prix possible d'une explosion combinatoire en présence de nombreux paramètres.

4.2.2 Optimisation

Une autre stratégie consiste, plutôt que de tenter de couvrir l'espace de recherche, à évaluer itérativement la valeur optimale par mises à jour successives en fonction d'un critère de performance. Cette approche sollicite donc des méthodes d'optimisation, domaine très large couvrant le problème de la recherche d'extrema. Les méthodes de programmation mathématique (linéaire, quadratique, etc...) ne peuvent être utilisées ici puisque le problème n'est pas exprimé analytiquement, tandis que les algorithmes dérivés de la descente de gradient sont applicables. De nombreuses approches de ce type existent dans la littérature [26], parmi lesquelles la Méthode de Newton, la descente de gradient stochastique ou le gradient conjugué. Néanmoins, une étude des méthodes d'optimisation sort du domaine de cette thèse et nous nous restreindrons à l'usage de la plus simple, la descente (ou montée) de gradient. On peut résumer celle-ci par l'algorithme 1 présenté ci-dessous.

La condition d'arrêt peut également s'appliquer à la différence de performance entre deux itérations. La descente de gradient implique de pouvoir calculer le gradient de la mesure de performance par rapport aux paramètres du noyau. C'est là la principale restriction liée à cette méthode.

Algorithme 1 Optimisation

$k_{\boldsymbol{\Theta}}$ le noyau paramétré par ...
$\boldsymbol{\Theta}$, de valeur initiale $\boldsymbol{\Theta}_0$.
$\mathcal{P}(k_{\boldsymbol{\Theta}})$ mesure de performance sur le noyau $k_{\boldsymbol{\Theta}}$
λ pas d'avancement
ϵ paramètre d'arrêt

répéter
  Estimer le gradient $\Delta_{\boldsymbol{\Theta}}\mathcal{P}(k_{\boldsymbol{\Theta}})$
  $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} + \lambda \cdot \Delta_{\boldsymbol{\Theta}}\mathcal{P}(k_{\boldsymbol{\Theta}})$
jusqu'à $\Delta_{\boldsymbol{\Theta}}\mathcal{P}(k_{\boldsymbol{\Theta}}) \leq \epsilon$

4.2.3 Recherche par voisinage

Momma et Bennett proposent [161] pour l'affinage de paramètres une recherche par pas successifs qui constitue un compromis intéressant entre optimisation et recherche par maillage pour le cas de critères non dérivables. A chaque itération k le critère est évalué sur des points voisins de la valeur actuelle, dont la disposition forme ce que les auteurs appellent un *pattern*, et le point actuel est déplacé au point voisin minimisant le critère, jusqu'à convergence. Le *pattern* le plus simple consiste à sélectionner les points à une distance Δ_k, décroissante, pour chacune des directions axiales de l'espace.

Ce critère permet de s'affranchir de la contrainte de dérivabilité et restreint fortement l'espace de recherche par rapport à la recherche par maillage. Toutefois, il reste très coûteux lorsque l'espace est de dimension élevée, et n'offre aucune garantie quant à la globalité du maximum trouvé.

En pratique on préférera se limiter ici à des critères dérivables permettant l'usage de la descente de gradient.

4.3 Critères d'évaluation basés sur l'erreur de généralisation

Nous présentons dans cette section la plupart des critères couramment utilisés par l'évaluation des noyaux. Ceux-ci consistent en général en l'expression d'une borne supérieure sur l'erreur *Leave-One-Out*. On trouvera dans [47], [90] et [66] une vue d'ensemble assez didactique des critères qui suivent.

4.3.1 Erreur sur un ensemble de validation

Si l'on dispose de suffisamment de données, la solution la plus simple consiste à estimer le taux d'erreur d'une machine SVM sur un ensemble de validation, dont les exemples sont distincts de ceux de l'ensemble d'apprentissage. Soit un ensemble de validation de p éléments $\mathcal{S}_V = \{(\boldsymbol{x}'_i, y'_i)\}_{i\in[1,\dots,p]}$, on définit l'erreur par :

$$\mathcal{P}_{val} = \frac{1}{p}\sum_{i=1}^{p} H\left(-y'_i\, f(\boldsymbol{x}'_i)\right),$$

où H est la fonction de Heaviside ($H(x) = 1$ si $x > 0$, et $H(x) = 0$ sinon).

Ce critère est simple mais il restreint le volume de données d'apprentissage et suppose que l'ensemble de validation est caractérisé par la même distribution sous-jacente, ce qui peut être difficile à valider sur des données réelles. On a donc peu de garanties sur l'absence de biais du critère.

4.3.2 Validation croisée

La validation croisée est une variante plus robuste du critère précédent. Les exemples d'un ensemble de base sont partagés en k sous-ensembles répartis aléatoirement. Chacun des k sous-ensembles est utilisé comme ensemble de validation pour calculer l'erreur d'un classifieur appris

sur l'union des $k-1$ autres sous-ensembles. Le critère de validation croisée est la moyenne des erreurs obtenues sur les k itérations.

Cette démarche permet ainsi de réduire le biais possible entre les ensembles d'apprentissage et de validation, mais accroît le temps de calcul d'un facteur k.

4.3.3 Erreur *Leave-One-Out*

L'erreur *Leave-One-Out* (ou LOO), littéralement « un laissé dehors », peut être vue comme une validation croisée poussée à l'extrême. Elle consiste à évaluer le taux d'erreur en classifiant chaque exemple $\boldsymbol{x}_i$ de l'ensemble $\mathcal{S}$ par le classifieur SVM appris sur l'ensemble $\mathcal{S}^{\setminus \boldsymbol{x}_i}$ comprenant tous les autres exemples, soit :

$$\mathcal{P}_{LOO} = \frac{1}{n} \sum_{i=1}^{n} H\left(-y_i\, f^i(\boldsymbol{x}_i)\right),$$

où f^i est la fonction de décision du classifieur SVM appris sur l'ensemble $\mathcal{S}^{\setminus \boldsymbol{x}_i}$.

L'estimation de l'erreur LOO est connue pour être presque non biaisée [142] (le presque faisant référence au fait que l'erreur porte sur $n-1$ échantillons au lieu de n) mais celui-ci est extrêmement coûteux puisqu'il nécessite à priori l'apprentissage de n classifieurs. Si l'on appelle f^0 le classifieur appris sur tous les exemples, on a :

$$\begin{aligned} \mathcal{P}_{LOO} &= \frac{1}{n} \sum_{i=1}^{n} H\left(-y_i\, f^0(\boldsymbol{x}_i) + y_i \left[f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i)\right]\right) && (4.1) \\ &= \frac{1}{n} \operatorname{Card}\left\{i, y_i \left[f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i)\right] > y_i\, f^0(\boldsymbol{x}_i)\right\}. && (4.2) \end{aligned}$$

Cette expression permet, dans le cas des SVM, de définir une borne supérieure à l'erreur LOO en bornant l'expression $y_i \left[f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i)\right]$. On peut remarquer que l'exclusion d'un vecteur non-support de l'ensemble d'aprentissage ne modifie pas la fonction de décision, d'où $f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i) = 0$ pour $\boldsymbol{x}_i \notin \mathcal{S}_{SV}$ (où $\mathcal{S}_{SV}$ est l'ensemble des vecteurs de support pour le classifieur f^0). Seul l'apprentissage des classifieurs f^i avec $\boldsymbol{x}_i \in \mathcal{S}_{SV}$ est donc nécessaire pour calculer l'erreur *Leave-One-Out*.

Malgré ce constat, le calcul de l'erreur LOO demeure prohibitif. De nombreuses méthodes [119][64][173][110] ont été suggérées qui permettent d'en alléger le calcul, généralement en bornant à l'excès l'expression $y_i \left[f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i)\right]$. Nous détaillons par la suite quelques-unes des plus courantes.

4.3.4 Nombre de Vecteurs de Support

Dans le cas de SVM à marge dure, le premier terme de la fonction de Heaviside dans l'équation 4.1 est borné supérieurement puisque $y_i\, f^0(\boldsymbol{x}_i) \geq 1$. La fonction de Heaviside étant croissante monotone, on a donc :

$$\mathcal{P}_{LOO} \leq \frac{1}{n} \sum_{i=1}^{n} H\left(y_i \left[f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i)\right] - 1\right).$$

Nous avons vu que $f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i) = 0$ pour tout vecteur « non support », on peut donc restreindre la somme précédente aux vecteurs de support :

$$\mathcal{P}_{LOO} \leq \frac{1}{n} \sum_{i/\boldsymbol{x}_i \in \mathcal{S}_{SV}} H\left(y_i \left[f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i)\right] - 1\right).$$

On peut ainsi approximer grossièrement l'erreur LOO en bornant chaque fonction de Heaviside par 1, ce qui donne l'estimée $\mathcal{P}_{NSV}$ (pour *Number of Support Vectors*) :

$$\mathcal{P}_{LOO} \leq \mathcal{P}_{NSV} = \frac{\operatorname{Card} \mathcal{S}_{SV}}{n},$$

où $\operatorname{Card} \mathcal{S}_{SV}$ est le nombre de vecteurs de support. Cette estimation de l'erreur est très simple dans sa formulation mais discontinue par rapport aux paramètres, ce qui empêche l'application de méthodes d'optimisation usuelles sur ce critère.

4.3.5 Estimée $\xi\alpha$

Joachims [115] fournit une borne supérieure à l'erreur LOO ne dépendant que des variables calculées durant l'apprentissage des SVM. Il montre en effet que

$$\mathcal{P}_{LOO} \leq \mathcal{P}_{\xi\alpha} = \frac{1}{n} \operatorname{Card} \left\{ i, 2\alpha_i R^2 + \xi_i \geq 1 \right\},$$

où R est une estimation du rayon minimal de la sphère contenant tous les exemples dans l'espace transformé (on trouvera dans l'annexe A les différentes techniques d'estimation du rayon R). Les grandeurs α_i et ξ_i sont respectivement les facteurs de Lagrange et les variables d'écarts définis dans le problème d'optimisation des SVM à marge souple ; leur calcul n'induit pratiquement aucun coût supplémentaire après apprentissage d'un SVM. Toutefois, comme pour $\mathcal{P}_{NSV}$, ce critère est un dénombrement et n'est pas dérivable.

4.3.6 Borne Rayon-Marge

Il est également montré [230] que dans le cas de données séparables, l'erreur de généralisation (qui est estimée de manière presque non-biaisée par l'erreur LOO) est bornée par la valeur suivante :

$$\mathcal{P}_{RM} = \frac{1}{n} \frac{R^2}{M^2} = \frac{1}{n} R^2 \|\boldsymbol{w}\|^2,$$

où R est le rayon défini précédemment, M est la marge du classifieur SVM, et $\boldsymbol{w}$ le vecteur normal de l'hyperplan de séparation. Cette borne est déduite d'une majoration par x de la fonction de Heaviside $H(x-1)$, qui permet de supprimer les discontinuités. Le critère $\mathcal{P}_{RM}$ appelé borne Rayon-Marge (*Radius-Margin bound*) est donc dérivable. Soit un paramètre à P composantes $\boldsymbol{\Theta} = [\theta_1 \dots \theta_P]$,

$$\frac{\partial R^2 \|\boldsymbol{w}\|^2}{\partial \boldsymbol{\Theta}} = R^2 \frac{\partial \|\boldsymbol{w}\|^2}{\partial \boldsymbol{\Theta}} + \|\boldsymbol{w}\|^2 \frac{\partial R^2}{\partial \boldsymbol{\Theta}}.$$

On déduit immédiatement de l'expression de $\boldsymbol{w}$ (éq. 3.31), celle de la dérivée de $\|\boldsymbol{w}\|^2$:

$$\frac{\partial \|\boldsymbol{w}\|^2}{\partial \boldsymbol{\Theta}} = -\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \frac{\partial k(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{\Theta}}.$$

L'estimation du rayon R est traitée dans l'annexe A. On suppose que celui-ci peut s'exprimer en fonction des exemples, par la combinaison linéaire suivante :

$$R^2 = \sum_{i=1}^{n} \beta_i k(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i,j=1}^{n} \beta_i \beta_j k(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{4.3}$$

On en déduit :

$$\frac{\partial R^2}{\partial \boldsymbol{\Theta}} = \sum_{i=1}^{n} \beta_i \frac{\partial k(\boldsymbol{x}_i, \boldsymbol{x}_i)}{\partial \boldsymbol{\Theta}} - \sum_{i,j=1}^{n} \beta_i \beta_j \frac{\partial k(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{\Theta}}.$$

Ce critère est très utilisé dans la littérature et permet d'affiner les paramètres avec précision. Cependant la majoration de la fonction de Heaviside par x accentue la contribution des erreurs importantes. De plus, on rappelle que même si le critère se comporte bien de façon générale il repose sur la supposition que les données d'apprentissage sont séparables, ce qui remet en question sa pertinence théorique dans le cas de données non séparables (ce qui est généralement le cas).

4.3.7 Borne sur l'étendue

Vapnik et Chapelle [46] [231] définissent le concept d'*étendue* (*span*) d'un vecteur support $\boldsymbol{x}_p$, comme la distance (dans l'espace transformé) entre celui-ci et l'espace Λ_p :

$$S_p = d(\Phi(\boldsymbol{x}_p), \Lambda_p) = \min_{\boldsymbol{x} \in \Lambda_p} (\Phi(\boldsymbol{x}_p), \Phi(\boldsymbol{x})),$$

où Λ_p est défini comme un sous ensemble de l'espace engendré par les autres vecteurs de support, délimité par une contrainte additionnelle :

$$\Lambda_p = \left\{ \sum_{i \neq p, \alpha_i > 0} \lambda_i \Phi(\boldsymbol{x}_i), \quad \sum_{i \neq p} \lambda_i = 1 \right\}.$$

Le *span* est ainsi une mesure de la distance entre un vecteur support et les autres. Intuitivement, plus cette mesure est réduite, et moins la procédure de *Leave-One-Out* sur cet exemple est susceptible de produire une erreur. On peut montrer [231] que si l'ensemble des vecteurs de support ne change pas entre les classifieurs f^0 et f^p (on reprend ici les notations de la section 4.3.3), alors

$$y_p \left(f^0(\boldsymbol{x}_p) - f^p(\boldsymbol{x}_p) \right) = \alpha_p S_p^2.$$

La borne sur l'étendue se déduit donc de l'équation 4.1 :

$$\begin{aligned} \mathcal{P}_{span} &= \frac{1}{n} \sum_{p,\, \boldsymbol{x}_p \in \mathcal{S}_{SV}} H\left(\alpha_p S_p^2 - y_p f^0(\boldsymbol{x}_p) \right) \\ &= \frac{1}{n} \operatorname{Card} \left\{ \alpha_p S_p^2 > y_p f^0(\boldsymbol{x}_p \right\}. \end{aligned}$$

À noter que S_p^2 s'exprime aisément en fonction du noyau, en introduisant $\lambda_p = -1$:

$$S_p^2 = \min_{\lambda_i, i \neq p} \left\{ k \left(\sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i, \sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i \right), \quad \sum_{i=1}^{p} \lambda_i = 0 \right\}.$$

Cette borne fournit une estimation très précise de l'erreur LOO et est de loin la plus pertinente parmi celles présentées. Cependant elle présente l'inconvénient d'être discontinue par rapport aux paramètres, du fait de la présence d'une énumération. Cette contrainte est prise en compte [47] en appliquant un traitement sigmoïdal qui lisse la réponse du critère, au prix d'une complexification de ce dernier. Celui-ci est malheureusement déjà très coûteux puisqu'il implique une inversion de la matrice de Gram.

Nous n'avons donc pas exploité ce critère, malgré sa pertinence, en raison de sa trop grande complexité.

4.4 Critères basés sur la séparation de classes

Nous avons exploré durant cette thèse l'usage de critères non pas basés sur une estimation de l'erreur *Leave-One-Out* mais sur la séparabilité des classes dans l'espace transformé. Le principal critère est celui de l'Alignement du noyau que nous présentons en section 4.4.1. Nous verrons que l'expression de ce critère, construit sur une base algébrique simple, s'accompagne d'une interprétation géométrique pertinente qui fait écho au Discriminant Linéaire de Fisher, introduit dans la section 3.2. Nous présenterons par la suite d'autres critères explicitement basés sur le critère de Fisher.

4.4.1 Critère d'Alignement

On reprend ici les notations présentées dans la section 3.1 : soit un ensemble d'apprentissage $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1\ldots n}$ dont les n_1 premiers exemples appartiennent à la classe 1 ($\mathcal{S}_1 = \{(\boldsymbol{x}_i, y_i = +1)\}_{i=1,\ldots,n_1}$) et les n_2 suivants à la classe 2 ($\mathcal{S}_2 = \{(\boldsymbol{x}_i, y_i = -1)\}_{i=n_1+1,\ldots,n_n}$), et un noyau k. On pourra trouver par la suite la notation abusive n_i, où $n_i = n_1$ si $(\boldsymbol{x}_i, y_i) \in \mathcal{S}_1$ et $n_i = n_2$ si $(\boldsymbol{x}_i, y_i) \in \mathcal{S}_2$. La matrice de Gram $\boldsymbol{K}$ pour le noyau k et l'ensemble $\mathcal{S}$ est définie par $[\boldsymbol{K}]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

On définit la matrice *cible* (*target matrix*), décrivant la matrice de Gram idéale pour le problème, comme $\boldsymbol{K}^* = \boldsymbol{y}\boldsymbol{y}^T$, où $\boldsymbol{y} = [y_1, \ldots, y_n]^T$ est le vecteur des labels de classes. On peut décomposer les deux matrices introduites en blocs de classes :

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{K_{11}} & \boldsymbol{K_{12}} \\ \boldsymbol{K_{21}} & \boldsymbol{K_{22}} \end{pmatrix} \qquad \boldsymbol{K^*} = \begin{pmatrix} \mathbf{1} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{1} \end{pmatrix}, \tag{4.4}$$

où $\mathbf{1}$ est une matrice dont toutes les composantes sont égales à 1. Les dimensions de cette matrice sont implicites, si bien qu'on se passera en général d'en préciser les dimensions.

Afin d'évaluer la pertinence d'un noyau pour la tâche de classification décrite par l'ensemble d'apprentissage, Cristianini et al. [56] définissent un nouveau critère basé sur une mesure de similarité exprimée par le produit scalaire de Frobenius, qui est défini entre deux matrices $\boldsymbol{A}$ et $\boldsymbol{B}$ (de termes $[\boldsymbol{A}]_{ij} = a_{ij}$ et $[\boldsymbol{B}]_{ij} = b_{ij}$) par :

$$\langle \boldsymbol{A}, \boldsymbol{B} \rangle_F = \sum_{i,j} a_{ij} b_{ij}.$$

On pourra également utiliser une notation alternative basée sur le produit de Hadamard (terme à terme) $\bullet$ et l'opérateur $\Sigma(\boldsymbol{A}) = \sum_{i,j} a_{ij}$:

$$\langle \boldsymbol{A}, \boldsymbol{B} \rangle_F = \Sigma(\boldsymbol{A} \bullet \boldsymbol{B}).$$

Les auteurs définissent ainsi le critère d'Alignement du noyau k (que nous appellerons indifféremment Alignement ou KTA, pour *Kernel Target Alignment*) comme le produit de Frobenius normalisé entre la matrice de Gram $\boldsymbol{K}$ et la matrice cible $\boldsymbol{K}^*$:

$$\mathcal{A}(\boldsymbol{K}, \boldsymbol{K^*}) = \frac{\langle \boldsymbol{K}, \boldsymbol{K^*} \rangle_F}{\|\boldsymbol{K^*}\|_F \|\boldsymbol{K}\|_F}, \tag{4.5}$$

où $\|\cdot\|_F$ est la norme associée au produit de Frobenius. On peut remarquer que

$$\|\boldsymbol{K^*}\|_F = \sqrt{\sum_{i,j=1}^{n} 1} = n.$$

La maximisation du critère d'Alignement a donc pour but d'accroître la similarité entre la matrice de Gram et la matrice cible idéale, ce qui se traduit conjointement par l'accroissement conjoint de la similarité (mesurée par la fonction noyau) entre les exemples de même classe, et sa réduction entre les exemples de classes opposées. La normalisation du produit de Frobenius permet de soustraire au critère l'influence d'un facteur d'échelle, et restreint ainsi le critère d'Alignement à un intervalle standard :

$$-1 \leq \mathcal{A}(\boldsymbol{K}, \boldsymbol{K}^*) \leq 1. \tag{4.6}$$

Dans le cas de classes mal proportionnées ($n_1 \gg n_2$ ou inversement) on peut compenser la représentation des classes en utilisant la matrice cible *pondérée* $\hat{\boldsymbol{K}}^* = \hat{\boldsymbol{y}}\hat{\boldsymbol{y}}^{\boldsymbol{T}}$ où $\hat{y}_i = \frac{y_i}{n_i}$. On a alors :

$$\hat{\boldsymbol{K}}^* = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{y_i}{n_i}\frac{y_j}{n_j} k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{4.7}$$

$$= \begin{pmatrix} \frac{1}{n_1^2}\mathbf{1} & -\frac{n_1}{n_2}\mathbf{1} \\ -\frac{n_1}{n_2}\mathbf{1} & \frac{1}{n_2^2}\mathbf{1} \end{pmatrix} = \frac{1}{n_1 n_2} \begin{pmatrix} \frac{n_2}{n_1}\mathbf{1} & -\mathbf{1} \\ -\mathbf{1} & \frac{n_1}{n_2}\mathbf{1} \end{pmatrix} \tag{4.8}$$

$$\text{et} \quad ||\hat{\boldsymbol{K}}^*||_F = \frac{n}{n_1 n_2}. \tag{4.9}$$

4.4.1.1 Interprétation géométrique

En développant le produit de Frobenius entre la matrice de Gram et la matrice cible pondérée, en terme de produits scalaires dans l'espace transformé, il est possible de faire ressortir une

interprétation géométrique du critère d'Alignement :

$$\begin{aligned}
\left\langle \boldsymbol{K}, \hat{\boldsymbol{K}}^* \right\rangle_F &= \left\langle \begin{pmatrix} \boldsymbol{K_{11}} & \boldsymbol{K_{12}} \\ \boldsymbol{K_{21}} & \boldsymbol{K_{22}} \end{pmatrix}, \begin{pmatrix} \frac{1}{n_1^2}\mathbf{1} & -\frac{n_1}{n_2}\mathbf{1} \\ -\frac{n_1}{n_2}\mathbf{1} & \frac{1}{n_2^2}\mathbf{1} \end{pmatrix} \right\rangle_F \\
&= \frac{1}{n_1^2} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{S}_1} k(\boldsymbol{x_i}, \boldsymbol{x_j}) + \frac{1}{n_2^2} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{S}_2} k(\boldsymbol{x_i}, \boldsymbol{x_j}) - \frac{2}{n_1 n_2} \sum_{\boldsymbol{x}_i \in \mathcal{S}_1} \sum_{\boldsymbol{x}_j \in \mathcal{S}_2} k(\boldsymbol{x_i}, \boldsymbol{x_j}) \\
&= \frac{1}{n_1^2} \Big(\sum_{\boldsymbol{x}_i \in \mathcal{S}_1} \phi(\boldsymbol{x}_i) \Big)^2 + \frac{1}{n_2^2} \Big(\sum_{\boldsymbol{x}_j \in \mathcal{S}_2} \phi(\boldsymbol{x}_j) \Big)^2 - \frac{2}{n_1 n_2} \sum_{\boldsymbol{x}_i \in \mathcal{S}_1} \phi(\boldsymbol{x}_i) \sum_{\boldsymbol{x}_j \in \mathcal{S}_2} \phi(\boldsymbol{x}_j) \\
&= \left(\frac{1}{n_1} \sum_{\boldsymbol{x}_i \in \mathcal{S}_1} \phi(\boldsymbol{x}_i) - \frac{1}{n_2} \sum_{\boldsymbol{x}_j \in \mathcal{S}_2} \phi(\boldsymbol{x}_j) \right)^2 \\
&= \left\| \boldsymbol{\mu}_1^\Phi - \boldsymbol{\mu}_2^\Phi \right\|^2,
\end{aligned}$$

où $\boldsymbol{\mu}_c^\Phi = \frac{1}{n_c} \sum_{\boldsymbol{x}_i \in \mathcal{S}_c} \phi(\boldsymbol{x}_i)$ est le centre des exemples de la classe c dans l'espace transformé.

La maximisation de l'Alignement se traduit donc dans l'espace transformé par la maximisation de la distance dite *inter-classes* entre les centres des deux classes, approche suivie par [251] sans référence au KTA. On retrouve ainsi le numérateur du critère de Fisher (équation 3.3) exprimant le vecteur normal de l'hyperplan de séparation optimale. Néanmoins, tandis que le critère de Fisher fait intervenir dans son dénominateur les covariances des exemples des classes, le dénominateur du critère d'Alignement est difficilement interprétable géométriquement puisqu'il fait intervenir des produits scalaires au carré dans l'espace transformé :

$$||\boldsymbol{K}||_F = \sqrt{\sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{S}} \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle^2}.$$

Ceci étant, on peut montrer [56] que la mesure d'Alignement est proportionnelle à la distance inter-classes et inversement proportionnelle aux distances intra-classes.

4.4.1.2 Fenêtres de Parzen

Nous apportons ici une interprétation nouvelle du critère d'Alignement en mettant en lumière une relation particulière du noyau gaussien RBF avec les fenêtres de Parzen. En 1962, Parzen propose [177] une nouvelle méthode d'estimation de la densité de probabilité basée sur ce qu'il appelle les fonctions noyaux, qui n'ont à priori pas de lien direct avec les noyaux impliqués dans les SVM. Nous emploierons de préférence le terme équivalent de « fenêtres de Parzen », pour éviter toute ambiguité.

Le problème posé est celui de l'estimation d'une densité de probabilité $f(\boldsymbol{x})$ à partir de n échantillons i.i.d. $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ d'une variable aléatoire $\boldsymbol{x}$. Une méthode courante est l'estimation par histogramme mais celle-ci présente l'inconvénient de fournir une réponse fortement discontinue et de support restreint.

On appelle *fenêtres de Parzen* toute fonction $K(y)$ intégrable à valeurs réelles non-négatives, si elle remplit les conditions suivantes :

- K est de somme unitaire, $\int_{-\infty}^{\infty} K(y)dy = 1$
- K est symétrique, $K(-y) = K(y)$

L'estimation de densité consiste simplement à sommer les contributions de la fenêtre au voisinage des exemples de l'échantillon, soit :

$$\hat{f}_n(\boldsymbol{x}) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right).$$

Cette méthode intuitive, statistiquement justifiée par Parzen, permet ainsi de lisser la contribution de chaque exemple sur la densité de probabilité estimée. En pratique on choisit en général des fenêtres dont le maximum est atteint en 0, et décroissants au voisinage, comme les exemples suivants :

- Fenêtre uniforme $K_U(y) = \frac{1}{2}\mathbf{1}_{\{|y|\leq 1\}}$
- Fenêtre triangulaire $K_{tri}(y) = (1-|y|)\mathbf{1}_{\{|y|\leq 1\}}$
- Fenêtre cosinus $K_{cos}(y) = \frac{\pi}{4}\cos\left(\frac{\pi}{2}y\right)\mathbf{1}_{\{|y|\leq 1\}}$
- Fenêtre gaussienne $K_G(y) = (2\pi)^{-\frac{1}{2}}e^{-\frac{1}{2}y^2}$

On remarque que la fenêtre uniforme équivaut à une estimation par histogramme, en relâchant la contrainte de pas fixe entre les *bins*. Parmi les fenêtres présentées, la fenêtre gaussienne est particulièrement intéressante puisqu'elle est continue et infiniment dérivable en tout point. De plus, on remarque que le lien avec le noyau gaussien RBF présenté précédemment est immédiat puisque

$$k_{\text{rbf}}(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\right) = \sqrt{2\pi}\,K_G\left(\frac{\|\boldsymbol{x}-\boldsymbol{y}\|}{\sigma}\right).$$

Dans le cas d'un problème à deux classes, on peut ainsi caractériser, à l'aide de l'estimation de Parzen sur la fenêtre gaussienne, la densité de probabilité pour chacune des classes :

$$\begin{aligned}\hat{f}(\boldsymbol{x}|y=+1) &= \frac{1}{n_1\sigma}\sum_{\boldsymbol{x}_i\in\mathcal{S}_1} K_G\left(\frac{\|\boldsymbol{x}_i-\boldsymbol{x}\|}{\sigma}\right)\\ \hat{f}(\boldsymbol{x}|y=-1) &= \frac{1}{n_2\sigma}\sum_{\boldsymbol{x}_i\in\mathcal{S}_2} K_G\left(\frac{\|\boldsymbol{x}_i-\boldsymbol{x}\|}{\sigma}\right).\end{aligned}$$

Un problème de discrimination est d'autant plus facile à résoudre qu'il est séparable. Nous définissons ainsi un critère de séparabilité $\mathcal{P}_{sep}$ sur chaque exemple en soustrayant les densités de probabilité des deux classes :

$$\begin{aligned}\mathcal{P}_{sep}(\boldsymbol{x}_j,y_j) &= \hat{f}(\boldsymbol{x}_j|y=y_j) - \hat{f}(\boldsymbol{x}_j|y\neq y_j)\\ &= y_j\sum_{y_i\in\{-1,+1\}} y_i\,\hat{f}(\boldsymbol{x}_j|y=y_i).\end{aligned}$$

Ce critère peut être interprété comme un détecteur d'*outlier*. En effet, il mesure l'adéquation de la classe y associée à l'exemple $\boldsymbol{x}$ par rapport aux densités de probabilité des deux classes dans son voisinage. On remarque que $\mathcal{P}_{sep}$ évolue dans l'intervalle $[-1;+1]$ et est d'autant plus proche de 1 que la probabilité que l'exemple $\boldsymbol{x}_i$ soit de classe y_i est élevée. Ainsi, si l'on somme le critère $\mathcal{P}_{sep}$ sur tous les exemples de l'ensemble d'apprentissage en le pondérant par le nombre d'exemples de chaque classe, on définit un critère d'estimation de séparabilité de l'ensemble $\mathcal{S}$:

$$\begin{aligned}\mathcal{P}_{sep}(\mathcal{S}) &= \frac{1}{n_1}\sum_{(\boldsymbol{x}_j,y_j)\in\mathcal{S}_1}^{n}\mathcal{P}_{sep}(\boldsymbol{x}_j,y_j) + \frac{1}{n_2}\sum_{(\boldsymbol{x}_j,y_j)\in\mathcal{S}_2}^{n}\mathcal{P}_{sep}(\boldsymbol{x}_j,y_j)\\ &= \sum_{j=1}^{n}\frac{y_j}{n_j}\sum_{y_i\in\{-1,+1\}} y_i\,\hat{f}(\boldsymbol{x}_j,y=y_i)\\ &= \sum_{j=1}^{n}\frac{y_j}{n_j}\left[y_1\frac{1}{n_1\sigma}\sum_{\boldsymbol{x}_i\in\mathcal{S}_1}K_G\left(\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|}{\sigma}\right) + y_2\frac{1}{n_2\sigma}\sum_{\boldsymbol{x}_i\in\mathcal{S}_2}K_G\left(\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|}{\sigma}\right)\right]\\ &= \frac{1}{\sigma}\sum_{j=1}^{n}\sum_{i=1}^{n}\frac{y_j}{n_j}\frac{y_i}{n_i}K_G\left(\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|}{\sigma}\right).\end{aligned}$$

D'où, d'après les relations 4.10 et 4.7 :

$$\mathcal{P}_{sep}(\mathcal{S}) = \frac{1}{\sqrt{2\pi}\sigma}\left\langle \boldsymbol{K},\hat{\boldsymbol{K}}^*\right\rangle_F,$$

où $\boldsymbol{K}$ est la matrice de Gram du noyau k_{rbf} de paramètre σ, et $\hat{\boldsymbol{K}}^*$ la matrice cible de l'ensemble $\mathcal{S}$.

Nous avons donc montré que dans le cas de noyaux (au sens des SVM) respectant les conditions de Parzen, comme c'est le cas pour le noyau gaussien RBF, le produit de Frobenius, qui constitue le critère non normalisé du KTA, peut être interprété comme une mesure de séparabilité de l'ensemble d'apprentissage, liée à l'estimation de Parzen de la densité de probabilité des deux classes.

4.4.1.3 Dérivation

De par l'expression très simple du produit de Frobenius, la dérivation du critère d'Alignement est immédiate. Si l'on considère un noyau $k_{\boldsymbol{\Theta}}$ caractérisé par les paramètres $\boldsymbol{\Theta} = [\theta_1, \ldots, \theta_p]$, et la matrice de Gram correspondante $\boldsymbol{K}_{\boldsymbol{\Theta}}$, alors :

$$\frac{\partial}{\partial \theta_p} \langle \boldsymbol{K}_{\boldsymbol{\Theta}}, \boldsymbol{K}^* \rangle_F = \left\langle \partial_{\theta_p} \boldsymbol{K}_{\boldsymbol{\Theta}}, \boldsymbol{K}^* \right\rangle_F \tag{4.10}$$

$$\frac{\partial}{\partial \theta_p} ||\boldsymbol{K}_{\boldsymbol{\Theta}}||_F = \frac{\left\langle \partial_{\theta_p} \boldsymbol{K}_{\boldsymbol{\Theta}}, \boldsymbol{K}_{\boldsymbol{\Theta}} \right\rangle_F}{||\boldsymbol{K}_{\boldsymbol{\Theta}}||_F}, \tag{4.11}$$

où l'on a défini les matrices $\partial_{\theta_p} \boldsymbol{K}_{\boldsymbol{\Theta}} = [\partial_{\theta_p} k_{\boldsymbol{\Theta}}(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{ij}$. Ainsi en calculant ces matrices on peut dériver l'Alignement par rapport à $\boldsymbol{\Theta}$:

$$\frac{\partial}{\partial \theta_p} \mathcal{A}(\boldsymbol{K}_{\boldsymbol{\Theta}}, \boldsymbol{K}^*) = \frac{\left\langle \partial_{\theta_p} \boldsymbol{K}_{\boldsymbol{\Theta}}, \boldsymbol{K}^* \right\rangle_F}{||\boldsymbol{K}_{\boldsymbol{\Theta}}||_F ||\boldsymbol{K}^*||_F} - \frac{\langle \boldsymbol{K}_{\boldsymbol{\Theta}}, \boldsymbol{K}^* \rangle_F \left\langle \boldsymbol{K}_{\boldsymbol{\Theta}}, \partial_{\theta_p} \boldsymbol{K}_{\boldsymbol{\Theta}} \right\rangle_F}{||\boldsymbol{K}_{\boldsymbol{\Theta}}||_F^3 ||\boldsymbol{K}^*||_F}. \tag{4.12}$$

On peut ainsi appliquer les techniques d'optimisation présentées précédemment sur le critère d'Alignement pour affiner les paramètres du noyau $k_{\boldsymbol{\Theta}}$.

4.4.1.4 Critique de l'Alignement

Nous avons expliqué que la normalisation du produit de Frobenius restreint l'Alignement à l'intervalle $[-1; +1]$ (équation 4.6). Pourtant si l'on développe l'expression de la matrice cible,

$$\boldsymbol{K}^* = \boldsymbol{y}\boldsymbol{y}^T,$$

on en déduit la positivité du produit de Frobenius

$$\langle \boldsymbol{K}, \boldsymbol{K}^* \rangle_F = \boldsymbol{y}\boldsymbol{K}\boldsymbol{y}^T \geq 0,$$

puisque comme nous l'avons vu, tout noyau respectant la condition de Mercer a sa matrice de Gram semi-définie positive sur tout ensemble d'exemples. Les termes de normalisation de l'Alignement étant positifs, on en déduit

$$0 \leq \mathcal{A}(\boldsymbol{K}, \boldsymbol{K}^*) \leq 1.$$

On peut remarquer en outre que certains noyaux, comme le noyau gaussien RBF, ont toujours une valeur positive, ce qui implique donc que les exemples sont restreints, dans l'espace transformé, à un cône d'angle borné par $\frac{\pi}{2}$ puisque tous les produits scalaires y sont positifs. Ainsi, dans le meilleur des cas, les exemples de classes opposées sont orthogonaux ($k(\boldsymbol{x}_+, \boldsymbol{x}_-) = 0$). La matrice de Gram optimale a donc pour valeur :

$$\boldsymbol{K}_{opt} = \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}.$$

L'Alignement est donc borné par la valeur suivante :

$$\mathcal{A}(\boldsymbol{K}_{opt}, \boldsymbol{K}^*) \leq \frac{\sqrt{n_1^2 + n_2^2}}{n}.$$

Soit, dans le cas de classes également réparties,

$$\mathcal{A}(\boldsymbol{K}_{opt}, \boldsymbol{K}^*) \leq \frac{1}{\sqrt{2}}.$$

On remarque donc que l'intervalle dans lequel évolue la mesure d'Alignement est fortement déterminé par le choix du noyau, ce qui remet en cause sa fiabilité, par exemple pour comparer la pertinence de deux noyaux différents. Cette carence s'explique très simplement en termes géométriques. Dans le cas du noyau gaussien, les exemples sont situés dans un cône dont l'extrémité est à l'origine de l'espace ; ils sont par ailleurs situés sur l'intersection de ce cône avec la sphère unité puisque

$$k_{\text{rbf}}(\boldsymbol{x}, \boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}\|^2}{2\sigma^2}\right) = 1.$$

L'origine est donc excentrée par rapport au centre de l'ensemble des exemples. Or la définition de la matrice cible suppose que, dans l'idéal, les exemples de classes opposés sont situés de part et d'autre de l'origine. En d'autres termes, la maximisation du critère d'Alignement est une condition suffisante à l'optimisation des performances, mais non nécessaire.

C'est là la critique principale au critère d'Alignement que l'on retrouve dans la littérature [167][188][189] et qui peut d'ailleurs être adressée également aux Machines à Vecteurs de Support [152]. Une solution à ce problème consiste à translater les exemples dans l'espace transformé pour les centrer autour de l'origine, comme le font par exemple Meila [152] ou Pothin et Richard [189]. Mais ce genre de solution implique généralement un procédure d'optimisation portant sur autant de coefficients que d'exemples dans la base. La procédure devient donc trop coûteuse dans le cas de bases d'apprentissage conséquentes (plusieurs milliers d'exemples).

4.4.2 Séparabilité dans l'espace Transformé (KCS)

Construit sur des bases algébriques, le produit de Frobenius présent dans le critère d'Alignement est en fait, comme nous l'avons vu, égal à la mesure de distance inter-classes dans l'espace transformé que l'on retrouve au numérateur du critère de Fisher. Il est en fait possible d'exprimer pleinement le critère de Fisher dans l'espace transformé. On le retrouve également dans la littérature, désigné par les termes de « critère de séparabilité des classes », défini de la façon suivante :

$$J = \frac{\operatorname{tr} \boldsymbol{S}_b}{\operatorname{tr} \boldsymbol{S}_w}, \tag{4.13}$$

où $\boldsymbol{S_b}$ est la matrice de *dispersion inter-classes* (b pour *between-class scatter*) et $\boldsymbol{S_w}$ la matrice de *dispersion intra-classe* (w pour *within-class scatter*). Ces dernières ont les expressions suivantes :

$$\boldsymbol{S}_b = \frac{1}{n} \sum_{c=1,2} n_c (\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T \tag{4.14}$$

$$\boldsymbol{S}_w = \sum_{c=1,2} \sum_{\boldsymbol{x}_i \in \mathcal{S}_c} (\boldsymbol{x}_i - \boldsymbol{\mu}_c)(\boldsymbol{x}_i - \boldsymbol{\mu}_c)^T, \tag{4.15}$$

où $\boldsymbol{\mu}_c = \frac{1}{n_c} \sum_{\boldsymbol{x}_i \in \mathcal{S}_c} \boldsymbol{x}_i$ est le centre des exemples de la classe c et $\boldsymbol{\mu}$ le centre de l'ensemble des exemples ($\boldsymbol{\mu} = \frac{1}{n} \sum_{\boldsymbol{x}_i \in \mathcal{S}} \boldsymbol{x}_i = \frac{1}{n} (n_1 \boldsymbol{\mu}_1 + n_2 \boldsymbol{\mu}_2)$). Dans le cas de classes également réparties ($\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \frac{\boldsymbol{\mu}}{2}$), on retrouve dans le quotient des deux matrices de dispersion une expression très proche du critère de Fisher énoncé en section 3.2 :

$$\boldsymbol{w}_F = \frac{\boldsymbol{S}_b}{\boldsymbol{S}_w} = \frac{1}{4} \frac{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^2}{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2},$$

où l'on suppose que la matrice de dispersion intra-classe $\boldsymbol{S_w}$ est inversible.

Les matrices $\boldsymbol{S_b}$ et $\boldsymbol{S_w}$ pouvant s'exprimer exclusivement en terme de produits scalaires sur les exemples, il est possible [252][240] d'y substituer l'usage d'une fonction noyau pour « kerneliser » l'expression du critère J (équation 4.13) :

$$\mathcal{J} = \frac{\mathbf{1}_n^T \boldsymbol{B} \mathbf{1}_n}{\mathbf{1}_n^T \boldsymbol{W} \mathbf{1}_n} = \frac{\Sigma(\boldsymbol{B})}{\Sigma(\boldsymbol{W})}, \tag{4.16}$$

où l'on rappelle que l'opérateur Σ correspond à la somme de tous les termes d'une matrice. Les matrices kernelisées de dispersion inter-classes ($\boldsymbol{B}$) et intra-classes ($\boldsymbol{W}$), introduites dans la relation précédente, ont les expressions suivantes :

$$\boldsymbol{B} = \begin{pmatrix} \frac{1}{n_1}\boldsymbol{K}_{11} & \mathbf{0} \\ \mathbf{0} & \frac{1}{n_2}\boldsymbol{K}_{22} \end{pmatrix} - \boldsymbol{K} \tag{4.17}$$

$$\boldsymbol{W} = \begin{pmatrix} k_{11} & 0 & \cdots & 0 \\ 0 & k_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k_{nn} \end{pmatrix} - \begin{pmatrix} \frac{1}{n_1}\boldsymbol{K}_{11} & \mathbf{0} \\ \mathbf{0} & \frac{1}{n_2}\boldsymbol{K}_{22} \end{pmatrix}. \tag{4.18}$$

On utilise ici la décomposition de la matrice de Gram par blocs de classes introduite dans l'équation 4.4. Le critère $\mathcal{J}$ introduit est donc une mesure de séparabilité dans l'espace transformé que nous désignerons par l'acronyme KCS (pour *Kernel Class Separability*).

Dérivation et régularisation

Le critère $\mathcal{J}$ est également aisément dérivable et sa dérivée par rapport au paramètre θ_p implique la matrice $\partial_{\theta_p}\boldsymbol{K}_{\boldsymbol{\Theta}}$ introduite précédemment.

Toutefois, dans le cas particulier du noyau RBF gaussien, l'expression du critère KCS implique une instabilité numérique dans la maximisation de $\mathcal{J}$ en provoquant systématiquement la convergence des deux dispersions vers 0 (soit $\operatorname{tr} S_b \xrightarrow[\sigma\to\infty]{} 0$ et $\operatorname{tr} S_w \xrightarrow[\sigma\to\infty]{} 0$, dont le rapport tend vers 1, borne supérieure du critère). On contourne ce problème en appliquant une régularisation sur le dénominateur :

$$\tilde{\mathcal{J}} = \frac{\mathbf{1}_n^T \boldsymbol{B} \mathbf{1}_n}{\mathbf{1}_n^T \boldsymbol{W} \mathbf{1}_n} = \frac{\Sigma(\boldsymbol{B})}{\Sigma(\boldsymbol{W}) + \epsilon}, \tag{4.19}$$

qui évite la convergence $\mathcal{J} \xrightarrow[\sigma\to\infty]{} 1$ sans avoir à faire appel aux techniques d'optimisation sous contraintes, qui complexifieraient l'algorithme. Nous verrons par la suite que cette procédure de régularisation est également nécessaire lorsque nous introduirons le critère KCS pour la sélection automatique de descripteurs (voir section 7.5.4).

4.5 Facteur d'erreur C

Le facteur d'erreur C est un paramètre particulier puisqu'il est le seul à ne pas intervenir dans la définition du noyau. Nous avons vu dans la section 3.6, que celui-ci intervient différemment dans les Machines à Marge Souple selon le choix de la norme appliquée sur les variables d'écart. On peut cependant tenter d'en donner une interprétation intuitive :

- Lorsque $C \to \infty$, la tolérance aux erreurs de classification est de plus en plus rigide. On voit que pour les deux problèmes $L1$ et $L2$ (équations 3.30 et 3.29), on retrouve alors l'expression du problème à Marge Dure. Le problème est en effet équivalent dans le cas de données séparables puisque les variables d'écart sont nulles.
- Lorsque $C \to 0$, le système tolère les erreurs jusqu'à ne plus distinguer les exemples des deux classes. Le cas extrême $C = 0$ implique d'ailleurs $\alpha_i = 0\ \forall i$ pour les $L1$ SVM, ce qui signifie que la fonction de décision ne dépend plus des exemples d'apprentissage. De même pour les $L2$ SVM où la matrice $\boldsymbol{H}$ devient négligeable dans l'expression du problème dual (équation 3.29).

La valeur optimale de C constituera donc un compromis entre la tolérance aux *outliers* et la minimisation de l'erreur.

Si l'on utilise la norme $L2$, l'expression du problème est la même que dans le cas des Machines à Marge Dure, en ajoutant la constante $\frac{1}{C}$ aux termes diagonaux de la matrice de Gram. La plupart des méthodes d'optimisation des paramètres étant basées sur la matrice de Gram, il

est donc possible d'assimiler la constante C à un hyper-paramètre du noyau dans le cas des $L2$ SVM.

Toutefois, concernant la norme $L1$, plus généralement employée, le raisonnement précédent ne s'applique pas. Le contrôle du risque structurel est d'ailleurs moins simple dans le cadre des Machines à Marge Souple car les résultats de la théorie de Vapnik-Chervonenkis ne s'appliquent pas tels quels. Néanmoins, Steinwart a montré [221] que pour tout $\epsilon > 0$, il existe une valeur C_ϵ telle que pour tout $C \geq C_\epsilon$, le risque de la fonction de décision obtenue par $L1$ SVM n'excède pas de plus de ϵ le risque fonctionnel minimal.

Il existe peu de travaux dans la littérature sur la détermination automatique de la valeur optimale de C. Nous apportons tout de même ici une réponse dans le cas des $L1$ SVM.

4.5.1 Valeur de Joachims

Dans son logiciel *SVMlight* [113], Joachims propose la valeur par défaut suivante :

$$C_{def} = \frac{1}{\bar{R}^2}, \tag{4.20}$$

où $\bar{R}$ est la distance moyenne des exemples à l'origine dans l'espace transformé. Cette valeur est similaire au rayon R introduit précédemment, qui concerne la sphère minimale contenant tous les exemples dans l'espace transformé. Plusieurs méthodes existent pour évaluer ces grandeurs, qui sont présentées dans l'annexe A. Celle qu'emploie Joachims est présentée dans la section A.3.

Validation expérimentale

Nous montrons par une courte expérience qu'en pratique la valeur de Joachims constitue le meilleur compromis entre complexité et performance. Afin de valider celle-ci nous apprenons une machine SVM à noyau gaussien RBF (où le paramètre σ est fixé arbitrairement à 1) pour chaque valeur C d'un ensemble de 25 valeurs réparties logarithmiquement entre 10^{-6} et 10^6. Pour chaque machine nous évaluons l'erreur d'apprentissage, qui du fait du sur-apprentissage, se révèle fortement biaisée, et l'erreur *Leave-one-out*, qui constitue, comme nous l'avons expliqué, l'estimateur le plus fiable de l'erreur de généralisation. L'évolution du taux de vecteurs de support parmi les exemples d'apprentissage, ainsi que le temps d'apprentissage nous apporterons également des informations utiles quant à la complexité de la phase d'apprentissage.

Nous avons appliqué cette évaluation sur trois problèmes de discrimination. Chacun d'entre eux se résume à une base de données contenant les exemples des deux classes, caractérisés par une série des descripteurs propres aux problèmes. Nous avons ainsi exploité deux bases publiques disponibles sur le dépôt UCI [16], destiné à offrir des données communes à la communauté scientifique en apprentissage statistique : *Spambase* qui décrit un problème de détection de mails parasites, et *Ionosphere*, qui concerne la détection d'une structure dans l'ionosphère. Une troisième base a été constituée sur les données du corpus ESTER (présenté dans la partie expérimentale finale, section 10.1.1) et décrit un problème de discrimination entre parole et musique pures, caractérisé par une grande collection de descripteurs. On trouvera plus de détail sur ces bases expérimentales dans l'annexe B. Celles-ci seront par ailleurs exploitées à nouveau dans la section suivante (4.6), sur les critères de sélection, ainsi que dans le chapitre 7 traitant des méthodes de sélection de descripteurs.

Les figures 4.5, 4.6 et 4.7 montrent les résultats de l'expérience sur les bases respectives *Spambase*, *Ionosphere* et *parole/musique*. On constate en premier lieu que les grandeurs représentées ont un profil commun dans les trois cas. Ainsi lorsque $\log_{10}(C)$ est inférieur à une certaine valeur (autour de -2), le taux d'erreur d'apprentissage et l'erreur *Leave-one-out* sont égaux et constants, de même que le taux de vecteurs de support, très élevé (de l'ordre de 80%). Ceci s'explique par le fait que la pénalisation des erreurs (des variables d'écart) est négligeable devant la minimisation de la marge, dont il résulte un hyperplan de séparation quelconque, et une large proportion d'exemples mal classifiés (et donc de vecteurs de support au delà de la marge). On constate par ailleurs que le temps d'apprentissage décroît très vite lorsque C augmente.

Passé le seuil autour de -2, on constate une rapide décroissance des deux erreurs, provenant du fait que les variables d'écart participent à l'optimisation. La réduction du nombre d'exemples mal

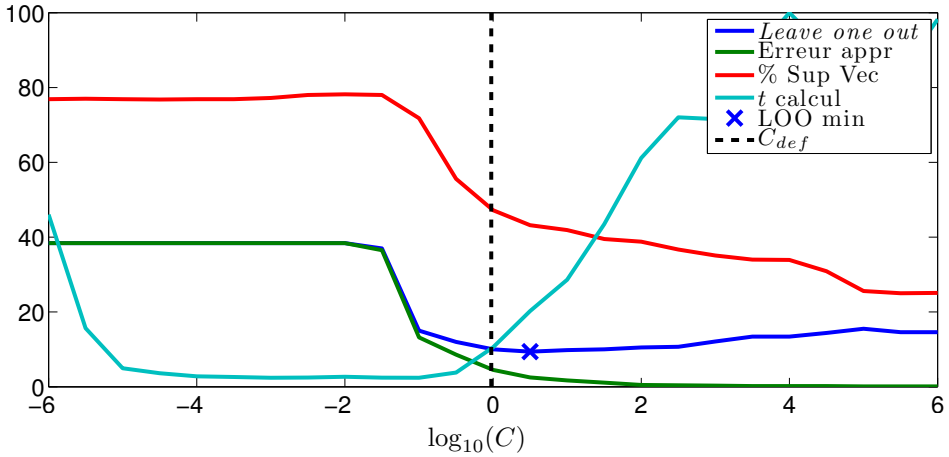


FIGURE 4.5 – Évolution des erreurs *Leave-one-out* et d'apprentissage, ainsi que du taux de vecteurs de support et du temps d'apprentissage (ici en pourcentage par rapport au temps maximal observé), par rapport aux variations du facteur C, sur la base *Spambase*. La ligne noire pointillée indique la valeur C_{def} définie par Joachims, et la croix bleue le minimum de l'erreur *Leave-one-out*. Le temps de calcul relatif (*t* calcul) figure ici en pointillés à titre informatif.

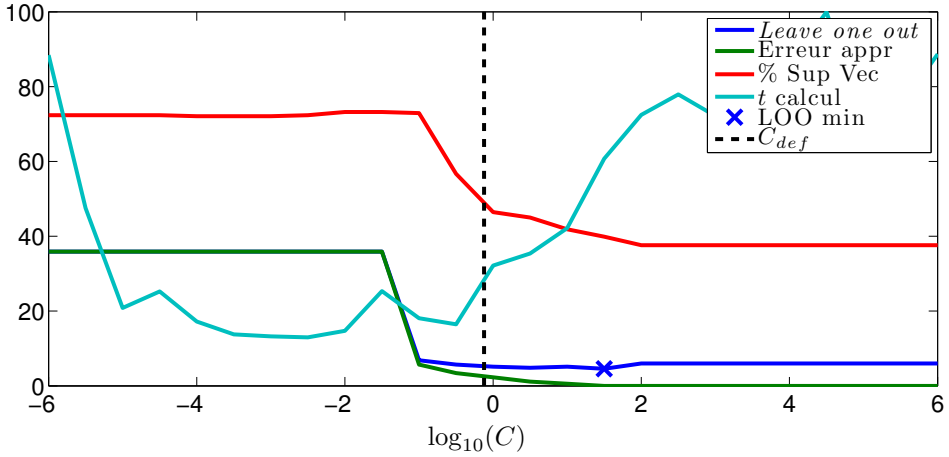


FIGURE 4.6 – Résultats de la même expérience sur la base *Ionosphere*.

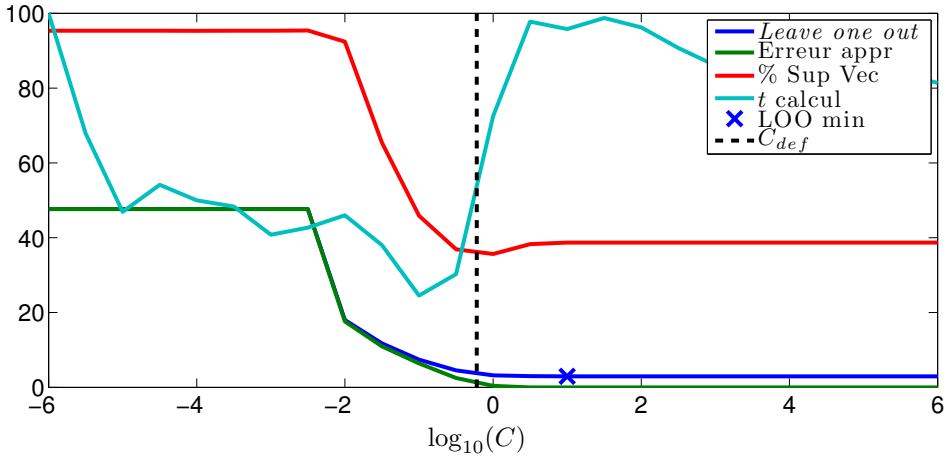


FIGURE 4.7 – Résultats de la même expérience sur la base *parole/musique*.

classifiés implique nécessairement la réduction du nombre de vecteurs de support (puisque, nous le rappelons, tout exemple mal classifié est un vecteur de support auquel est associé une variable d'écart non nulle). Cependant, plus on augmente le facteur C, plus le compromis entre marge et variables d'écart est complexe, et plus la convergence de l'algorithme d'apprentissage est longue, d'où un temps de calcul qui augmente de manière quasi monotone.

Tandis que l'erreur d'apprentissage décroît de manière monotone avec l'augmentation de C, on constate que l'erreur *Leave-one-out* augmente de nouveau après avoir dépassé un minimum global (marqué d'une croix bleue sur les figures), ce qui traduit le phénomène de sur-apprentissage des exemples de la base, que l'erreur d'apprentissage ne permet pas de constater.

La ligne verticale pointillée représente la valeur C_{def} calculée sur la base des exemples. On constate dans les trois cas que celle-ci est proche du minimum de l'erreur LOO, ou au moins, que la valeur de l'erreur LOO y est très proche de sa valeur minimale, ce qui confirme la pertinence de cette valeur en termes de performances. Enfin, on remarque dans les trois cas également, que le léger incrément d'erreur apporté par C_{def} est compensé par une nette réduction en temps de calcul par rapport à la valeur minimisant l'erreur LOO.

La valeur de Joachims constitue donc un excellent compromis entre complexité et performances, et se révèle très simple et rapide à calculer.

4.5.2 Inclusion du facteur C dans les critères

L'expérience précédente nous montre l'influence déterminante du facteur C sur les performances des Machines à Vecteurs de Support. Pourtant les critères de séparabilité introduits dans la section 4.4 (alignement et séparabilité de classes) sont exclusivement basés sur la matrice de Gram, qui synthétise l'action du noyau sur les exemples d'apprentissage. Ils n'incluent donc pas le facteur d'erreur C, qui est un paramètre extérieur au noyau.

En introduisant les machines à marge souple dans ce document (section 3.6) nous avons mentionné les deux principaux paradigmes $L1$ et $L2$ qui diffèrent selon que l'on applique la puissance $k = 1$ ou $k = 2$ à la somme des variables d'écart, pour constituer la pénalité totale des exemples mal classifiés, pondérée par le facteur C.

Le cas $L1$ est le plus généralement employé parce qu'il permet de conserver le même problème d'optimisation que dans le cas des machines à marge dure, en ajoutant seulement une borne supérieure sur les facteurs de Lagrange (voir équation 3.30). On peut montrer cependant que le cas $L2$ est équivalent au seul ajout de la constante $\frac{1}{C}$ sur les termes diagonaux de la matrice de Gram (équation 3.29). On définit donc la matrice de Gram *ajustée* par le facteur C comme suit :

$$\boldsymbol{K}_C = \boldsymbol{K} + \frac{1}{C}\boldsymbol{I}$$

Bien que nous employions exclusivement le paradigme $L1$ dans l'implémentation des SVM, nous proposons d'appliquer le principe de l'approche $L2$ sur les matrices de Gram exploitées pour le calcul de l'Alignement et du critère KCS. Les deux problèmes à marge souple ne sont pas équivalents formellement, de sorte que les paramètres n'ont théoriquement pas la même influence. Toutefois, nous verrons dans la section expérimentale 4.6 qu'en pratique cette opération renforce la fiabilité des critères sus-cités.

En suivant le raisonnement précédent, on en déduit que la matrice de Gram non-ajustée est équivalente à un problème où $C = \infty$, soit une pénalisation infinie des exemples hors de la marge, ce qui revient à appliquer un modèle à marge dure sur un problème non-séparable. En pratique on trouve tout de même une solution dans ce cas, mais celle-ci est sous-optimale.

Cet apport se révèle par ailleurs d'un coût négligeable puisque son application revient à ajouter

un terme constant aux grandeurs impliquées dans les critères :

$$\begin{aligned}
\langle \boldsymbol{K}_C, \boldsymbol{K}^* \rangle_F &= \langle \boldsymbol{K}, \boldsymbol{K}^* \rangle_F + \frac{n}{C} \\
\|\boldsymbol{K}_C\|_F^2 &= \|\boldsymbol{K}\|_F^2 + \frac{n}{C^2} \\
\Sigma(\boldsymbol{B}_C) &= \Sigma(\boldsymbol{B}) - \frac{n-2}{C} \\
\Sigma(\boldsymbol{W}_C) &= \Sigma(\boldsymbol{W}) + \frac{n-2}{C}
\end{aligned}$$

où $\boldsymbol{B}_C$ et $\boldsymbol{W}_C$ sont le résultat de l'application des formules 4.17 et 4.18 sur la matrice de Gram ajustée ; $\Sigma(\boldsymbol{B})$ et $\Sigma(\boldsymbol{W})$ interviennent dans la définition du critère KCS (équation 4.16).

4.6 Evaluation des critères de sélection de noyau

L'expérience pratique décrite dans cette section confirme les remarques avancées sur les avantages et les limites des critères proposés pour l'évaluation du noyau. Nous étudions pour cela la recherche du paramètre $\hat{\sigma}$ optimal sur un noyau RBF gaussien. A titre expérimental, on déploie donc une recherche par maillage (dont la complexité reste ici raisonnable sur une dimension) en faisant varier le paramètre σ sur 25 valeurs réparties logarithmiquement entre 0.1 et 20, qui constituent des bornes assez larges pour ce paramètre. On rappelle que l'introduction du facteur $\frac{1}{d}$ dans l'expression des noyaux (voir section 3.5.3) réduit largement le champ des valeurs optimales mesurées pour σ, qui se trouvent généralement dans le voisinage de 1.

La valeur optimale retenue est celle minimisant le critère, dans le cas des critères d'estimation de l'erreur *Leave-one-out*, décrits dans la section 4.3, ou le maximisant, dans le cas des critères de séparabilité de classes décrits dans la section 4.4.

On a également calculé l'erreur *Leave-one-out* exacte pour chacune des valeurs de σ, qui nous sert de référence pour estimer l'erreur de généralisation.

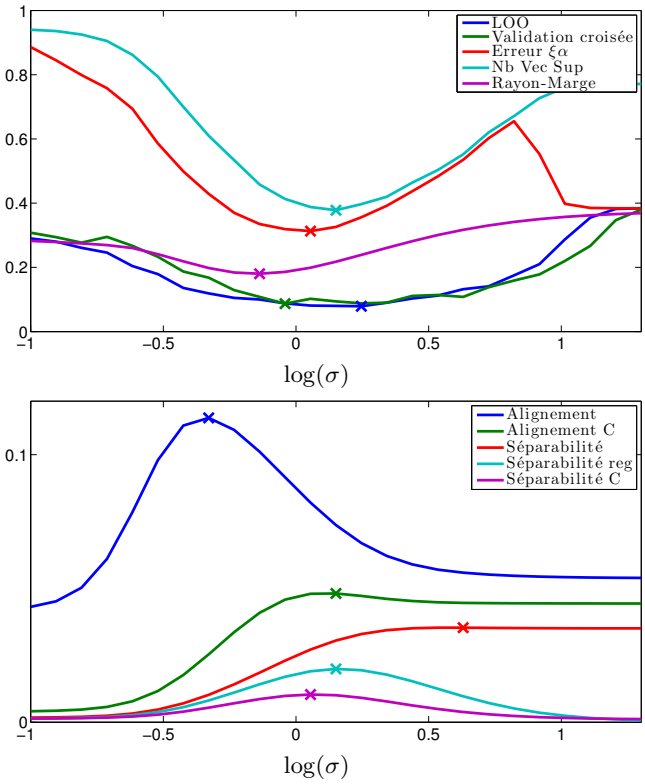
Nous avons appliqué cette évaluation sur les bases introduites dans la section 4.5.1, auxquelles nous ajoutons la base *Lymphoma*. Ces bases, ainsi que les problèmes qu'elles modélisent, sont décrits en détail dans l'annexe B.

Dans un premier temps nous avons fixé le facteur d'erreur à $C = 1$ dans toutes les étapes (apprentissage des SVM, et ajustement éventuel des matrices de Gram, d'après la procédure proposée dans la section 4.5.2). Les figures 4.8, 4.9 et 4.10 illustrent respectivement le résultat de l'expérience sur les bases *Spambase*, *Ionosphere* et *Parole/musique*.

Nous n'avons pas superposé les critères d'erreur et de séparabilité, dont la comparaison n'a pas de sens. Les critères d'erreurs sont représentés dans les figures supérieures, et comparés à l'erreur *Leave-one-out*, tandis que les critères de séparabilité apparaissent dans les figures inférieures, et ne sont comparables entre eux que par la seule localisation du maximum (leurs valeurs ne sont pas comparables). On remarquera que le logarithme de σ est utilisé ici en abscisses. Le tableau de droite qui accompagne chaque couple de figures indique dans la seconde colonne le temps de calcul t (en secondes), la valeur $\hat{\sigma}$ qui minimise ou maximise le critère, ainsi que la valeur de l'erreur *Leave-one-out* pour le $\hat{\sigma}$ estimé.

Sur les figures, la lettre C suivant les noms des critères d'alignement et de séparabilité (KCS) indique que la procédure d'ajustement de la matrice de Gram est appliquée. « Séparabilité reg » indique quant à lui l'application de la procédure de régularisation pour éviter la divergence du critère de séparabilité lors de la phase d'optimisation. La validation croisée est appliquée sur une division de la base d'apprentissage en dix sous-ensembles.

On constate en premier lieu que la validation croisée constitue sans conteste l'estimateur le plus précis de l'erreur LOO, mais au prix d'une complexité prohibitive. Nous rappelons que la validation croisée ne permet pas d'appliquer de recherche par optimisation et implique donc nécessairement une recherche par maillage. Nous précisons par ailleurs que si, en théorie, le calcul de l'erreur LOO est beaucoup plus coûteux que celui de la validation croisée, nous avons exploité ici l'implémentation optimisée de Joachims [113], tandis que la validation croisée n'est calculée que par apprentissages



Critère	t	$\hat{\sigma}$	**LOO**
LOO	2.59	1.76	7.9%
Validation croisée	1.94	0.91	8.8%
Erreur $\xi\alpha$	0.19	1.13	8.1%
Nb Vec Sup	0.19	1.41	8.0%
Rayon-Marge	0.27	0.73	10.0%
Alignement	0.11	0.47	11.9%
Alignement C	0.1	1.41	8.0%
Séparabilité	0.1	4.26	13.2%
Séparabilité reg	0.1	1.41	8.0%
Séparabilité C	0.1	1.13	8.1%

FIGURE 4.8 – Comparaison des valeurs optimales $\hat{\sigma}$ du paramètre σ estimées par chaque critère sur la base *Spambase*, et de l'erreur *Leave-One-Out* pour ces valeurs, avec $C = 1$.

successifs de SVM, ce qui explique le rapport relativement faible entre les temps de calcul de ces deux erreurs.

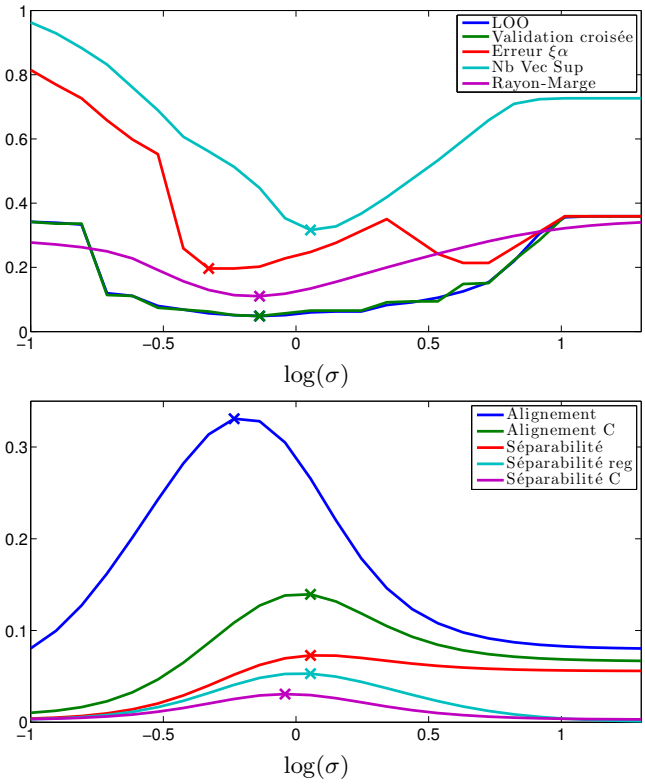
Le nombre de vecteurs de support (Nb Vec Sup) et l'erreur $\xi\alpha$ sont moins gourmants en temps de calcul et constituent bien des bornes supérieures à l'erreur LOO mais on constate en pratique que ces deux bornes sont bien trop larges et surtout que le point minimum est peu corrélé au minimum de l'erreur LOO. De plus, ces critères nécessitent tous deux l'apprentissage d'un SVM et ne sont pas non plus dérivables.

On observe par contre que le critère Rayon-Marge fournit une borne à l'erreur LOO plus resserrée et donc la valeur minimisante est cette fois proche du minimum idéal, sauf dans le cas *Spambase*, où l'estimation est moins précise. Le calcul de la matrice de Gram, ainsi que du rayon R, explique le coût additionnel (le temps de calcul est multiplié par un facteur 1,5) du Rayon-Marge par rapport aux critères précédents.

Les critères proposés (synthétisés dans la partie inférieure des tableaux de résultats) se distinguent nettement des précédents par leur coût en temps de calcul réduit (de l'ordre d'un facteur 3 par rapport au Rayon-Marge), qui résulte de l'absence d'une phase d'apprentissage de SVM dans le calcul. Le calcul de la matrice de Gram est la seule opération coûteuse pour ces critères. L'écart reste cependant moins marqué sur la base *Ionosphere* du fait de sa taille très réduite (les coûts annexes constants y sont donc prédominants).

On remarque que les deux critères proposés ont chacun des limites dans leur forme originelle. Ainsi le critère d'alignement sous-évalue systématiquement la valeur $\hat{\sigma}$ optimale par rapport au minimum de l'erreur LOO, ce qui se traduit par une augmentation raisonnable de l'erreur LOO. En revanche, le critère de séparabilité est ici victime du phénomène de divergence sur le noyau gaussien RBF décrit précédemment, si bien que sur les bases *Spambase* et *parole/musique*, le point maximisant se trouve largement surestimé et produit ainsi une erreur LOO encore plus importante que l'alignement.

La prise en compte du facteur C par la procédure d'ajustement proposée se révèle déterminante sur les deux critères. Sur les bases *Spambase* et *Parole/musique*, on observe ainsi que les valeurs $\hat{\sigma}$ estimées sont très proches des minima de l'erreur LOO (l'égalité stricte, quand elle est observée, est la conséquence du maillage et ne peut être considérée que comme un indice de proximité). Le



Critère	t	$\hat{\sigma}$	**LOO**
LOO	0.2	0.73	4.8%
Validation croisée	0.2	0.73	4.8%
Erreur $\xi\alpha$	0.02	0.47	5.7%
Nb Vec Sup	0.02	1.13	6.0%
Rayon-Marge	0.04	0.73	4.8%
Alignement	0.03	0.58	5.1%
Alignement C	0.03	1.13	6.0%
Séparabilité	0.03	1.13	6.0%
Séparabilité reg	0.02	1.13	6.0%
Séparabilité C	0.03	0.91	5.1%

FIGURE 4.9 – Comparaison des valeurs optimales $\hat{\sigma}$ du paramètre σ estimées par chaque critère sur la base *Ionosphere*, et de l'erreur *Leave-One-Out* pour ces valeurs, avec $C = 1$.

résultat en termes d'erreur est moins évident sur la base *Ionosphere* mais la proximité du minimum demeure cependant encourageante. La procédure de régularisation du critère de séparabilité corrige également le biais déviant mais dans une moindre mesure que l'ajustement.

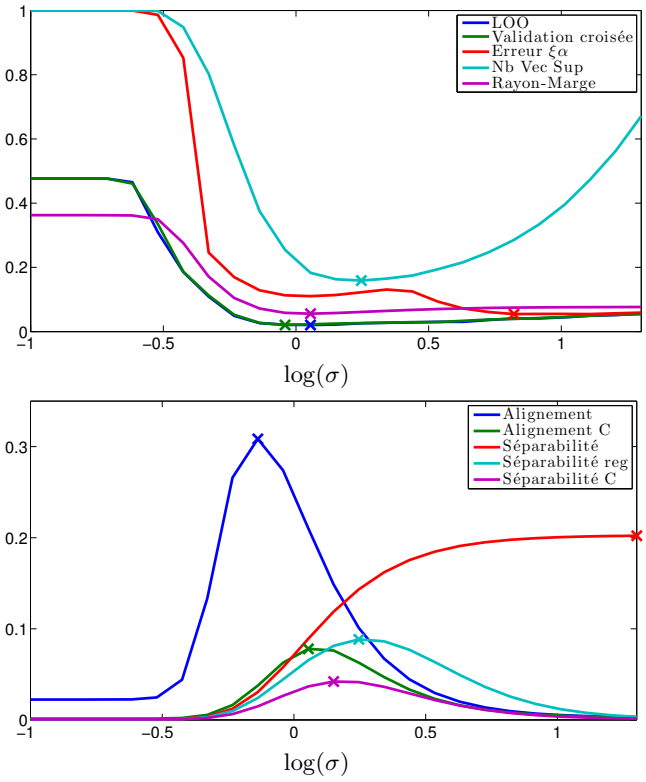
Nous concluons cette expérience par l'application du même protocole sur la base *Lymphoma*. Dans ce cas le facteur C n'est pas fixé d'avance et prend à chaque itération la valeur optimale C_{def} définie par Joachims (voir section 4.5.1). C'est également cette valeur qui est utilisée pour l'ajustement des matrices de Gram. Les résultats sur la base *Lymphoma* sont synthétisés dans la figure 4.11.

La base *Lymphoma* constitue un test plus difficile puisqu'elle contient très peu d'exemples (96) et un nombre élevé de descripteurs (4026) fortement corrélés entre eux. Ceci a pour conséquence de rendre l'allure de la courbe de l'erreur $\xi\alpha$ plus chaotique (dont le minimum à la borne inférieure du maillage, $\hat{\sigma} = 0.1$ est complètement erroné) et réduit considérablement l'ampleur du minimum de l'erreur LOO (vers $\hat{\sigma} = 3.4$) au delà duquel l'erreur ne croît pas vraiment, ce qui complique la tâche pour les autres critères. Ce phénomène est d'ailleurs également dû à la variabilité du facteur C. En effet, l'adaptation du facteur C aux paramètres du noyau (ici le paramètre σ) permet de réduire l'influence de ces derniers sur les performances des SVM. En pratique, donc, si l'usage de C_{def} est préférable, l'affinage des paramètres se trouve alors plus sensible.

Ainsi le critère Rayon-Marge rencontre bien son minimum dans le « bassin » minimal de l'erreur LOO, mais ce minimum reste éloigné du minimum idéal, sans pour autant trop pénaliser les performances. De même, du fait du nombre réduit d'exemples, on constate que l'erreur par validation croisée est plus bruitée que dans les cas précédents et oscille autour de la valeur LOO, tout en restant un bon estimateur.

Les remarques sur les critères proposés demeurent globalement les mêmes. Néanmoins, malgré la divergence apparente des critères de séparabilité (avec ou sans ajustement), il est difficile de juger cette dernière puisque la valeur de σ la plus élevée du maillage produit en pratique une erreur LOO minimale, due au « bassin » de l'erreur LOO que nous avons mentionné. On note cependant que dans ce cas encore l'ajustement de la matrice de Gram permet d'améliorer sensiblement les performances du critère d'alignement.

On constate enfin que les rapports de temps de calcul entre les critères différent par rapport aux

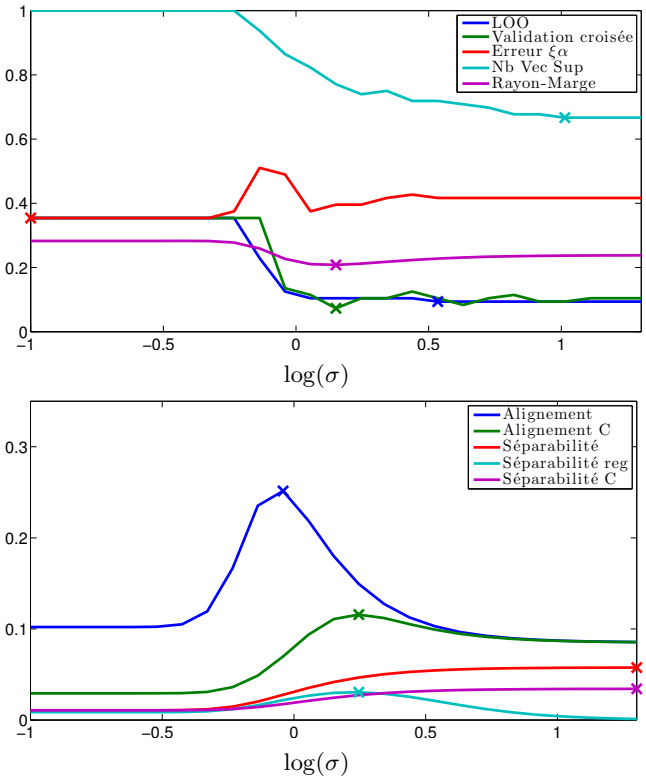


Critère	t	$\hat{\sigma}$	**LOO**
LOO	4.03	1.13	2.0%
Validation croisée	12.5	0.91	2.1%
Erreur $\xi\alpha$	1.53	6.63	4.1%
Nb Vec Sup	1.53	1.76	2.6%
Rayon-Marge	1.99	1.13	2.0%
Alignement	0.64	0.73	2.6%
Alignement C	0.63	1.13	2.0%
Séparabilité	0.62	20.00	5.7%
Séparabilité reg	0.63	1.76	2.6%
Séparabilité C	0.62	1.41	2.2%

FIGURE 4.10 – Comparaison des valeurs optimales $\hat{\sigma}$ du paramètre σ estimées par chaque critère sur la base *Parole/musique*, et de l'erreur *Leave-One-Out* pour ces valeurs, avec $C = 1$.

cas précédent, à cause du nombre élevé de descripteurs qui a une influence directe sur la complexité du calcul de la matrice de Gram. Or, celle-ci n'est calculée qu'une fois dans l'implémentation de Joachims pour le calcul de l'erreur LOO, tandis qu'elle est recalculée à chacune des dix itérations de la validation croisée, ce qui explique que cette dernière soit plus coûteuse dans notre expérience. De même, l'évaluation de la valeur C_{def} est pénalisée par la dimension des exemples, ce qui explique la différence mesurée entre les critères d'alignement et de séparabilité avec et sans ajustement de la matrice de Gram.

Nous avons ainsi introduit plusieurs critères de sélection de noyau, jusqu'ici inusités dans le domaine de la classification audio, de manière à soustraire au processus d'apprentissage la phase habituelle de recherche par maillage. Nous avons montré la pertinence des critères d'Alignement et de Séparabilité des Classe sur des exemples concrets, par rapport aux autres méthodes de la littérature, tout en montrant le gain apporté par notre proposition d'ajustement des matrices de Gram.



Critère	t	$\hat{\sigma}$	**LOO**
LOO	0.59	3.42	9.4%
Validation croisée	0.85	1.41	10.4%
Erreur $\xi\alpha$	0.09	0.10	35.4%
Nb Vec Sup	0.08	10.31	9.4%
Rayon-Marge	0.17	1.41	10.4%
Alignement	0.11	0.91	12.5%
Alignement C	0.19	1.76	10.4%
Séparabilité	0.11	20.00	9.4%
Séparabilité reg	0.11	1.76	10.4%
Séparabilité C	0.20	20.00	9.4%

FIGURE 4.11 – Comparaison des valeurs optimales $\hat{\sigma}$ du paramètre σ estimées par chaque critère sur la base *Lymphoma*, et de l'erreur *Leave-One-Out* pour ces valeurs, avec $C = C_{def}$.

Chapitre 5

Stratégies multi-classes

Sommaire

5.1 Combinaisons de SVM 70
5.1.1 Estimation des probabilités a posteriori 70
5.1.2 Approche *Un contre tous* (OVA) 71
5.1.3 Approche *Un contre un* (OVO) 71
5.1.4 Codes Correcteurs d'Erreur (ECOC) 73
5.1.5 Classification hiérarchique 73
5.1.5.1 Graphe Acyclique Direct (DAGSVM) 73
5.1.5.2 Dendogrammes (DSVM) 74
5.1.5.3 Dendogrammes hybrides 74
5.1.5.4 Probabilités a posteriori par pondérations successives 75
5.2 Reformulation des SVM 76
5.3 Discussion et Conclusion 77

Nous abordons maintenant la question de l'adaptation, pour un problème multi-classes, des Machines à Vecteurs de Support, originellement conçues sur un paradigme de discrimination binaire. Nous emploierons par la suite le terme *multi-classes* pour désigner une classification impliquant plus de deux classes. Ce problème est antérieur à la création des SVM puisqu'il est légitimement posé par toute méthode discriminative, en particulier par la séparation par hyperplan linéaire. Ainsi, de nombreuses méthodes ont été proposées qui permettent de combiner les résultats de classifieurs binaires pour formuler une réponse multi-classes. Nous présenterons celles-ci dans la section 5.1. De nombreuses propositions ont également été faites pour reformuler les Machines à Vecteurs de Support dans un cadre multi-classes. Nous présenterons dans la section 5.2 quelques une des plus citées, qui aboutissent chacune à un nouveau problème d'optimisation. Nous discuterons par la suite, en section 5.3, des mérites comparatifs de ces deux paradigmes (combinaison et reformulation) et justifierons notre choix de nous restreindre au premier. Le lecteur intéressé par cette question particulière pourra consulter les références [95], [197] et [107] pour une comparaison détaillée des différentes approches.

Cette étude nous permettra ainsi de distinguer les méthodes permettant l'évaluation de probabilités a posteriori, qui nous seront utiles par la suite. Nous proposerons une méthodologie de classification basée sur le parcours d'un arbre de classification hybride combinant les approches «Un contre un» et par dendogramme.

On considérera dans la suite de ce chapitre que le label y_i associé à l'exemple $\boldsymbol{x}_i$ prend ses valeurs dans $[1, \ldots, C]$, où C est le nombre de classes impliquées dans le problème.

5.1 Combinaisons de SVM

5.1.1 Estimation des probabilités a posteriori

Nous présentons dans les sections suivantes les différentes stratégies proposées pour combiner les résultats de plusieurs machines bi-classes sur un problème multi-classes.

Toutefois il nous faut avant tout pour cela appliquer sur chaque SVM binaire un traitement destiné à en tirer un résultat probabiliste. On peut se restreindre à la mise en place d'un algorithme n'évaluant que la classe optimale associée à un exemple donné, mais nous verrons au chapitre 8, que l'on apporte un gain significatif aux performances des SVM en appliquant un post-traitement sur leurs résultats. Les post-traitements que nous présenterons exploitent les probabilités a posteriori associées aux classes du problème :

$$p_c(\boldsymbol{x}_i) = p(y_i = c \,|\, \boldsymbol{x}_i).$$

On construira donc, dans la mesure du possible, des algorithmes de combinaison dont le résultat est un vecteur contenant les probabilités a posteriori estimées.

Cependant, les SVM sont construites sur la séparation et non sur l'estimation de probabilités. En effet, la fonction de décision

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x})$$

fournit des valeurs non-bornées et non-calibrées sur la droite réelle, et n'est construite que pour opérer une prise de décision sur le signe de la valeur de sortie : $\hat{y} = \text{sign}(f(\boldsymbol{x}))$. On peut cependant raisonnablement avancer l'hypothèse que plus la valeur est éloignée de 0, plus la classe estimée est fiable. Une des premières méthodes proposées [103] pour transformer la sortie des SVM en valeur probabiliste consiste à modéliser les valeurs de sortie de chaque classe par une gaussienne normalisée de manière à obtenir $P(y = c|f(\boldsymbol{x}) = 0) = 0.5$ pour chacune des classes $c = +1$ et $c = -1$. Néanmoins cette méthode est affaiblie par le fait que l'hypothèse de gaussianité sur les densités de probabilités est rarement respectée.

Partant du constat empirique que les densités de probabilités conditionnelles de chaque classe (pour les valeurs $f(\boldsymbol{x})$) sont exponentielles dans la marge, Platt propose [185] de modéliser la probabilité de la classe positive par une forme sigmoïdale :

$$P\left(y = 1 \,|\, f(\boldsymbol{x})\right) = \frac{1}{1 + \exp\left(A\, f(\boldsymbol{x}) + B\right)}.$$

La probabilité de la classe négative étant implicite :

$$\begin{aligned} P\left(y = -1 \,|\, f(\boldsymbol{x})\right) &= 1 - P\left(y = 1 \,|\, f(\boldsymbol{x})\right) \\ &= \frac{\exp\left(A\, f(\boldsymbol{x}) + B\right)}{1 + \exp\left(A\, f(\boldsymbol{x}) + B\right)}. \end{aligned}$$

Les paramètres A et B sont fixés en maximisant l'estimation de vraisemblance sur les exemples (f_i, y_i) de l'ensemble d'apprentissage (avec $f_i = f(\boldsymbol{x}_i)$) :

$$\min_{A,B} \quad -\sum_{i=1}^{n} t_i \log(p_i) + (1 - t_i) \log(1 - p_i),$$

où l'on a défini les valeurs $t_i = \dfrac{y_i + 1}{2}$ et $p_i = \dfrac{1}{1 + \exp\left(A\, f_i + B\right)}$.
Le problème de minimisation peut être résolu à l'aide de n'importe quelle méthode d'optimisation.

Par la suite on désignera par f^* la composition de la fonction de décision et du post-traitement sigmoïdal, soit :

$$f^*(\boldsymbol{x}) = \frac{1}{1 + \exp\left(A\, f(\boldsymbol{x}) + B\right)}.$$

5.1.2 Approche *Un contre tous* (OVA)

Le premier algorithme multi-classes employé pour les SVM [208][236] est également le plus simple. Il consiste à utiliser un classifieur binaire pour chaque classe. Celui-ci est appris pour discriminer les exemples de la classe des exemples de l'ensemble des autres classes, d'où son nom de *Un contre tous* (ou *One versus All*, OVA). Si l'on désigne par f_c la fonction de décision du classifieur concernant la classe c, l'algorithme OVA choisit donc la classe maximisant les valeurs prises par les fonctions de décisions :

$$\hat{y} = \arg\max_{1 \leq c \leq C} f_c(\boldsymbol{x}).$$

On remarque que le vecteur $[f_1(\boldsymbol{x}), \ldots, f_C(\boldsymbol{x})]$ ne saurait constituer un vecteur de probabilités a posteriori puisque leur somme n'est pas unitaire. On pourra, en normalisant celles-ci, fournir l'estimation suivante des probabilités a posteriori :

$$\hat{p}_c(\boldsymbol{x}) = \frac{f_c(\boldsymbol{x})}{\sum_{k=1}^{C} f_k(\boldsymbol{x})}.$$

Cependant, les fonctions de décisions étant indépendantes, ces probabilités peuvent être très biaisées et n'ont pas de fondement statistique solide.

En pratique, l'algorithme OVA, se révèle très efficace pour la prise de décision, malgré sa simplicité. Cette question, largement débattue dans la littérature, sera discutée en section 5.3. Les défauts majeurs de cet algorithme restent son inadéquation pour l'estimation de probabilités a posteriori, et le fait que des fonctions de décision peuvent être peu fiables si certaines classes disposent de beaucoup moins d'exemples d'apprentissage que les autres.

5.1.3 Approche *Un contre un* (OVO)

Lorsque le nombre de classes est trop élevé, le problème de séparation OVA peut devenir trop complexe, engendrant ainsi des classifieurs mal calibrés. On peut donc espérer mieux contrôler la complexité des surfaces de décision en se restreignant à l'usage de classifieurs appris sur des couples de classes. Les figures 5.1 et 5.2 illustrent cette différence sur un problème simple n'impliquant que 3 classes. On peut ainsi constater que le formalisme *un contre tous* (à gauche) peine à déterminer un plan de séparation adéquat pour la discrimination 1 *vs* 2&3 tandis le problème ne se pose pas dans une approche par paires (à droite).

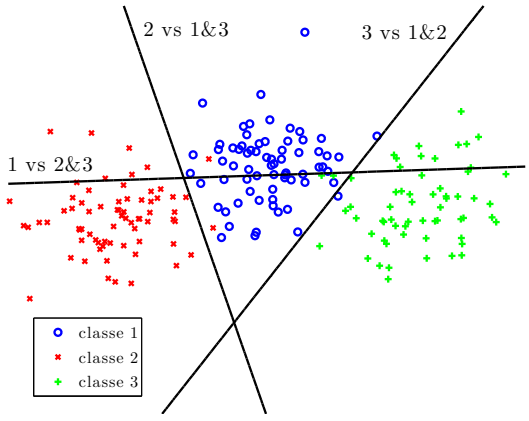


FIGURE 5.1 – Hyperplans de séparation par approche *un contre tous* (OVA) sur 3 classes.

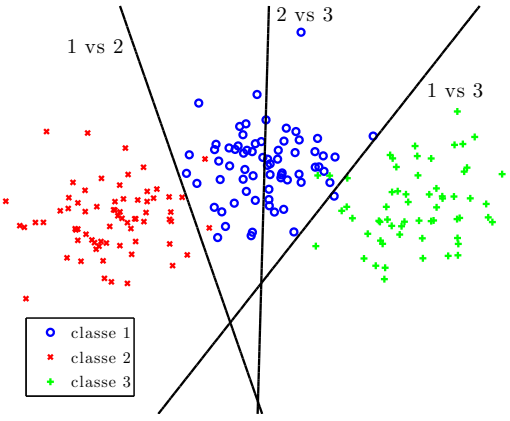


FIGURE 5.2 – Hyperplans de séparation par approche *un contre un* (OVO) sur 3 classes.

L'approche *un contre un*, généralement attribuée à Knerr et al. [126] et Friedman [81], se base sur les résultats des classifieurs séparant chacune des $\frac{C(C-1)}{2}$ paires sur les C classes.

On désigne par f_{kl} la fonction de décision du classifieur appris pour discriminer la classe positive k et la classe négative l ; on peut donc également considérer le classifieur $f_{lk} = -f_{kl}$, pour simplifier

les notations à venir. Friedman propose la règle de décision multi-classes suivante, basée sur un vote majoritaire sur l'ensemble des classes :

$$\hat{y} = \underset{1\leq k\leq C}{\arg\max} \sum_{l=1}^{C} H\left(f_{kl}(\boldsymbol{x})\right), \tag{5.1}$$

où H est la fonction de Heaviside. La stratégie du vote majoritaire associée aux classifieurs par paires est employée pour la première fois sur les SVM par Kressel [127], et est depuis largement reprise dans la littérature.
Toutefois ce paradigme présente deux défauts majeurs :

- De nombreuses régions de l'espace de décision (où évolue le vecteur $[f_{kl}(\boldsymbol{x})]_{k>l}$) sont indécidables, parculièrement lorsque le nombre de classes est réduit. Par exemple, dans un problème à 3 classes, si $f_{12}(\boldsymbol{x}) > 0$, $f_{23}(\boldsymbol{x}) > 0$ et $f_{31}(\boldsymbol{x}) > 0$, toutes les classes maximisent le critère. Le cas d'un problème à 4 classes où 2 classes maximisent le critère, sans configuration cyclique, est également plausible et problématique.
- La présence de la fonction de Heaviside introduit une forte discontinuité dans le critère. Il est donc impossible d'espèrer en tirer une estimation fiable des probabilités a posteriori.

Hastie et Tibshirani ont proposé [103] un algorithme permettant d'estimer les probabilités a posteriori sur une approche OVO, qui résout ainsi ces deux problèmes. Celui-ci se base sur une procédure d'optimisation sur les probabilités a posteriori p_i, minimisant la distance entre les résultats probabilisés $f^*_{kl}(\boldsymbol{x})$ (ici notés r_{kl}) des classifieurs et une estimation μ_{kl} de ces valeurs, calculée à partir des $\boldsymbol{p} = [p_i]_{1\leq i\leq C}$. Ainsi si on définit μ_{kl} comme l'estimée de la probabilité conditionnelle de la classe k, sachant que la classe est k ou l :

$$\mu_{kl} = \mathrm{E}(r_{kl}) = \frac{p_k}{p_k + p_l},$$

soit,

$$\log \mu_{kl} = \log(p_k) - \log(p_k + p_l).$$

On souhaite mettre à jour les p_i de manière à minimiser la distance entre les r_{kl} (pour rappel $r_{kl} = f^*_{kl}(\boldsymbol{x})$) et les μ_{kl}. La mesure choisie par Hastie et Tibshirani est la distance moyenne de Kullback-Leibler entre les deux vecteurs, soit :

$$\ell(\boldsymbol{p}) = \sum_{k<l} n_{kl} \left[r_{kl} \log \frac{r_{kl}}{\mu_{kl}} + (1 - r_{kl}) \log \frac{1 - r_{kl}}{1 - \mu_{kl}} \right],$$

celle-ci est pondérée par les valeurs n_{kl} qui représentent le nombre d'exemples utilisés pour l'apprentissage du classifieur f_{kl}. Ce terme permet de compenser les effets d'une disproportion entre les données disponibles pour les différentes classes, et peut être uniforme sur l'ensemble des classes, dans le cas de classes à peu près équilibrées, sans nuire aux performances de l'algorithme.

Du calcul du gradient du critère $\ell(\boldsymbol{p})$, on déduit la méthode d'optimisation PWC (*Pair-Wise Classification*) synthétisée par l'algorithme 2. En pratique on utilise pour l'initialisation des probabilités $\hat{p}_k$ les valeurs normalisées de la procédure de Friedman (équation 5.1), soit $\hat{p}_k = \frac{\sum_{l=1}^{C} H\left(f_{kl}(\boldsymbol{x})\right)}{\sum_{k=1}^{C}\sum_{l=1}^{C} H\left(f_{kl}(\boldsymbol{x})\right)}$.
La convergence est en général très rapide. Une alternative a cependant été proposée [101] qui permet la détermination des probabilités a posteriori sans itération. On pourra toutefois noter que Hastie et Tibshirani précisent qu'une seule itération est suffisante si l'on ne s'intéresse qu'à la probabilité majoritaire (c'est-à-dire la seule décision multi-classes).

Plusieurs auteurs formulent la critique que l'approche OVO prend en compte les résultats de tous les classifieurs, y compris ceux pour lesquels la classe d'un exemple donné n'est pas concernée, ce qui introduit une part importante d'informations non-pertinentes qui peut pénaliser le processus. Ainsi, l'algorithme O-PWC se propose [136] de combiner les résultats des C algorithmes PWC pour chacun desquels seuls les couples impliquant une classe donnée sont pris en compte (au travers des poids n_{kl}). Garcia-Pedrajas et Ortiz-Boyer [84] proposent également de combiner les approches OVO et OVA, sans toutefois réellement justifier leur démarche d'un point de vue théorique, ni mettre en valeur dans leurs résultats un réel gain de performances.

Algorithme 2 PWC, *PairWise Classification*

$r_{kl} = f^*_{kl}(\boldsymbol{x})$
Initialiser les estimées $\hat{p}_k$.
Initialiser les $\hat{\mu}_{kl}$:
$$\hat{\mu}_{kl} = \frac{p_k}{p_k + p_l}$$
répéter
Mise à jour des $\hat{p}_k$:
$$\hat{p}_k \leftarrow \hat{p}_k \frac{\sum_{k \neq l} n_{kl} r_{kl}}{\sum_{k \neq l} n_{kl} \hat{\mu}_{kl}}.$$
Normalisation des probabilités :
$$\hat{\boldsymbol{p}} \leftarrow \hat{\boldsymbol{p}} / \sum \hat{p}_k$$
Mise à jour des $\hat{\mu}_{kl}$:
$$\hat{\mu}_{kl} = \frac{p_k}{p_k + p_l}$$
jusqu'à convergence des $\hat{p}_k$

5.1.4 Codes Correcteurs d'Erreur (ECOC)

Dietterich et Bakiri développent dans [65] un cadre plus général pour la combinaison de classifieurs binaires. Sans imposer de contrainte a priori sur la collection de classifieurs exploités, ils proposent de baser la fusion de résultats sur un principe inspiré de la théorie des codes correcteurs d'erreurs. On définit ainsi une matrice binaire $\boldsymbol{M} = [m_{cn}] \in \mathcal{M}_{C,N}(\{-1, 1\})$ contenant C lignes (où C est le nombre de classes) et N colonnes (N étant le nombre de classifieurs binaires f_n impliqués). Chaque vecteur ligne $\boldsymbol{M}_c$ de la matrice $\boldsymbol{M}$ est un *mot-code* pour une classe donnée. Chaque exemple $\boldsymbol{x}$, après classification par les N classifieurs, se voit attribuer un vecteur $\boldsymbol{f} = [f_1(\boldsymbol{x}), \ldots, f_N(\boldsymbol{x})]$ regroupant les résultats des fonctions de décision.

Le principe de la méthode ECOC (*Error Correcting Output Codes*) consiste à attribuer à l'exemple $\boldsymbol{x}$ la classe minimisant la distance L^1 (distance de Manhattan) entre son mot-code $\boldsymbol{M}_c$ et le mot-code $\boldsymbol{f}$ de l'exemple :

$$\hat{y} = \underset{1 \leq c \leq C}{\arg\min}\, L1(\boldsymbol{f}, \boldsymbol{M}_c) = \underset{1 \leq c \leq C}{\arg\min} \sum_{n=1}^{N} |f_n(\boldsymbol{x}) - m_{cn}|.$$

Allwein et al. [14] apportent un regard intéressant sur les ECOC en étendant l'espace des matrices de codes aux matrices "ternaires", soit $m_{cn} \in \{-1, 0, 1\}$, où la valeur $m_{cn} = 0$ introduite représente le fait que le classifieur n n'apporte aucune information sur la classe c. Cette extension permet d'inclure les méthodes OVO et OVA dans le cadre théorique des ECOC.

Le point central de la méthode ECOC est la définition de la matrice de codes $\boldsymbol{M}$, qui doit être définie avec soin pour représenter le problème posé. Cependant, en définitive, l'approche ECOC apporte peu par rapport aux approches OVA et OVO, dans un cas comme le notre, où le nombre de classes est assez restreint. On n'entreprendra donc pas d'étude expérimentale sur celle-ci.

5.1.5 Classification hiérarchique

Les méthodes présentées jusqu'ici sont basées sur des combinaisons de classifieurs indépendants qui peuvent être traités en parallèle. Nous présentons ici quelques méthodes multi-classes où l'ordre des classifieurs est déterminant. Celles-ci sont basées sur une structure d'arbre (graphes connexes acycliques) dont les nœuds représentent les différents classifieurs ; on parlera également de *classification hiérarchique*.

5.1.5.1 Graphe Acyclique Direct (DAGSVM)

La première contribution basée sur les graphes de décision s'appuie sur une structure particulière proposée par Platt et al. [186], qui réduit le nombre de classifications dans le processus de décision. On associe au nœud racine l'ensemble des classes. Chaque nœud décrit un classifieur portant sur la première et la dernière des classes associées. La règle de décision implique deux branches, associées

chacune à la négation d'une classe, qui est ôtée de la liste du noeud fils. La figure 5.3 clarifie cette structure pour un exemple à 4 classes.

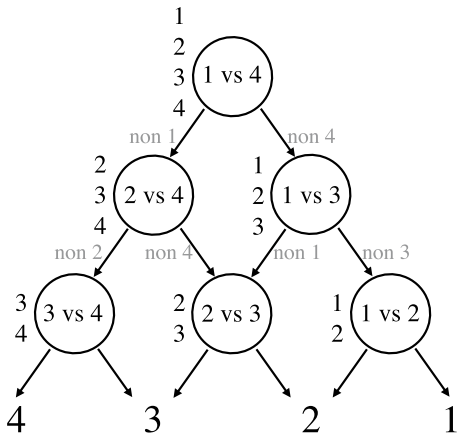


FIGURE 5.3 – Structure du Graphe Acyclique Direct (DAGSVM) défini pour un problème à 4 classes. À chaque nœud est associé un discriminateur (n vs m) et la liste des classes restantes (agencée verticalement à gauche du nœud).

Les auteurs nomment ce modèle DAGSVM (pour *Direct Acyclic Graph SVM*). Celui-ci repose sur l'élimination d'une classe à chaque nœud de discrimination. La feuille terminale de l'arbre indique l'unique classe restante, résultat de la classification. On voit que cette structure implique exactement les mêmes classifieurs par paires que l'approche OVO. Si l'ordre des éliminations successives a théoriquement une influence sur les résultats, les auteurs avancent, d'après leurs observations empiriques, que celle-ci est très modérée et non systématique.

Cette approche a le mérite de se baser sur les classifieurs par paires, supposés plus robustes que les classifieurs OVA, tout en n'impliquant que C classifications pour la prise de décision, au lieu des $C(C-1)/2$ nécessaires pour l'approche OVO. Néanmoins, comme la méthode OVO avec vote majoritaire, elle ne fournit aucune estimation des probabilités a posteriori.

5.1.5.2 Dendogrammes (DSVM)

La structure d'arbre de classification permet également la mise en place d'un algorithme [21] par discriminations successives sur des unions de classes, à l'aide d'un *dendogramme*. Les auteurs du DSVM (*Dendogram SVM*) regroupent itérativement les classes par *clustering bottom-up* sur les exemples d'apprentissage. Le processus, basé sur une mesure de proximité entre les groupes de classes, permet ainsi la construction d'un dendogramme décrivant le processus *top-down* de classification, illustré par l'exemple figure 5.4.

La structure du dendogramme renseigne sur les classifieurs nécessaires à la classification. Ceux-ci associent aux nœuds les plus profonds des paires de classes, et aux autres nœuds des paires impliquant des unions de classes (qui regroupent les exemples de plusieurs classes). On trouve une formulation équivalente sous le nom de « *Half against Half* » [130], où le dendogramme est construit sur un paradigme *top-down*, où chaque nœud de classification est choisi de manière à équilibrer les deux groupes de classes discriminées.

Cette approche implique un très net gain en complexité puisque la classification n'implique que $\lceil \log_2 C \rceil$ discriminations successives. Toutefois elle ne fournit qu'un indice de classe estimée et ne permet pas d'estimer les probabilités a posteriori.

5.1.5.3 Dendogrammes hybrides

Nous proposons une extension du cadre de l'approche par dendogramme en combinant ce dernier à l'approche *One-vs-One*. On peut en effet étendre l'arbre de classification à des arbres non-binaires en traitant d'éventuels nœuds non-binaires (c'est-à-dire à plus de deux branches filles) par une approche multi-classes non hiérarchisée. Le cadre *One-vs-One* est ici le mieux indiqué puisqu'il nous permet, contrairement à l'approche *One-vs-All*, d'estimer les probabilités a posteriori des classes

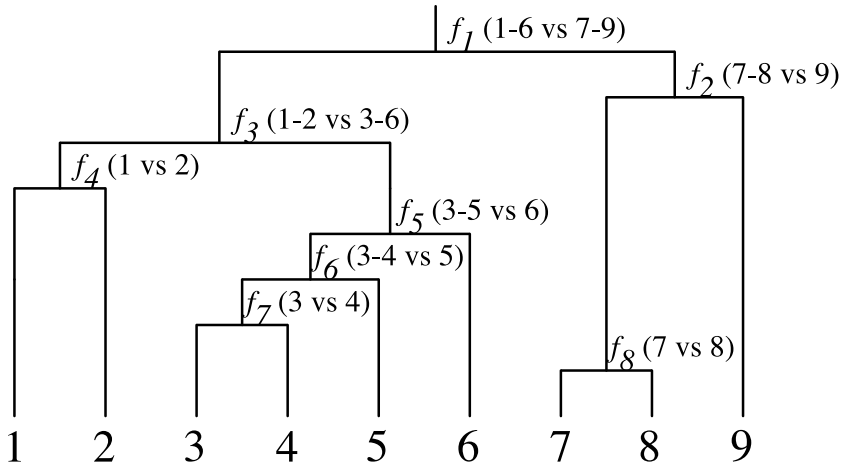


FIGURE 5.4 – Dendogramme de classification *top-down* sur un exemple de problème à 9 classes. Chaque noeud du dendogramme est associé à un classifieur binaire f_n.

impliquées.

Cette approche constitue ainsi ce que nous appelons le *dendogramme hybride*.

5.1.5.4 Probabilités a posteriori par pondérations successives

La plupart des méthodes multi-classes présentées jusque-là se focalisent sur le problème de l'estimation de la classe des exemples, laissant de côté la question des probabilités a posteriori. Nous proposons une méthode simple permettant d'estimer ces probabilités sur l'approche par dendogramme.

Celle-ci est basée sur un parcours récursif de l'arbre de classification. On supposera qu'à chaque nœud est associé un index n unique et un classifieur f_n. Le résultat probabilisé du classifieur f_n sur l'exemple $\boldsymbol{x}$ est noté $f_n^*(\boldsymbol{x})$. Le nœud racine est associé à l'index 1. La figure 5.5 fournit un exemple de dendogramme indexé pour un problème à 6 classes (les index de nœuds sont en gris clair et les index de classes en bleu).

Le déroulement de notre méthode sur un exemple $\boldsymbol{x}$ est le suivant :

- Le nœud racine (d'indice 1) reçoit une probabilité d'entrée égale à $p_{in}^1 = 1$.
- Tout nœud fils n reçoit une probabilité p_{in}^n de son nœud parent.
- Le nœud n produit une probabilité de sortie $p_{out,\pm}^n$ pour les 2 classes traitées par le classifieur f_n, pondérée par la probabilité d'entrée :

$$\begin{aligned} p_{out,+}^n &= p_{in}^n \, f_n^*(\boldsymbol{x}) \\ p_{out,-}^n &= p_{in}^n \, (1 - f_n^*(\boldsymbol{x})) \, . \end{aligned}$$

- Si le nœud n a deux nœuds fils m_+ et m_-, ses probabilités de sorties sont transmises en entrée des nœuds fils :

$$\begin{aligned} p_{in}^{m_+} &= p_{out,+}^n \\ p_{in}^{m_-} &= p_{out,-}^n . \end{aligned}$$

La concaténation des probabilités de sortie de toutes les feuilles constitue l'estimée des probabilités a posteriori. Ainsi on obtient sur l'exemple de la figure 5.5 :

$$\begin{aligned} \hat{\boldsymbol{p}}(\boldsymbol{x}) &= \left[p_{out,+}^4, p_{out,-}^4, p_{out,+}^5, p_{out,-}^5, p_{out,+}^3, p_{out,-}^3 \right] \\ &= \left[f_1 f_2 f_4, f_1 f_2 (1 - f_4), f_1 (1 - f_2) f_5, f_1 (1 - f_2)(1 - f_5), (1 - f_1) f_3, (1 - f_1)(1 - f_3) \right] , \end{aligned}$$

où l'on utilise la notation allégée $f_n = f_n^*(\boldsymbol{x})$.

Cet algorithme consiste tout simplement en une mise à jour récursive des probabilités par pondérations successives. Le processus garantit par ailleurs que le vecteur estimé a bien une somme

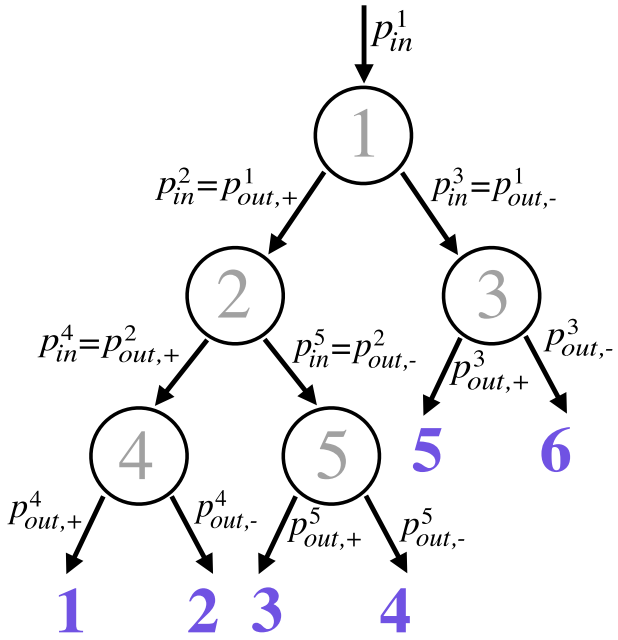


FIGURE 5.5 – Exemple d'arbre de classification pour l'estimation de probabilités a posteriori. La figure illustre la transmission des probabilités de sorties d'un nœud vers la probabilité d'entrée des nœuds fils. Les index de nœuds sont en gris clair, les index de classes en bleu aux feuilles de l'arbre.

unitaire. Il est par ailleurs applicable sur les dendogrammes hybrides puisque l'approche *one-vs-one* produit également des probabilités a posteriori.

Etude de l'application aux DAGSVM

L'application à l'approche DAGSVM est moins cohérente. En effet, le graphe n'ayant pas une structure d'arbre (puisqu'un nœud peut avoir plusieurs parents), il existe plusieurs estimations possibles pour certaines classes. Ainsi, en se basant sur l'exemple de la figure 5.3, on obtient les estimées suivantes :

$$\hat{p_1} = (1-f_{14})(1-f_{24})(1-f_{34}) \qquad \hat{p_4} = f_{14}f_{13}f_{12}$$

$$\begin{aligned}\hat{p_2} &= (1-f_{14})(1-f_{24})f_{34} \\ &= (1-f_{14})f_{24}(1-f_{23}) \\ &= f_{14}(1-f_{13})(1-f_{23})\end{aligned} \qquad \begin{aligned}\hat{p_3} &= (1-f_{14})f_{24}f_{23} \\ &= f_{14}(1-f_{13})f_{23} \\ &= f_{14}f_{13}(1-f_{12})\end{aligned}$$

De plus les probabilités estimées ne sont pas de somme unitaire. Il est bien sûr possible de les normaliser mais ce constat affaibli la pertinence de l'estimation pour l'approche DAGSVM.

5.2 Reformulation des SVM

Nous avons décrit plusieurs méthodes pour combiner les résultats de différents classifieurs bi-classes pour une tâche multi-classes. Plusieurs auteurs ont cependant tenté de reformuler directement les Machines à Vecteurs de Support sur ce type de problèmes. Cette tâche est loin d'être évidente puisque, comme nous l'avons vu, les SVM reposent sur le principe de la maximisation de la marge, dont la définition même repose sur le principe de la séparation linéaire. On trouve ainsi plusieurs alternatives dans la littérature, dont les références [197] et [95] couvrent un large éventail. Nous présentons ici brièvement les principales d'entre elles.

La première approche est introduite en 1998 par Weston et Watkins [246]. Elle consiste à déterminer simultanément les C fonctions de décision « un contre tous » f_c :

$$f_c(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_{c,i}\, y_i\, k(\boldsymbol{x}, \boldsymbol{x}_i) + b_c.$$

L'idée de base est de contraindre les variables d'écart, non plus indépendamment pour chaque fonction, mais relativement aux résultats des autres. Ainsi, le problème comprend $n \times (C-1)$ variables d'écart ξ_{ic} (où n est le nombre d'exemples) relatives à chaque classe $c \neq y_i$, où y_i est la classe de l'exemple $\boldsymbol{x}_i$, avec les contraintes suivantes :

$$f_{y_i}(\boldsymbol{x}_i) \geq f_c(\boldsymbol{x}_i) + 1 - \xi_{ic} \qquad \text{et} \qquad \xi_{ic} \geq 0$$

Si la formulation du problème est simple, elle pose plusieurs difficultés pour sa mise en application. En premier lieu la formulation du problème dual (non présentée ici) ne permet pas de se débarrasser des constantes introduites dans le problème primal, comme c'est le cas pour les SVM bi-classes. De plus, les auteurs ne proposent aucune implémentation efficace de l'algorithme d'optimisation, qui ne peut tirer partie des techniques permettant de rendre efficace l'apprentissage SVM traditionnel. Le problème posé est donc beaucoup plus complexe, du fait de la multiplication du nombre de variables d'écart, et donc difficilement applicable sur des données réelles.

Bredensteiner et Bennett ont proposé, sans lien avec Weston et Watkins, une approche [37] basée sur les mêmes contraintes liant les fonctions de décisions entre elles :

$$f_{y_i}(\boldsymbol{x}_i) \geq f_c(\boldsymbol{x}_i) + 1 - \xi_{ic}.$$

Soit, dans le cas linéaire :

$$(\boldsymbol{w}_{y_i} - \boldsymbol{w}_c)^T\, \boldsymbol{x}_i \geq (b_c - b_{y_i}) + 1 - \xi_{ic}.$$

Cette relation les mène à proposer la valeur $\frac{2}{\|\boldsymbol{w}_c - \boldsymbol{w}_d\|}$ comme mesure de séparabilité entre les classes c et d. Ils suggèrent donc la minimisation du terme $\|\boldsymbol{w}_c - \boldsymbol{w}_d\|$ sur toutes les paires (c,d), regularisée par le terme $\sum_{c=1}^{C} \|\boldsymbol{w}_c\|^2$.L'expression duale du problème permet de substituer aux produits scalaires la fonction noyau. L'équivalence formelle avec l'approche de Weston et Watkins a par la suite été démontrée par différents auteurs [95][107].

Crammer et Singer proposent [55] par la suite une approche simplifiant considérablement le modèle de Weston et Watkins. Au lieu de définir une pénalité ξ_{ic} pour chaque couple (i,c), une seule pénalité est définie pour la valeur $f_c(\boldsymbol{x})$ maximale. Ainsi on retrouve une unique variable d'écart pour chaque exemple. Le problème d'optimisation s'exprime alors de la manière suivante :

$$\begin{aligned} \min_{f_1,\ldots,f_C} \quad & \tfrac{1}{2} \textstyle\sum_{c=1}^{C} \|\boldsymbol{w}_c\|^2 + C_{pen} \sum_{i=1}^{n} \xi_i \\ \text{sous les contraintes} \quad & f_{y_i}(\boldsymbol{x}_i) \geq f_c(\boldsymbol{x}_i) + 1 - \xi_i. \\ & \xi_i \geq 0 \end{aligned}$$

Les auteurs proposent en outre un algorithme de décomposition du problème d'optimisation, permettant une implémentation efficace de leur approche multi-classes.

5.3 Discussion et Conclusion

Le bilan essentiel de la courte présentation précédente est qu'il n'existe pas à l'heure actuelle de formulation multi-classes des SVM qui fasse consensus au sein de la communauté. Celle-ci se limite encore essentiellement aux méthodes par combinaisons présentées dans la section 5.1, beaucoup plus simples à mettre en place et bien moins coûteuses en temps de calcul.

En vérité, on peine à trouver dans la littérature des résultats qui justifient l'usage des SVM reformulées plutôt que celui des méthodes par combinaisons. Les résultats expérimentaux de Weston et Watkins [246] ne montrent pas d'amélioration des performances par rapport à une simple approche OVA ou OVO, et se focalisent surtout sur la réduction du nombre de vecteurs de support.

On trouvera de plus dans [107] un test comparatif impliquant les méthodes de Weston & Watkins, de Crammer & Singer, OVO, OVA et les DAGSVM, duquel il ne ressort pas d'avantage particulier pour les méthodes reformulées.

Rifkin et Klautau [197] ont également écrit un plaidoyer très didactique en faveur de la méthode un contre tous (OVA), souvent dédaignée dans la littérature. Reproduisant avec rigueur les expériences de nombreux articles, ils montrent que cette méthode, de loin la plus simple de toutes, est tout à fait comparable en performances aux alternatives considérées (OVA, ECOC avec diverses configurations, ainsi que les méthodes reformulées présentées précédemment). En définitive, leur constat est que les méthodes se valent globalement, si l'on prend la peine d'affiner avec soin les paramètres des classifieurs SVM impliqués dans le processus. Cette conclusion justifie donc l'attention particulière que nous portons à cette question dans le chapitre 4.

Les implémentations libres de méthodes par SVM réformulées sont pour l'instant rares et très coûteuses en temps de calcul. Aussi, au vue des résultats expérimentaux énoncés ci-dessus, nous n'avons porté notre attention que sur les méthodes multi-classes par combinaisons de SVM.

Il est difficile d'évaluer les méthodes multi-classes en dehors d'un contexte applicatif, aussi nous reportons l'évaluation comparée de ces dernières au chapitre d'évaluation 10, dont la section 10.3 présente une comparaison de différentes taxonomies de classification sur des corpus audio.

Toutefois, le résultat théorique essentiel de ce chapitre réside dans notre proposition d'une méthode de classification hybride combinant l'approche *one-vs-one* aux arbres de classification hiérarchiques, couplé à une procédure d'estimation des probabilités a posteriori. Cette dernière contribution démarque l'approche proposée de la plupart des méthodes de l'état de l'art (comme l'approche *one-vs-all* ou les DAGSVM), qui ne permettent pas d'estimer ces probabilités.

Deuxième partie

Caractérisation audio

Introduction de la partie II

Nous avons présenté dans la partie précédente les Machines à Vecteurs de Support, qui constituent le cœur de notre système de classification, ainsi que les questions relatives à leur mise en œuvre optimale sur un problème impliquant plusieurs classes. Ce faisant, nous avons laissé de côté la nature des vecteurs exemples exploités pour l'apprentissage, qui revient à optimiser une surface de séparation dans un espace donné.

Pourtant la question du choix de cet espace est essentielle. En effet, aussi minutieux que puisse être l'affinage des SVM, leurs performances demeurent en définitive bornées par l'erreur de Bayes :

$$P_B = \int_{\mathbb{R}^d} \left[1 - P(c_B(\boldsymbol{x})|\boldsymbol{x})\right] p(\boldsymbol{x}) \boldsymbol{dx},$$

où c_B est la classe déterminée par la règle de décision de Bayes, qui minimise le risque fonctionnel :

$$c_B(\boldsymbol{x}) = \underset{1 \leq c \leq C}{\arg\max}\, P(c|\boldsymbol{x}).$$

On en déduit que plus les probabilités conditionnelles $P(c|\boldsymbol{x})$ sont uniformes, autrement dit plus les régions associées aux différentes classes se chevauchent dans l'espace d'entrée, plus l'erreur minimale est importante. Les performances de classification sont donc fortement gouvernées par le choix des paramètres caractérisant les exemples.

Nous explorerons dans cette partie la question de la caractérisation des données. L'introduction de la donnée audio, dans le chapitre 6, nous permettra dans un premier temps de dresser une architecture globale de notre système pour la mise en œuvre des Machines à Vecteurs de Support sur le problème particulier de la classification audio. Après avoir présenté la façon dont on extrait à partir d'un signal une collection d'exemples associés à des instants donnés, nous présenterons dans le chapitre une large collection de descripteurs audio, choisis pour leur aptitude supposée à discriminer au mieux les classes étudiées (à savoir la parole, la musique et le chant).

Nous verrons ensuite que s'il est préférable de disposer d'un large éventail de descripteurs afin de caractériser au mieux les classes étudiées, en fournir un trop grand nombre aux SVM présente de nombreux défauts. Aussi, afin de réduire le nombre de descripteurs considérés pour l'apprentissage et d'en garder le sous-ensemble le plus pertinent, nous étudierons le problème de la sélection automatique de descripteurs dans le chapitre 7. Puis nous présenterons plusieurs algorithmes à cet effet, dont certains sont le fruit du travail de cette thèse.

Chapitre 6

Application sur un signal audio

Sommaire

6.1 Architecture du système de classification 83
6.2 Analyse du signal en trames 84
6.3 Intégration temporelle 84
6.4 Normalisation des descripteurs 86
6.5 Liste des descripteurs employés 87
6.6 Discussion 89

Ayant introduit la théorie des Machines à Vecteurs de Support, nous nous intéressons maintenant à leur mise en œuvre sur le problème spécifique de la classification audio. Après avoir présenté l'architecture globale du système dans la section 6.1, nous verrons en section 6.2 comment le signal audio est traité en entrée pour se conformer au cadre théorique exposé précédemment. La constitution des exemples d'apprentissage pour les SVM se fait par le calcul de descripteurs audio, dont nous présenterons en section 6.5 le panel choisi pour caractériser au mieux les classes mises en jeu. Une courte discussion sur ces derniers (section 6.6) nous permettra de mettre en évidence plusieurs modalités dominantes de description du signal, dont les descripteurs sont fortement corrélés. Ce constat nous conduira donc à nous intéresser dans le chapitre suivant au problème de la sélection automatique de descripteurs.

6.1 Architecture du système de classification

L'architecture du système mis en place est résumée dans la figure 6.1.

Nous avons traité, dans les chapitres 3 et 4, de la question de l'apprentissage des SVM ainsi que de la sélection du noyau optimisant les performances par rapport à un ensemble d'apprentissage. Nous aborderons dans ce chapitre la constitution de l'ensemble d'apprentissage, par l'extraction de descripteurs audio calculés sur le signal audio du corpus d'apprentissage après un découpage en trames.

De par la nature discriminative des SVM, nous avons vu dans le chapitre 5 qu'il est nécessaire de combiner plusieurs discriminateurs dans une situation impliquant plus de deux classes. Le processus hiérarchique de combinaison des classifieurs est synthétisé dans une taxonomie multi-classes, que l'on retrouve en haut à gauche de la figure 6.1. Cette taxonomie implique un ensemble de classifieurs, chacun étant destiné à discriminer une paire de classes donnée. Sur chacune de ces paires on appliquera donc, de manière indépendante, les traitements contenus dans l'encadré jaune de la figure.

Ainsi, nous verrons dans le chapitre 7 qu'à la phase d'extraction générale des descripteurs succède une sélection des descripteurs les plus pertinents, qui est bien sûr propre à la paire de classes considérée. Un modèle SVM est par la suite appris sur les descripteurs sélectionnés.

L'application du système sur le corpus audio d'évaluation (ou de test) suivra la même séparation en processus distincts propres à chaque paire. Ainsi on n'extraira cette fois-ci que les descripteurs sélectionnés, puis on classifiera les exemples inconnus au moyen du modèle SVM appris. Ensuite,

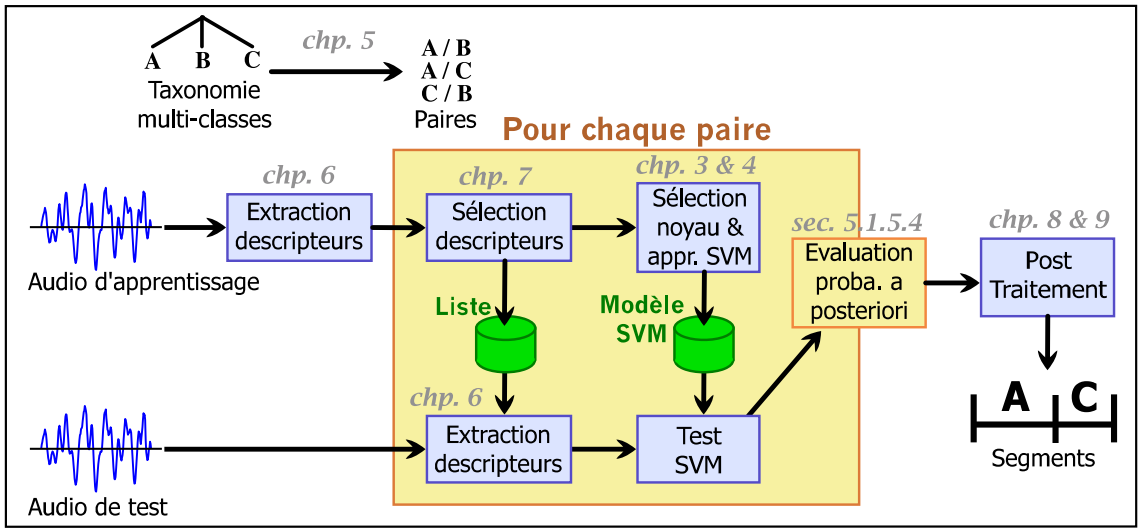


FIGURE 6.1 – Architecture générale du processus de classification audio par combinaison de SVM.

en suivant la méthode présentée dans la section 5.1.5.4, on estime les probabilités a posteriori qui nous permettent d'appliquer l'un des procédés de post-traitement dynamiques qui seront présentés dans les chapitres 8 et 9.

6.2 Analyse du signal en trames

La tâche de classification porte sur un signal audio numérique fini de L échantillons. On supposera par la suite que celui-ci est échantillonné à $f_s = 16$ kHz. Le signal est représenté par un vecteur $\boldsymbol{x}$ de L échantillons : $\boldsymbol{x} = [x_1, \ldots, x_L]^T$. La valeur de chaque échantillon indépendamment des autres n'apporte aucune information concernant les propriétés du signal à un instant donné. On doit donc considérer un ensemble de N échantillons successifs appelé *trame*. On admettra qu'un signal audio est stationnaire sur une durée inférieure à 40 ms. Afin d'optimiser le calcul de la FFT (Transformation de Fourier Rapide), on choisira donc la puissance de 2 la plus élevée satisfaisant cette contrainte, soit $N = 2^{\lfloor \log_2(f_s \times 0.04) \rfloor} = 512$ ce qui correspond à des trames de 32 ms.

On utilise généralement un pas d'avancement de R échantillons entre les trames, qui est inférieur à la taille de la fenêtre, afin d'accroître la précision temporelle de la classification ; on parle alors de trames *chevauchantes*. On choisit ici $R = \frac{N}{2} = 256$.

Le signal sera donc caractérisé par un ensemble de valeurs calculées sur ces trames temporelles. Nous verrons qu'une partie implique le spectre fréquentiel défini par l'analyse de Fourier. On calcule donc les 256 composantes d'amplitude a_k et de phase ϕ_k associées à chaque *bin* fréquentiel d'indice k, après pondération de la trame par une fenêtre de Hamming.

La classification sur des exemples caractérisant des trames temporelles constitue le paradigme de base de l'apprentissage statistique. Cette approche est généralement appelée *sac de trames* (*bag of frames*).

6.3 Intégration temporelle

Nécessaire pour caractériser les propriétés instantanées du signal, le découpage en trames courtes reste cependant lacunaire puisque nombre de phénomènes acoustiques n'ont de sens que sur une portée temporelle plus longue ; par exemple, en musique et en parole, le trémolo et le vibrato sont des grandeurs impliquant des modulations d'amplitude ou de fréquence sur une durée de l'ordre d'une seconde. On peut d'ailleurs montrer que la durée nécessaire à un humain pour la reconnaissance de genre musical est de l'ordre de 0.5 à 3 secondes [184]. On trouvera dans [243] une étude comparative de plusieurs horizons temporels pour la classification de genres musicaux.

La prise en compte d'une échelle temporelle plus étendue se fait généralement au travers de deux moyens [155] : soit par l'inclusion de descripteurs dit « long-terme », directement calculés sur des trames longues de l'ordre d'une seconde, également nommées *fenêtres de texture* par Tzanetakis et Cook [229] et d'autre auteurs [41][164], soit par l'intégration statistique des valeurs des descripteurs court-terme sur les trames longues. Nous présentons ci-dessous ces deux possibilités que nous exploitons conjointement.

Trames longues

Nous définissons, par opposition aux *trames courtes* introduites précédemment, des *trames longues* d'une durée d'une seconde, temps correspondant au consensus global sur la fenêtre de texture citée précédemment. Afin de pouvoir synchroniser ces deux modalités, les trames longues ont pour taille le nombre exact d'échantillons impliqués dans une série de N_{mul} trames courtes, soit la longueur N_l suivante :

$$N_l = (N_{\text{mul}} - 1)R + N.$$

L'avancement R_l entre deux trames longues est également choisi de manière à ce que chaque trame longue soit synchronisée avec le début d'une trame courte, d'où :

$$R_l = R_{\text{mul}}R,$$

où R_{mul} est le nombre de trames courtes d'avancement entre chaque trame longue. Afin d'obtenir des trames longues de l'ordre d'une seconde, on choisit $N_{\text{mul}} = 60$ (soit $N_l/f_s = 0.976$ s). Les trames se chevauchent, comme pour les trames courtes, d'environ 50%, soit $R_{\text{mul}} = 30$. On parlera respectivement, pour les descripteurs calculés sur les trames courtes et longues, de descripteurs court-terme et long-terme.

La figure 6.2 illustre la synchronisation induite par les grandeurs introduites.

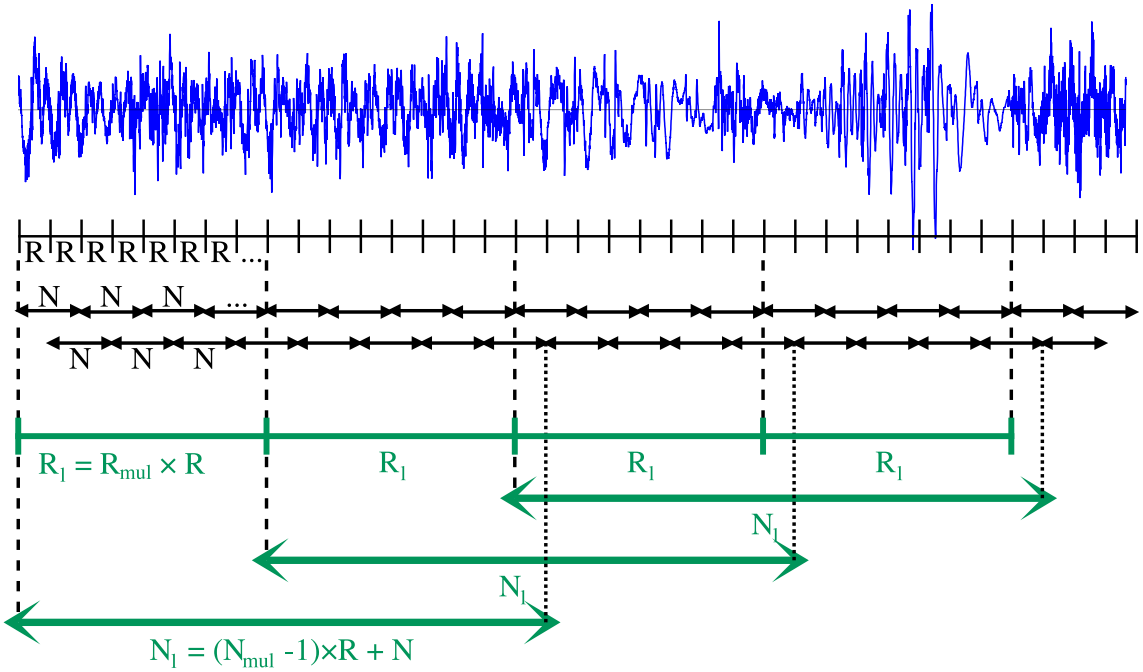


FIGURE 6.2 – Représentation de la synchronisation entre les frontières de trames courtes (segments fléchés en noir) et de trames longues (segments fléchés en vert).

Intégration statistique

Le découpage en trames courtes est nécessaire pour évaluer les propriétés instantanées définies par certains descripteurs. Cependant, la précision induite, de l'ordre de la dizaine de millisecondes,

est largement supérieure aux besoins pratiques. Nous verrons par la suite que la campagne d'évaluation ESTER tolère une erreur de 0.25 s sur les frontières de segments de classification. Plusieurs études [184][7] établissent d'ailleurs que la durée nécessaire à un être humain pour classifier un extrait audio est de cet ordre de grandeur.

De plus, d'un point de vue pratique, la classification sur trames courtes est extrêmement coûteuse en ressources et implique un nombre très élevé d'exemples (de l'ordre de 200000 par heure de signal). Étant donné que l'on se limite à quelques dizaines de milliers d'exemples pour l'apprentissage des SVM [1], on en déduit que le corpus d'apprentissage serait caractérisé de manière très parcellaire sur une base de plusieurs dizaines d'heures.

En pratique on choisit donc non pas de considérer les descripteurs eux-mêmes, mais certaines grandeurs statistiques calculées sur un nombre de trames suffisamment significatif. On exploite ici la synchronisation entre trames courtes et longues en calculant ces mesures statistiques sur les trames courtes couvertes par chaque trame longue. De plus, une comparaison de plusieurs échelles temporelles d'intégration statistique montre [182] que les meilleurs résultats sont obtenus pour 1 s, ce qui correspond à la longueur de nos trames longues. À chaque trame longue est donc associé un vecteur de descripteurs composé de statistiques de descripteurs court-terme et de descripteurs long-terme sans traitement statistique.

Ainsi, si $x(n)$ est la suite des valeurs d'un descripteur court-terme, où n est l'index de trame, on calcule les descripteurs long-terme $X(m)$, indexés par l'indice de trame longue m (on suppose que les indices débutent à 0) :

$$X(m) = f\left(x(mR_{\text{mul}}), x(mR_{\text{mul}}+1), \ldots, x(mR_{\text{mul}}+N_{\text{mul}}-1)\right).$$

f représente ici le traitement statistique appliqué ; on parle également d'intégration de descripteurs, dans un sens plus large. Nous n'exploitons dans cette étude que les grandeurs les plus couramment exploitées [140][30][229] : la moyenne et l'écart type.

Les bornes minimales et maximales sont également exploitées par certains auteurs [48] mais celles-ci sont trop sensibles à d'éventuelles valeurs marginales excentrées. Nous avons d'ailleurs montré dans une étude précédente [191] que leur inclusion dans le processus de classification parole/musique n'améliore pas les performances.

D'autres intégrations ont été explorées dans la littérature. En particulier, Meng a proposé [154] l'exploitation des coefficients d'un modèle auto-régressif appris sur les descripteurs d'une trame longue et fournit par ailleurs une étude comparative impliquant divers procédés d'intégration [155]. Toutefois, les coefficients d'un modèle AR sont connus pour être instables et leur évolution est discontinue par rapport aux variations du signal. On trouvera également dans [116] une étude assez exhaustive des différents procédés d'intégration sur une tâche de reconnaissance automatique des instruments de musique. Les résultats présentés ne montrent cependant pas d'avantage clair et systématique à utiliser des méthodes d'intégration plus complexes et confirment ainsi notre choix de ne pas explorer plus en profondeur ce sujet.

6.4 Normalisation des descripteurs

Les descripteurs obtenus après dérivation et intégration temporelle proviennent de modalités différentes et leur dynamique est très hétérogène. Pourtant, dans tous les noyaux usuels que nous exploitons, les descripteurs sont mis en concurrence au travers de sommes à pondérations uniformes, par exemple $k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{d=1}^{D} x_d y_d$ dans le cas du noyau linéaire, ou $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(\sigma^{-2} \sum_{d=1}^{D} (x_d - y_d)^2\right)$ pour le noyau RBF gaussien, où D est le nombre de composantes. Ainsi il est clair qu'un descripteur de moyenne largement supérieure à celle d'un autre descripteur couvrira ce dernier et le rendra presque « muet » dans l'expression de la fonction noyau.

On normalise donc les descripteurs de manière à réduire les disparités statistiques. La méthode la plus classique [225], que nous employons ici, consiste à homogénéiser les statistiques de premier

1. Cette limitation est due à un compromis entre performances et temps de calcul, rendu nécessaire par la complexité quadratique de la phase d'apprentissage.

et de deuxième ordre. Ainsi, si l'on note $x_{i,d}$ la composante d'indice d de l'exemple $\boldsymbol{x}_i$, on estime la moyenne μ_d et la déviation standard σ_d du descripteur d par les estimateurs statistiques classiques :

$$\begin{aligned} \mu_d &= \frac{1}{n}\sum_{i=1}^{n} x_{i,d} \\ \sigma_d^2 &= \frac{1}{n-1}\sum_{i=1}^{n} \left(x_{i,d} - \mu_d\right)^2 . \end{aligned}$$

Les composantes normalisées prennent donc l'expression suivante :

$$\hat{x}_{i,d} = \frac{x_{i,d} - \mu_d}{\sigma_d}.$$

Malgré le consensus autour de cette méthode de normalisation, il est important de rappeler que celle-ci se base sur un postulat de gaussianité des distributions des descripteurs. Or, dans le cas de distributions moins régulière, cette procédure de normalisation ne permet pas d'obtenir, comme souhaité dans l'idéal, des données gaussiennes centrées. Cette hypothèse est confirmée en pratique dans de nombreux cas de descripteurs non-bornées. Toutefois, les descripteurs bornés ou semi-bornés (l'exemple le plus courant est celui de descripteurs positifs, comme c'est le cas pour des mesures d'énergie), ne satisfont généralement pas le modèle gaussien, et s'approchent plutôt d'un modèle de distribution Gamma. L'effet de la normalisation « gaussienne » sur une distribution Gamma n'est pas évident et ne sera pas couvert dans ce document.

Une manière courante [12] pour s'affranchir de la distribution des données consiste à substituer à la valeur $x_{i,d}$ la valeur de fonction de répartition $F(x_{i,d})$, estimée sur l'ensemble des exemples. Ainsi, on a la garantie d'obtenir pour toutes les composantes une distribution quasi-uniforme sur l'intervalle $[0; 1]$. Il est équivalent, à un facteur multiplicatif près, de substituer au descripteur son rang parmi les valeurs de l'ensemble d'apprentissage, triées par ordre croissant, comme le proposent Stolcke et al. [222]. Toutefois, nous n'avons pas constaté un effet notable de ces techniques de normalisations alternatives sur les performances des SVM, nous n'explorons donc pas cette question plus en détail dans ce document. On trouvera dans [12] une étude comparative des différentes méthodes de normalisation de données.

6.5 Liste des descripteurs employés

Nous présentons brièvement dans cette section la collection de descripteurs réunis pour les tâches de classification traitées dans ce document. Ceux-ci sont regroupés selon la modalité de calcul. On distingue ainsi des descripteurs spectraux, calculés sur le spectre estimé par FFT, des descripteurs temporels, calculés directement sur le signal audio, des descripteurs cepstraux et des descripteurs perceptifs, basés sur des modèles de propriétés psychoacoustiques de l'audition humaine. Les descripteurs employés étant pour la plupart bien connus de la communauté, leur présentation détaillée a été reportée dans l'annexe C.

Pour certains des descripteurs proposés, l'évolution temporelle de la valeur peut être aussi significative, voire plus, que la valeur elle-même. Ainsi, on ajoute à la plupart des descripteurs les estimations des dérivées premières et secondes, qui forment elles-mêmes de nouveaux descripteurs.

Descripteurs spectraux

Les descripteurs spectraux sont calculés à partir du spectre estimé par la Transformée de Fourier Discrète (TFD), qui est définie, sur une trame de N échantillons, de la façon suivante :

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-2j\pi k \frac{n}{N}} \quad \forall k \in [0, \ldots, N-1].$$

Le calcul de la TFD est précédé de la pondération du signal de trame par une fenêtre de Hamming, qui limite l'étalement des pics spectraux. En pratique, seuls les amplitudes $a_k = |X(k)|$ sont utilisées dans les descripteurs spectraux présentés ci dessous :

- Les *moments statistiques spectraux* : cette série de descripteurs est basée sur le barycentre et les moments d'ordre i d'un modèle probabiliste du spectre ; elle contient :
  - le *centroïde spectral*,
  - la *largeur spectrale*,
  - l'*asymétrie spectrale* (ou *Skewness*),
  - la *platitude spectrale* (ou *Kurtosis*).
- *Descripteurs MPEG-7* : plusieurs descripteurs liés au standard MPEG-7 [3] sont ici exploités :
  - le *rapport spectral*, qui constitue une alternative à la mesure de platitude spectrale,
  - la *platitude d'amplitude spectrale (ASF)*,
  - le *facteur de crête spectral (SCF)*, proposé par Peeters [178] qui, bien qu'il ne fasse pas partie du standard MPEG-7, reste très proche des deux descripteurs précédents, dans sa définition.
- La *pente spectrale*, qui représente le taux de décroissance spectrale.
- La *décroissance spectrale*, qui mesure la décroissance des amplitudes spectrales.
- La *fréquence de coupure*, liée à une mesure de quantile de l'énergie spectrale.
- Le *flux spectral*, défini par Scheirer et Slaney [207], comme une mesure de variation spectrale entre trames consécutives.
- Les *coefficients LPC (Linear Prediction Coding)*, caractérisant un modèle source-filtre pour le codage audio.
- Les *sous-bandes en octaves*, proposés par Essid pour la reconnaissance d'instruments de musique [72], et destinés à capturer la structure spectrale de sons instrumentaux. Ils se composent des deux sous-groupes suivants :
  - les intensités de signaux de sous-bandes en octaves, nommées OBSI (cf. annexe C)
  - et les rapports d'intensité de signaux de sous-bandes en octaves, nommés OBSIR.
- Plusieurs mesures de *modulation d'amplitude* sont exploités pour caractériser les phénomènes de trémolo et de rugosité, qui se manifestent respectivement sur les bandes de fréquences entre 4 et 8 Hz et entre 10 et 40 Hz. Quatre critères sont définis pour chaque bande :
  - la *fréquence AM* du pic maximal,
  - l'*amplitude AM* du pic maximal,
  - l'*amplitude AM heuristique* du pic maximal, par rapport à la bande de fréquences,
  - et le *produit AM* de la fréquence et de l'amplitude AM.

Descripteurs temporels

Les descripteurs suivants sont basés exclusivement sur la forme d'onde du signal audio dans une trame courte (ou longue si précisée) et ne font pas intervenir le spectre.

- Le *taux de passage par zéro (ZCR)*, proposé par Kedem [118], et dont on peut montrer la corrélation au centroïde spectral.
- Le *taux de passage par zéro long-terme*, calculé sur les trames long-terme.
- Les *moments statistiques temporels*, qui reprennent les mêmes moments que ceux définis sur le spectre. Ils sont calculés :
  - sur les trames court-terme,
  - sur les trames long-terme,
  - sur l'enveloppe des trames longues, estimée par le biais de la transformée de Hilbert.
- Les *coefficients d'autocorrélation*.

Descripteurs cepstraux

Le *cepstre* complexe d'un signal est défini à l'origine en 1963 par Bogert et al. [33], comme la transformée de Fourier du logarithme du spectre, soit :

$$C(\tau) = C(z(t)) = \mathcal{F}\left\{\ln\left(\mathcal{F}\left\{z(t)\right\}\right)\right\}.$$

Originellement proposé pour l'étude de phénomènes d'échos, le cepstre permet l'observation des variations du spectre. Dans sa forme actuelle, le cepstre réel est généralement formulé avec une transformée de Fourier inverse :

$$C(\tau) = C(x(t)) = \left|\mathcal{F}^{-1}\left\{\ln\left(\left|\mathcal{F}\left\{x(t)\right\}\right|^2\right)\right\}\right|^2.$$

Ceci permet d'exprimer le cepstre dans le domaine temporel. Son usage est particulièrement répandu dans le domaine du traitement de la parole, car cette dernière peut être modélisée par un modèle source-filtre $x(n) = (s * h)(n)$. En effet le produit de convolution se traduisant par un produit simple dans le domaine fréquentiel, on montre que le cepstre complexe est un morphisme transformant l'opérateur de convolution en somme :

$$\begin{aligned} C(x) &= \mathcal{F}^{-1}\left\{\ln\left(\mathcal{F}\left\{s * h\right\}\right)\right\} && (6.1)\\ &= \mathcal{F}^{-1}\left\{\ln\left(\mathcal{F}\left\{s\right\}\right) + \ln\left(\mathcal{F}\left\{h\right\}\right)\right\} && (6.2)\\ &= C(s) + C(h). && (6.3) \end{aligned}$$

Ainsi, dans les cas proches de la voix, où la source $s(n)$ est un peigne fréquentiel, tandis que les résonances apportées par le filtre $h(n)$ ont un spectre beaucoup plus lisse, généralement assimilée à une enveloppe spectrale, les cepstres de la source et du filtre sont pratiquement disjoints et donc séparables. Ceci étant, beaucoup de sources sonores musicales ne rentrent pas dans le cadre du modèle source-filtre et ne sont pas aussi aisément interprétables par l'analyse cepstrale.

Nous exploitons pour la tâche de classification audio deux types de descripteurs basés sur la représentation cepstrale :

- Les *coefficients MFCC*, basés sur une échelle mel des fréquences, et où l'on substitue une DCT à la Transformée de Fourier.
- Les *coefficients cepstraux à Q constant*, basés sur une échelle liée à la répartition logarithmique des hauteurs musicales, qui implique en outre l'adaptation des largeurs de bandes par rapport à la fréquence centrale.

Descripteurs perceptifs

- Nous exploitons deux mesures liées au *pitch* (c'est-à-dire la perception de la fréquence fondamentale dominante), calculées par l'algorithme YIN de Cheveigné et al. [60] :
  - la *fréquence fondamentale* F_0,
  - et la *mesure de périodicité*, qui caractérise les spectres harmoniques.
- Enfin, trois descripteurs sont liés à la mesure d'*intensité perceptive* proposée par Moore et al. [162] et appelée *loudness* :
  - la *loudness spécifique relative*, constituée de coefficients d'intensité sur les bandes de fréquences perceptives,
  - l'*acuité perceptive* (ou *sharpness*), proposée par Peeters [178]), qui est l'équivalent du centroïde spectral sur la loudness,
  - et l'*étalement perceptif* (ou *spread*), également proposé par Peeters, défini comme une mesure de contraste sur la loudness.

6.6 Discussion

Les descripteurs présentés ont avant tout été choisis pour respecter deux contraintes essentielles liées à notre cadre particulier.

En premier lieu le système mis en place est conçu pour fonctionner en temps réel ; il ne doit pas impliquer de descripteurs coûteux. Ont donc été rejetés de nombreux descripteurs basés sur une analyse spectrale plus poussée par un modèle psychoacoustique de l'audition [194][157], impliquant une forte charge en temps de calcul ; ou d'autres qui permettent de mieux appréhender le matériau musical éventuel. Ainsi on trouve dans la littérature certaines formes de « transcriptions ébauchées » à travers la recherche de trajectoires de pics spectraux [121]. Ces approches ont par ailleurs généralement l'inconvénient de se baser sur des algorithmes de type Viterbi, impliquant les trames futures dans la prise de décision.

Cette contrainte temporelle est la seconde restriction que nous nous imposons. Une réponse en temps réel implique une décision régulière et instantanée, ou au moins avec retard constant, qui exclue donc l'usage de descripteurs temporellement alignés sur des événements particuliers, comme les attaques. On trouve par exemple dans la littérature plusieurs propositions impliquant une estimation du tempo ou de la structure rythmique [111][41], que nous avons exclues de notre cadre expérimental, puisqu'elles font intervenir une modalité temporelle beaucoup plus large (de

l'ordre de plusieurs secondes). Certains auteurs, comme Lachambre et al. [129], se basent également sur une décomposition temporelle du signal en segments « homogènes » de tailles variables servant chacun de support au calcul d'une valeur.

Beaucoup de ces propositions émanent de domaines annexes, plus directement liés à la musique, comme la reconnaissance de genres ou d'ambiances musicales. Nous supposons ici, sans toutefois l'étayer par un constat psychoacoustique, que les classes impliquées (parole, musique, chant) sont identifiables par un être humain de manière quasi instantanée et surtout hors de tout contexte. Les classes considérées sont plus clairement définies et peu ambigües [2], contrairement à d'autres domaines, comme la reconnaissance de genre musical, qui fait intervenir des notions sémantiques et cognitives.

La plupart des descripteurs réunis ici sont d'usage très courant dans la littérature. On notera en outre que les traitements statistiques décrits dans la section 6.3 font apparaître certaines propriétés qui peuvent sembler absentes de notre liste. Par exemple le taux de trames à basse énergie ou le taux de trames silencieuses, que l'on trouve assez fréquemment [207][138][109] dans la littérature, est équivalent à la moyenne de l'énergie des trames court-terme. De plus, certains descripteurs, difficilement interprétables tels quels, comme les moments temporels, prennent leur sens si l'on considère leur moyenne ou leur variance. Ainsi la variance de la *largeur temporelle* constitue une alternative à l'estimation du taux de trames à basse énergie.

En définitive on remarquera que de nombreux descripteurs, parmi ceux présentés, sont fortement redondants dans leur définition. On peut en effet grouper ceux-ci en 4 modalités principales :

- *Centre spectral* : estimation du centre de gravité du spectre. Par exemple le centroïde spectral, le ZCR, le ZCR long-terme, et l'acuité perceptive.
- *Répartition spectrale* : description de l'allure ou de la répartition spectrale. Par exemple la largeur, l'asymétrie, la platitude, le rapport, la pente, et la décroissance spectraux, la fréquence de coupure, la platitude d'amplitude spectrale, le facteur de crête spectrale, ou encore l'étalement perceptif.
- *Énergies de sous-bandes* : de nombreux descripteurs fournissent une estimation de l'énergie de sous-bandes spectrales, généralement grâce à l'usage de banc de filtres, ou à travers l'estimation d'enveloppes spectrales. On peut ainsi citer les descripteurs OBSI et OBSIR, les coefficients LPC, MFCC, à Q constant, ainsi que la *loudness* spécifique relative.
- *Estimation de pitch* : enfin, une série de descripteurs fournissent une estimation de la fréquence fondamentale dominante, comme le F_0 estimé par YIN, et les coefficients d'autocorrélation.

Les autres descripteurs expriment des caractéristiques plus particulières, comme la mesure d'apériodicité estimée par YIN, les mesures de trémolo et de rugosité par modulation d'amplitude, le flux spectral, décrivant les variations du spectre, ou encore les moments statistiques temporels.

Bien que ces descripteurs soient pertinents pour la discrimination parole/musique, on remarque que la littérature s'attarde très peu sur le problème des classes mixtes. En effet, l'observation comparatée de spectres de parole et de musique met en évidence de nombreuses différences assez claires que beaucoup d'auteurs ont tenté de traduire par des mesures quantifiées. Néanmoins, la plupart de ces observations perdent leur sens lorsque l'on superpose les signaux des deux classes. Or nous verrons par la suite, dans la partie IV sur l'évaluation, que la majeure partie des erreurs du système provient précisément de confusions sur cette situation (la parole accompagnée de musique). On peut bien sûr être tenté de faire appel à des techniques de séparation de sources mais celles-ci sont généralement beaucoup plus complexes et coûteuses que les méthodes impliquées dans la classification audio. De plus, de nombreux algorithmes de séparation de sources font généralement appel à des techniques de classification afin d'identifier les régions pertinentes à séparer, ce qui inverse totalement la méthodologie.

2. Certains problèmes demeurent par exemple sur la détection du chant, comme l'identification du rap, ou de certaines prosodies de chant proches de la parole, comme le Gainsbourg des années 80.

Chapitre 7

Sélection de descripteurs

Sommaire

7.1 Introduction 91
7.2 Taxonomie des algorithmes de sélection 92
7.2.1 Classement ou sélection 92
7.2.2 Notion de pertinence 93
7.2.3 Stratégies de recherches 93
7.2.4 Paradigmes 94
7.3 Méthodes filtres classiques 95
7.3.1 Coefficients de Pearson et critère de Fisher 95
7.3.2 Inertia Ratio Maximization using Feature Space Projection (IRMFSP) . 96
7.3.3 Test de Kolmogorov-Smirnov 96
7.4 Méthodes à noyaux 97
7.4.1 Feature Selection concaVe (FSV) 97
7.4.2 Approximation of the zeRO-norm Minimization (AROM) 98
7.4.3 Recursive Feature Extraction (RFE) 99
7.4.4 Sélection par minimisation de la borne Rayon-Marge (R2W2) 99
7.5 Propositions d'algorithmes efficaces de sélection 100
7.5.1 Sélection pondérée basée sur le critère d'Alignement (SAS) 100
7.5.2 Sélection Pondérée basée sur le produit de Frobenius (SFS) 101
7.5.3 Sélection Forward basée sur le critère d'Alignement (FAS) 101
7.5.4 Sélection Pondérée sur le critère de Séparabilité (SCSS) 102
7.5.5 Sélection sur le Discriminant de Fisher Kernelisé (KFDS) 103
7.5.6 Avantages des méthodes proposées 104
7.6 Synthèse 105
7.7 Expériences comparatives 105
7.7.1 Données artificielles 106
7.7.2 Données réelles 107
7.7.2.1 Spambase 108
7.7.2.2 Ionosphere 109
7.7.2.3 Lymphoma 109
7.7.2.4 Classification parole/musique 110
7.7.3 Coût en temps de calcul 111
7.8 Commentaires 112

7.1 Introduction

Nous avons introduit dans le chapitre 6 une large collection de descripteurs destinés à caractériser au mieux les classes mises en jeu pour la segmentation d'un flux audio. Nous avons fait le

constat que beaucoup d'entre eux décrivent des modalités très proches et peuvent donc présenter de fortes redondances. Diversifier les descripteurs reste bien sûr souhaitable, mais peut se révéler contre-productif lors de la phase de classification, en premier lieu à cause de la malédiction de la dimension, que nous avions évoquée dans la section 2.3.4 de l'état de l'art.

De plus, malgré le soin porté dans le choix des descripteurs mis en jeu, il est possible d'introduire une certaine proportion de descripteurs non pertinents qui bruitent l'information considérée par le classifieur. L'introduction d'une composante non corrélée à la tâche considérée peut en effet avoir un impact néfaste sur la mesure des distances entre exemples, qui constitue pourtant l'outil de base de la majorité des méthodes de classification, particulièrement des SVM.

Enfin, d'un point de vue pratique, l'utilisation de descripteurs non pertinents, ou au moins redondants, introduit une complexité inutile (tant calculatoire qu'en termes de mémoire) dans la phase de classification par le calcul coûteux et superflu de ces descripteurs.

La sélection des descripteurs les plus pertinents pour une tâche donnée est un problème à part entière dans le domaine de l'apprentissage statistique, qui a beaucoup occupé la communauté scientifique durant les dernières décennies. Outre la résolution des difficultés exposées précédemment, elle peut apporter une meilleure compréhension d'un problème par l'interprétation des descripteurs les plus pertinents.

On conserve dans ce chapitre les notations introduites dans la section 3.1 en se concentrant sur un problème de discrimination : on travaille donc sur un ensemble d'apprentissage $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1\ldots n}$, où les exemples $\boldsymbol{x}_i \in \mathbb{R}^D$ sont décrits par D composantes dimensionnelles correspondant chacune à un descripteur donné ($\boldsymbol{x}_i = [x_{i,1}\,, \ldots, x_{i,D}]^T$), et sont associés à un label $y_i \in \{+1, -1\}$. On utilisera également le vecteur des labels $\boldsymbol{y} = [y_1, \ldots, y_n]^T$ et les vecteurs des exemples pour chaque descripteur $\boldsymbol{x}_{\cdot,d} = [x_{1,d}\,, \ldots, x_{n,d}]^T$.

Nous abordons dans la section 7.2 la question de la définition de la pertinence d'un descripteur, qui met en évidence la complexité du problème posé. L'explosion combinatoire qui en découle est généralement contournée par le moyen de stratégies de recherche que nous énumérons par la suite. La relation avec le classifieur est également formalisée par une taxonomie classique qui fait apparaître l'importance de la prise en compte du comportement du classifieur dans le processus de sélection des descripteurs.

Nous détaillerons dans la section 7.3 quelques algorithmes classiques principalement basés sur une mesure de corrélation, pour nous concentrer ensuite, dans la section 7.4, sur des méthodes prenant en compte le comportement des machines à vecteurs de support dans la sélection. L'étude des algorithmes de la littérature nous permet finalement de proposer plusieurs algorithmes basés sur certains des critères de performances abordés dans le chapitre 4.

7.2 Taxonomie des algorithmes de sélection

7.2.1 Classement ou sélection

On peut définir le problème de la sélection de descripteurs comme la recherche d'un sous-groupe des S descripteurs les plus susceptibles d'optimiser la tâche de classification ultérieure, parmi une collection originale de D descripteurs. Plusieurs questions se posent alors :

- Souhaite-t-on déterminer le sous-groupe optimal pour un nombre $S < D$ donné ?
- Souhaite-t-on déterminer le sous-groupe optimal pour *tout* nombre $S < D$ possible ?
- Souhaite-t-on déterminer automatiquement le nombre S de descripteurs conjointement au sous-groupe ?

Ces trois problèmes sont en réalité tout à fait différents d'un point de vue méthodologique, et soulignent l'hétérogénéité de la sélection de descripteurs, qui recouvre en réalité deux problèmes différents :

- Le **classement** (*variable ranking*) vise à ranger les descripteurs par ordre croissant (ou décroissant) de pertinence pour la tâche donnée.
- La **sélection** de sous-ensemble (*subset selection*) vise à extraire de la liste originale un sous-ensemble de descripteurs pertinents, dont la taille est déterminée manuellement ou automatiquement.

Nous verrons que certains des algorithmes présentés sont fondamentalement liés à la notion de classement. D'autres sont souvent présentés dans la littérature sous la forme de problèmes de sélection de sous-ensemble, mais nous les présenterons systématiquement sous forme d'algorithmes de classement, de manière à pouvoir définir un protocole expérimental commun. Cette transformation implique en général d'ignorer les étapes de seuillage introduites par les auteurs.

7.2.2 Notion de pertinence

Les deux problèmes précédents ont pour trait commun de se baser sur la notion centrale de *pertinence*, qui peut être considérée comme une mesure de l'efficacité d'un descripteur pour une tâche donnée. Pourtant, même si elle semble intuitive, celle-ci est difficile à définir analytiquement. De nombreuses propositions ont été formulées dans la littérature, synthétisées par John et al. [117]. Les auteurs montrent que pour un simple problème de OU exclusif (XOR) sur des données corrélées, où l'on définit 5 variables booléennes x_1, x_2, x_3, x_4 et x_5, avec deux couples inversement correllés ($x_4 = \bar{x}_2$ et $x_5 = \bar{x}_3$), associées à un label $y = x_1 \otimes x_2$ (où $\otimes$ représente le OU exclusif), toutes les définitions considérées sont fausses. Ils en déduisent une distinction formelle entre *pertinence forte*, relative à un descripteur indispensable à la tâche (c'est-à-dire dont l'exclusion pénalise celle-ci), et *pertinence faible*, relative à un descripteur utile à la tâche, mais auquel peut cependant se substituer un autre descripteur. La notion de pertinence faible fait apparaître le fait que la pertinence d'un descripteur n'est pas une propriété intrinsèque et ne peut être jugée indépendamment des autres descripteurs mis en jeu.

Guyon et al. [97] montrent par ailleurs, par le biais d'un exemple concret également basé sur le problème du OU exclusif, que deux descripteurs strictement non- pertinents lorsqu'ils sont utilisés chacun seul, peuvent se révéler pertinents lorsqu'ils sont exploités ensemble (dans des cas de séparation non-linéaire). On retrouve ce constat dans une note de Toussaint [226] qui montre que les k pires descripteurs individuels peuvent se révéler meilleurs ensemble que les k meilleurs descripteurs individuels ; nous appellerons par la suite ce phénomène *interpertinence*.

Guyon et al. se penchent en outre sur une autre idée reçue qui consiste à considérer que deux descripteurs corrélés n'apportent pas d'information supplémentaire, en montrant également par un exemple simple que ce raisonnement n'est pas systématiquement vrai.

Nous ne cherchons pas ici à traiter en détail la question de la définition formelle de la pertinence, sur laquelle on pourra trouver plus d'informations dans [160] et [31]. Toutefois, ces considérations montrent les difficultés sous-jacentes à la sélection de descripteurs qui, par une approche « force brute », se heurte au classique problème d'explosion combinatoire : il n'est en général pas possible d'évaluer les $\binom{D}{S} = \frac{D!}{S!(D-S)!}$ possibilités pour sélectionner le sous-ensemble optimal de S descripteurs parmi les D. Ce constat s'aggrave si l'on considère toutes les valeurs S possibles. Il est donc nécessaire d'adopter une stratégie de recherche.

7.2.3 Stratégies de recherches

Webb propose [241] une taxonomie des différentes stratégies généralement employées pour la sélection de descripteurs, reprise par Wang et Chen [239] :

BIN (*Best Individual N*) : C'est la stratégie la plus simple. L'efficacité de chaque descripteur est mesurée indépendamment par un critère donné. La complexité est donc réduite (de l'ordre $O(D)$) et peut par ailleurs largement profiter d'un traitement en parallèle. Les descripteurs sont rangés par ordre décroissant de pertinence, d'après la mesure effectuée. Bien sûr la contrepartie à ce moindre coût computationel est l'absence de prise en compte des dépendances éventuelles entre les descripteurs. On risque ainsi de retrouver de nombreux descripteurs redondants dans les mieux notés, et de passer à côté des phénomènes d'interpertinence présentés plus haut.

SEQ (*SEQuential*) : Afin de prendre en compte les interdépendances entre descripteurs on peut adopter une stratégie séquentielle, qui a pour principe de sélectionner itérativement les descripteurs ou des groupes de descripteurs. On commence ainsi par sélectionner le descripteur le plus pertinent. Par la suite, à chaque itération la sélection prendra en compte (d'une manière non précisée) la liste

des descripteurs déjà choisis pour mesurer la pertinence des descripteurs restants. On évite ainsi la sélection de multiples descripteurs redondants, au prix néanmoins d'un fort accroissement de complexité, de l'ordre $O\left(\frac{D^2}{2}\right)$, puisque l'on doit évaluer à chaque itération tous les descripteurs restants, ou $O\left(SD - \frac{S^2}{2}\right)$ si l'on s'arrête à S descripteurs. On notera toutefois que dans le cas extrême (et bien sûr théorique) du OU exclusif, le problème des descripteurs interpertinents n'est pas résolu par cette stratégie.

À l'approche *forward* décrite ici, on peut substituer une approche *backward*, où les descripteurs les moins pertinents sont itérativement supprimés. Bien que l'approche *backward* soit beaucoup plus coûteuse si $S \ll D$ (on retrouve alors l'ordre $O\left(\frac{D^2}{2}\right)$), Guyon et al. [97] expliquent que cette dernière est moins susceptible d'être biaisée puisque dès la première itération l'effet conjoint de tous les descripteurs est pris en compte, ce qui n'est pas le cas de l'approche *forward*.

PO (*Parameter Optimization*) : La troisième stratégie repose sur une procédure d'optimisation. En pondérant chaque composante (chaque descripteur d) d'un exemple $\boldsymbol{x}$ par un facteur w_d (soit $\boldsymbol{x_w} = \boldsymbol{x} \bullet \boldsymbol{w}$, où $\bullet$ est le produit terme à terme), on minimise un critère donné par mises à jours successives des poids w_d, jusqu'à convergence (nous présenterons dans la suite plusieurs exemples associés à des critères différents). On considère alors les poids w_d comme une mesure de pertinence des descripteurs qui sont ordonnés par ordre décroissant. Cette approche présente l'avantage de faire intervenir la totalité des descripteurs simultanément à chaque itération, ce qui permet de mieux prendre en compte les problèmes d'interdépendances et de redondances. La complexité d'ordre $O(ID)$ est beaucoup plus réduite que pour l'approche séquentielle, si l'on suppose que le nombre d'itérations I est négligeable devant le nombre de descripteurs D. Néanmoins, cette approche suppose l'usage d'un critère dérivable par rapport aux poids w_d. De plus il n'existe pas de preuve théorique que ces poids soient une mesure fiable de pertinence ; en particulier on n'a aucune garantie de la consistance des poids si l'on ôte une grande partie des descripteurs (ce qui est a priori le but souhaité). Certains auteurs se contentent de supprimer les descripteurs dont les poids tendent vers 0, mais cette approche a tendance à supprimer plus de descripteurs qu'il ne faudrait.

7.2.4 Paradigmes

Nous avons jusque-là considéré la sélection de descripteurs en dehors de tout contexte. Pourtant celle-ci précède et détermine l'utilisation d'une méthode de classification, puisqu'elle est dans l'idéal menée pour optimiser cette dernière. On retrouve généralement dans la littérature [97][117][31] une taxonomie distinguant trois paradigmes de sélection de descripteurs, déterminés par la relation avec le classifieur :

Filtres

Les filtres (*filters*) sont considérés comme indépendant du processus de classification. La sélection de descripteurs peut être vue comme une étape de pré-traitement des données qui ne fait pas appel au classifieur. Elle est donc « universelle » de ce point de vue et présente a priori le désavantage de ne pouvoir prendre en compte le biais éventuel introduit par tel ou tel classifieur.

Enveloppeurs

La distinction entre filtres et enveloppeurs (*wrappers*) est introduite par John et al. [117] pour prendre en compte la notion de pertinente faible qu'ils introduisent. Le principe des enveloppeurs est d'inclure le classifieur, comme une boîte noire, dans le processus de sélection. Celle-ci s'opère donc par itérations successives, tirant parti des résultats de classification et prenant ainsi en compte le comportement propre du classifieur. Cependant cette approche est généralement beaucoup plus coûteuse que les filtres, puisqu'elle implique de nombreuses phases d'apprentissage, en plus des calculs directement liés à la sélection de descripteurs.

Sélection embarquée

Les algorithmes de sélection embarquée (*embedded methods*) concernent une classe de classifieurs pour lesquels la sélection de descripteurs fait directement partie du processus de classification. L'exemple des arbres de décision de type CART (*Classification And Regression Trees*) est le plus typique de cette approche. Les sélections embarquées ont le mérite, par rapport aux enveloppeurs qui traitent le classifieur comme une boîte noire, de ne pas nécessiter de corpus de validation distinct du corpus d'apprentissage pour valider les performances du classifieur.

Il est communément admis que les approches filtres sont indépendantes du classifieur mis en jeu. Pourtant, tout critère de mesure de pertinence impliqué dans la sélection induit en réalité une hypothèse sur le processus de classification. Nous présenterons dans la suite de ce chapitre de nouvelles méthodes de sélection de descripteurs, adaptées aux SVM, qui n'incluent aucun apprentissage du classifieur tout en basant la sélection sur les critères de performances introduits dans le chapitre 4, directement liés aux SVM.

7.3 Méthodes filtres classiques

De nombreuses méthodes de sélection de descripteurs ont été proposées sur la base d'une mesure de corrélation entre le descripteur et le label associé à l'exemple. Elles reposent en général sur une approche *filtre* de classement de pertinence. Nous détaillons quelques-unes de ces méthodes dans cette section.

7.3.1 Coefficients de Pearson et critère de Fisher

On exploite en statistiques le coefficient de corrélation de Pearson entre deux variables aléatoires X et Y :

$$\mathcal{R} = \frac{\mathrm{cov}(X,Y)}{\sqrt{\mathrm{var}(X)\,\mathrm{var}(Y)}},$$

où cov et var sont respectivement les mesures de covariance et de variance. On peut ainsi estimer empiriquement la corrélation entre le vecteur des labels $\boldsymbol{y}$ et le vecteur $\boldsymbol{x}^d$ des exemples pour le descripteur d :

$$\mathcal{R}(d) = \frac{\sum_{i=1}^{n}(x_{i,d}-\bar{x}_d)(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_{i,d}-\bar{x}_d)^2\sum_{i=1}^{n}(y_i-\bar{y})^2}},$$

où $\bar{x}_d = \frac{1}{n}\sum_{i=1}^{n} x_{i,d}$ est le centre des exemples du descripteur d et $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ le centre des labels.

Dans le cas du problème de discrimination ($y_i \in \{+1;-1\}$), si l'on suppose les deux classes également réparties, on a $\bar{y}=0$, et $\sum_{i=1}^{n}(y_i-\bar{y})^2 = n$, on peut alors montrer que, à un facteur $\sqrt{2}$ près, les coefficients ont pour valeur :

$$R_{corr}(d) = \frac{\mu_{1,d}-\mu_{2,d}}{\sqrt{\sigma_{1,d}^2+\sigma_{2,d}^2}},$$

où $\mu_{c,d}$ et $\sigma_{c,d}^2$ sont respectivement le centre et la variance pour le descripteur d des exemples de la classe c. Cette expression est généralement employée pour définir les coefficients de corrélation de Pearson exploités pour ranger les descripteurs par ordre de pertinence.

On remarquera néanmoins que la corrélation telle quelle est a priori mal définie puisqu'un descripteur inversement corrélé au label se voit attribuer le pire score de pertinence alors qu'il apporte autant d'information que son opposé. On préfère donc généralement employer le carré des coefficients de Pearson, que Bishop [29] nomme critère de Fisher :

$$R_{Fisher}(d) = \frac{|\mu_{1,d}-\mu_{2,d}|^2}{\sigma_{1,d}^2+\sigma_{2,d}^2}.$$

En effet la maximisation de ce critère dans sa forme générale ($R = \frac{|\mu_1 - \mu_2|^2}{\sigma_1^2 + \sigma_2^2}$) est directement liée à l'Analyse Discriminante de Fisher introduite dans le prélude aux Machines à Vecteurs de Support, en section 3.2. On exploite donc ce dernier pour quantifier la séparabilité des classes dans l'espace unidimensionnel défini par chaque descripteur.
On a donc introduit ici une approche filtre couplée à une stratégie de recherche BIN.

7.3.2 Inertia Ratio Maximization using Feature Space Projection (IRMFSP)

Peeters propose [181] une méthode séquentielle de classement de descripteurs, dont le principe est proche du critère de Fisher. Il y ajoute une phase qui, à chaque itération, permet de prendre en compte les descripteurs déjà sélectionnés. Il se base en premier lieu sur le critère de séparabilité que nous avons introduit dans la section 4.4.2. On rappelle que ce dernier est le rapport des *dispersions inter-classes* et *intra-classes* (équations 4.14 et 4.15, page 58) qui, sur des données uni-dimensionnelles, prennent les expressions suivantes :

$$S_b = \frac{1}{n} \sum_{c=1,2} n_c \left(\mu_{c,d} - \mu_d\right)^2 \tag{7.1}$$

$$S_w = \sum_{c=1,2} \sum_{\boldsymbol{x}_i \in \mathcal{S}_c} \left(x_{i,d} - \mu_{c,d}\right)^2 . \tag{7.2}$$

On peut en outre introduire la mesure de *dispersion globale* (ou totale), qui mesure la dispersion de tous les exemples par rapport au centre global μ_d :

$$S_T = \frac{1}{n} \sum_{i=1}^{n} \left(x_{i,d} - \mu_d\right)^2 . \tag{7.3}$$

Peeters propose donc, comme critère de pertinence pour les descripteurs, le rapport entre les dispersions inter-classes et globale (qu'il nomme *inerties*, d'où le terme *Inertia Ratio*) :

$$R_{IR}(d) = \frac{\sum_{c=1,2} n_c \left(\mu_{c,d} - \mu_d\right)^2}{\sum_{i=1}^{n} \left(x_{i,d} - \mu_d\right)^2}.$$

L'apport principal de sa contribution consiste à choisir le descripteur maximisant le critère à chaque itération, puis d'appliquer une procédure d'orthogonalisation par rapport aux descripteurs sélectionnés sur les descripteurs restants afin de réduire la corrélation entre ces derniers et le descripteur sélectionné.

Ainsi, si l'on suppose qu'à l'itération k, le descripteur d'indice d_k maximise le critère R_{IR}, alors pour tout descripteur restant d'indice e, on applique la mise à jour suivante par rapport au vecteur normalisé $\tilde{\boldsymbol{x}}_{\cdot,d_k} = \frac{\boldsymbol{x}_{\cdot,d_k}}{\|\boldsymbol{x}_{\cdot,d_k}\|}$:

$$\boldsymbol{x}_{\cdot,e} \longleftarrow \boldsymbol{x}_{\cdot,e} - \left(\boldsymbol{x}_{\cdot,e}^T \tilde{\boldsymbol{x}}_{\cdot,d_k}\right) \tilde{\boldsymbol{x}}_{\cdot,d_k}.$$

La méthode IRMFSP apporte ainsi une stratégie de recherche séquentielle sur un critère proche du critère de Fisher. Celle-ci reste très peu coûteuse de par la simplicité du critère exploité. Cependant, la phase d'orthogonalisation introduite nécessite un certain nombre d'exemples pour être statistiquement fiable. De plus cette fiabilité décroît fortement à mesure que les effets des orthogonalisations successives se cumulent. On peut donc finir par travailler sur des descripteurs totalement bruités après un certain nombre d'itérations.

7.3.3 Test de Kolmogorov-Smirnov

Il est également possible de construire une mesure de pertinence sur la base du test de Kolmogorov-Smirnov. Ce dernier est un test d'hypothèse utilisé en statistiques pour déterminer si un échantillon suit une loi définie par sa fonction de répartition $F_X(x) = \mathcal{P}(X \leq x)$, où X est une variable aléatoire modélisant ici le comportement d'un descripteur donné. Il est défini comme le maximum de la différence absolue entre deux fonctions de répartitions, l'une supposant la classe positive, l'autre non ; soit :

$$KS(d) = \sqrt{n} \max_{1 \leq i \leq n} \left|F_{X_d}(x_{i,d}) - F_{X_d}(x_{i,d} | \, y = +1)\right| .$$

On estime la fonction de répartition F_{X_d} à partir des échantillons de $\boldsymbol{x}_{\cdot,d}$ par :

$$F_{X_d}(x) = \frac{1}{n} \operatorname{Card} \{x_{i,d} \mid x_{i,d} < x\}_{1 \leq i \leq n} .$$

Le critère ainsi défini ne fait donc aucune supposition sur la répartition des observations (contrairement à la modélisation gaussienne implicite dans les deux approches précédentes), et permet ainsi la constitution d'une approche filtre avec stratégie BIN pour la sélection de descripteurs. On notera toutefois qu'un algorithme a été proposé [28] pour adapter ce critère à une stratégie séquentielle et ainsi prendre en compte les redondances entre descripteurs.

Il existe de nombreuses autres approches de type filtres dans la littérature, par exemple basées sur l'Information Mutuelle [253][255], que nous ne détaillerons pas ici. Notre propos est avant tout de montrer la pertinence des méthodes prenant en compte la classification par SVM, par contraste avec les méthodes filtres classiques uniquement basées sur des mesures de corrélation ou d'information entre les descripteurs ou entre les descripteurs et les labels de classe.

7.4 Méthodes à noyaux

7.4.1 Feature Selection concaVe (FSV)

Parallèlement au développement des Machines à Vecteurs de Support, Mangasarian a dirigé de nombreux travaux sur la séparation linéaire par programmation linéaire [147]. Il formule ainsi l'algorithme de Programmation Linéaire Robuste [23] (RLP *Robust Linear Programming*) pour la détermination d'un hyperplan linéaire optimal, même dans le cas de données non-séparables linéairement :

$$\begin{array}{ll} \min\limits_{\boldsymbol{w},\gamma,\boldsymbol{y},\boldsymbol{z}} & \dfrac{\mathbf{1}^T\boldsymbol{y}}{n_1} + \dfrac{\mathbf{1}^T\boldsymbol{z}}{n_2} \\ & \\ \text{sous les contraintes} & \begin{array}{c} -\boldsymbol{A}\boldsymbol{w} + \gamma\mathbf{1} + \mathbf{1} \leq \boldsymbol{y} \\ \boldsymbol{B}\boldsymbol{w} - \gamma\mathbf{1} + \mathbf{1} \leq \boldsymbol{z} \\ \boldsymbol{y} \geq \mathbf{0}, \boldsymbol{z} \geq \mathbf{0}, \end{array} \end{array} \tag{7.4}$$

où $\mathbf{1}$ est un vecteur dont les composantes sont égales à 1 (et permet d'exprimer la norme $L1$ $\|\boldsymbol{x}\|_1 = \sum x_i = \mathbf{1}^T\boldsymbol{x}$), n_c le nombre d'exemples de la classe c, et $\boldsymbol{A} \in \mathbb{R}^{n_1 \times D}$ et $\boldsymbol{B} \in \mathbb{R}^{n_2 \times D}$ les matrices dont les lignes contiennent respectivement les exemples des classes 1 et 2, et $\boldsymbol{w}$ le vecteur normal de l'hyperplan de séparation.

Mangasarian et Bradley proposent par la suite [36][35] de résoudre de manière conjointe le problème de la sélection de descripteurs par l'ajout d'un terme contraignant la minimisation du nombre de composantes non-nulles du vecteur $\boldsymbol{w}$. L'expression à minimiser devient donc, sous les mêmes contraintes :

$$\min_{\boldsymbol{w},\gamma,\boldsymbol{y},\boldsymbol{z}} \quad (1-\lambda)\left(\frac{\mathbf{1}^T\boldsymbol{y}}{n_1} + \frac{\mathbf{1}^T\boldsymbol{z}}{n_2}\right) + \lambda \|\boldsymbol{w}\|_0 , \tag{7.5}$$

où $\|\cdot\|_0$ est la « *norme zéro* » [1] et est égale au nombre de composantes non nulles. Le paramètre additionnel λ fixe le compromis entre les deux termes à minimiser.

Toutefois la dite norme zéro pose problème parce qu'elle n'est ni continue, ni dérivable. Les auteurs résolvent ce problème en approchant cette dernière par une exponentielle inverse :

$$\|\boldsymbol{w}\|_0 = \mathbf{1}^T\left(\mathbf{1} - e^{-\alpha\boldsymbol{v}}\right),$$

où $\boldsymbol{v}$ est un vecteur aux termes positifs bornant à l'excès les composantes de $\boldsymbol{w}$ (il définit donc une nouvelle contrainte $-\boldsymbol{v} \leq \boldsymbol{w} \leq \boldsymbol{v}$). L'exponentielle inverse est choisie par les auteurs pour sa simplicité et sa concavité qui garantit de bonnes propriétés de convergence.

L'algorithme FSV (*Feature Selection concaVe*) [35] ainsi défini est un exemple de méthode *embarquée* impliquant dans un même processus d'optimisation l'apprentissage du classifieur et

1. l'appellation courante de norme est ici abusive puisqu'elle ne vérifie pas la propriété d'homogénéité, à savoir $\|\lambda\boldsymbol{x}\| = |\lambda| \cdot \|\boldsymbol{x}\|$.

la sélection de descripteurs. Les auteurs font par ailleurs apparaître un problème d'optimisation de SVM en remplaçant la norme zéro par une classique norme $L2$ (terme additionnel $\frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$). On retrouve alors le principe de maximisation de la marge, mais l'algorithme se révèle incapable d'annuler des composantes du vecteur $\boldsymbol{w}$ et n'opère donc plus de sélection.

7.4.2 Approximation of the zeRO-norm Minimization (AROM)

L'utilisation des composantes du vecteur normal $\boldsymbol{w}$ pour le classement de descripteurs a été largement exploitée dans d'autres publications sur la sélection de descripteurs liée aux SVM. Comme nous l'avons expliqué, l'optimisation de Machines à Vecteurs de Support consiste en la détermination d'un vecteur $\boldsymbol{w}$ optimal, exprimé à partir des exemples $(\boldsymbol{x}_i, y_i)$ et des multiplicateurs de Lagrange α_i :

$$\boldsymbol{w} = \sum_i \alpha_i y_i \Phi(\boldsymbol{x}_i). \tag{7.6}$$

Le problème majeur lié à l'usage de $\boldsymbol{w}$ provient du fait que la fonction Φ n'est pas explicite pour la plupart des noyaux, on ne peut donc exprimer numériquement les composantes du vecteur. Plusieurs propositions contournent cette difficulté en se restreignant au noyau linéaire, parmi lesquelles la méthode d'Approximation de Minimisation de la norme zéro (AROM *Approximation of the zeRO-norm Minimization*) [244].

Se concentrant sur le problème de la minimisation de la norme zéro, les auteurs font le constat que l'approche de Bradley et Mangasarian souffre d'une grande complexité lorsque le nombre de descripteurs est élevé, et laisse de plus ouverte la question de la détermination du paramètre α. Ils proposent ainsi de substituer à ce problème la minimisation de la grandeur suivante :

$$\min_{\boldsymbol{w}} \quad \sum_{d=1}^{D} \ln |w_d|,$$

et montrent que le minimum atteint est pratiquement égal au minimum de $\|\boldsymbol{w}\|_0$. On peut constater intuitivement que la fonction ln favorise les composantes proches de zéro, forçant ainsi la minimisation du nombre de composantes non nulles. En appliquant la méthode de Franke et Wolke de descente de gradient, ils montrent que le problème converge vers un minimum local. Celui-ci est atteint par mises à jour successives du vecteur $\boldsymbol{w}$ (initialisé par exemple à $\boldsymbol{w} = \mathbf{1}$) en résolvant le problème suivant :

$$\begin{array}{ll} \min\limits_{\hat{\boldsymbol{w}}} & \sum_{d=1}^{D} |\hat{w}_d| \\ \text{sous les contraintes} & y_i \left(\hat{\boldsymbol{w}}^T (\boldsymbol{x}_i \bullet \boldsymbol{w}) + b\right) \geq 1, \end{array} \tag{7.7}$$

où $\bullet$ est le produit terme à terme (dit de Hadamard). La mise à jour du vecteur $\boldsymbol{w}$ se fait simplement en multipliant à chaque itération les composantes terme à terme avec le vecteur $\hat{\boldsymbol{w}}$ évalué : $\boldsymbol{w} \leftarrow \boldsymbol{w} \bullet \hat{\boldsymbol{w}}$, jusqu'à la convergence de $\boldsymbol{w}$. Nous désignerons par la suite cette méthode par AROM $L1$, du fait de l'usage de la norme correspondante.

Les auteurs proposent ensuite une approximation très efficace de l'algorithme en substituant à $L1$ la norme $L2$, ce qui mène à la formulation duale (on peut également exploiter la formulation primale) d'un problème d'apprentissage de SVM :

$$\begin{array}{ll} \min\limits_{\alpha_i} & \sum_{i=}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\boldsymbol{w} \bullet \boldsymbol{x}_i\right)^T \left(\boldsymbol{w} \bullet \boldsymbol{x}_j\right) \\ \text{sous les contraintes} & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & \alpha_i \geq 0 \end{array} \tag{7.8}$$

La mise à jour du vecteur $\boldsymbol{w}$ se fait de la même manière à partir du vecteur $\hat{\boldsymbol{w}}$, que l'on déduit des α_i estimés par la relation 7.6. Cette méthode, nommée AROM $L2$, est très avantageuse par rapport à la précédente puisqu'elle bénéficie de toutes les techniques de décomposition proposées dans la littérature pour l'apprentissage des SVM.

Les formulations 7.7 et 7.8 permettent en outre d'interpréter la méthode AROM comme une mise à jour de coefficients de pondération (les composantes w_d) sur les descripteurs. Ainsi un descripteur qui intervient peu dans le processus de classification sera pénalisé jusqu'à ne plus être pris en compte par ce dernier (lorsque le poids w_d est nul). La méthode AROM peut donc être considérée comme une méthode de sélection embarquée avec stratégie de recherche par optimisation de paramètres (PO).

7.4.3 Recursive Feature Extraction (RFE)

Par un développement en série de Taylor de la fonction objectif des SVM ($J = \frac{1}{2}\|\boldsymbol{w}\|^2$) au voisinage de son optimum, Guyon et al. [99] montrent que l'estimation de la dérivée $\frac{\partial J}{\partial w_d}$ justifie l'usage de la grandeur $R(d) = w_d^2$ comme critère de classement des descripteurs.

Néanmoins, se basant sur les constats énoncés en section 7.2.2, ils estiment qu'une simple stratégie BIN basée sur un tel classement peut se révéler largement sous-optimale, puisqu'elle équivaut à exclure un grand nombre de descripteurs en même temps, ce qui biaise la pertinence du critère énoncé. Il proposent donc une méthode à stratégie séquentielle *backward* (SEQ) d'éliminations successives des descripteurs.

L'algorithme RFE (*Recursive Feature Extraction*) consiste donc à éliminer à chaque itération le descripteur minimisant le critère $R(d) = w_d^2$ après apprentissage d'un SVM sur les descripteurs restants. La structure itérative permet de mettre à jour le classement des critères de pertinence après chaque élimination de descripteur.

Bien qu'efficace, la méthode se révèle ainsi beaucoup plus coûteuse que la méthode AROM, sans pourtant que ce coût soit réellement justifié par un écart notable de performances. Les auteurs proposent notamment d'éliminer plusieurs descripteurs à chaque itération pour en réduire la complexité, mais sans apporter de réponse théorique au nombre optimal de descripteurs à éliminer.

7.4.4 Sélection par minimisation de la borne Rayon-Marge (R2W2)

Il est également possible d'appliquer une stratégie PO (beaucoup plus économique que la stratégie SEQ déployée dans l'algorithme RFE) basé sur les facteurs de pondération introduits dans l'algorithme AROM, sans avoir à évaluer le vecteur normal de l'hyperplan, que nous noterons $\boldsymbol{w}_h$ dans cette section. Weston et al. ont proposé [47][245] un algorithme de sélection de descripteurs basé sur les critères d'évaluation de SVM introduits dans les sections 4.3.6 et 4.3.7. Nous nous concentrons ici sur la borne Rayon-Marge, la borne sur l'étendue étant beaucoup trop coûteuse quoique plus resserrée. La borne Rayon-Marge, dont on rappelle l'expression :

$$\mathcal{P}_{RM} = \frac{1}{n}\frac{R^2}{M^2} = \frac{1}{n}R^2\|\boldsymbol{w}_h\|^2,$$

a pour avantage de ne faire intervenir que la norme quadratique du vecteur $\boldsymbol{w}_h$, qui ne nécessite pas de connaître la fonction Φ puisque $\|\boldsymbol{w}_h\|_2^2 = k(\boldsymbol{w}_h, \boldsymbol{w}_h)$, où k est le noyau impliqué dans le classifieur SVM.

Les auteurs utilisent, de manière similaire à la méthode AROM, un vecteur de pondération $\boldsymbol{w}$ (qui ici n'est pas lié au vecteur normal $\boldsymbol{w}_h$) dont on peut résumer l'effet en introduisant le noyau pondéré $k_{\boldsymbol{w}}$:

$$k_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{w} \bullet \boldsymbol{x}, \boldsymbol{w} \bullet \boldsymbol{y}).$$

On peut ainsi exprimer la dérivée de la borne Rayon-Marge par rapport aux composantes w_d du vecteur $\boldsymbol{w}$:

$$\frac{\partial R^2\|\boldsymbol{w}_h\|^2}{\partial w_d} = R^2\frac{\partial\|\boldsymbol{w}_h\|^2}{\partial w_d} + \|\boldsymbol{w}_h\|^2\frac{\partial R^2}{\partial w_d},$$

avec :

$$\begin{aligned}\frac{\partial\|\boldsymbol{w}_h\|^2}{\partial w_d} &= -\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j\frac{\partial k_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial w_d}\\ \frac{\partial R^2}{\partial w_d} &= \sum_{i=1}^{n}\beta_i\frac{\partial k_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{x}_i)}{\partial w_d} - \sum_{i,j=1}^{n}\beta_i\beta_j\frac{\partial k_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial w_d},\end{aligned}$$

où l'on utilise l'expression du rayon R introduite dans l'équation 4.3. La dérivée $\frac{\partial k_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial w_d}$ est généralement évidente pour les noyaux usuels et ne pose donc pas de problème.

La sélection se fait donc par minimisation du critère Borne-Marge, en suivant une descente de gradient sur les composantes w_d. Les auteurs accélèrent l'algorithme en fixant à chaque itération les pires poids à zéro, arrêtant celui-ci lorsque seuls S poids non-nuls subsistent. On peut cependant

déduire de l'algorithme un classement de tous les descripteurs, sans sélection de sous-groupe, après convergence de la borne Rayon-Marge. Nous désignerons par la suite cette méthode par l'acronyme R2W2.

Comme nous le verrons dans la section expérimentale 7.7, la méthode R2W2 forme un très bon compromis entre performances et complexité. En effet, la stratégie de recherche PO garantie une complexité proportionnelle à la dimension, et se révèle donc, si l'algorithme converge en peu d'itérations, beaucoup plus rapide que la méthode RFE. De plus, contrairement aux autres méthodes liées aux SVM que nous avons présentées, celle-ci peut prendre en compte tout type de noyaux (pour peu que ce dernier soit dérivable par rapport aux w_d, ce qui est le cas de tous les noyaux usuels), et donc adapter la sélection de descripteurs à des cas non séparables linéairement.

La méthode est de type enveloppeur parce qu'elle implique l'apprentissage de SVM dans le processus de sélection (nécessaire pour évaluer le rayon R et le vecteur $\boldsymbol{w}_h$). Nous proposons dans la suite de ce chapitre plusieurs nouvelles méthodes de type filtre dont le principe est proche de R2W2. Nous verrons qu'elle réduisent la complexité en se passant de l'apprentissage de SVM.

7.5 Propositions d'algorithmes efficaces de sélection

7.5.1 Sélection pondérée basée sur le critère d'Alignement (SAS)

Toutes les méthodes de sélection présentées dans la section précédente impliquent l'apprentissage d'un SVM afin d'évaluerle critère utilisé. Nous avons présenté dans la section 4.4.1 le critère d'Alignement (ou KTA) qui permet d'évaluer les performances d'un noyau pour une tâche de classification donnée. Nous proposons ici un algorithme de sélection de descripteurs, basé sur la maximisation de l'Alignement par la mise à jour des composantes du vecteur de pondération $\boldsymbol{w}$. On rappelle que le KTA a pour expression :

$$\mathcal{A}(\boldsymbol{K}, \boldsymbol{K^*}) = \frac{\langle \boldsymbol{K}, \boldsymbol{K^*} \rangle_F}{\|\boldsymbol{K^*}\|_F \|\boldsymbol{K}\|_F}, \tag{7.9}$$

où $\boldsymbol{K}^* = \boldsymbol{y}\boldsymbol{y}^T$ est la matrice cible, et $\langle \cdot, \cdot \rangle_F$ le produit de Frobenius défini par $\langle \boldsymbol{A}, \boldsymbol{B} \rangle_F = \sum_{i,j} a_{ij} b_{ij}$. On définit ici la matrice de Gram pondérée $\boldsymbol{K}_{\boldsymbol{w}}$ du noyau pondéré $k_{\boldsymbol{w}}$. La dérivée de l'Alignement, pour la matrice de Gram pondérée, par rapport à la composante w_d du vecteur $\boldsymbol{w}$, a pour expression (équation 4.12, appliquée sur $\boldsymbol{K}_{\boldsymbol{w}}$) :

$$\frac{\partial}{\partial w_d} \mathcal{A}(\boldsymbol{K}_{\boldsymbol{w}}, \boldsymbol{K^*}) = \frac{\langle \partial_{w_d} \boldsymbol{K}_{\boldsymbol{w}}, \boldsymbol{K^*} \rangle_F}{\|\boldsymbol{K}_{\boldsymbol{w}}\|_F \|\boldsymbol{K^*}\|_F} - \frac{\langle \boldsymbol{K}_{\boldsymbol{w}}, \boldsymbol{K^*} \rangle_F \langle \boldsymbol{K}_{\boldsymbol{w}}, \partial_{w_d} \boldsymbol{K}_{\boldsymbol{w}} \rangle_F}{\|\boldsymbol{K}_{\boldsymbol{w}}\|_F^3 \|\boldsymbol{K^*}\|_F}. \tag{7.10}$$

Seule la matrice $\partial_{w_d} \boldsymbol{K}_{\boldsymbol{w}} = [\partial_{w_d} k_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{ij}$ nous fait défaut pour évaluer la dérivée de l'Alignement. Nous la noterons par la suite $\partial_d \boldsymbol{K}_{\boldsymbol{w}}$ pour simplifier les notations, de même pour la dérivée $\partial_d k_{\boldsymbol{w}}$ du noyau pondéré par rapport à la composante w_d.

Tout comme pour l'algorithme R2W2, on doit évaluer la dérivée du noyau par rapport au poids w_d. L'introduction d'une décomposition des noyaux en *noyaux dimensionnels* κ^d permet de simplifier l'expression des dérivées. On détaille ici les décompositions naturelles pour les noyaux usuels :

- **Linéaire :**
  $k_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{w} \bullet \boldsymbol{x})^T (\boldsymbol{w} \bullet \boldsymbol{y}) = \sum_d w_d^2 \kappa^d(\boldsymbol{x}, \boldsymbol{y})$
  $\kappa^d(\boldsymbol{x}, \boldsymbol{y}) = x_d \cdot y_d$
  $\partial_d k_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}) = 2\, w_d\, \kappa^d(\boldsymbol{x}, \boldsymbol{y})$

- **RBF Gaussien :**
  $k_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{||\boldsymbol{w} \bullet (\boldsymbol{x} - \boldsymbol{y})||^2}{2\sigma^2}\right) = \exp\left(-\sum_d w_d^2 \kappa^d(\boldsymbol{x}, \boldsymbol{y})\right)$
  $\kappa^d(\boldsymbol{x}, \boldsymbol{y}) = \frac{(x_d - y_d)^2}{2\sigma^2}$
  $\partial_d k_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}) = -2\, w_d\, \kappa^d(\boldsymbol{x}, \boldsymbol{y})\, k_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y})$

- **Polynomial :**
  $k_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}) = \chi_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y})^\delta$
  $\chi_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}) = 1 + c\, (\boldsymbol{w} \bullet \boldsymbol{x})^T (\boldsymbol{w} \bullet \boldsymbol{y}) = 1 + c \sum_d w_d^2 \kappa^d(\boldsymbol{x}, \boldsymbol{y})$
  $\kappa^d(\boldsymbol{x}, \boldsymbol{y}) = x_d \cdot y_d$
  $\partial_d k_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}) = 2\, \delta\, c\, w_d\, \kappa^d(\boldsymbol{x}, \boldsymbol{y})\, \chi_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y})^{\delta - 1}$

Il ressort de ces décompositions, particulièrement pour les noyaux linéaires et RBF gaussien, que le calcul des dérivées se déduit du noyau et des noyaux dimensionnels avec un coût additionnel très modéré.

L'algorithme SAS (*Scaled Alignment Selection*) de sélection que nous proposons consiste donc à déduire de la maximisation de l'Alignement le classement des descripteurs par ordre décroissant des poids w_d. On utilise pour la procédure d'optimisation un simple algorithme de montée de gradient avec initialisation de tous les poids à 1. Il s'agit d'une approche *filtre* (puisque aucun apprentissage de SVM n'est impliqué dans la sélection) liée à une stratégie PO.

On remarquera que dans le cas du noyau RBF gaussien pondéré, le paramètre σ est implicitement fixé par la détermination des facteurs de poids. En effet, si l'on définit le vecteur de paramètres $\boldsymbol{\Theta} = (\sigma, \boldsymbol{w})$, soit une valeur arbitraire $\tilde{\sigma}$, et $\tilde{\boldsymbol{\Theta}} = \left(\tilde{\sigma}, \frac{\tilde{\sigma}}{\sigma}\boldsymbol{w}\right)$, alors

$$k_{\boldsymbol{\Theta}}(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\sum_i w_i^2 (x_i - y_i)^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_i \left(\frac{\tilde{\sigma}}{\sigma} w_i\right)^2 (x_i - y_i)^2}{2\tilde{\sigma}^2}\right) = k_{\tilde{\boldsymbol{\Theta}}}(\boldsymbol{x}, \boldsymbol{y}).$$

Notre proposition n'est pas la première à faire intervenir le critère d'Alignement pour la sélection de descripteurs. La principale contribution sur le sujet [166] est basée sur une minimisation conjointe de l'opposé de l'Alignement et de la norme zéro du vecteur normal de l'hyperplan $\boldsymbol{w}_h$, s'inspirant de la stratégie proposée par Bradley et Mangasarian pour l'algorithme FSV (section 7.4.1). Néanmoins, la fonction objectif n'étant pas convexe, elle est décomposée comme une différence de deux fonctions convexes, qui permet l'usage d'une technique de minimisation spécifique (DCA, *Difference of Convex functions minimization Algorithm*, proposé dans [224]). La minimisation s'opère par une double boucle ; chaque étape de la boucle intérieure implique donc le calcul de l'Alignement, ce qui rend l'algorithme beaucoup plus complexe et coûteux que l'algorithme SAS proposé ici. De plus, afin de faire apparaître la décomposition comme différence de deux fonctions convexes, les auteurs suppriment le dénominateur de normalisation de l'Alignement, se justifiant par le fait que dans le cas du noyau RBF gaussien, la matrice de Gram est déjà bornée. Nous montrons dans la partie expérimentale l'importance de ce dénominateur en comparant l'algorithme SAS au SFS proposé ci-dessous.

7.5.2 Sélection Pondérée basée sur le produit de Frobenius (SFS)

L'algorithme SFS (*Scaled Frobenius Selection*) suit exactement le même principe que l'algorithme SAS, en substituant au critère d'Alignement le dit *critère de Frobenius*, qui se résume au produit de Frobenius entre la matrice de Gram et la matrice cible, c'est-à-dire un critère d'Alignement non normalisé :

$$\mathcal{F}(\boldsymbol{K}, \boldsymbol{K}^*) = \langle \boldsymbol{K}, \boldsymbol{K}^* \rangle_F . \tag{7.11}$$

Cette algorithme n'est défini ici que pour mettre en évidence sa moindre efficacité par rapport au critère d'alignement complet, comme le montrera la partie expérimentale.

L'absence de normalisation se révèle d'ailleurs immédiatement préjudiciable pour l'usage du noyau linéaire puisque dans de nombreux cas, le critère de Frobenius diverge vers l'infini lorsque les poids w_d augmentent arbitrairement. La maximisation étant impossible en pratique, on se limitera donc pour la méthode présente aux noyaux non-linéaires.

7.5.3 Sélection Forward basée sur le critère d'Alignement (FAS)

Les approches par noyau pondéré, de même que la méthode R2W2, offrent un très bon compromis entre performances et complexité. Toutefois, comme nous l'avons précisé, la stratégie PO ne permet pas de lier d'un point de vue théorique la pertinence des S meilleurs descripteurs au classement de leurs poids parmi les D descripteurs originaux.

La décomposition des noyaux linéaires et RBF gaussien en noyaux dimensionnels, présentée plus haut dans la section 7.5.1, offre un grand avantage puisqu'elle permet l'évaluation de l'effet individuel d'un descripteur additionnel sur une matrice de Gram. Les noyaux sont ici dans leur forme originale non pondérée. On définit les *matrices de Gram dimensionnelles* $\boldsymbol{\kappa}^d = \left[\kappa^d(\boldsymbol{x}_i, \boldsymbol{x}_j)\right]_{ij}$, afin de transposer les relations énoncées sous forme matricielle.

Dans le cas du noyau linéaire, la contribution individuelle se fait par sommation des matrices de Gram dimensionnelles, avec $\kappa_d(\boldsymbol{x}, \boldsymbol{y}) = x_d \cdot y_d$:

$$\boldsymbol{K} = \sum_{d=1}^{D} \boldsymbol{\kappa}^d.$$

Dans le cas du noyau RBF gaussien, la contribution individuelle se fait par le produit des matrices de Gram dimensionnelles ($\bigotimes$ représente le produit terme à terme de différentes matrices), avec $\kappa^d(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{(x_d - y_d)^2}{2\sigma^2}\right)$:

$$\boldsymbol{K} = \bigotimes_{d=1}^{D} \boldsymbol{\kappa}^d.$$

Ce constat nous permet de définir une stratégie de recherche séquentielle (SEQ) *forward* de sélections successives de descripteurs, basée sur le critère d'Alignement. Nous proposons ainsi la méthode de sélection FAS (*Forward Alignement Selection*), détaillée dans l'algorithme 3.

Algorithme 3 Forward Aligment Selection

D le nombre de descripteurs
S le nombre de descripteurs à sélectionner
Initialisation : $\boldsymbol{K}_0 \leftarrow 0$, liste de descripteurs restants $I_1 \leftarrow 1, \ldots, D$.
Calcul des matrices de Gram dimensionnelles $\boldsymbol{\kappa}^i \ \ \forall i \in I_1$.
pour $d = 1$ à S **faire**
  pour $i \in I_d$ **faire**
    Noyau linéaire : $\boldsymbol{K}_d^i = \boldsymbol{K}_{d-1} + \boldsymbol{\kappa}^i$
    ou RBF gaussien : $\boldsymbol{K}_d^i = \boldsymbol{K}_{d-1} \bullet \boldsymbol{\kappa}^i$
    Calcul de l'Alignement : $A_d^i = \mathcal{A}(\boldsymbol{K}_d^i, \boldsymbol{K}^*)$.
  fin pour
  Sélection du descripteur de rang d : $i_d = \arg\max_{i \in I_d} A_d^i$
  $\boldsymbol{K}_d = \boldsymbol{K}_d^{i_d}$
  $I_{d+1} = I_d \setminus \{i_d\}$
fin pour
résultat : Liste ordonnée des descripteurs sélectionnés $i_1, \ldots, i_S$.

La stratégie séquentielle favorise ainsi la sélection des premiers descripteurs en considérant toutes les possibilités parmi les D descripteurs disponibles. Le calcul préalable des matrices de Gram dimensionnelles offre de plus une importante réduction du coût de calcul, l'essentiel des boucles de l'algorithme consistant donc en sommes ou produits terme à terme (donc parallélisables) de matrices. Cependant la complexité reste beaucoup plus élevée que l'algorithme pondéré SAS, en particulier lorsque le nombre de descripteurs D est élevé, comme nous l'avons précisé dans la présentation de la stratégie de recherche séquentielle (section 7.2.3). De plus l'algorithme tend à accumuler les erreurs au fil des itérations, et devient donc de moins en moins fiable à mesure que le nombre de descripteurs sélectionnés augmente.

Cependant pour une sélection restreinte ($S \ll D$), l'algorithme FAS obtient de meilleurs résultats que l'approche SAS, tout en impliquant une complexité raisonnable (en $O(SD)$).

7.5.4 Sélection Pondérée sur le critère de Séparabilité (SCSS)

Sur la base des critiques formulées à l'égard du critère d'Alignement (section 4.4.1.4), nous avons introduit le critère de Séparabilité de Classes « Kernelisé » (KCS), défini comme le rapport des dispersions inter-classes et intra-classes dans l'espace transformé. On rappelle l'expression du critère KCS, défini par rapport aux matrices $\boldsymbol{B}$ et $\boldsymbol{W}$, exprimées dans les équations 4.17 et 4.18 :

$$\mathcal{J} = \frac{\mathbf{1}_n^T \boldsymbol{B} \mathbf{1}_n}{\mathbf{1}_n^T \boldsymbol{W} \mathbf{1}_n} = \frac{\Sigma(\boldsymbol{B})}{\Sigma(\boldsymbol{W})},$$

où l'opérateur Σ est égal à la somme des composantes d'une matrice ($\Sigma(\boldsymbol{A}) = \sum_{ij} a_{ij}$).

Dans le cas du noyau pondéré $k_{\boldsymbol{w}}$, on peut donc en déduire l'expression de la dérivée par rapport au poids w_d :

$$\frac{\partial \mathcal{J}}{\partial w_d} = \frac{\Sigma(\partial_{w_d}\boldsymbol{B})\Sigma(\boldsymbol{W}) - \Sigma(\boldsymbol{B})\Sigma(\partial_{w_d}\boldsymbol{W})}{\Sigma(\boldsymbol{W})^2},$$

avec :

$$\Sigma(\partial_{w_d}\boldsymbol{B}) = \Sigma(\partial_{w_d}\boldsymbol{K_{11}}) + \Sigma(\partial_{w_d}\boldsymbol{K_{12}}) - \Sigma(\partial_{w_d}\boldsymbol{K}), \tag{7.12}$$

$$\Sigma(\partial_{w_d}\boldsymbol{W}) = \sum_{i=1}^{n} \partial_{w_d} k_{ii} - \Sigma(\partial_{w_d}\boldsymbol{K_{11}}) - \Sigma(\partial_{w_d}\boldsymbol{K_{12}}). \tag{7.13}$$

On a ôté l'indice $\boldsymbol{w}$ de la matrice de Gram $\boldsymbol{K_w}$ et de ses sous-blocs pour clarifier les notations, mais ici le critère est bien calculé sur la matrice de Gram du noyau pondéré.

On propose donc l'algorithme SCSS (*Scaled Class Separability Selection*) basé sur le même principe que l'algorithme SAS, où l'on substitue le critère de Séparabilité dans l'espace transformé ($\mathcal{J}$) au critère d'Alignement.

Cependant, bien que le critère de séparabilité soit a priori plus fiable que le critère d'Alignement (on a vu que le premier prend en compte la dispersion intra-classes, tandis que le second ne fait intervenir que la mesure de dispersion inter-classe), son expression introduit une instabilité numérique dans la maximisation de $\mathcal{J}$ que nous avons évoquée dans la section 4.4.2. On appliquera donc ici à nouveau la procédure de régularisation proposée dans cette précédente section pour éviter ce problème. Nous verrons cependant dans la partie expérimentale que le critère KCS n'apporte pas de gain en pratique par rapport au critère d'Alignement pour l'approche par optimisation sur noyau pondéré.

Notons qu'un algorithme de sélection de descripteur basé sur le critère KCS a également été proposé par Wang [239]. Néanmoins, afin de contourner le problème de régularisation, l'auteur avance que le critère KCS est borné inférieurement par la grandeur $\operatorname{tr} S_b$ (qui se trouve en fait être son numérateur), et base en conséquence toutes ses expériences sur ce critère plus simple, qui se trouve être, comme nous l'avons montré, le produit de Frobenius entre la matrice de Gram et la matrice cible (soit le critère de Frobenius, comme nous l'avons appelé dans la section 7.5.2). La dispersion intra-classes est alors absente du critère employé. On peut donc considérer que le critère KCS n'est pas réellement exploité par l'auteur.

7.5.5 Sélection sur le Discriminant de Fisher Kernelisé (KFDS)

La dernière méthode que nous proposons s'appuie sur un modèle très proche du critère de séparation des classes, basé sur les matrices de dispersion dans l'Analyse Discriminante de Fisher Kernelisée (*Kernel Fisher Discriminant Analysis* KFDA).

Le problème de l'Analyse Discriminante de Fisher a été introduit dans la section 3.2 et consiste en la détermination d'un hyperplan (de vecteur normal $\boldsymbol{w}_h$) de séparation entre deux classes, autrement dit d'un axe $\boldsymbol{w}_h$ de projection des données maximisant le critère de l'équation 4.13 ($J = \frac{\operatorname{tr} \boldsymbol{S}_b}{\operatorname{tr} \boldsymbol{S}_w}$). Le problème peut ainsi être formulé sous la forme de la maximisation du critère $J(\boldsymbol{w}_h)$ suivant :

$$J(\boldsymbol{w}_h) = \frac{\boldsymbol{w}_h^T \boldsymbol{S}_b \boldsymbol{w}_h}{\boldsymbol{w}_h^T \boldsymbol{S}_w \boldsymbol{w}_h}.$$

où l'on retrouve les matrices de dispersion inter-classes et intra-classes (équations 4.14 et 4.15, l'expression suivante de $\boldsymbol{S}_b$ est égale à un facteur près à celle de l'équation 4.14) :

$$\begin{aligned} \boldsymbol{S}_b &= (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T, \\ \boldsymbol{S}_w &= \sum_{c=1,2} \sum_{\boldsymbol{x}_i \in \mathcal{S}_c} (\boldsymbol{x}_i - \boldsymbol{\mu}_c)(\boldsymbol{x}_i - \boldsymbol{\mu}_c)^T. \end{aligned}$$

Mika et al. ont montré [158] qu'il est possible de formuler ces dernières exclusivement en termes de produits scalaires, ce qui permet, de manière similaire aux SVM, d'étendre le champ des surfaces

de séparation à des surfaces plus complexes, en leur substituant une fonction noyau k. Soit Φ la fonction de transformation relative au noyau k, on « kernelise » les matrices de dispersion de la manière suivante :

$$\begin{aligned} \boldsymbol{S}_b^\Phi &= (\boldsymbol{\mu}_1^\Phi - \boldsymbol{\mu}_2^\Phi)(\boldsymbol{\mu}_1^\Phi - \boldsymbol{\mu}_2^\Phi)^T, \\ \boldsymbol{S}_w^\Phi &= \sum_{c=1,2} \sum_{\boldsymbol{x}_i \in \mathcal{S}_c} (\Phi(\boldsymbol{x}_i) - \boldsymbol{\mu}_c^\Phi)(\Phi(\boldsymbol{x}_i) - \boldsymbol{\mu}_c^\Phi)^T, \end{aligned}$$

où l'on a introduit les centres des classes dans l'espace transformé $\boldsymbol{\mu}_c^\Phi = \frac{1}{n_c} \sum_{\boldsymbol{x}_i \in \mathcal{S}_c} \Phi(\boldsymbol{x}_i)$. Les résultats de la Théorie des Noyaux Reproduisants nous permettent d'affirmer que le vecteur $\boldsymbol{w}_h$ se trouve dans l'espace engendré par les exemples de l'ensemble d'apprentissage, soit : $\boldsymbol{w}_h = \sum_{i=1}^n \alpha_i \Phi(\boldsymbol{x}_i)$, ce qui permet [244] de reformuler les termes du critère $J(\boldsymbol{w}_h)$:

$$\begin{aligned} \boldsymbol{w}_h^T \boldsymbol{S}_b^\Phi \boldsymbol{w}_h &= \boldsymbol{\alpha}^T \boldsymbol{M} \boldsymbol{\alpha}, \\ \boldsymbol{w}_h^T \boldsymbol{S}_w^\Phi \boldsymbol{w}_h &= \boldsymbol{\alpha}^T \boldsymbol{N} \boldsymbol{\alpha}, \end{aligned}$$

où l'on a introduit les deux matrices $\boldsymbol{M}$ et $\boldsymbol{N}$. On décompose la matrice de Gram $\boldsymbol{K}$ en deux blocs verticaux, chacun relatif à une classe, soit $\boldsymbol{K} = [\boldsymbol{K}_1 \boldsymbol{K}_2]$, avec $[\boldsymbol{K}_c]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ pour $i = [1, \ldots, n]$ et $\boldsymbol{x}_j \in \mathcal{S}_c$. De plus, on définit les vecteurs $\boldsymbol{M}_c$ comme les moyennes de colonnes des matrices $\boldsymbol{K}_c$, soit $[\boldsymbol{M}_c]_i = \frac{1}{n_c} \sum_{\boldsymbol{x}_j \in \mathcal{S}_c} k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Cette décomposition nous permet d'exprimer les matrices introduites :

$$\begin{aligned} \boldsymbol{M} &= (\boldsymbol{M}_1 - \boldsymbol{M}_2)(\boldsymbol{M}_1 - \boldsymbol{M}_2)^T \\ \boldsymbol{N} &= \sum_{c=1,2} \boldsymbol{K}_c \left(\boldsymbol{I} - \frac{1}{n_c} \boldsymbol{1}_{n_c} \right) \boldsymbol{K}_c^T. \end{aligned}$$

La matrice $\boldsymbol{1}_{n_c}$ étant une matrice de taille $n_c \times n_c$ dont les termes sont égaux à 1.

L'Analyse du Discriminant de Fisher Kernelisé (KFDA) consiste donc en la maximisation du critère J suivant, par rapport aux coefficients α_i :

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^T \boldsymbol{M} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \boldsymbol{N} \boldsymbol{\alpha}}.$$

Le problème est résolu de manière analogue à l'algorithme LDA non-kernelisé, par la recherche des vecteurs propres de la matrice $\boldsymbol{N}^{-1}\boldsymbol{M}$.

L'extraction d'un vecteur $\boldsymbol{w}_h$ de séparation nous mène à appliquer une transposition de la méthode AROM (basée sur les SVM), sur le résultat précédent. On propose l'algorithme KFDS (*Kernel Fisher Discriminant Selection*) qui consiste à mettre à jour itérativement les facteurs de pondération de descripteurs du vecteur $\boldsymbol{w}$ par un produit terme à terme avec le vecteur $\boldsymbol{w}_h$ ($\boldsymbol{w} \leftarrow \boldsymbol{w} \bullet \boldsymbol{w}_h$), puis à appliquer à l'itération suivante l'algorithme d'analyse de Fisher sur les descripteurs pondérés (on utilise donc la matrice de Gram pondérée $\boldsymbol{K}_{\boldsymbol{w}}$).

Cependant, afin de pouvoir exprimer le vecteur $\boldsymbol{w}_h$, l'algorithme proposé est soumis aux mêmes contraintes qu'AROM (transformée Φ explicite), et l'on limite donc pour l'instant son usage au noyau linéaire.

7.5.6 Avantages des méthodes proposées

Volume en mémoire

L'un des principaux avantages des méthodes proposées est leur adaptabilité en matière d'espace mémoire. En effet, la plupart des méthodes de sélection de descripteurs ne peuvent être appliquées sur un nombre trop élevé (plusieurs milliers) d'exemples puisqu'elles impliquent des structures numériques trop grandes pour être contenues dans des volumes classiques de mémoire vive.

Les deux critères proposés, de même que leurs dérivées par rapport aux facteurs de pondération, diffèrent des autres critères, du fait qu'ils sont exclusivement basés sur des termes additifs (des traces et des sommes). Ils peuvent donc être calculés itérativement au moyen d'une décomposition

en blocs des matrices de Gram ; le calcul de la matrice cible n'est d'ailleurs pas nécessaire puisqu'il équivaut, si le bloc est homogène en termes de classes, qu'à un éventuel changement de signe. Les besoins en mémoire peuvent donc être arbitrairement faibles, selon le compromis choisi avec la complexité (par exemple le pré-calcul des matrices de Gram dimensionnelles est largement profitable en termes de complexité, mais suppose un coût en stockage conséquent).

De plus, les matrices de Gram étant symétriques, seule la moitié des blocs non diagonaux est nécessaire pour l'évaluation des critères, ce qui apporte un gain en complexité supplémentaire.

Ces remarques ne s'appliquent pas cependant au dernier algorithme (KFDS), qui implique une inversion de matrice.

Complexité

L'approche pondérée basée sur le critère d'Alignement (SAS, section 7.5.1) est proposée ici comme une alternative à la méthode R2W2 présentée plus haut (section 7.4.4). La méthode R2W2 est basée sur une double boucle d'optimisation, chaque itération externe implique donc :

1. L'apprentissage d'un SVM pour évaluer les facteurs α_i et la norme $\|\boldsymbol{w}\|^2$.
2. L'évaluation de la matrice de Gram $\boldsymbol{K}$.
3. Une boucle d'optimisation par programmation quadratique pour évaluer le rayon R.
4. Le calcul des matrices dérivées $\partial_{w_d}\boldsymbol{K}$ pour chaque facteur de pondération w_d.

Les méthodes basées sur l'Alignement (KTA) et la Séparabilité des Classes (KCS) ne nécessitent que les étapes 2 et 4 de la boucle R2W2, sans impliquer l'apprentissage de SVM ou la résolution de problèmes de programmation quadratique, qui sont tous les deux des phases particulièrement coûteuses de l'algorithme. Les mesures de temps de calcul prodiguées dans la section expérimentale 7.7 confirmeront que les méthodes proposées sont plus rapides que R2W2, et présentent des performances comparables, voire meilleures dans certains cas.

De la même manière, l'algorithme KFDS (section 7.5.5), proposé comme alternative à la méthode AROM (section 7.4.2), se révèle beaucoup plus rapide que cette dernière, la résolution d'un système matriciel $\boldsymbol{Ax} = \boldsymbol{B}$ étant moins coûteuse que l'apprentissage d'un SVM, nécessaire dans les itérations de la méthode AROM.

7.6 Synthèse

Le tableau 7.1 synthétise la taxonomie des différents algorithmes de sélection de descripteurs présentés dans ce chapitre. La troisième colonne précise, dans le cas des méthodes adaptées aux SVM, quels types de noyaux sont pris en compte, la quatrième indique si l'algorithme est de type filtre, enveloppeur ou embarqué (selon la taxonomie présentée dans la section 7.2.4), enfin la cinquième précise la stratégie de recherche associée (parmi les stratégies énumérées dans la section 7.2.3).

On peut voir que les algorithmes proposés sont exclusivement des méthodes de type filtre, ce qui explique leur moindre complexité, mais prenant toutefois en compte les spécificités des SVM, et adoptant en majorité une stratégie d'optimisation, également largement exploitée dans les méthodes existantes basées sur les SVM.

7.7 Expériences comparatives

Nous présentons dans cette section un protocole expérimental, à la fois sur des données synthétiques et réelles, destiné à comparer les diverses approches pour la sélection de descripteurs et à valider les algorithmes proposés dans ce document. Nous exploiterons toutes les méthodes énumérées dans la synthèse de la section 7.6 précédente, à l'exception de IRMFSP, Kolomogorov-Smirnov et FSV, dont les résultats sont globalement décevants par rapport aux méthodes plus

	Section	Noyaux	Paradigme	Stratégie de recherche
Méthodes existantes				
Fisher	7.3.1	·	Filtre	BIN
IRMFSP	7.3.2	·	Filtre	SEQ
Kolmogorov-Smirnov	7.3.3	·	Filtre	BIN
FSV	7.4.1	Linéaire	Embarquée	PO
AROM	7.4.2	Linéaire	Embarquée	PO
RFE	7.4.3	Linéaire	Enveloppeur	SEQ
R2W2	7.4.4	Tous	Enveloppeur	PO
Méthodes proposées				
SAS	7.5.1	Tous	Filtre	PO
SFS	7.5.2	Non-linéaire	Filtre	PO
FAS	7.5.3	Linéaire & RBF	Filtre	SEQ
SCSS	7.5.4	Tous	Filtre	PO
KFDS	7.5.5	Linéaire	Filtre	PO

TABLE 7.1 – Tableau récapitutif des méthodes de sélection de descripteurs présentées dans ce chapitre, avec renvoi aux sections correspondantes. Y sont précisés les types de noyaux pris en compte, le type d'algorithme et la stratégie de recherche.

récentes. Nous conservons néanmoins le critère de Fisher, qui sert de référence, en raison de son usage très courant dans la littérature, et de son coût extrêmement réduit en temps de calcul.

Les commentaires porteront sur les courbes d'erreur, plus lisibles que les tableaux de résultats. On remarquera que sur toutes les figures, les méthodes proposées sont représentées par des pointillés et les méthodes de l'état de l'art sont en traits plein.

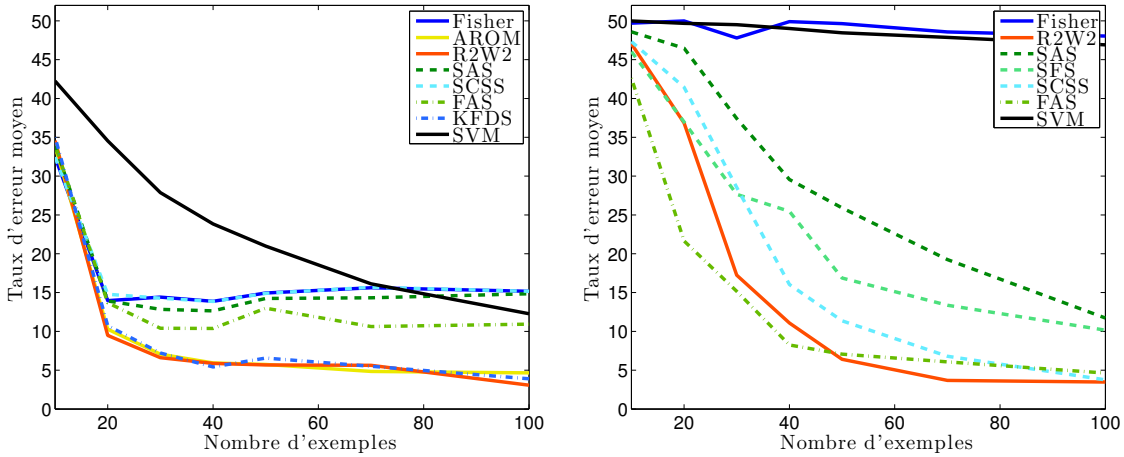
7.7.1 Données artificielles

Nous avons reproduit ici l'expérience décrite dans [245] et [47], incluant les protocoles de synthèse des données et d'évaluation des résultats. L'expérience compare les performances des différents algorithmes sur un problème linéairement séparable (que nous désignerons par « problème linéaire ») et un problème non-séparable linéairement (que nous désignerons par « problème non-linéaire »), par l'évaluation des SVM sur deux descripteurs sélectionnés parmi un ensemble conséquent de descripteurs redondants ou non-pertinents. Les approches basées sur des noyaux emploient respectivement un noyau linéaire et un noyau RBF gaussien pour chacun des cas. Nous précisons que sur toutes les expériences impliquant un noyau linéaire, l'approche par noyau pondéré sur le critère de Frobenius (SFS) n'est pas considérée car la définition du critère implique une divergence des facteurs de pondération vers l'infini lors de la phase d'optimisation.

Le problème linéaire réunit 202 descripteurs synthétiques dont seulement 6 ne sont pas du bruit, et sont corrélés entre eux par groupes de 3. Le problème non-linéaire regroupe 52 descripteurs dont seulement 2 ne sont pas du bruit, mais définissent des distributions multi-gaussiennes non-séparables linéairement. Nous invitons le lecteur à consulter l'article original [245] pour une description plus détaillée de la synthèse des données.

L'expérience consiste à observer l'évolution des performances lorsque le nombre n d'exemples d'apprentissage (générés aléatoirement) varie entre 10 et 100. A chaque itération les deux meilleurs descripteurs sont sélectionnés, un classifieur SVM est appris sur ces deux composantes, à partir des mêmes exemples d'apprentissage, et le taux d'erreur est calculé sur un ensemble de 500 exemples de test (générés selon la même distribution). La procédure est répétée de manière à évaluer l'erreur moyenne sur 40 itérations. La figure 7.1 montre les performances des différentes méthodes sur les deux problèmes. La courbe noire pleine indique les résultats obtenus en conservant tous les descripteurs pour l'apprentissage des SVM, afin d'évaluer l'amélioration apportée par la sélection de descripteurs.

Le problème linéaire, dont les résultats apparaissent sur la figure 7.1(a), illustre le principal défaut des méthodes proposées basées sur l'alignement (SAS, SFS et FAS) et le critère de séparabilité des classes (SCSS) dans leur incapacité à trouver la meilleure solution en présence de descripteurs



(a) noyau linéaire (b) noyau non-linéaire

FIGURE 7.1 – Comparaison des performances sur un problème séparable linéairement (a) et un problème non-séparable linéairement (b), impliquant un grand nombre de descripteurs bruités.

fortement redondants. Avec ces méthodes, de même qu'avec le critère de Fisher, le taux d'erreur tend vers 15% lorsque n augmente, parce que deux descripteurs pertinents mais redondants sont sélectionnés, au lieu de deux descripteurs indépendants apportant plus d'information. L'approche *forward* sur l'alignement (FAS) apporte cependant une amélioration sensible des performances par rapport à l'approche pondérée (SAS). Par contre les approches R2W2, AROM et la méthode proposée sur le discriminant de Fisher kernelisé (KFDS) se montrent plus aptes à prendre en compte ces redondances et présentent des résultats comparables.

Le problème non-linéaire (figure 7.1(b)) n'implique pas de redondance et est centré sur la capacité à détecter la complémentarité de deux descripteurs indépendants pour la séparation des classes. Sans surprise, le critère de Fisher est ici totalement incapable de distinguer les descripteurs pertinents, de même que les autres méthodes basées sur un noyau linéaire (AROM et le Fisher kernelisé KFDS) dont les résultats ne sont pas affichés pour une meilleure lisibilité, mais similaires à ceux du critère de Fisher. L'approche *forward* sur l'alignement (FAS) obtient ici les meilleurs résultats, avec l'approche R2W2. Ce constat montre la pertinence de l'approche *forward*, comparée aux approches par noyaux pondérés (SAS, SFS et SCSS), pour la sélection d'un nombre réduit de descripteurs. On remarque pour finir que le critère de séparabilité de classes (SCSS) apparaît ici plus efficace que le critère d'alignement (SAS), mais ce constat ne se retrouve malheureusement pas sur les données réelles. On remarque également que l'usage du critère de Frobenius (SFS) par rapport à l'Alignement complet (SAS) réduit les performances du système. Ce résultat sera confirmé par les expériences suivantes.

7.7.2 Données réelles

Nous avons également testé les méthodes proposées sur des données réelles. Nous réemployons ici les bases de données introduites dans la section 4.6 et présentées en détail dans l'annexe B, dont trois sont des bases disponibles sur le dépôt public UCI [16], et la dernière une base construite à partir de nos données sur le problème de classification *parole/musique*. Les bases ont été choisies pour couvrir un éventail assez diversifié de configurations, en termes de nombre de descripteurs originaux et de nombre d'exemples d'apprentissage. Ainsi, les bases *Ionosphere* et *Spambase* offrent un ensemble modéré d'exemples et de descripteurs (de l'ordre de la centaine) tandis que la base *Lymphoma*, tirée d'une expérience décrite dans [244] et [99], est caractérisée par une très vaste collection de descripteurs (plusieurs milliers) et un nombre réduit d'exemples (quelques dizaines). La base *parole/musique* se distingue par un nombre très conséquent d'exemples (20000).

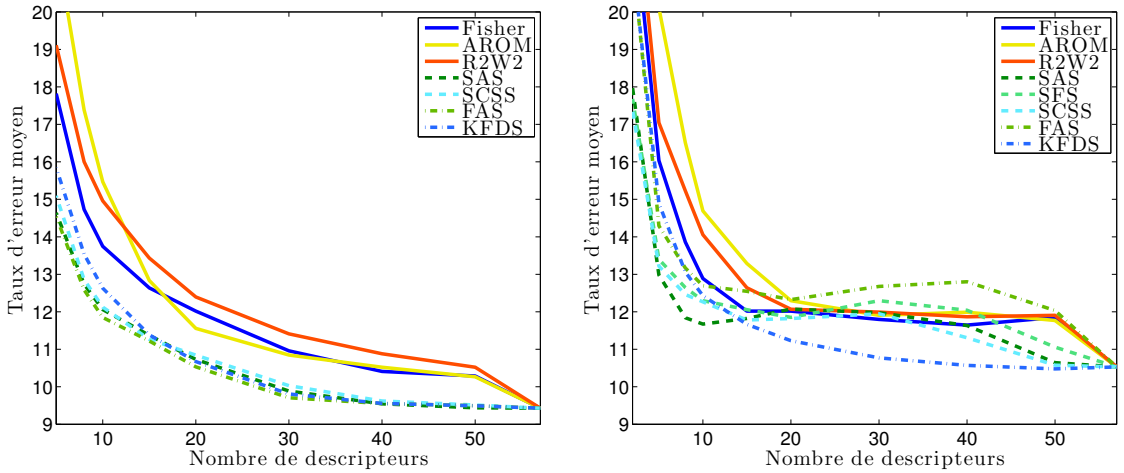
Le protocole d'évaluation est proche de celui déployé sur les données artificielles. La sélection de descripteurs est appliquée sur un ensemble d'apprentissage de n_{appr} exemples tirés aléatoirement parmi les n exemples de la base (qui contient n_1 et n_2 exemples pour chacune des deux classes). Un

classifieur SVM est ensuite appris sur le même ensemble d'apprentissage dont on a sélectionné les D descripteurs les plus pertinents, D étant ici le paramètre variable de l'évaluation, alors que l'on faisait varier n_{appr} pour les problèmes sur données synthétiques. Le taux d'erreur moyen est calculé sur 30 itérations de ce processus, et l'on fournira généralement les résultats avec un noyau linéaire et un noyau non-linéaire (RBF gaussien). Le tableau 7.2 résume les principales caractéristiques des bases exploitées ici.

Base	Nb d'exemples n (n_1/n_2)	n_{appr}	n_{test}	Nb descripteurs
Artificiel linéaire	synthétique	10 à 100	500	202
Artificiel non-linéaire	synthétique	10 à 100	500	52
Lymphoma	96 (34 / 62)	60	36	4026
Ionosphere	351 (126 / 225)	250	101	34
Spambase	4601 (2788 / 1813)	500	1000	57
Parole/musique	20000 (10000 / 10000)	500	500	321

TABLE 7.2 – Caractéristiques comparées des bases employées pour l'évaluation.

7.7.2.1 Spambase



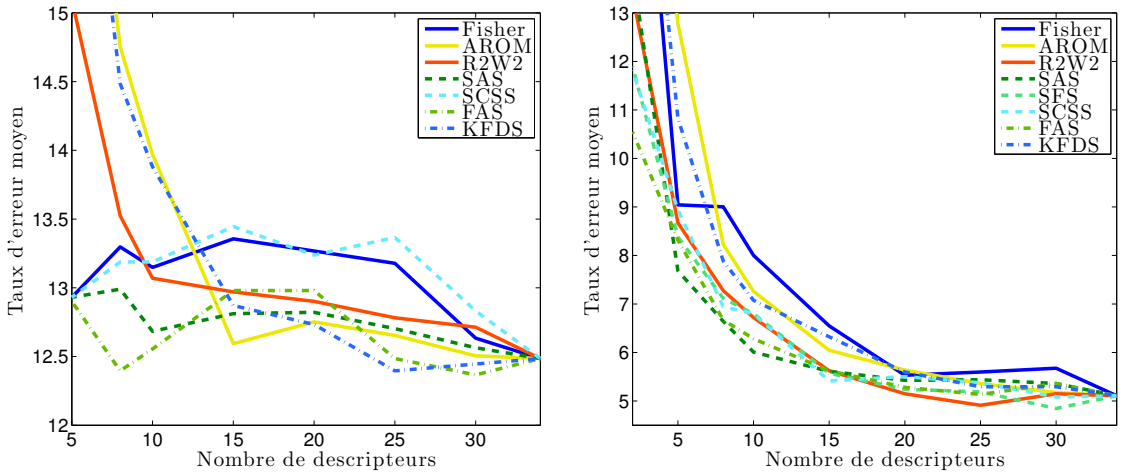
(a) noyau linéaire (b) noyau RBF gaussien

FIGURE 7.2 – Comparaison des performances entre les différentes méthodes sur la base *Spambase*.

Les résultats observés sur la base *Spambase*, illustrés par la figure 7.2, montrent en premier lieu la séparabilité linéaire du problème, étant donné que le noyau RBF gaussien n'apporte aucune amélioration par rapport au noyau linéaire. Cet exemple ne semble pas impliquer de descripteurs non pertinents (ou bruités) pénalisant la classification, puisque le taux d'erreur décroît de manière monotone lorsque le nombre de descripteurs sélectionnés augmente. Pour les deux noyaux, on constate que les algorithmes proposés apportent un net avantage par rapport aux approches existantes. Ainsi, dans le cas linéaire, on observe un réduction de 3% du taux d'erreur pour $d = 10$ sur les 15% d'erreur mesurés avec l'approche R2W2 (soit une réduction relative du taux d'erreur de l'ordre de 20%). Les très bonnes performances de l'approche basée sur le discriminant de Fisher kernelisé (KDFS) sont en outre surprenantes lorsqu'elle est suivie d'une classification avec noyau non-linéaire (figure 7.2 b), bien que l'algorithme KDFS soit basé exclusivement sur un noyau linéaire, ce qui tend à confirmer la séparabilité linéaire du problème. Le handicap de l'approche *forward* FAS sur le noyau RBF gaussien (qui sera confirmé par la suite), comparé aux approches par noyaux pondérés, montre que la complexité de cet algorithme n'apporte pas d'amélioration notable sur de petits ensembles d'apprentissage, à cause de la modélisation trop parcellaire des distributions de classes. L'écart de performances entre les approches pondérées SAS et SFS (où

le dénominateur de l'alignement est omis) sur le noyau RBF gaussien confirme par ailleurs la pertinence du terme de normalisation dans l'expression de l'alignement.

7.7.2.2 Ionosphere



(a) noyau linéaire (b) noyau RBF gaussien

FIGURE 7.3 – Comparaison des performances entre les différentes méthodes sur la base *Ionosphere*. On remarquera la différence d'échelle en ordonnées entre les deux figures (a) et (b).

Contrairement au cas précédent, le base *Ionosphere*, dont les résultats sont reportés sur la figure 7.3, constitue clairement un problème non-séparable linéairement. On constate en effet que le taux d'erreur reste supérieur à 12.5% avec le noyau linéaire, tandis que le noyau RBF gaussien permet de réduire celui-ci jusqu'à environ 5%. Les résultats mitigés des approches linéaires (Fisher, Fisher kernelisé et AROM) par rapport aux autres approches, viennent confirmer cet constat. Nous préférons donc ici nous concentrer sur les résultats mesurés avec le noyau RBF gaussien (figure 7.3(b)). À nouveau la pente décroissante quasi-monotone du taux d'erreur, que l'on retrouve sur toutes les méthodes, exclue l'hypothèse de la présence de descripteurs bruités pénalisant la classification. Les résultats confirment globalement les observations portées sur la base *Spambase*, à savoir une légère baisse des performances lorsque l'on substitue à l'alignement standard sur noyau pondéré (SAS) le critère allégé de Frobenius (SFS), ainsi que l'absence d'amélioration lorsque l'on emploie l'algorithme *forward* (FAS) très coûteux, par rapport à l'approche par noyau pondéré (SAS). De même, on n'observe pas d'amélioration lorsque l'on emploie le critère de séparabilité de classes (SCSS), pourtant théoriquement mieux justifié que le critère d'alignement. Toutefois, tous les algorithmes proposés ici, à l'exception du déterminant de Fisher kernelisé (KFDS) qui ne peut prendre en compte qu'un noyau linéaire, permettent d'obtenir des performances comparables à celles mesurées avec l'approche R2W2, tout en impliquant une charge de calcul plus réduite, comme nous le montrerons dans la section 7.7.3.

7.7.2.3 Lymphoma

La base de données *Lymphoma* est constituée d'un nombre très réduit d'exemples caractérisés par plusieurs milliers de composantes, cas typique des données génétiques traitées en bioinformatique. Ce problème particulier nous permet d'évaluer le comportement des algorithmes proposés sur de très larges collections de descripteurs, généralement fortement redondantes. L'approche *forward* sur critère d'alignement (FAS) n'a pas été appliquée ici du fait de sa complexité quadratique par rapport au nombre de descripteurs. Nous reproduisons ici l'expérience décrite dans [244] pour l'évaluation de la méthode AROM, qui se restreignait donc à l'usage du seul noyau linéaire. La méthode SFS n'est donc pas testée ici. La figure 7.4 montre les résultats obtenus pour les différentes méthodes, avec un noyau linéaire.

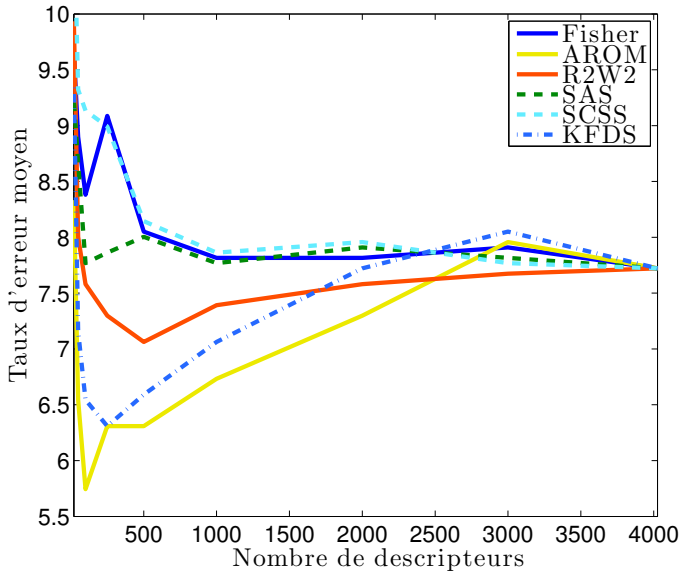


FIGURE 7.4 – Comparaison des performances entre les différentes méthodes sur la base *Lymphoma*.

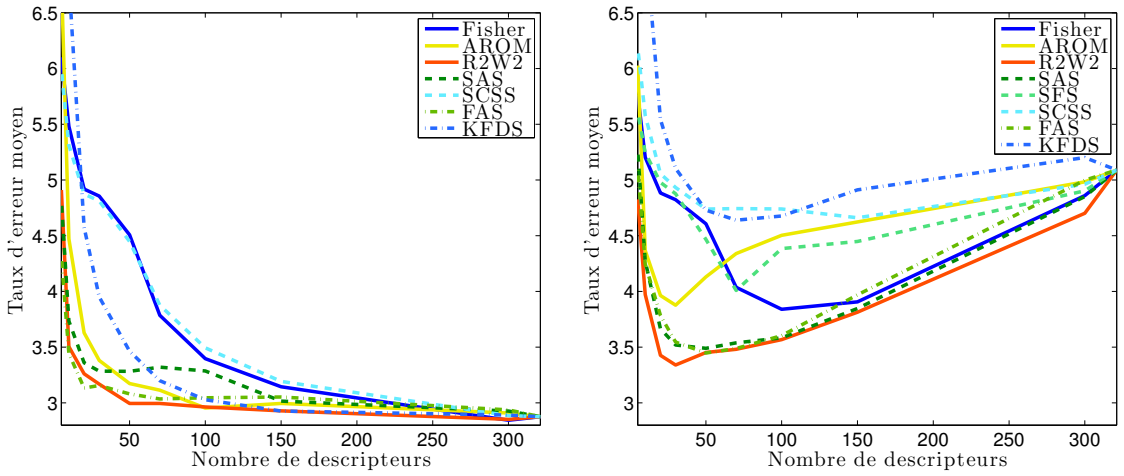
Les minima observés sur les courbes d'erreur des méthodes R2W2, AROM et Fisher kernelisé (KFDS) confirment le fait que la base contient un grand nombre de descripteurs non-pertinents, dont l'écrémage améliore les performances de classification. On constate d'ailleurs que le processus de sélection se révèle largement bénéfique en restreignant fortement le nombre de descripteurs. Ceci s'explique par le fait que l'information apportée par les descripteurs pertinents se trouve « diluée » dans la part dominante de bruit portée par le reste des descripteurs. De plus, les mauvais résultats obtenus avec le critère de Fisher montrent la forte interdépendance des descripteurs pertinents dans l'optimisation du problème, qui ne peut être mesurée indépendamment sur chacun d'entre eux.

L'expérience montre les limites des approches par noyau pondéré présentées ici (SAS, SFS, SCSS) sur de trop larges ensembles de descripteurs. En effet, l'optimisation conjointe sur l'ensemble des facteurs de pondération se révèle ici suboptimale et inefficace face à ce genre de configurations. Cependant la méthode basée sur le discriminant de Fisher kernelisé (KFDS), bien que légèrement inférieure à l'approche AROM, réussit dans ce cas à améliorer les performances tout en supprimant des descripteurs (on constate une réduction absolue de l'erreur de l'ordre de 1.5% sur les 7.7% d'erreur mesurés sans sélection), contrairement aux approches par noyau pondéré, et surpasse en outre les performances de l'approche R2W2.

7.7.2.4 Classification parole/musique

Nous terminons cette évaluation par une expérience similaire sur le problème central de cette thèse : la classification parole/musique. Ici, comme indiqué dans l'annexe B.3, la base ne contient que des exemples calculés sur des trames de parole ou de musique pure. Les descripteurs employés dans la base sont ceux que nous avons décrits dans le chapitre précédent.

Les résultats obtenus avec un noyau linéaire, représentés sur la figure 7.5(a), n'indiquent pas la présence de descripteurs non-pertinents pénalisant la classification à haute dimension. La pente très faible de l'erreur au-delà de 100 descripteurs constitue néanmoins une preuve de la forte redondance entre ces derniers ; nous avons vu en effet dans la section 6.6 que beaucoup d'entre eux traduisent des propriétés très similaires. On peut ainsi interpréter les descripteurs redondants comme une unique variable fortement pondérée qui, à travers l'amplification exponentielle implicite du noyau RBF gaussien, devient un fort handicap dans le cas non-linéaire. Les résultats, dans ce second cas, illustrés par la figure 7.5(b), confirment cette interprétation puisque que l'on constate que les méthodes les plus efficaces présentent un minimum très marqué autour de $D = 30$. Cette forte redondance explique ainsi les faibles performances du critère de Fisher avec le noyau linéaire,



(a) noyau linéaire (b) noyau RBF gaussien

FIGURE 7.5 – Comparaison des performances entre les différentes méthodes sur la base *parole/musique*.

puisque les descripteurs sont alors évalués indépendamment.

Bien que l'approche R2W2 se révèle la plus efficace, les méthodes proposées basées sur le critère d'alignement (SAS et FAS) présentent des résultats tout à fait comparables, l'écart étant compensé par le gain en temps de calcul. À nouveau on constate, en particulier sur le noyau RBF gaussien, que le critère de Frobenius (SFS) est clairement pénalisé par rapport à l'alignement normalisé (SAS). On observe enfin (figure 7.5 a) que l'approche *forward* sur l'alignement (FAS) est plus efficace à basse dimension que l'approche par noyau pondéré, ce qui confirme sa pertinence pour la sélection d'un ensemble très restreint de descripteurs.

7.7.3 Coût en temps de calcul

Le tableau 7.3 indique les temps de calcul en secondes moyennés sur l'ensemble des itérations, pour chaque méthode impliquée dans les expériences décrites. Toutes les durées que figurent ici proviennent des expériences sur noyau RBF gaussien, à l'exception de la base *Lymphoma*. Les méthodes proposées figurent en italique et la valeur soulignée pour l'approche FAS sur la base *Lymphoma* n'a été calculée que sur une seule itération. Les calculs ont été effectués sur un MacBook Core 2 Duo à 2.16 GHz avec 2 Giga-Octets de mémoire, sur un seul cœur du processeur.

Ces mesures sont bien sûr largement dépendantes de l'implémentation de chaque méthode. Aussi, afin d'homogénéiser au mieux cette comparaison, tous les SVM impliqués dans les diverses méthodes sont basés sur l'outil *SVMlight* de Thorten Joachims [114]. Les temps de calculs sont également directement liés au nombre d'itérations des processus d'optimisation (sauf pour l'approche *forward* FAS).

On constate que le discriminant de Fisher kernelisé (KFDS) constitue un compromis intéressant entre complexité et performances par rapport à l'approche AROM. Les deux méthodes sont construites sur le même principe et leur comportement est comparable sur les différentes expériences détaillées plus haut. Le pas de gradient et la condition d'arrêt sont strictement identiques dans les implémentations des deux algorithmes. On peut donc considérer l'algorithme KFDS, proposé ici, comme une alternative plus rapide à l'approche AROM, en particulier sur de larges ensembles de descripteurs (par exemple KFDS ne prend en moyenne que 12.5 s sur la base *Lymphoma*, tandis qu'AROM nécessite 130 s), au prix d'une légère baisse des performances.

Les expériences nous ont également montré que la méthode à noyau pondéré sur l'alignement (SAS) est globalement comparable en performances à l'approche R2W2, et même meilleure dans certains cas. De plus, les temps de calcul mesurés nous permettent de constater que la méthode proposée est plus rapide, en particulier en présence d'un grand nombre d'exemples d'apprentissage, comme sur la base *parole/musique*, où la méthode R2W2 nécessite 281 s de calcul tandis que la sélection par SAS s'exécute en seulement 54.6 s. Le coût en temps de calcul est à peu près le même

Base	**Lymphoma**	**Ionosphere**	**Spambase**	**Par/mus**
n_{appr}	60	250	500	500
Nb de descripteurs D	4026	34	57	321
Noyau	Linéaire	RBF	RBF	RBF
Fisher	19ms	2ms	3ms	10ms
AROM	130.0	2.6	4.8	12.3
Fisher kernelisé (KFDS)	*12.5*	*0.9*	*3.6*	*5.0*
R2W2	387.9	24.1	113.4	281.3
Frobenius pondéré (SFS)	.	*2.4*	*10.8*	*68.0*
Alignment pondéré (SAS)	*103.8*	*3.5*	*13.0*	*54.6*
Sép classes (SCSS)	*103.7*	*2.7*	*12.6*	*108.5*
Alignment forward (FAS)	*7432*	*3.9*	*28.2*	*1122*

TABLE 7.3 – Comparaison des temps de calcul moyens (en secondes) des méthodes impliquées dans l'évaluation (Par/mus désigne ici la base Parole/musique).

pour toutes les approches par noyau pondéré (SAS, SFS, SCSS), mais le tableau nous confirme le coût très élevé de la méthode *forward* (FAS) lorsqu'elle est appliquée sur un grand nombre de descripteurs (plus de 2 heures de temps d'exécution sur la base *Lymphoma*), ce qui proscrit son usage pour ce genre de situations.

7.8 Commentaires

Nous avons proposé des alternatives fiables aux méthodes de l'état de l'art pour la sélection de descripteurs, adaptées à la discrimination par Machines à Vecteurs de Support. Alors que la plupart des méthodes existantes sont de type enveloppeur (*wrapper*) et se basent donc sur des phases d'apprentissage par SVM pour l'évaluation des descripteurs pertinents, le récent critère d'alignement du noyau nous permet ici d'évaluer directement les performances d'un noyau par rapport à une base d'apprentissage donnée. Ce critère, proposé à l'origine pour la sélection de noyau, est pour l'instant peu exploité dans le domaine de la sélection de descripteurs. La méthode SAS, basée sur une stratégie de recherche par optimisation sur les facteurs de pondération du noyau, se révèle très efficace et comparable en performances aux méthodes les plus récentes tirant parti des noyaux, pour un coût moindre en temps de calcul. De plus, l'expression additive du critère, et l'absence de techniques complexes de programmation mathématique (comme la minimisation par programmation quadratique) en permettent une implémentation rapide et scalable, qui peut s'appliquer sur des ensembles d'apprentissage arbitrairement grands.

L'inclusion de la mesure de dispersion intra-classes a également été étudiée, à travers l'usage du critère de séparabilité de classes. Mais ce critère n'apporte pas les résultats escomptés avec l'approche par noyau pondéré. La tentative de régularisation pour éviter la convergence vers zéro des facteurs de pondération n'est malheureusement pas suffisante ici pour constituer une alternative fiable au critère d'alignement. Toutefois la pertinence de la mesure de dispersion complète (c'est-à-dire incluant les termes intra et inter-classes) est validée en suivant une autre approche basée sur le discriminant de Fisher kernelisé, similaire à la méthode AROM. La méthode KFDS, proposée ici, se révèle comparable en performances à AROM, pour un coût en temps de calcul largement réduit. Son usage reste toutefois limité au noyau linéaire, et son application à un ensemble plus large de noyaux constitue une perspective intéressante pour ces travaux.

Nous avons évoqué l'équivalence, démontrée par Shashua [216], entre la solution d'un SVM et la solution du discriminant linéaire de Fisher sur l'ensemble des vecteurs de support dans l'espace transformé. Ce constat ouvre une perspective très attirante qui consisterait à restreindre aux seuls vecteurs de support obtenus après un apprentissage SVM les méthodes de sélection de descripteurs basées sur la séparabilité de classes kernelisée ou le discriminant de Fisher kernelisé. Cependant, l'objectif étant de réduire le nombre de descripteurs, il nous faut pouvoir garantir le fait que le sous-ensemble des vecteurs de support demeuré inchangé sur un sous-ensemble de descripteurs. Des travaux dans ce sens permettraient ainsi, nous l'espérons, de réduire encore la complexité des

algorithmes proposés tout en éliminant les exemples inutiles ou non-pertinents pour la sélection de descripteurs.

Troisième partie

Approches dynamiques

Introduction de la partie III

L'approche par découpage en trames permet de caractériser le signal audio de manière homogène pour fournir aux SVM les vecteurs de descripteurs nécessaires au processus de classification. Chaque vecteur est donc classé indépendamment de tous les autres, dans un cadre de classification dit *statique*. Pourtant, contrairement à d'autres domaines d'application, les exemples sont fortement corrélés puisque les trames voisines sont liées entre elles par la proximité temporelle et par un contenu commun, du fait que celles-ci se chevauchent de moitié.

De plus, malgré l'effort porté sur la caractérisation du signal audio, certaines classes demeurent proches et peuvent se recouvrir partiellement dans l'espace des descripteurs. Ceci implique nécessairement la présence de trames mal classifiées qui, replacées dans leur contexte temporel, peuvent être aisément corrigées si celles-ci apparaissent isolées ; on parle alors d'erreurs marginales ou accidentelles, ou d'*outliers*.

Cette partie présente plusieurs méthodes destinées à introduire un cadre dynamique (c'est-à-dire prenant en compte la dimension temporelle) dans le processus présenté jusqu'ici.

Nous présentons dans le chapitre 8 plusieurs algorithmes de post-traitement destinés à corriger ces erreurs de classification en tirant parti des corrélations entre trames voisines. Si l'on trouve dans la littérature quelques heuristiques simples sur les labels de classe, ce formalisme est limité par une énorme perte d'information concernant les classes non détectées. Ainsi en exploitant les probabilités a posteriori estimées par les approches SVM multi-classes, on peut déployer des algorithmes plus complexes de post-traitement, allant du simple filtrage à une approche par Modèles de Markov Cachés, que nous adaptons pour cette tâche particulière.

Nous introduisons également dans le chapitre 9 une approche hybride combinant la classification supervisée aux algorithmes de segmentation dite « aveugle », destinée à délimiter dans le signal des régions au contenu acoustique supposé homogène. La segmentation aveugle repose sur la détection de rupture, technique largement explorée dans les domaines de la reconnaissance et du suivi de locuteur. Nous présenterons quelques-unes de ces approches dans ce chapitre, dont certaines exploitent le modèle des SVM à une classe, avatar des machines à noyaux pour la détection de rupture.

Chapitre 8

Algorithmes de post-traitement

Sommaire

8.1 Règles heuristiques 119
8.2 Filtrages 120
8.3 Modèles de Markov Cachés (HMM) 122
8.3.1 Description théorique 122
8.3.2 Estimation de la séquence d'états 123
8.3.2.1 Approche classique par mélange de gaussiennes 124
8.3.2.2 Proposition de post-traitement par HMM 124
8.3.3 Estimation du modèle λ 125
8.4 Hidden Semi-Markov Models 125

8.1 Règles heuristiques

On trouve de nombreux exemples dans la littérature de règles heuristiques destinées à corriger la présence d'éventuelles erreurs de classification sur la séquence des classes estimées $\hat{y}_i \in \{1, \ldots, C\}$ associées à la suite d'exemples $\boldsymbol{x}_i$, où i suit la séquence temporelle des trames. Ces heuristiques ciblent généralement la correction d'erreurs marginales (*outliers*). On peut définir ces dernières de manière informelle comme la présence accidentelle d'un label A dans une séquence longue de labels $B \neq A$.

La règle la plus simple [121][139] consiste donc à simplement remplacer toutes les occurrences de la séquence ABA par la séquence AAA :

- $ABA \rightarrow AAA$

Certains auteurs [260][261] préfèrent au préalable réunir les trames consécutives de même label en segments homogènes pour prendre en compte la durée de ces segments dans la détection d'erreurs marginales. Néanmoins ce procédé revient de fait à induire des règles heuristiques sur des séquences plus longues afin de prendre en compte un voisinage plus large de labels.

Ainsi, dans [52], Chou et Gu définissent un ensemble de règles plus complexes portant sur des séquences de longueurs variables s'échelonnant de 3 à 7 trames, par exemple :

- $AABAA \rightarrow AAAAA$
- $AABBAAA \rightarrow AAAAAAA$

celles-ci sont complétées par des règles empiriques plus fines guidées par la nature des classes mise en jeu. Dans le cadre d'un problème portant sur les quatre classes de parole, musique, chant et bruit, respectivement représentées par les labels « S », « M », « A » et « N » [1], les auteurs définissent par exemple certaines règles destinées à corriger les transitions erronées entre le bruit et les classes de parole et de chant (on note $[A|B]$ un label pouvant prendre indifféremment les valeurs A ou B) :

1. bien que le symbole « A » pour le chant soit contre-intuitif, nous reproduisons ici les notations des auteurs.

1. $N[M|A]SSS \rightarrow NSSSS$
2. $SSS[M|A]N \rightarrow SSSSN$
3. $N[M|S]AAA \rightarrow NAAAA$
4. $AAA[M|S]N \rightarrow AAAAN$
5. $NN[M|A][M|A]SSS \rightarrow NNSSSSS$
6. $SSS[M|A][M|A]NN \rightarrow SSSSSNN$

Les règles 2, 4 et 6 sont les symétriques des règles 1, 3 et 5. On voit que l'auteur choisit ici d'avantager implicitement les classes non bruitées lors de transitions marginales. On peut construire ainsi de nombreuses règles plus complexes encore pour prendre en compte les particularités de chacune des classes par rapport aux autres. Il devient cependant de plus en plus difficile de contrôler l'absence de contradiction entre les règles heuristiques édictées. Il est en outre aisé de construire des séquences indécidables au regard des règles habituelles, en particulier l'alternance entre deux classes, ou certains schémas de transition plus complexes :

- $AAAABABABABBBB$
- $AAAACBACCCC$
- $AAAABBABBAAAA$

Zhang et al. [262] contournent ce problème en reclassifiant les trames de transition marginale (qu'ils définissent comme des sous-séquences $X_1 \neq X_2 \neq \ldots \neq X_n$ dans une séquence $AAX_1 \ldots X_nBB$), en ajoutant la contrainte d'homogénéité de classe (soit $X_1 = X_2 = \ldots = X_n$). Toutefois si la reclassification suit le même processus de classification, cette correction n'a pour seul effet que d'élire la classe majoritaire parmi les labels X_i. Or, si l'on considère que les labels sont erronés, le vote majoritaire prend le risque d'étendre l'erreur sur toutes les trames X_i.

Les règles heuristiques apparaissent de fait comme un pis aller dans un processus où le choix prématuré des labels entraîne une perte importante d'information pour le post-traitement. La seule information sur les classes non choisies pour une trame donnée provient de l'énumération de labels des trames voisines qui, du fait de la discrétisation des valeurs, se heurte aux limitations classiques du vote majoritaire. L'utilisation des probabilités a posteriori comme base du post-traitement permet ainsi de systématiser la prise de décision, en écartant les cas ambigus, et en prenant en compte la vraisemblance des classes non majoritaires. La figure 8.1 donne une illustration de ce phénomène sur un exemple de transition marginale : tandis que les seuls labels (en haut) ne permettent pas de fixer la frontière entre les classes, l'allure des probabilités a posteriori (en bas) nous montre l'évolution homogène croissante de la classe B, qui croise l'évolution décroissante de la classe A, plus bruitée, en particulier par de claires erreurs d'estimation sur la probabilité de la classe C aux trames 5 et 7.

Nous proposons dans la suite de ce chapitre plusieurs algorithmes de post-traitement sur les probabilités a posteriori estimées à partir des approches SVM multi-classes. On notera $\boldsymbol{p}_i = [p_c(i)]_{1 \leq c \leq C}$ le vecteur de probabilités a posteriori associé à la trame d'indice i, et calculé à partir du vecteur de descripteurs $\boldsymbol{x}_i$. La prise de décision se fera selon le principe de maximisation de la vraisemblance :

$$\hat{y}_i = \arg\max_{1 \leq c \leq C} \tilde{p}_c(i),$$

à partir des probabilités corrigées $\tilde{p}_c(i)$ obtenues par post-traitement des probabilités a posteriori $p_c(i)$.

8.2 Filtrages

La méthode de post-traitement la plus simple consiste à lisser les probabilités en appliquant un filtrage moyenneur sur une fenêtre dite glissante couvrant L trames successives. On choisit en général un nombre impair de trames afin de prendre en compte un nombre égal de trames *passées* et *futures*. Le filtrage prend la forme suivante :

$$\tilde{p}_c(i) = \frac{1}{L} \sum_{j=0}^{L-1} p_c(i + j - \frac{L-1}{2}) \qquad \forall c \in [1, \ldots, C].$$

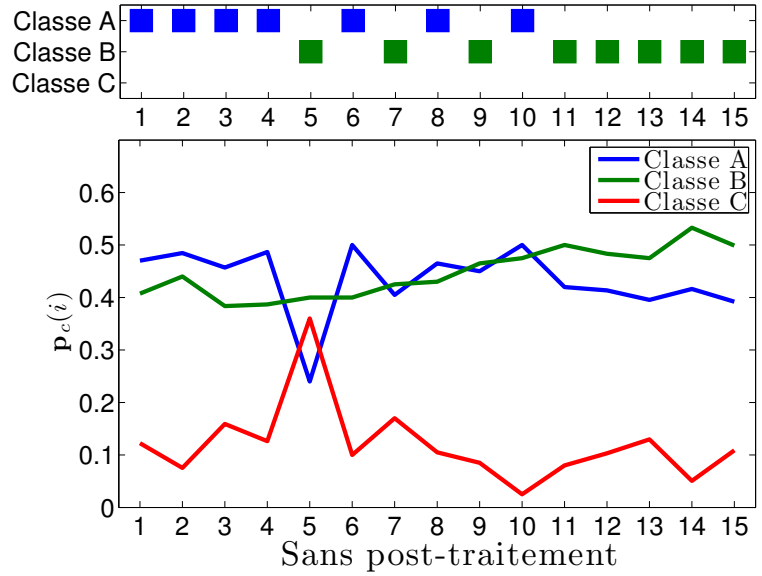


FIGURE 8.1 – Exemple de probabilités a posteriori sur une transition marginale entre les classes A et B. Les labels estimés sans post-traitement sont indiqués dans le cadre supérieur.

La fenêtre couvre donc les trames d'indice $i - \frac{L-1}{2}$ à $i + \frac{L-1}{2}$. Le choix de la longueur L est bien entendu déterminant et sera mené empiriquement par comparaison des performances sur un ensemble de validation. Les figures 8.2(a) et 8.2(b) comparent l'effet du filtre moyen sur l'exemple précédent pour des longueurs de fenêtre respectives de 3 et 7 trames.

On remarque que les probabilités résultantes ne respectent pas la contrainte stochastique $\sum_c \tilde{p}_c(i) = 1$, mais ce constat est de portée mineure puisqu'une normalisation éventuelle des probabilités n'a aucune influence sur la classe de probabilité maximale.

Cependant, le filtre moyenneur, bien qu'ayant l'avantage d'être linéaire et donc propice à une implémentation efficace, garde le défaut d'être relativement sensible aux brusques variations accidentelles. Ainsi on peut observer sur la figure 8.2(a) (concernant un filtre sur 3 trames) que les pics accidentels de la classe C sont amoindris mais restent visibles, bien que n'ayant aucun effet sur la décision finale ; en revanche, on voit que les variations accidentelles de la classe A restent suffisamment présentes pour maintenir une transition marginale.

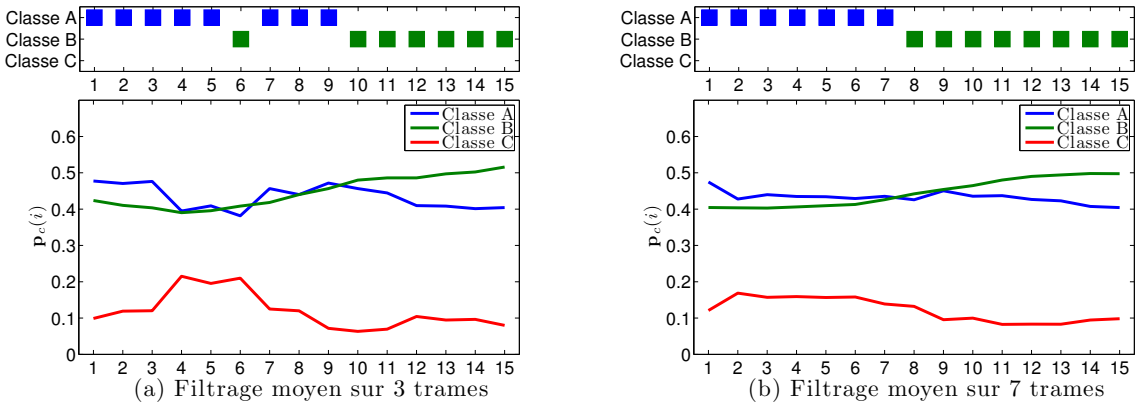
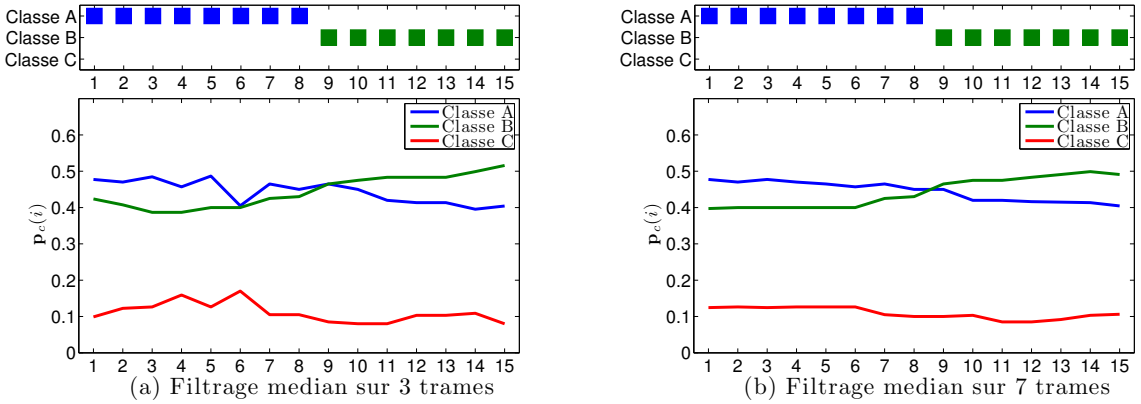


FIGURE 8.2 – Probabilités $\tilde{p}_c(i)$ après filtrage moyen sur des fenêtres glissantes de (a) 3 et (b) 7 trames. Les labels des classes majoritaires sont indiqués dans le cadre supérieur.

Le filtrage médian, généralement attribué à Gustav Fechner [122], est une alternative courante [2] pour corriger les défauts du filtre moyenneur. La médiane d'une distribution de probabilité est définie comme la valeur pour laquelle la fonction de densité de probabilité est égale à $\frac{1}{2}$. Sur un nombre impair d'exemples, on définit la valeur *médiane* comme l'exemple parmi ceux-ci qui sépare les autres en nombres égaux d'exemples inférieurs et supérieurs à ce dernier. Dans le cas d'un nombre pair d'exemples, on utilise généralement la valeur moyenne des deux exemples séparant les autres.

Il résulte que le résultat du filtrage médian n'est pas influencé par les valeurs aberrantes, si celles-ci sont suffisamment minoritaires. On peut d'ailleurs montrer que la valeur médiane est le point minimisant les déviations absolues des exemples. Les figures 8.3(a) et 8.3(b), illustrant l'effet du filtrage médian pour des fenêtres de 3 et 7 trames, montrent que ce dernier estime l'allure non bruitée des courbes de manière plus lisse et pour des fenêtres de taille moindre. On observe par exemple que le pic accidentel en trame 5 sur la classe A, que l'on retrouve après filtrage moyen sur 7 trames et qui implique ainsi une trame d'avance sur l'estimation de la transition, n'a pas cet effet après filtrage médian. En pratique on exploitera un filtre médian sur 9 trames longues (soit une fenêtre d'environ 5 secondes) ; cette envergure a été déterminée empiriquement.



(a) Filtrage median sur 3 trames (b) Filtrage median sur 7 trames

FIGURE 8.3 – Probabilités $\tilde{p}_c(i)$ après filtrage médian sur des fenêtres glissantes de (a) 3 et (b) 7 trames. Les labels des classes majoritaires sont indiqués dans le cadre supérieur.

8.3 Modèles de Markov Cachés (HMM)

Le lissage des probabilités par filtrage médian conduit généralement à une amélioration notable des performances. Néanmoins celui-ci est totalement aveugle au regard des classes considérées et ne peut donc prendre en considération la nature de classes impliquées dans une transition donnée. Pourtant, par exemple dans le contexte d'une émission de radio, la transition passagère de la musique pure vers la voix chantée sur fond musical est beaucoup plus vraisemblable que vers un segment de parole pure.

Les *Modèles de Markov* formalisent la modélisation des transitions entre un ensemble fini d'états. Nous proposons ici, après une brève description théorique de ce modèle stochastique et de ses applications courantes, des solutions de post-traitement des probabilités reposant sur ce dernier.

8.3.1 Description théorique

On modélise un processus par une suite d'états y_i évoluant dans un ensemble fini $S_1, \ldots, S_C$ (dans notre cas, l'état y_i représente la classe acoustique associée à la trame i, on parlera par la suite de *nombre d'instants* $j - i$ pour quantifier la durée qui sépare les états des trames i et j). Un processus Markovien consiste en une modélisation stochastique de la séquence d'états telle que la

2. que l'on trouve en particulier dans le domaine du traitement d'images.

probabilité d'être à un état S_c à l'instant i (soit $y_i = S_c$) ne dépend que de l'état occupé à l'instant précédant ($y_{i-1} = S_d$). De plus cette probabilité est indépendante de l'instant i. On peut donc définir les constantes a_{cd} suivantes :

$$a_{cd} = p(y_i = S_c | y_{i-1} = S_d) \quad 1 \leq c, d \leq C,$$

soumises aux contraintes stochastiques classiques, soit $a_{cd} \geq 0$ et $\sum_{c=1}^{C} a_{cd} = 1$. Si l'on introduit également les probabilités π_c de débuter la séquence sur l'état S_c (avec $\sum_{c=1}^{C} \pi_c = 1$), le modèle (a_{cd}, π_c) décrit entièrement le processus Markovien. On peut ainsi calculer la probabilité d'observer une séquence de n états $y_1, \ldots, y_n$:

$$p(y_1 = S_{c_1}, \ldots, y_n = S_{c_n}) = \pi_{c_1} \cdot \sum_{i=2}^{n} a_{c_i c_{i-1}}.$$

Le *Modèle de Markov Caché* (HMM *Hidden Markov Model*) [190] [3] substitue à la connaissance des états (qui sont désormais non observables, donc *cachés*), celle d'observations O_i dont la génération à chaque instant i est gouvernée par la *probabilité d'observation* b_c qui ne dépend que de l'état $y_i = S_c$. Soit un ensemble fini de M symboles d'observations $v_1, \ldots, v_M$, on a :

$$b_c(k) = p(O = v_k \,|\, y = S_c) \qquad 1 \leq c \leq C,\ 1 \leq k \leq M,$$

un processus ne sera plus caractérisé par sa séquence d'état $y_1, \ldots, y_n$ mais par la séquence d'observations $\boldsymbol{O} = O_1, \ldots, O_n$ avec $O_i = v_{k_i} \forall i$, ce qui suppose l'existence d'une séquence d'états inconnus ayant généré ces observations.

Pour résumer, un HMM est caractérisé par les paramètres suivants :

- Ses C **états** S_c, pour $1 \leq c \leq C$.
- L'alphabet des M **symboles** d'observations v_k, pour $1 \leq k \leq M$.
- Les C^2 **probabilités de transition** a_{cd}, synthétisées par la matrice $\boldsymbol{A} = [a_{cd}]_{cd}$.
- Les $M \times C$ **probabilités d'observation** $b_c(k)$, formant la matrice $\boldsymbol{B} = [b_c(k)]_{ck}$.
- Enfin, les **probabilités d'état initial** π_c, formant le vecteur $\boldsymbol{\pi} = [\pi_c]_c$.

L'ensemble des paramètres $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ constitue le modèle HMM.

8.3.2 Estimation de la séquence d'états

Rabiner [190], reprenant Ferguson, pose les trois problèmes traduisant le champ d'applications concrètes des HMM sur la base d'une séquence d'observations $\boldsymbol{O}$:

1. Connaissant le modèle λ, comment évaluer la probabilité d'observation $p(\boldsymbol{O} \,|\, \lambda)$?
2. Connaissant le modèle λ, comment déterminer la séquence d'états $\boldsymbol{Y} = y_1, \ldots, y_n$ la plus susceptible d'avoir généré les observations $\boldsymbol{O}$?
3. Comment paramétrer le modèle λ maximisant la probabilité $p(\boldsymbol{O} \,|\, \lambda)$?

Notre but étant de déterminer les classes acoustiques associées aux différentes trames, donc la séquence d'états $\boldsymbol{Y}$, c'est bien sûr le problème 2 qui nous intéresse ici.

Ce dernier se résout en employant l'*algorithme de Viterbi* [78][238] qui consiste à rechercher de manière inductive la séquence maximisant la probabilité d'observer les i premières observations. Ainsi si l'on pose :

$$\begin{aligned} \delta_1(c) &= \pi_c \, b_c(O_1) & 1 \leq c \leq C, \\ \delta_i(c) &= \max_{1 \leq d \leq C} [\delta_{i-1}(d) \, a_{cd}] \, b_c(O_i), & \end{aligned} \tag{8.1}$$

alors $\delta_i(c)$ mesure bien la probabilité maximale parmi toutes les séquences d'états d'observer la séquence $O_1, \ldots, O_i$. On calcule ainsi les probabilités $\delta_i(c)$ jusqu'à la trame n. Les valeurs $\psi_i(c)$

3. Si le désormais célèbre tutoriel de Rabiner est la référence la plus courante sur le sujet, les HMM sont originellement proposés par Baum et d'autres auteurs dans une série d'articles datant des années 60 et référencés dans ce même tutoriel.

suivantes sont renseignées conjointement afin de conserver la trace du parcours maximisant la vraisemblance :

$$\begin{aligned} \psi_1(c) &= 0 \qquad\qquad\qquad\qquad \text{par convention} \\ \psi_i(c) &= \underset{1\leq d\leq C}{\arg\max} \left[\delta_{i-1}(d)\, a_{cd}\right]. \end{aligned} \tag{8.2}$$

On déduit ensuite les états $\hat{y}_1, \ldots, \hat{y}_n$ du chemin optimal par induction arrière :

$$\begin{aligned} \hat{y}_n &= \underset{1\leq c\leq C}{\arg\max}\, \delta_n(c) \\ \hat{y}_{i-1} &= \psi_i\left(\hat{y}_i\right). \end{aligned}$$

Un avantage de l'algorithme de Viterbi est qu'il ne dépend que des observations présente et passées. On peut ainsi appliquer ce dernier en temps-réel pour calculer à chaque instant i les valeurs $\delta_i(c)$ en fonction des valeurs passées. Cependant, la séquence d'états optimale étant évaluée par induction arrière, une prise de décision en temps réel impliquerait que les états soient déterminés indépendamment. On peut compenser ce phénomène en introduisant un léger retard de décision, de manière à ne déterminer l'état à un instant i qu'une fois que l'on dispose de suffisamment de valeurs $\delta_i(c)$ d'avance.

Il nous reste à déterminer quelles sont les observations $\boldsymbol{O}$ et comment fixer les paramètres du modèle $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$.

8.3.2.1 Approche classique par mélange de gaussiennes

En règle générale, les HMM sont utilisés en exploitant les vecteurs de descripteurs $\boldsymbol{x}_i$ comme observations du processus. C'est l'application que l'on trouve dans la plupart des articles de la littérature exploitant les HMM pour le problème de classification [125][195][9]. L'exploitation de données réelles multi-dimensionnelles (et non tirées dans un alphabet fini de symbole) suppose cependant une adaptation de l'algorithme puisqu'il n'est pas possible de définir la matrice $\boldsymbol{B}$ des probabilités d'observations.

Celles-ci sont classiquement modélisées par l'estimation des densités de probabilités, notamment par un Modèle de Mélange de Gaussiennes (GMM, *Gaussian Mixture Model*) :

$$b_c(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_{cm}\, \mathcal{N}\left(\boldsymbol{x}, \boldsymbol{\mu}_{cm}, \boldsymbol{\Sigma}_{cm}\right) \quad 1 \leq c \leq C,$$

où $\mathcal{N}\left(\boldsymbol{x}, \boldsymbol{\mu}_{cm}, \boldsymbol{\Sigma}_{cm}\right)$ est la loi normale multidimensionnelle de centre $\boldsymbol{\mu}_{cm}$ et de matrice de covariance $\boldsymbol{\Sigma}_{cm}$, et les α_{cm} sont les coefficients positifs de pondération respectant les contraintes stochastiques $\sum_{m=1}^{M} \alpha_{cm} = 1$. Les paramètres du modèle de mélange sont estimés au moyen de l'algorithme EM [62] (*Expectation Maximization*).

On voit que l'extension des données d'observation au domaine continu ne change en rien l'algorithme de Viterbi (équation 8.1). Le choix du nombre de gaussiennes M est bien sûr important mais ne sera pas traité ici.

8.3.2.2 Proposition de post-traitement par HMM

On trouve de nombreux exemples dans la littérature [92][220], généralement dans le domaine de la reconnaissance de la parole, d'approches hybrides substituant les SVM aux GMM pour l'évaluation des probabilités $b_c(\boldsymbol{x})$. En déterminant les probabilités a posteriori $p(y = S_c \,|\, \boldsymbol{x})$ avec les SVM, on déduit les probabilités d'observation avec la règle de Bayes :

$$p(\boldsymbol{x} \,|\, y = S_c) = \frac{p(y = S_c \,|\, \boldsymbol{x}) \cdot p(\boldsymbol{x})}{p(y = S_c)},$$

où l'on peut supposer $p(\boldsymbol{x})$ uniforme ; les probabilités a priori $p(y = S_c)$ peuvent également être choisies uniformes ou déterminées à partir de la base d'apprentissage.

Nous proposons une autre approche où l'application du post-traitement par HMM est indépendante de la nature du classifieur. On exploite, comme observations, non plus les descripteurs mais les probabilités a posteriori déduites de la classification par SVM ou, dans un cas général, de tout autre processus de classification.

On modélise ainsi, de manière similaire à l'approche classique, la densité de probabilité des observations $b_c(\boldsymbol{x})$ par un mélange de M gaussiennes. Toutefois les probabilités $b_c(\boldsymbol{x})$, respectant la contrainte stochastique de somme unitaire, sont fortement corrélées et situées sur l'hyperplan $x_C = 1 - \sum_{i=1}^{C-1} x_c$, ce qui induit de fortes singularités dans la modélisation gaussienne. On réduit donc le vecteur d'observations d'une dimension en excluant la dernière composante, soit :

$$\boldsymbol{O}_i = [p(y = S_1|\boldsymbol{x}_i), \ldots, p(y = S_{C-1}|\boldsymbol{x}_i)] \quad \forall i.$$

Les probabilités a posteriori étant à priori fortement corrélées aux états du modèle, la modélisation peut se contenter d'un nombre assez limité de gaussiennes, voire d'une seule.

L'apprentissage du modèle gaussien nécessite l'estimation des probabilités a posteriori sur un ensemble de validation disjoint de l'ensemble d'apprentissage afin de prendre en compte le biais du système de classification dans le post-traitement.

8.3.3 Estimation du modèle λ

Contrairement aux applications en reconnaissance de la parole, où un modèle est appris pour chaque mot sur des états inconnus, ici nous avons l'avantage de connaître lors de l'apprentissage les états relatifs aux séquences, par le biais des annotations du corpus. Ainsi, si le corpus est constitué de K fichiers audio, où à chaque fichier d'indice k, de n_k trames, est associée la séquence des classes $y_1^k, \ldots, y_{n_k}^k$, on estime empiriquement à partir du corpus d'apprentissage les probabilités de transition a_{cd} et les probabilités d'état initial π_c (pour $1 \leq c, d \leq C$) :

$$\begin{aligned} a_{cd} &= \frac{1}{\sum_{k=1}^{K} n_k} \sum_{k=1}^{K} \mathrm{Card}\left\{y_i^k = c,\, y_{i-1}^k = d,\, 2 \leq i \leq n_k\right\} \\ \pi_c &= \frac{1}{K} \sum_{k=1}^{K} \mathrm{Card}\left\{y_1^k = c\right\}. \end{aligned}$$

On remarque que ces grandeurs ne dépendent que des annotations et que le processus de classification n'intervient aucunement. On peut donc utiliser la totalité du corpus d'apprentissage, pour accroître la confiance, sans introduire de biais dans l'estimation du modèle HMM.

8.4 Hidden Semi-Markov Models

Les HMM ont montré leur efficacité pour de nombreuses applications. Néanmoins le modèle implique que la probabilité de rester sur un état c durant d instants successifs, suit une distribution géométrique : $p(d) = a_{cc}^{d-1}(1 - a_{cc})$. Cette contrainte ne traduit pas nécessairement le modèle d'une application donnée. Dans notre cas, par exemple, sur des archives radiophoniques, les segments de parole ne suivent en aucun cas une loi géométrique et peuvent d'ailleurs atteindre des longueurs très conséquentes. La figure 8.4 illustre ce constat en représentant la distribution des nombres de trames consécutives pour les classes de musique (à gauche) et de parole (à droite) sur le corpus ESTER (que nous présenterons dans la section 10.1.1).

On constate que si les durées des segments de musique suivent approximativement une distribution géométrique, on trouve au contraire de nombreux segments longs de parole, qui rendent la modélisation géométrique inadéquate.

Les *modèles semi-markoviens cachés* [165] (HSMM, *Hidden Semi-Markov Model*), également appelés *modèles de segments* [174], sont généralement attribués à Ferguson qui introduit dès 81 la modélisation des durées d'états [75]. Ils étendent le modèle HMM pour relâcher la contrainte précédente, en associant à chaque état y_i, non plus une observation mais une séquence d'observations. Ainsi on ajoute au système une variable d'état supplémentaire ℓ_i associant à chaque état y_i le nombre d'observations $O_{i,1}, \ldots, O_{i,\ell_i}$ générées par le système à cet état ; on parle ainsi de

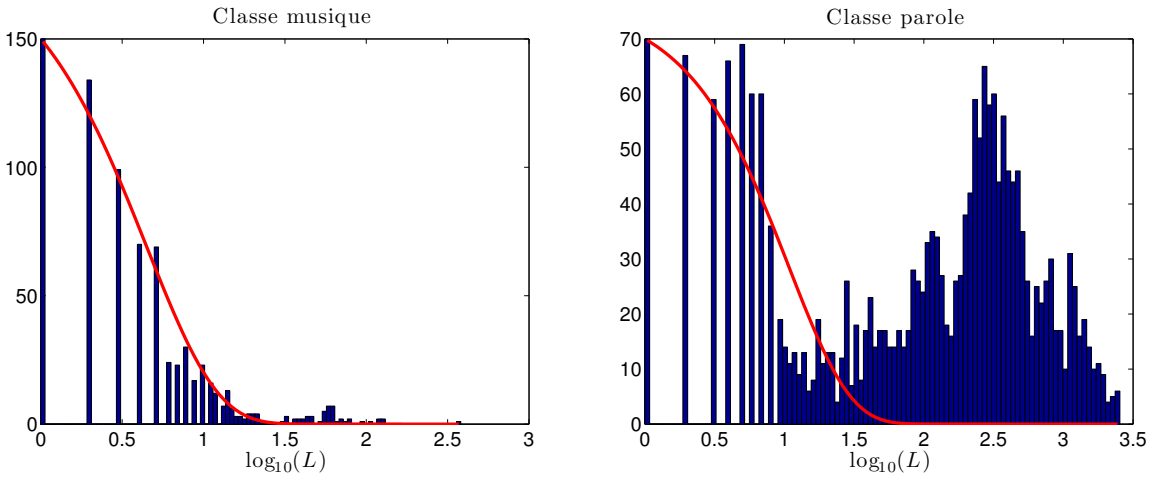


FIGURE 8.4 – Distribution des nombres de trames par segments (L) pour les classes de musique (à gauche) et de parole (à droite) sur le corpus ESTER. Les abscisses suivent une échelle logarithmique.

segments homogènes modélisés par la séquence des états. De plus, le système est caractérisé par deux nouvelles probabilités : $p(\ell = L \,|\, y = S_c)$ et $p(O_1, \ldots, O_\ell \,|\, y = S_c, \ell = L)$ gouvernant respectivement la distribution des longueurs de segments pour un état donné S_c et la probabilité d'observer la séquence d'observations $O_1, \ldots, O_L$ à l'état S_c si celle-ci est de longueur $\ell = L$.

L'ajout du paramètre de durée des segments complexifie considérablement le modèle, en premier lieu parce que la variabilité de la longueur des segments introduit un nouveau paramètre dans l'espace de recherche, qui implique un facteur multiplicatif D sur la complexité [254][256] (où D est la longueur maximale d'un segment) pour l'algorithme Forward-Backward [4], lequel n'est pas présenté ici mais apporte une réponse aux deux autres questions de la section 8.3.2. De plus le champ des probabilités d'observation se trouve largement élargi puisqu'il couvre la modélisation de séquences d'observations de longueurs variables. Ostendorf et al. [174] proposent un large panorama de modèles dynamiques possibles traduisant les caractéristiques de séquences d'observation. Toutefois, nous nous limiterons au cas d'observations indépendantes, soit :

$$p(O_1, \ldots, O_\ell \,|\, y_i = S_c, \ell = L) = \prod_{l=1}^{L} p(O_l \,|\, y_i = S_c),$$

où l'on exploitera les probabilités d'observation $p(O_l \,|\, y_i = S_c)$ définies pour le modèle HMM. Il est important de noter que l'hypothèse d'indépendance des observations est loin d'être arbitraire si l'on considère le fait que les observations proviennent du processus de classification par SVM, où chaque vecteur descripteur de trame est traité indépendamment des autres.

De plus, plutôt que de considérer que chaque état produit plusieurs observations, on introduit une variable d'état f_i qui décrit le nombre d'instants successifs passés sur l'état actuel à un instant i, ce qui nous permet de revenir au formalisme des HMM avec seulement une légère modification dans l'algorithme de Viterbi. On parlera par la suite de *segments* pour désigner une suite d'états de même classe, distincte des classes des segments adjacents.

On adapte ce dernier en court-circuitant les probabilités de transition a_{cc} d'un état à lui-même, desquelles résulte, comme nous l'avons expliqué, la distribution géométrique de la probabilité de rester dans un état c. Ainsi la probabilité de transition de l'état c à lui-même dépend, dans le cas des HSMM, de la probabilité $p(\ell = L \,|\, y = S_c)$ des durées de segments générés par l'état S_c, introduite précédemment. On note $e_c(L)$ la probabilité qu'un segment soit de longueur supérieure ou égale à L, appelée *probabilité de stagnation* :

$$e_c(L) = p(\ell \geq L \,|\, y = S_c) = \sum_{\ell=L}^{\infty} p(\ell = L \,|\, y = S_c),$$

4. également désigné sous le nom d'algorithme de Baum-Welch.

en supposant que l'on se trouve sur l'état c depuis $f_i = F$ instants, la probabilité d'y rester un instant de plus est donc égale à :

$$\begin{aligned} p(y_{i+1} = S_c \,|\, y_i = S_c,\, f_i = F) &= p(\ell \geq F+1 \,|\, \ell \geq F,\, y = S_c) && (8.3)\\ &= \frac{p(\ell \geq F+1,\, \ell \geq F,\, y = S_c)}{p(\ell \geq F,\, y = S_c)} && (8.4)\\ &= \frac{e_c(F+1)}{e_c(F)}. && (8.5) \end{aligned}$$

ce qui revient donc à substituer à la constante a_{cc} la valeur $\frac{e_c(F+1)}{e_c(F)}$, dépendant du nombre d'instants F déjà passés sur l'état actuel, à l'instant i. Si l'on reprend l'expression de $\delta_i(c)$ (équation 8.1), on propose ainsi la règle d'induction suivante :

$$\delta_i(c) = \max_{1 \leq d \leq C} [\delta_{i-1}(d)\, \tilde{a}_{cd}]\, b_c(O_i),$$

avec :

$$\tilde{a}_{cd} = \begin{cases} a_{cd} & \text{si } c \neq d \\ \frac{e_c(f_i(c)+1)}{e_c(f_i(c))} & \text{si } c = d \end{cases}$$

où l'on introduit la variable $f_i(c)$, gardant la trace du nombre d'instants depuis le dernier changement d'état, également renseignée par induction durant l'algorithme :

$$\begin{aligned} f_1(c) &= 1 \\ f_i(c) &= \begin{cases} 1 & \text{si } \psi_i(c) \neq c \\ f_{i-1}(c) + 1 & \text{sinon.} \end{cases} \end{aligned}$$

Les probabilités de stagnation $e_c(\ell)$, que nous avons introduites dans le modèle HSMM, sont également estimées empiriquement à partir des annotations du corpus d'apprentissage. On regroupe dans un premier temps la séquence des classes par trames $y_1^k, \ldots, y_{n_k}^k$ du fichier k en une séquence de S_k segments de longueur $l_1^k, \ldots, l_{S_k}^k$ et de classe homogène $c_1^k, \ldots, c_{S_k}^k$ (avec $c_i^k \neq c_{i+1}^k$). On estime à partir des segments, les probabilités de stagnation :

$$e_c(\ell) = \frac{1}{l_{T,c}} \sum_{k=1}^{K} \text{Card}\left\{c_s^k = c, l_s^k \geq \ell\right\}_{1 \leq s \leq S_k},$$

où $l_{T,c}$ est un facteur de normalisation égal au nombre total de sous-séquences de segments partant de débuts de segments de classe c, sur l'ensemble du corpus :

$$l_{T,c} = \sum_{k=1}^{K} \sum_{s | c_s^k = c} \sum_{l=1}^{l_s^k} l.$$

Ce dernier garantit le respect de la contrainte $e_c(1) = 1\ \forall c$.

Même sur un corpus conséquent, le nombre de segments longs n'est jamais très élevé, si bien que généralement les grandeurs $e_c(\ell)$ ont une allure en escalier pour de grandes valeurs de ℓ. On résout ce problème par un lissage classique ou par une simple interpolation affine entre les points de changement. On peut également chercher à modéliser statistiquement $e_c(\ell)$, mais un modèle trop pauvre peut se révéler équivalent au formalisme des HMM (c'est-à-dire à une modélisation par une loi exponentielle).

On a donc proposé un algorithme simple, sans coût additionnel prohibitif par rapport aux HMM, pour prendre en compte explicitement les durées des segments de classe homogène dans l'algorithme de Viterbi. Cependant, les résultats sont très décevants en pratique puisque l'application du post-traitement par HSMM n'apporte aucun gain en performances notable par rapport au modèle HMM. Ce résultat semble montrer que l'inadéquation théorique des HMM aux segments longs ne se traduit pas par un réel handicap en pratique. Nous n'inclurons donc pas les HSMM dans le chapitre d'évaluation, puisque les résultats sont globalement les mêmes qu'avec les HMM.

Chapitre 9

Approche hybride par segmentation aveugle

Sommaire

9.1 Principe 129
9.2 Détection de rupture 130
9.3 Méthodes classiques 131
9.3.1 Un exemple d'approche métrique : divergence de Kullback Leibler . . . 131
9.3.2 Rapport de vraisemblance généralisé (GLR) 132
9.3.3 Critère d'Information Bayésienne (BIC) 133
9.4 Mesures probabilistes dans les espaces RKHS 134
9.5 SVM à une classe 135
9.5.1 Principe des SVM à une classe 136
9.5.2 Rapport de vraisemblance par SVM1C (LLR) 137
9.5.3 Kernel Change Detection (KCD) 138
9.5.4 Mise à jour incrémentale des SVM à une classe 140
9.6 Recherche de maxima pour la détection de rupture 140

9.1 Principe

Il est possible d'introduire la dimension temporelle dans le processus en combinant l'approche statique de la classification à une méthode dynamique de découpage du flux audio en segments dont le contenu est acoustiquement homogène. Cette notion de segment rejoint celle introduite dans le chapitre précédent dans la présentation des modèles de segments associés aux HSMM (section 8.4).

Ce découpage, appelé segmentation, se fait non par l'analyse directe du contenu des trames, mais par la recherche des points de changement délimitant les segments successifs. Cette méthodologie est généralement employée dans le domaine du suivi ou de l'identification de locuteurs [128][6][34][74][227][10], de manière à mettre en évidence les tours de paroles ne contenant qu'un locuteur à la fois. La reconnaissance de locuteur est un problème particulier qui dépasse le cadre de cette thèse, parce qu'il implique un nombre très important de classes (de locuteurs), dont il est en général impossible de connaître la totalité, comme par exemple sur des bulletins d'informations radiophoniques, où potentiellement n'importe qui peut être présent dans une interview. Cette contrainte implique donc l'usage de techniques de segmentation dite aveugles (ou non-supervisées) qui ne reposent pas sur l'apprentissage préalable des modèles des classes susceptibles d'être observées. Nous en expliquerons le principe dans la section 9.2.

Le processus de segmentation aveugle nous permet, en reprenant les notations de la section 8.4, de répartir la séquence des trames $i = 1, \ldots, n_k$ du fichier k, en S_k segments successifs dont les indices de trames initiaux sont respectivement $i_1^k, \ldots, i_{S_k}^k$ (avec bien sûr $i_1^k = 1$ et par convention

$i_{S_k+1} = n_k + 1$). Contrairement au post-traitement par HSMM, nous ne connaissons pas les classes $c_1^k, \ldots, c_{S_k}^k$ associées aux segments mais nous supposons ces derniers homogènes par rapport aux classes acoustiques considérées. On remarque que l'on a également substitué aux longueurs, la donnée équivalente des indices de début de trames qui convient mieux dans le cadre présent puisque la segmentation aveugle consiste en la détermination des indices i_s^k.

L'approche hybride peut alors se faire de deux manières différentes :

- Comme **pré-traitement** : c'est la méthode que l'on trouve généralement dans la littérature. Chaque segment est classifié dans son ensemble par une unique prise de décision. Ainsi la conjonction des différentes trames du segment permet d'accroître la confiance dans la décision. Les intégrations temporelles des descripteurs sont ainsi plus fiables puisqu'elles portent sur une période plus large.
- Comme **post-traitement** : c'est l'approche que nous suivons dans ce document. L'information de segmentation intervient ici après la classification, sur la donnée des probabilités a posteriori, comme pour les approches de post-traitement présentées dans le chapitre précédent. On associe à chaque segment s la classe $\hat{c}_s$ maximisant la somme des probabilités sur les trames du segment. Soit :

$$\hat{c}_s = \underset{1 \leq c \leq C}{\arg\max} \sum_{i=i_s^k}^{i_{s+1}^k - 1} p_c(i),$$

où l'on rappelle que $p_c(i)$ est la probabilité a posteriori évaluée sur la trame i pour la classe c, et i_s^k est l'indice de la première trame du segment s. Comme pour l'approche par pré-traitement, la conjonction des décisions sur les trames d'un segment renforce la fiabilité de la décision. Dans le cas d'une erreur, elle peut cependant avoir l'effet contraire et contaminer l'ensemble des trames d'un segment.

Les frontières entre segments étant à priori caractérisées par une transition plus ou moins brusque d'un modèle acoustique à un autre, on détermine celles-ci par des algorithmes de détection de rupture, dont nous présentons le principe dans la section suivante.

9.2 Détection de rupture

Le mécanisme de la détection de rupture est généralement assez simple. On suppose que l'on observe le signal contenu dans une fenêtre d'analyse W de n échantillons représentés par les vecteurs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, et que l'on souhaite examiner l'hypothèse d'une rupture à l'indice t . On exploite pour cela les signaux des sous-fenêtres antérieure et postérieure, respectivement $W_1 = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}]$ et $W_2 = [\boldsymbol{x}_t, \ldots, \boldsymbol{x}_n]$, de tailles respectives $n_1 = t$ et $n_2 = n - t$. La figure 9.1 résume la configuration des fenêtres d'analyse mises en jeu.

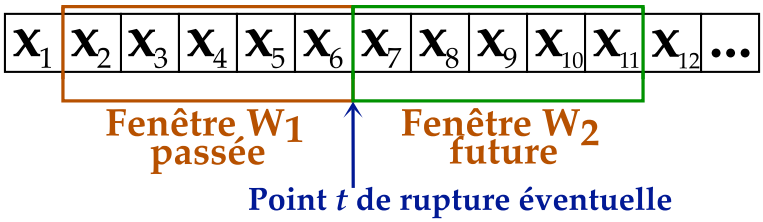


FIGURE 9.1 – Exemple de configuration pour les fenêtres d'analyse passée et future, respectivement W_1 et W_2 (ici de mêmes tailles $n_1 = n_2 = 5$), par rapport à l'instant d'hypothèse t.

Les algorithmes proposés pour la détection de rupture se basent sur une modélisation stochastique des signaux des fenêtres d'analyse. Ainsi on suppose que les exemples de la fenêtre d'analyse W sont les réalisations d'une variable aléatoire gouvernée par la loi de probabilité $P_0(\boldsymbol{X})$; de même, les lois de probabilité $P_1(\boldsymbol{X})$ et $P_2(\boldsymbol{X})$ gouvernent respectivement la réalisation des exemples des fenêtres W_1 et W_2.

Une taxonomie communément admise dans la littérature [49][120][250] distingue les quatre modalités suivantes pour la détection de rupture :

- **Approche énergétique** : en se basant sur la supposition que les tours de parole sont généralement séparés par de courtes périodes de silence entre les locuteurs, certains algorithmes rudimentaires ne basent la segmentation que sur le seuillage d'un critère d'énergie court terme [249]. Cette approche, déjà contestée dans le domaine de la parole pure, n'est pas pertinente sur un signal audio quelconque, où la présence de silences intermédiaires est plutôt l'exception que la règle.
- **Approche par modèles** : consiste à modéliser chacune des classes mises en jeu afin de classifier le contenu des fenêtres W_1 et W_2 sur le critère du maximum de vraisemblance [17]. En plus de ne pas être aveugle, cette approche est le processus exactement inverse de ce que nous recherchons ici puisque la classification est utilisée comme outil de segmentation.
- **Approche métrique** : la détermination des frontières entre segments est basée sur la recherche des maxima locaux d'une métrique qui évalue la similarité entre les modèles des fenêtres W_1 et W_2 [217][89]. C'est l'approche que nous suivrons ici.
- **Approche sur critère d'information** : cette approche est similaire à la précédente mais substitue aux critères métriques, nécessitant un seuil de décision, une mesure d'information appelée Critère d'Information Bayésienne (BIC, pour *Bayesian Information Criterion*), pour laquelle le seuil est implicite [49][45][61], comme nous le verrons par la suite. Ce critère implique en outre un autre paradigme de détection. Tandis que l'approche métrique évalue une mesure de distance entre les deux fenêtres W_1 et W_2, cette approche compare les hypothèses d'absence et de présence de rupture. Nous examinons cette approche plus en détail dans la section suivante.

On trouve également dans la littérature plusieurs propositions d'algorithmes hybrides combinant les avantages complémentaires de deux approches ; par exemple la proposition de Kemp et al. [120] qui consiste en une approche par modèles basée sur un premier traitement par approche métrique, ou encore l'algorithme SEQDAC de Cheng et Wang [50]. De nombreux algorithmes hybrides [263][257][61][51] combinent le critère BIC à d'autres métriques comme la statistique T^2 (test basé sur les statistiques de premier et de second ordre de deux modèles probabilistes).

Nous présentons dans la section suivante quelques-uns des algorithmes classiques de détection de rupture, notamment le critère BIC, très largement exploité, afin de montrer leurs implications sur la stratégie de recherche de points de rupture multiples dans un signal. Nous introduirons par la suite certains critères plus récents et plus élaborés tirant parti des résultats sur les espaces à noyaux reproduisants et les machines à noyaux dans les sections 9.4 et 9.5.

9.3 Méthodes classiques

9.3.1 Un exemple d'approche métrique : divergence de Kullback Leibler

Le principe de l'approche métrique consiste à déterminer les points de rupture par une mesure de distance entre les fenêtres voisines W_1 et W_2, soumise à un seuil de détection adéquat. La stratégie de recherche appliquée est généralement celle des *fenêtres glissantes* [257][153][128][34], qui consiste à échantilloner la mesure de distance entre les fenêtres W_1 et W_2 de tailles fixes et égales à M, en glissant itérativement ces dernières d'un pas fixe du début à la fin de la séquence de trames. La figure 9.2 illustre le principe de la recherche par fenêtres glissantes, que nous suivons dans notre cadre expérimental.

Le choix de la métrique est un problème ouvert, et les sections suivantes apporteront diverses propositions pour ce point. Siegler et al. [217] proposent par exemple l'usage de la divergence de Kullback-Leibler, définie par :

$$KL(P \,|\, Q) = E_P \left[\log P(\boldsymbol{X}) - \log Q(\boldsymbol{X})\right],$$

où E_P est l'espérance par rapport à la probabilité $P(\boldsymbol{X})$; soit, sur les densités de probabilités p et

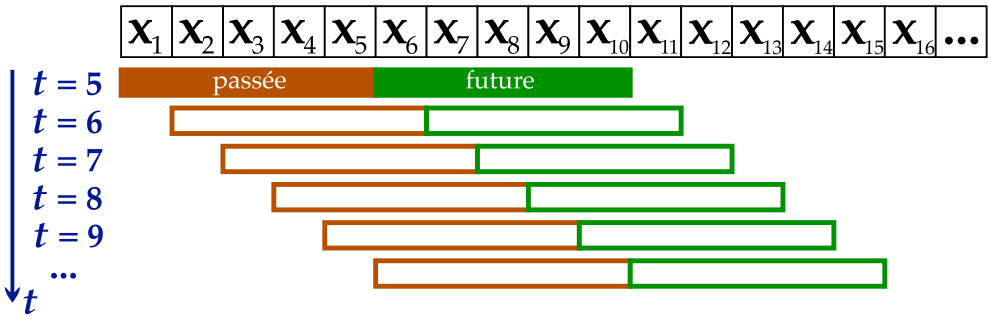


FIGURE 9.2 – Illustration de la recherche de rupture par fenêtres adjacentes glissantes.

q :

$$KL(P\,|\,Q) = \int_{\boldsymbol{X}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} d\boldsymbol{x}.$$

Le terme de divergence met en lumière le caractère non symétrique de cette « semi-métrique ». On emploie donc en général la variante symétrisée de la mesure de Kullback-Leibler :

$$\begin{aligned} KL2(P,Q) &= KL(P\,|\,Q) + KL(Q\,|\,P) \\ &= \int_{\boldsymbol{X}} [p(\boldsymbol{x}) - q(\boldsymbol{x})] \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} d\boldsymbol{x}. \end{aligned}$$

Dans le cas de la détection de rupture, les probabilités P et Q sont estimées par des modèles gaussiens $(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ et $(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ appris sur les exemples des fenêtres W_1 et W_2. On peut ainsi exprimer analytiquement la métrique $KL2$ dans le cas de distributions gaussiennes :

$$KL2(t) = \frac{1}{2} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right) (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2} \operatorname{tr} \left[\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1 - 2\boldsymbol{I}\right].$$

On obtient donc un signal $KL2(M+1), \ldots, KL2(n-M)$ de distances entre fenêtres, sur lequel on appliquera l'heuristique de recherche de pics maxima présentée dans la section 9.6.

9.3.2 Rapport de vraisemblance généralisé (GLR)

Le critère GLR (*Generalized Likelihood Ratio*), introduit par Gish et Schmidt pour la segmentation aveugle [89], repose sur le test d'hypothèse introduit pour les approches sur critère d'information, à savoir la comparaison entre les deux hypothèses suivantes :

- H_0 : il n'y a pas de rupture. Tous les exemples de la fenêtre W sont donc générés par l'unique modèle $P_0(\boldsymbol{X})$.
- H_1 : il y a une rupture à l'instant t. Les exemples des fenêtres W_1 et W_2 sont donc respectivement générés par les modèles $P_1(\boldsymbol{X})$ et $P_2(\boldsymbol{X})$.

Le critère en question se base donc sur le rapport de vraisemblance entre les deux hypothèses, où l'on considère les échantillons i.i.d. :

$$r(t) = \frac{\prod_{i=1}^{n} P_0(\boldsymbol{x}_i)}{\prod_{i=1}^{t-1} P_1(\boldsymbol{x}_i) \prod_{i=t}^{n} P_2(\boldsymbol{x}_i)}. \tag{9.1}$$

On estime les probabilités par un modèle gaussien évalué sur les exemples des fenêtres considérées. Ainsi, chaque probabilité P_h est modélisée par une loi gaussienne $\mathcal{N}(\boldsymbol{\mu}_h; \boldsymbol{\Sigma}_h)$ de centre $\boldsymbol{\mu}_h$ et de matrice de covariance $\boldsymbol{\Sigma}_h$. En définitive on exploite généralement le logarithme du critère précédent calculé sur les modèles gaussiens estimés [61] :

$$R(t) = -\frac{n}{2} \log |\boldsymbol{\Sigma}_0| + \frac{n_1}{2} \log |\boldsymbol{\Sigma}_1| + \frac{n_2}{2} \log |\boldsymbol{\Sigma}_2|. \tag{9.2}$$

On constate que si l'on utilise un modèle de même complexité pour les trois probabilités P_0, P_1 et P_2, ces deux dernières seront nécessairement plus précises et le rapport de vraisemblance

logarithmique $R(t)$ est donc systématiquement négatif, ce qui implique la nécessité de déterminer un seuil de décision empirique entre les hypothèses H_0 et H_1.

Ajmera et al. proposent [10], pour s'affranchir de la nécessité d'un tel seuil, de modéliser la probabilité P_0 par un mélange de deux gaussiennes, tout en conservant une unique gaussienne pour les modèles P_1 et P_2. Ainsi les deux hypothèses sont de même complexité, ce qui permet de montrer que le seuil de décision se situe naturellement à 0 (pour le rapport logarithmique). Néanmoins, le Critère d'Information Bayésienne est une alternative plus généralement suivie dans la littérature.

9.3.3 Critère d'Information Bayésienne (BIC)

En 1972, Akaike [11] est le premier à proposer un critère d'information théorique, appelé AIC (*Akaike Information Criterion*) permettant de prendre en compte la complexité du modèle dans les tests d'hypothèses impliquant un rapport de vraisemblance. Il ajoute à la mesure de vraisemblance une pénalité k mesurant le nombre de paramètres libres du modèle, soit pour un modèle donné de loi P :

$$AIC(P) = \log P(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) - k.$$

Dans le cas d'un modèle gaussien, on dénombre le nombre suivant de paramètres libres pour la moyenne $\boldsymbol{\mu}$ et la matrice de covariance $\boldsymbol{\Sigma}$:

$$k = d + \frac{d(d+1)}{2}.$$

Par la suite, Schwarz propose [215] le Critère d'Information Bayésienne qui pénalise plus fortement les modèles construits sur une large collection d'exemples en ajoutant un facteur multiplicatif $\log n$ au facteur de pénalité ; on y joint généralement un facteur multiplicatif λ de manière à contrôler le compromis entre la vraisemblance et la complexité du modèle, bien que celui-ci soit absent de la proposition originale de Schwartz. Ainsi le critère devient :

$$BIC(P) = \log P(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) - \lambda \frac{k}{2} \log n.$$

Schwarz justifie ce critère par le fait qu'il est asymptotiquement optimal pour le choix de modèle, tandis que le critère AIC à tendance à choisir le modèle le plus complexe lorsque $n \to \infty$. Rissanen montre par ailleurs [198] que, pour $\lambda = 1$, le critère BIC est égal à la MDL (*Minimum Description Length*), grandeur en Théorie de l'Information décrivant le nombre de bits minimum nécessaires pour coder le modèle, ce qui rejoint la notion de complexité du modèle.

Le critère fut plus tard introduit dans le contexte de la détection de rupture [49]. Le test d'hypothèse consiste à évaluer la différence des valeurs BIC entre l'hypothèse de rupture et de non-rupture, sur des modèles gaussiens ; soit :

$$\Delta BIC(t) = R(t) + \frac{1}{2}\lambda \left(d + \frac{d(d+1)}{2} \right) \log n. \tag{9.3}$$

où, $R(t)$ est le critère GLR introduit dans l'équation 9.2. Les auteurs revendiquent la supériorité du critère ΔBIC sur les approches métriques, du fait que celui-ci prend en compte la complexité des modèles et permet ainsi d'appliquer un seuil naturel de décision à 0. Cependant, malgré la valeur théorique $\lambda = 1$, le facteur λ constitue en pratique un paramètre supplémentaire à déterminer [227][183], qui se substitue au seuil de décision.

La taille de la fenêtre d'analyse est un point essentiel dans le comportement du critère BIC et doit fixer un compromis entre les deux contraintes suivantes :

- Une fenêtre trop large est susceptible de contenir plus d'un point de rupture, ce qui affecte directement le taux d'omissions (MD, *Missed Detections*).
- Une fenêtre trop étroite comporte peu d'exemples, ce qui affaiblit l'estimation des modèles.

Cette seconde contrainte constitue l'une des limites du critère BIC. En effet, ce dernier est choisi pour son comportement asymptotique optimal, mais le facteur de pénalité propre au critère ($k \log n$) favorise les modèles les plus simples [263] lorsque le modèle est estimé sur un nombre restreint d'exemples. De plus, le fait qu'il soit exclusivement basé sur des statistiques du second ordre (les matrices de covariances) accroît cette faiblesse, puisque l'estimation du modèle se trouve elle-même pénalisée [227][44]. On trouve ainsi plusieurs propositions d'approches hybrides appliquant une première étape de détection moins fiable mais plus simple, dont le seuil est fixé de manière à minimiser le taux d'omissions, basée par exemple sur le critère GLR [108], la statistique T^2 [264] [263], ou une approche métrique [61][50].

Un autre inconvénient majeur du critère BIC est sa complexité. En effet, le calcul des inverses des matrices de covariances est une opération lourde (de l'ordre de $O(\frac{n^3}{6})$ en utilisant la décomposition de Cholesky). On peut cependant réduire ce coût par des heuristiques, comme la mise à jour incrémentale des matrices de covariance [45][218]. Nous nous contenterons de suivre l'exemple d'Ajmera et al. [10] qui n'exploitent que des matrices de covariance diagonales.

Il est important de préciser que l'algorithme BIC s'accompagne à l'origine [49] d'une stratégie de recherche par maxima locaux par élargissement itératif des fenêtres d'analyse, qui permet d'affiner progressivement les modèles de distributions. Cependant, s'il est théoriquement plus pertinent, cet algorithme n'est pas adapté à un traitement en ligne (ou *online*, c'est-à-dire avec un retard de réponse borné) des données ; aussi nous préférons n'employer que la méthode des fenêtres glissantes, ce qui nous permet en outre de comparer les différents critères de distance sur les mêmes bases méthodologiques.

9.4 Mesures probabilistes dans les espaces RKHS

Les critères présentés dans la section précédente reposent sur des mesures probabilistes basées sur des modèles gaussiens. Nous avons présenté comme exemple la divergence de Kullback-Leibler symétrisée (KL2), mais il existe pléthore de mesures probabilistes alternatives, parmi lesquelles nous nous intéresserons également à la distance de Bhattacharrya [27], définie de la manière suivante :

$$d_B(p_1, p_2) = -\log\left(\int_{\boldsymbol{X}} \sqrt{p_1(\boldsymbol{x})p_2(\boldsymbol{x})}d\boldsymbol{x}\right).$$

Il est également possible d'exprimer analytiquement cette dernière dans le cas gaussien :

$$d_B(p_1, p_2) = \frac{1}{8}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)^T \left[\frac{1}{2}\left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\right)\right]^{-1} \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right) + \frac{1}{2}\log\frac{\left|\frac{1}{2}\left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\right)\right|}{\sqrt{\left|\boldsymbol{\Sigma}_1\right|\left|\boldsymbol{\Sigma}_2\right|}}.$$

Le modèle gaussien est très largement exploité pour ses excellentes propriétés de régularités, sa concision et sa simplicité théorique qui permet généralement d'exprimer analytiquement les mesures probabilistes. Il reste néanmoins très restrictif et peut se révéler inadéquat en présence de données réelles.

Zhou et Chellappa tirent parti [265] des résultats de la théorie des Espaces à Noyaux Reproduisants (RKHS, pour *Reproducing Kernel Hilbert Spaces*), brièvement introduite dans la section 3.5.2, pour étendre le champ des modèles employés sur les fenêtres d'analyse. De la même manière que l'introduction des noyaux permettait de modéliser implicitement des surfaces de décisions plus complexes dans le processus discriminatif des SVM, il est possible d'exploiter le fameux *kernel trick* sur les mesures probabilistes classiques.

On rappelle que les noyaux respectant la condition de Mercer permettent de construire un espace fonctionnel de Hilbert par la transformation suivante (équation 3.23) :

$$\begin{array}{rcl} \Phi : \mathcal{X} & \rightarrow & \mathbb{R}^{\mathcal{X}} \\ \boldsymbol{x} & \mapsto & k(.\,, \boldsymbol{x}) \end{array}$$

On montre par ailleurs qu'il est possible de définir un produit scalaire sur cet espace qui reproduit le comportement de la fonction noyau :

$$\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle = \langle k(\cdot, \boldsymbol{x}), k(\cdot, \boldsymbol{y}) \rangle = k(\boldsymbol{x}, \boldsymbol{y}).$$

Cette propriété particulière justifie l'appellation d'Espace de Hilbert à Noyaux Reproduisants (RKHS), et montre que l'action d'un noyau de Mercer est équivalente au calcul d'un produit scalaire dans l'espace RKHS, la transformation de l'espace d'origine à ce dernier étant gouvernée par la fonction implicite Φ dont l'expression analytique n'est pas nécessaire. Ce dernier constat constitue ce qu'on appelle le *kernel trick*. On peut en outre montrer que l'espace RKHS est de dimension largement supérieure (voire infinie) à celle de l'espace d'origine, ce qui permet de renforcer la validité de l'hypothèse de gaussianité, comme le montrent les auteurs [265].

On montre que les moyennes et les covariances dans l'espace reproduisant, estimées à partir des exemples des fenêtres W_1 et W_2, prennent les expressions suivantes :

$$\begin{aligned} \hat{\boldsymbol{\mu}}_i &= \boldsymbol{\Phi}_i \, \mathbf{s} \\ \hat{\boldsymbol{\Sigma}}_i &= \boldsymbol{\Phi}_i \, \mathbf{J}\mathbf{J}^T \boldsymbol{\Phi}_i^T, \end{aligned}$$

où l'on a introduit, pour obtenir une expression matricielle, le vecteur des exemples dans l'espace transformé $\boldsymbol{\Phi}_i^T = \left[\Phi(\boldsymbol{x}_{i,1}), \ldots, \Phi(\boldsymbol{x}_{i,n_i})\right]^T$, le vecteur moyennant $\mathbf{s}$ et la matrice de centrage $\mathbf{J}$, définis par :

$$\mathbf{s} = \frac{1}{n_i}\mathbf{1} \qquad\qquad \mathbf{J} = \frac{1}{\sqrt{n_i}} \left(\mathbf{I}_{n_i} - \mathbf{s}\mathbf{1}^T\right).$$

Malheureusement la matrice $\hat{\boldsymbol{\Sigma}}_i$ n'est pas de rang plein puisqu'elle peut être exprimée comme le produit d'un matrice non carrée avec sa transposée ($\boldsymbol{A}\boldsymbol{A}^T$). Or les mesures probabilistes impliquent généralement (c'est le cas des deux mesures considérées ici) l'inverse des matrices de covariance. Zhou et Chellappa proposent ainsi d'approximer $\hat{\boldsymbol{\Sigma}}_i$ par la matrice suivante :

$$\boldsymbol{C}_i = \boldsymbol{\Phi}_i \, \mathbf{J}\mathbf{Q}\mathbf{Q}^T \mathbf{J}^T \boldsymbol{\Phi}_i^T + \rho \, \mathbf{I},$$

où $\mathbf{Q}$ est une matrice de dimension $r \times n_i$. La matrice $\boldsymbol{C}_i$ ainsi est régularisée et inversible. On trouvera dans l'article de Zhou et Cheppalla [265] le développement qui mène à l'expression suivante de l'inverse :

$$\boldsymbol{C}_i^{-1} = \rho^{-1} \left(\mathbf{I}_{n_i} - \mathbf{Q}\mathbf{B}\mathbf{Q}^T\right),$$

avec

$$\mathbf{B} = \rho \mathbf{I}_r + \mathbf{Q}^T \mathbf{J}^T \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i \, \mathbf{J}\mathbf{Q}.$$

La matrice $\mathbf{Q}$ est choisie de manière à ce que $\boldsymbol{C}_i^{-1}$ soit une bonne estimation de $\hat{\boldsymbol{\Sigma}}_i$, c'est-à-dire en conserve les r vecteurs propres principaux. On remarque que la plupart des matrices considérées dans ce développement ($\hat{\boldsymbol{\mu}}_i$, $\hat{\boldsymbol{\Sigma}}_i$, $\boldsymbol{\Phi}_i$, ...) ne sont pas exprimables en pratique puisque leurs valeurs sont définies dans l'espace transformé, qui peut être de dimension infinie. Mais la forme de l'inverse $\boldsymbol{C}_i^{-1}$ ne dépend que du produit $\boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i = [k(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{i,l})]_{kl} = \boldsymbol{K}$ qui n'est autre que la matrice de Gram définie sur les exemples de la fenêtre i. De même on peut montrer que l'expression des mesures probabilistes considérées s'exprime exclusivement en terme de produits scalaires dans l'espace transformé, et constitue ainsi une démonstration supplémentaire du fameux *kernel trick*.

9.5 SVM à une classe

L'approche précédente tire parti du *kernel trick* en étendant la pertinence du modèle gaussien par son application dans un espace de dimension supérieure. Nous avons vu que les machines à noyaux reposent sur la conjonction du *kernel trick* et du principe de maximisation de la marge, qui implique à la fois la minimisation du Risque Structurel et une sélection parcimonieuse des exemples essentiels pour la fonction de décision. Ce second principe n'est pas appliqué dans l'approche précédente.

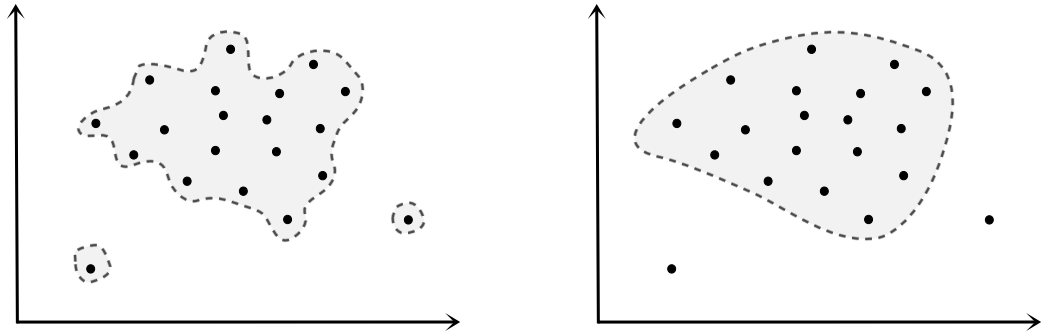
Nous présentons dans cette section les SVM à une classe, qui adaptent le formalisme des SVM pour la caractérisation du support d'une distribution donnée. Après en avoir présenté le principe, nous verrons comment ce dernier peut être exploité pour la détection de rupture.

9.5.1 Principe des SVM à une classe

Le problème posé par Schölkopf et al. dans [209] et [212] consiste à estimer à partir de réalisations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ le support d'une distribution de probabilité P donnée, c'est-à-dire à déterminer un sous-ensemble S de l'espace d'origine tel qu'on ait idéalement :

$$\begin{aligned} P(\boldsymbol{x}) &> 0 \qquad \forall\, \boldsymbol{x} \in S \\ P(\boldsymbol{x}) &= 0 \qquad \forall\, \boldsymbol{x} \notin S. \end{aligned}$$

Le problème se heurte aux mêmes écueils que le problème de classification. En effet, il est possible d'apprendre « par cœur » la distribution des exemples d'apprentissage (voir figure 9.3(a)) mais on se trouve alors en situation de sur-apprentissage et l'ensemble déterminé ne pourra se généraliser correctement sur des données inconnues. Il est donc nécessaire de lisser la frontière de l'ensemble S en régularisant le problème ; la figure 9.3(b) représente un exemple de solution mieux régularisée. Le principe de minimisation du risque structurel nous permet à nouveau de faire face à cette contrainte.



(a) (b)

FIGURE 9.3 – Estimation du support d'une distribution sur un cas simple à deux dimensions, présentant deux exemples marginaux (*outliers*). Comparaison entre (a) un cas de sur-apprentissage, et (b) un cas correctement régularisé.

On reformule le problème en l'assimilant à une classification *one-vs-all* sur une classe unique, déterminée par les exemples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. On leur associe par convention le label $y_i = +1$ qui correspond au résultat idéal de la fonction de décision f, les exemples hors du support de la distribution sont idéalement associés au résultat $f(\boldsymbol{x}) = -1$. On cherche donc à apprendre une fonction f telle que :

$$\begin{aligned} f(\boldsymbol{x}) &\geq 0 \qquad \forall\, \boldsymbol{x} \in S \\ f(\boldsymbol{x}) &< 0 \qquad \forall\, \boldsymbol{x} \notin S. \end{aligned}$$

La fonction noyau joue ici un rôle essentiel en transposant la distribution dans l'espace transformé. On peut se baser sur la haute dimension de cet espace pour supposer que les exemples sont localisés dans une moitié de l'espace dont l'origine est exclue (cette deuxième supposition est toujours vraie dans le cas du noyau RBF gaussien). Il en résulte que la tâche équivaut à l'apprentissage d'un hyperplan de séparation séparant de manière optimale les exemples et l'origine. On rejoint ainsi le cadre des SVM en exploitant la maximisation de la marge comme critère d'optimalité. La figure 9.4 illustre le problème de séparation dans l'espace transformé.

On transpose aisément le problème de minimisation du modèle SVM (se référer au chapitre 3) dans ce contexte :

$$\begin{aligned} \text{minimiser} \quad & \frac{1}{2}\left\|\boldsymbol{w}\right\|^2 + \frac{1}{\nu n}\sum_i \xi_i - \rho \\ \text{sous les contraintes} \quad & \boldsymbol{w}^T \Phi(\boldsymbol{x}_i) \geq \rho - \xi_i \qquad i = 1, \ldots, n \\ & \xi_i \geq 0, \end{aligned} \tag{9.4}$$

où $\boldsymbol{w}$ est le vecteur normal de l'hyperplan de séparation, ρ l'équivalent de la constante b, et ξ_i sont les variables d'écart pénalisant les erreurs de classification. Le paramètre $\nu \in]0, 1]$ s'inspire

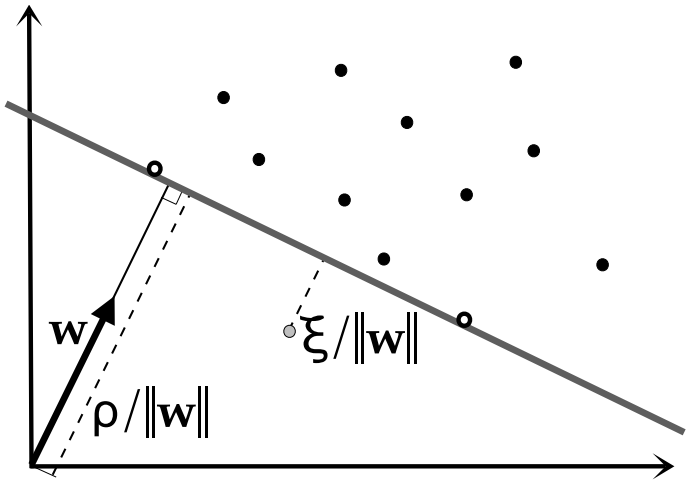


FIGURE 9.4 – Séparation des exemples avec l'origine par l'hyperplan (en gris foncé) défini par le vecteur normal $\boldsymbol{w}$, sur une projection schématique en 2D de l'espace transformé. La figure montre trois vecteurs de support, dont deux à la marge (en blanc) et un autre mal classifié (en gris), dont la distance avec l'hyperplan définit la pénalité ξ.
Cette figure s'inspire d'une figure de l'ouvrage de Schölkopf et Smola [212].

des ν-SVM [211] et permet de contrôler le compromis entre le risque empirique et la complexité du classifieur. On peut par ailleurs montrer [209][212] que ν est à la fois une borne supérieure pour le taux d'erreurs marginales et une borne inférieure pour le taux de Vecteurs de Support dans l'ensemble d'apprentissage, ces deux valeurs convergeant asymptotiquement vers ν lorsque $n \to \infty$.

Nous ne détaillons pas la résolution du problème d'optimisation 9.4. On obtient, de manière similaire aux SVM, un vecteur $\boldsymbol{w}$, moyenne des exemples pondérés par les facteurs de Lagrange α_i (introduits dans la formulation du problème dual). On rappelle que les exemples de facteur non-nul constituent les vecteurs de support du classifieur :

$$\boldsymbol{w} = \sum_i \alpha_i \Phi(\boldsymbol{x}_i).$$

La fonction de décision prend donc l'expression suivante :

$$f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T \Phi(\boldsymbol{x}) - \rho) = \text{sign}\left(\sum_i \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) - \rho\right).$$

Les SVM à une classe ainsi formulés (que l'on pourra désigner par *SVM1C* dans la suite de ce document) sont par la suite exploités par leurs inventeurs [214] pour la détection de nouveauté, en considérant simplement tout vecteur $\boldsymbol{x}$ comme « nouveau » (c'est-à-dire n'appartenant pas à la classe modélisée) si $f(\boldsymbol{x}) < 0$.

9.5.2 Rapport de vraisemblance par SVM1C (LLR)

Loosli et al. [137] exploitent les SVM à une classe sur la base du Rapport de Vraisemblance Généralisé (GLR), présenté précédemment dans la section 9.3.2. Ils adaptent ce dernier en excluant du test d'hypothèses la fenêtre globale W et son modèle $P_0(\boldsymbol{X})$, et supposent dans tous les cas que les échantillons de la fenêtres W_1 sont décrits par le modèle $P_1(\boldsymbol{X})$. Le test d'hypothèses se résume ainsi à évaluer l'égalité entre les distributions P_1 et P_2, ce qui revient à appliquer une approche métrique.

Le rapport de vraisemblance de l'équation 9.2 devient donc :

$$r(t) = \frac{\prod_{i=1}^{n} P_1(\boldsymbol{x}_i)}{\prod_{i=1}^{t-1} P_1(\boldsymbol{x}_i) \prod_{i=t}^{n} P_2(\boldsymbol{x}_i)} = \prod_{i=t}^{n} \frac{P_1(\boldsymbol{x}_i)}{P_2(\boldsymbol{x}_i)}.$$

Le dénominateur mesure la vraisemblance des exemples de la fenêtre W_2 sur la distribution calculée sur ces mêmes exemples. On peut donc considérer celui-ci comme constant, ou du moins de variations négligeables par rapport au numérateur, et simplifier ainsi le critère :

$$r(t) = \prod_{i=t}^{n} P_1(\boldsymbol{x}_i). \tag{9.5}$$

On remarque au passage que ce critère ne nécessite que l'apprentissage d'un unique modèle de probabilité, au lieu des trois modèles impliqués dans les critères GLR et BIC. On estime la distribution des exemples de la fenêtre W_1 au moyen d'un SVM à une classe. Les auteurs couplent la fonction de décision au modèle de famille exponentielle afin d'obtenir une estimation de la probabilité P_1 :

$$\hat{P}_1(\boldsymbol{x}) = \exp\left(\sum_{i=1}^{t-1} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) - g(\theta_0)\right),$$

où $g(\theta_0)$ est la fonction de log-partition, mais ne joue aucun rôle ici. En effet, le test de décision se limite à comparer le logarithme du critère $r(t)$ (équation 9.5) à un seuil s, qui inclut de fait la constante $g(\theta_0)$:

$$\sum_{j=t}^{n} \left(\sum_{i=1}^{t-1} \alpha_i k(\boldsymbol{x}_j, \boldsymbol{x}_i)\right) \gtrless s.$$

On a ainsi défini une mesure de vraisemblance des exemples de la fenêtre W_2 par rapport à la distribution du modèle P_1, qui s'exprime simplement à partir de la matrice de Gram.

9.5.3 Kernel Change Detection (KCD)

Désobry et al. proposent un autre algorithme de détection de rupture basé sur les SVM à une classe [63][74], qu'ils nomment *Kernel Change Detection* (KCD). Ils partent pour cela de l'hypothèse que le noyau est normalisé, c'est-à-dire respecte la condition $k(\boldsymbol{x}, \boldsymbol{x}) = 1 \ \forall \boldsymbol{x}$, sans perte de généralité puisque l'on peut normaliser n'importe quel noyau par la relation suivante :

$$k'(\boldsymbol{x}, \boldsymbol{y}) = \frac{k(\boldsymbol{x}, \boldsymbol{y})}{\sqrt{k(\boldsymbol{x}, \boldsymbol{x})\, k(\boldsymbol{y}, \boldsymbol{y})}}.$$

On peut montrer que le noyau k' respecte également la condition de Mercer. La normalisation a pour effet de restreindre la position des exemples sur la sphère unité dans l'espace transformé ($\|\Phi(\boldsymbol{x}_i)\|^2 = k(\boldsymbol{x}, \boldsymbol{x}) = 1$). L'apprentissage d'un SVM à une classe détermine donc la position d'un hyperplan séparant une section de la sphère et son centre, comme l'illustre la figure 9.5 sur une projection plane simplifiée.

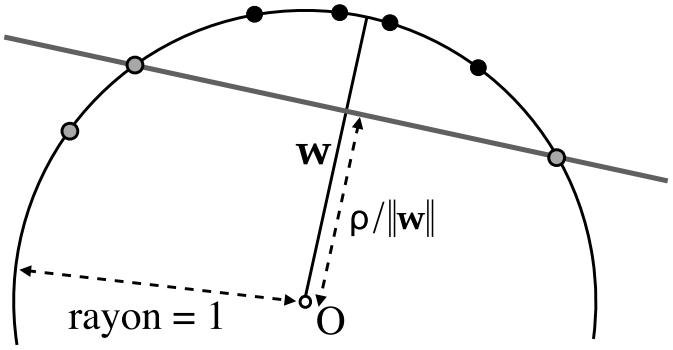


FIGURE 9.5 – La figure montre un exemple d'application de SVM à une classe. La contrainte de normalisation du noyau implique que, dans l'espace transformé, les exemples sont situés sur l'hypersphère de rayon 1. On peut y voir 3 vecteurs de support (en gris), dont deux sont à la marge, et le troisième mal classifié. *Cette figure s'inspire d'une figure de l'article de Desobry et al.* [63].

En suivant le paradigme de l'approche métrique, on apprend deux SVM à une classe modélisant chacun les exemples de l'une des fenêtres W_1 et W_2, en déterminant les hyperplans de séparation de vecteurs normaux respectifs $\boldsymbol{w}_1$ et $\boldsymbol{w}_2$. Il existe nécessairement un plan dans l'espace engendré par les deux vecteurs normaux ; l'intersection de la sphère unité avec ce dernier est un cercle $\mathcal{S}$ de rayon 1 et dont le centre est à l'origine $\boldsymbol{O}$, comme le montre la figure 9.6.

Les auteurs proposent de mesurer la dissimilarité entre les deux modèles sur la base de la distance d'arc entre les points $\boldsymbol{c}_1$ et $\boldsymbol{c}_2$, définis comme les intersections respectives des radiales sur les axes des vecteurs normaux $\boldsymbol{w}_1$ et $\boldsymbol{w}_2$, avec le cercle $\mathcal{S}$ (voir figure 9.6). Néanmoins, une telle mesure n'a de sens que si l'on prend en compte l'étalement des exemples d'une classe autour du « centre » $\boldsymbol{c}_i$ sur le cercle $\mathcal{S}$. Ainsi les auteurs introduisent un dénominateur normalisant la distance précédente par la distance entre les « centres » $\boldsymbol{c}_i$ et les points d'intersection $\boldsymbol{p}_i$ entre les hyperplans $\boldsymbol{H}_i$ et le cercle $\mathcal{S}$, s'inspirant ainsi, de leur propre aveu, du rapport de Fisher entre une statistique du premier ordre (distance entre les moyennes) et du second ordre (déterminant des matrices de covariance). Le critère de dissimilarité ainsi défini a donc l'expression suivante :

$$d_{KCD} = \frac{\widehat{\boldsymbol{c}_1\boldsymbol{c}_2}}{\widehat{\boldsymbol{c}_1\boldsymbol{p}_1} + \widehat{\boldsymbol{c}_2\boldsymbol{p}_2}},$$

où $\widehat{\boldsymbol{x}\boldsymbol{y}}$ représente la distance d'arc entre les points $\boldsymbol{x}$ et $\boldsymbol{y}$ situés sur le cercle $\mathcal{S}$. Ce dernier étant de rayon unitaire, la distance est égale à l'angle $\widehat{\boldsymbol{x}\boldsymbol{O}\boldsymbol{y}}$ exprimé en radian, défini comme l'arccosinus du produit scalaire entre les deux points. Soit :

$$\widehat{\boldsymbol{x}\boldsymbol{y}} = \widehat{\boldsymbol{x}\boldsymbol{O}\boldsymbol{y}} = \arccos k(\boldsymbol{x}, \boldsymbol{y}).$$

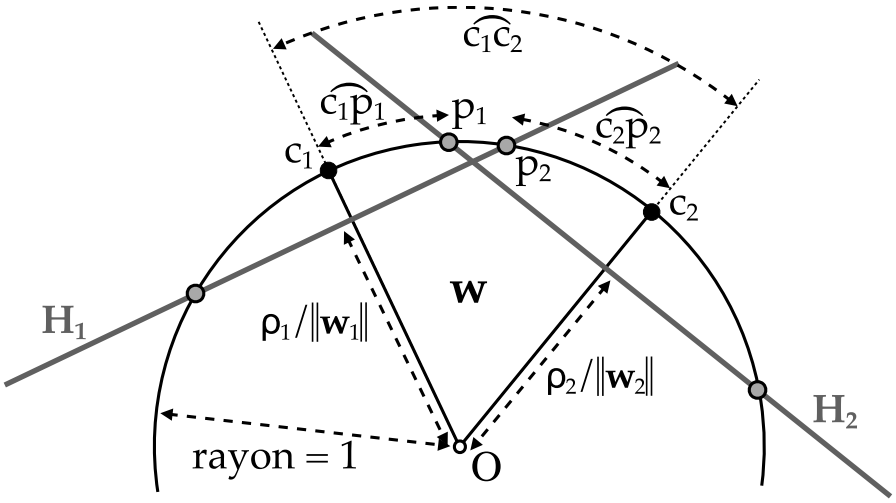


FIGURE 9.6 – Configuration dans l'espace transformé des hyperplans de séparation $\boldsymbol{H}_1$ et $\boldsymbol{H}_2$ pour les fenêtres d'analyse W_1 et W_2. Les distances d'arc $\widehat{\boldsymbol{c}_1\boldsymbol{c}_2}$, $\widehat{\boldsymbol{c}_1\boldsymbol{p}_1}$ et $\widehat{\boldsymbol{c}_2\boldsymbol{p}_2}$, définies à partir des points d'intersection des hyperplans avec l'hypersphère de rayon 1, permettent le calcul de la métrique KCD. *Cette figure s'inspire d'une figure de l'article de Desobry et al.* [63].

De l'expression du point $\boldsymbol{c}_i = \frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}$, on déduit :

$$\widehat{\boldsymbol{c}_1\boldsymbol{c}_2} = \arccos\left(\frac{k(\boldsymbol{w}_1, \boldsymbol{w}_2)}{\sqrt{k(\boldsymbol{w}_1, \boldsymbol{w}_1)\, k(\boldsymbol{w}_2, \boldsymbol{w}_2)}}\right).$$

L'angle $\widehat{\boldsymbol{c}_i\boldsymbol{O}\boldsymbol{p}_i}$ ayant pour cosinus la valeur $\frac{\rho_i}{\|\boldsymbol{w}_i\|}$, on déduit de même :

$$\widehat{\boldsymbol{c}_i\boldsymbol{p}_i} = \arccos\left(\frac{\rho_i}{\sqrt{k(\boldsymbol{w}_i, \boldsymbol{w}_i)}}\right).$$

Toutes les composantes du critère d_{KCD} sont ainsi exprimées en termes de produits scalaires via la fonction noyau, ce qui rend son calcul possible en employant les valeurs de la matrice de Gram.

9.5.4 Mise à jour incrémentale des SVM à une classe

Les deux algorithmes précédents emploient un ou deux SVM à une classe appris itérativement sur les exemples d'une fenêtre glissante. On peut remarquer que, dans le cas d'un pas réduit à un échantillon (ce qui est notre cas), une fenêtre W_i conserve $n_i - 2$ exemples en commun entre les instants t et $t+1$, en supprimant l'exemple $\boldsymbol{x}_t$ et en rajoutant l'exemple $\boldsymbol{x}_{t+n_i}$. Ceci implique que la structure du SVM varie peu puisqu'elle ne diffère au pire que de deux vecteurs de support. Il est possible de tirer parti de ce constat en n'effectuant pas à chaque itération l'apprentissage total du SVM1C.

Le problème d'optimisation des SVM consiste en la minimisation d'un critère sous contrainte, par le biais des multiplicateurs de Lagrange α_i qui, en définitive, déterminent la solution du problème, ainsi que le sous-ensemble des vecteurs de support. Il est donc possible pour l'apprentissage à l'instant $t+1$ de conserver les $\alpha_{i,t}$ relatifs au SVM1C de l'instant t, à l'exception de celui correspondant au vecteur supprimé, et d'initialiser le coefficient du nouveau vecteur à 0. Il en résulte un gain important en nombre d'itérations (et donc en temps de calcul) dans la procédure d'optimisation. On trouvera plus de détail sur cette question dans les articles annexes des auteurs du KCD [94][58].

9.6 Recherche de maxima pour la détection de rupture

Nous avons présenté plusieurs approches de détection de rupture applicables sur une recherche par fenêtres adjacentes glissantes. On suppose ici les fenêtres W_1 et W_2 de même longueur n_w. En exploitant l'une des métriques présentées, on obtient donc à partir de la séquence de N exemples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, une séquence de mesures de distance $[d(1), \ldots, d(n_d)]$ entre fenêtres antérieures et postérieures (avec $n_d = N - 2n_w$), la mesure $d(i)$ d'indice i correspondant à l'instant $i + n_w$, en raison du retard impliqué par la fenêtre antérieure.

La détection de points de rupture se fait généralement par la recherche de maxima locaux dépassant un seuil donné. Cependant, dans le contexte de programmes radiophoniques par exemple, les conditions acoustiques peuvent largement évoluer au sein d'un même fichier (généralement d'une étendue d'une heure) ; aussi il est profitable de prendre en compte les conditions d'enregistrement locales. De plus, la présence de pics secondaires au voisinage des maxima locaux peut parasiter la recherche. On reprend donc pour cela l'algorithme de filtrage non-linéaire proposé par Gillet [88], qui adapte des techniques usuelles en traitement d'image. Celui-ci consiste en 3 étapes successives (illustrées par la figure 9.7, page 142) :

1. **Filtrage médian** : On soustrait dans un premier temps au signal le résultat d'un filtrage médian à large échelle, calculé sur une fenêtre glissante centrée sur l'échantillon concerné, de manière à annuler d'éventuels *offsets* constants locaux sur la métrique. On choisit dans notre cas une fenêtre d'une minute, ce qui correspond à $n_{\text{filt}} = 120$ trames environ, soit :

$$d_{\text{med}}(i) = d(i) - \text{med}\left(d(j_{i,1}), \ldots, d(j_{i,2})\right),$$

avec

$$\begin{aligned} j_{i,1} &= \max(1, i - n_{\text{filt}}/2) \\ j_{i,2} &= \min(n_d, i + n_{\text{filt}}/2 - 1), \end{aligned}$$

où med désigne le filtrage non-linéaire médian. Les variables $j_{i,1}$ et $j_{i,2}$ servent uniquement à s'assurer que les fenêtres de filtrage médian sont correctement définies.

2. **Variance homogène** : Le signal $d_{med}(i)$ est ensuite divisé par la déviation standard locale calculée sur la même fenêtre glissante, afin d'équilibrer la balance des dynamiques sur l'ensemble du signal. On a donc :

$$d_{\text{var}}(i) = d_{\text{med}}(i) - \text{std}\left(d_{\text{med}}(j_{i,1}), \ldots, d_{\text{med}}(j_{i,2})\right),$$

où std désigne le filtrage non-linéaire de calcul de déviation standard.

3. **Détection des pics :** la détection de pics maxima est soumise à deux contraintes : un pic local doit être au delà d'un seuil donné τ et éloigné des pics voisins d'une durée minimale donnée, définie par un nombre de trames n_{max}. On choisit $n_{max} = 10$ dans notre cas. On répond à ces deux contraintes par l'intermédiaire du signal « plateau » d_{plat}, défini de la manière suivante à partir du signal d_{var} :

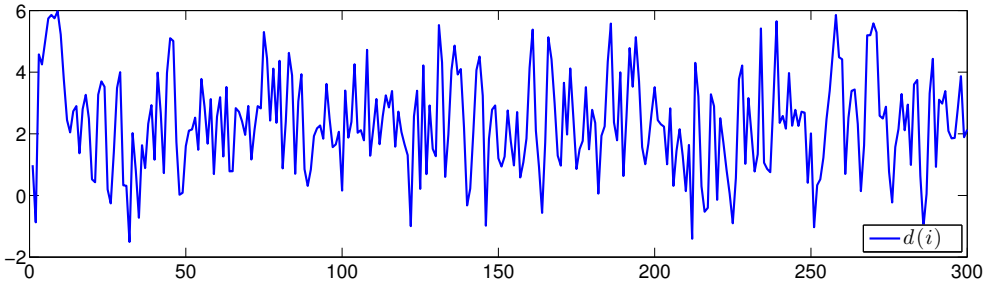
$$d_{\text{plat}}(i) = \max\left(d_{\text{var}}(k_{i,1}), \ldots, d_{\text{var}}(k_{i,2}), \tau\right),$$

où

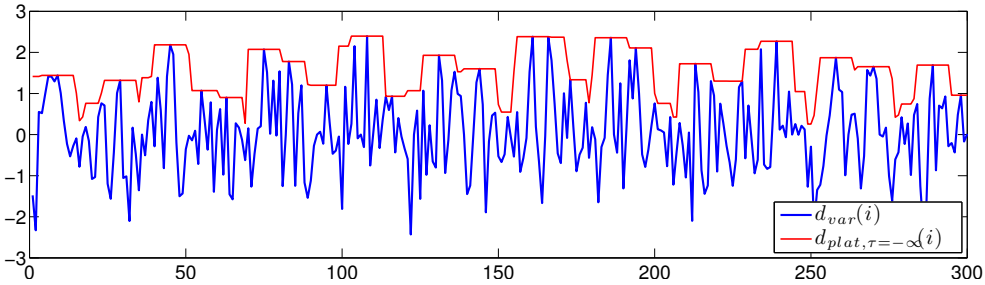
$$\begin{aligned} k_{i,1} &= \max(1, i - n_{\max}/2) \\ k_{i,2} &= \min(n_d, i + n_{\max}/2 - 1). \end{aligned}$$

Les variables $k_{i,1/2}$ jouent le même rôle que les $j_{i,1/2}$ introduits précédemment, pour la longueur de fenêtre $n_{\max}$. On détecte un maximum local quand $d_{\text{var}}(i) = d_{\text{plat}}(i)$.

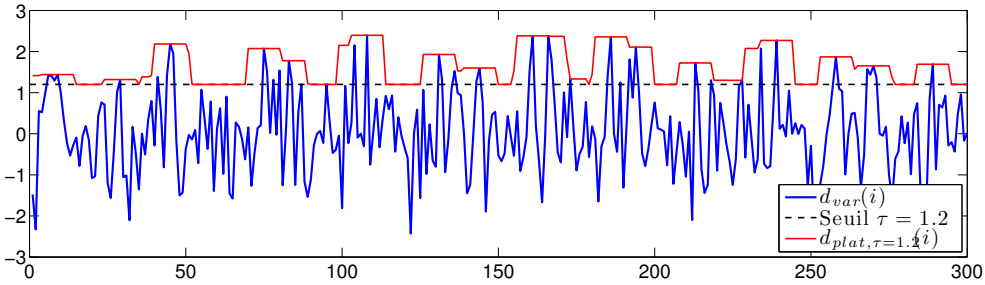
Nous avons introduit dans ce chapitre plusieurs méthodes de détection de rupture, dont la plupart sont étroitement liées à la théorie des Noyaux ou des Machines à Vecteurs de Support. Le chapitre suivant, qui traite de l'évaluation de nos propositions sur des corpus audio publics, comprendra une étude, en section 10.4, dans laquelle nous comparerons les différentes métriques proposées. En particulier, la détermination du seuil τ reste un point essentiel de la détection, qui détermine le compromis entre le nombre de frontières fausses et manquées. Ce point sera abordé de manière pratique en suivant la procédure classique d'estimation empirique de la valeur optimale sur un ensemble de validation.



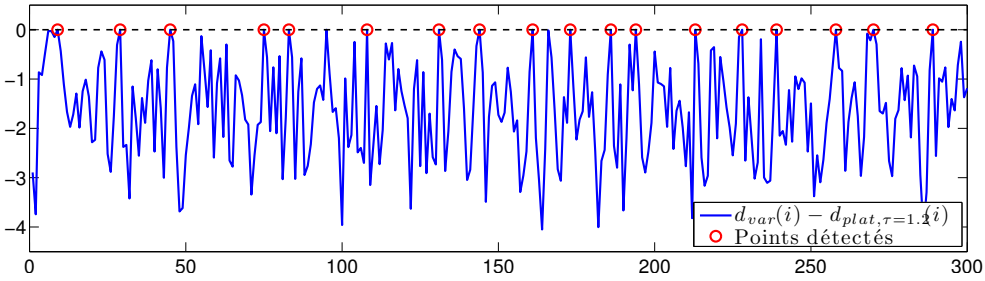
(a) Mesure de nouveauté $d(i)$ brute.



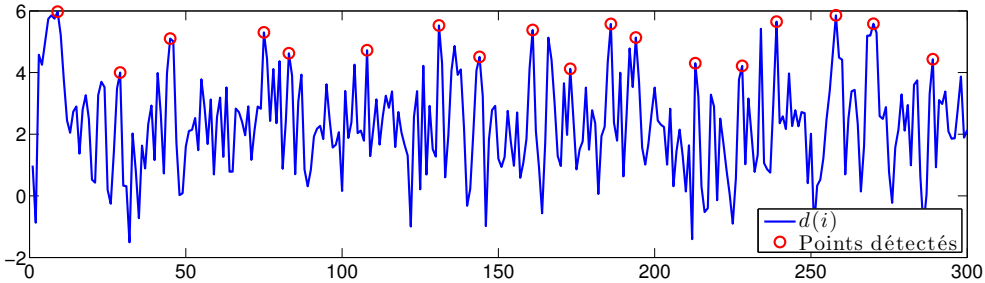
(b) Mesure de nouveauté $d_{var}(i)$ après normalisation, signal plateau (en rouge) $d_{plat,\tau=-\infty}(i)$ sans seuil pour illustrer le principe de filtrage des pics secondaires.



(c) L'application d'un seuil (ici $\tau = 1$) permet de réduire le nombre de maxima détectés.



(d) On observe que les maxima locaux sont détectés aux points où $d_{var}(i) - d_{plat,\tau}(i) = 0$.



(e) Représentation des maxima détectés sur la mesure originale $d_(i)$.

FIGURE 9.7 – Représentation des étapes successives pour la recherche de maxima locaux.

Quatrième partie

Évaluation et analyse

Chapitre 10

Évaluations

Sommaire

10.1 Corpora audio 146
10.1.1 Campagne ESTER 146
10.1.2 Campagne ESTER 2 148
10.1.3 Corpus de Scheirer 148
10.1.4 Corpus Jamendo pour la détection de chant 149
10.2 Protocole d'évaluation 149
10.3 Expérience 1 : comparaison des taxonomies 151
10.3.1 Affinage du noyau par la mesure d'Alignement 152
10.3.2 Résultats sur le corpus ESTER 1 153
10.4 Expérience 2 : post-traitements 154
10.5 Résultats à ESTER 2 et sur le corpus de Scheirer 157
10.6 Expérience 4 : Détection de voix chantée 160

La question de l'évaluation des algorithmes constitue le dernier point essentiel pour l'expérimentateur, dans la mise en place d'un système d'indexation automatique. On peut considérer celle-ci comme la conjonction de deux éléments principaux : un critère d'évaluation quantifiable, qui permet ainsi la comparaison numérique objective des méthodes, et un corpus de test, que l'on suppose représentatif du problème évalué, et qui fournit une base commune pour la comparaison des résultats numériques. Nous commencerons par détailler les corpora exploités dans cette étude pour la classification parole/musique et la détection de chant en section 10.1, puis nous présenterons le procotole et les critères d'évaluation utilisés dans la section 10.2.

Les sections suivantes détailleront les expériences mises en place pour évaluer notre système et valider les points théoriques présentés dans ce document. Ainsi, la première expérience, section 10.3 présentera nos travaux sur le corpus ESTER 1, à travers une étude sur les taxonomies multi-classes. L'expérience décrite en section 10.4 prolongera cette dernière par l'étude des algorithmes de post-traitements. Nous présenterons ensuite en section 10.1.2 les résultats de notre participation à la campagne d'évaluation ESTER 2, puis nous finirons par valider dans la section 10.6, l'application du système développé sur la tâche de détection de chant.

10.1 Corpora audio

Le contenu du corpus de test définit clairement le cadre expérimental, par la proportion relative des classes et leur disposition dans les fichiers, ce que l'expérimentateur va considérer comme une synthèse des difficultés afférentes au problème étudié. Ainsi par exemple, un corpus dans lequel une classe est sous-représentée favorisera implicitement les autres classes. Mais la proportion adéquate des classes dépend largement de l'application visée.

De plus, la constitution d'un bon corpus de test, dont la pertinence est reconnue par la communauté, permet à celle-ci de travailler sur une base d'évaluation commune et ainsi de comparer objectivement les résultats des diverses contributions. Une telle démarche s'accompagne généralement d'un corpus d'apprentissage commun afin de restreindre la variabilité aux algorithmes de classification. Pourtant, la plupart des corpora d'évaluation de la littérature ne sont pas rendus publics, principalement en raison de la protection des droits d'auteurs. Cependant, il existe heureusement plusieurs corpora dont le contenu est partagé par leurs auteurs, et que nous exploiterons dans cette étude.

Le meilleur exemple de corpus public reste cependant celui qui accompagne une campagne d'évaluation nationale ou internationale. En effet, devant le besoin d'un cadre d'évaluation comparative exprimé par la communauté en indexation audio, ces dernières années ont vu fleurir un bon nombre de ces campagnes d'évaluation. Leur but est non seulement de fournir aux participants un corpus dont le soin apporté à la constitution et à l'annotation est hors de portée des laboratoires de recherche, mais également d'imposer un protocole d'évaluation commun qui rend possible une comparaison entre les contributions, dont les modalités sont reconnues.

10.1.1 Campagne ESTER

La campagne d'évaluation ESTER [1] (Évaluation des Systèmes de Transcription enrichie d'Émissions Radiophoniques) est née en 2003 de la réunion d'intérêts communs à plusieurs laboratoires de recherche dans le domaine de la transcription automatique de la parole, et a été proposée par l'AFCP (Association Francophone de la Communication Parlée). La campagne définit un cadre commun pour les différents laboratoires en concurrence, dont les systèmes sont évalués par un acteur extérieur, représenté par le Centre d'Expertise Parisien de la DGA (Délégation Générale pour l'Armement).

La majorité des tâches définies concerne la transcription et l'indexation de la parole, et couvre toute la chaîne qui permet, à partir du signal audio, et en passant par la reconnaissance de locuteur et la transcription de la parole, d'obtenir une base textuelle indexée, axée sur la catégorisation automatique en entités nommées. La première de ces tâches, nommée SES (Segmentation en Événements Sonores) concerne en toute logique la localisation des segments de parole et de musique, qui permet d'appliquer les autres traitements sur les segments identifiés.

La première édition de la campagne ESTER s'est déroulée entre 2003 et 2005, depuis la publication du protocole d'évaluation et du corpus d'apprentissage [93] jusqu'à la publication des résultats comparatifs des participants [83]. Le corpus contient un certain nombre d'heures d'enregistrements d'informations radiophoniques annotées ainsi que des transcriptions textuelles de journaux. Nous n'exploitons que les enregistrements annotés dans le cadre de cette étude. Les documents sonores proviennent des radios suivantes : France Inter, France Info, RFI (Radio France International), RTM (Radio Télévision Marocaine), dont les proportions dans les corpora d'apprentissage et de test sont résumées dans le tableau 10.1.

Bien que les annotations fournies avec le corpus soient très minutieuses, l'effort a surtout été concentré sur la transcription de la parole et la délimitation des segments de classes acoustiques présente quelques erreurs. Nous avons donc reparcouru l'intégralité du corpus d'apprentissage, à l'aide de l'outil d'annotation *Transcriber* [2], et corrigé ces erreurs, ce qui nous a permis en outre d'affiner l'annotation pour distinguer les segments de chant et de parole sur fond bruité (que nous

1. On pourra se rendre sur le site dédié de l'AFCP : `http://www.afcp-parole.org/ester/index.html` pour trouver plus d'informations sur la campagne et le corpus ESTER.

2. Transcriber (`http://trans.sourceforge.net/`) est un outil libre de segmentation, d'annotation et de transcription dont nous avons détourné l'usage habituel pour l'annotation de segments audio.

Station	**Apprentissage**	**Test**
France Inter	35h	2h
France Info	10h	2h
RFI	25h	2h
RTM	20h	2h
France Culture	-	1h
France Musique	-	1h
Total	90h	10h

TABLE 10.1 – Contenu des ensembles d'apprentissage et de test de la campagne ESTER.

désignons par ParoleBr). Le tableau 10.2 synthétise les durées cumulées de chacune des classes pour les différents sous-ensembles du corpus ESTER. Les pourcentages sous les durées précisent la proportion de chaque classe dans le sous-ensemble. Cependant, bien que nous ayons annoté à titre personnel les sous-classes en question dans le corpus de test, aucune modification n'a été apportée à ce dernier lors de l'évaluation, afin de conserver la pertinence de la comparaison aux autres participants. Les 12 minutes manquantes au total par rapport aux 90 heures de données audio sont dues au fait que certains segments ne sont pas pris en compte (silence ou classe non définie).

On reprécise le sens des classes ici mises en jeu :

- **Chant** : désigne la présence de voix chantée, a priori en présence d'un fond musical instrumental.
- **Mix** : désigne la présence de voix sur fond musical.
- **Musique** : désigne la présence de musique sans voix chantée.
- **ParoleBr** : désigne la présence de voix parlée sur fond de bruit (par exemple enregistrements en extérieur).
- **Parole** : désigne la présence de voix parlée pure.

Ensemble	**Chant**	**Mix**	**Musique**	**ParoleBr**	**Parole**	**Total**
Apprentissage	0h38	8h02	1h50	4h48	64h33	79h53
	0.8%	*10.1%*	*2.3%*	*6.0%*	*80.8%*	
Test	0h02	1h14	0h15	0h31	7h51	9h54
	0.4%	*12.5%*	*2.6%*	*5.2%*	*79.3%*	
Total	0h41	9h16	2h05	5h19	72h25	89h48

TABLE 10.2 – Répartition des classes dans les sous-ensembles du corpus ESTER

Le constat le plus frappant est la sur-représentation de la classe de parole dans le corpus, qui est également due au fait que la transcription de parole est la tâche prédominante dans la campagne. Le corpus est en effet essentiellement constitué de bulletins d'informations radiophoniques. La forte proportion de parole sur musique par rapport à la musique provient des habillages musicaux qui accompagnent couramment la voix du présentateur, notamment durant la présentation des titres. On notera enfin la proportion non négligeable de parole bruitée dans le corpus, qui n'est pas prise en compte dans la campagne ESTER, mais qui nous permettra d'apporter une analyse plus fine des résultats.

Le corpus de la campagne ESTER nous sert de point de comparaison pour l'évaluation de nos contributions. Toutefois, il convient de rappeler que, celle-ci ayant été close avant le début de cette thèse, la portée de cette comparaison est d'un impact limité puisque nous avons nécessairement tiré le bénéfice des enseignements qu'apportent les résultats des autres participants, ainsi que des annotations disponibles de l'ensemble de test, qui étaient inconnues dans les conditions réelles de la campagne. Nous avons cependant eu la chance de pouvoir participer à la seconde édition de cette campagne, que nous décrivons ci-dessous.

10.1.2 Campagne ESTER 2

La campagne d'évaluation ESTER 2 a regroupé la plupart des acteurs de la première édition, en particulier les institutions organisatrices, auxquelles se sont greffés plusieurs acteurs industriels. Elle a débuté en janvier 2008 par la mise à disposition d'un ensemble d'apprentissage et d'un autre de développement, pour l'estimation des résultats. Après la diffusion de l'ensemble de test et la campagne de test courant novembre 2008, la campagne s'est terminée en avril 2009 sur un atelier de clôture et une publication des résultats des participants [82].

Le tableau 10.3 indique la répartition du corpus audio parmi les médias et les sous-ensembles qui le constituent. Un nouveau média a été introduit dans le corpus ESTER 2, la radio Africa 1, qui se caractérise par une prise de son plus bruitée que les autres radios, et qui vient donc compliquer la tâche de classification audio. TVME est le nouveau nom de la Radio Télévision Marocaine (RTM), qui était présente dans le corpus ESTER. L'essentiel du corpus provient de la radio RFI, avec environ 70 heures d'enregistrements.

Station	**Apprentissage**	**Développement**	**Test**
France Inter	26h40	2h40	3h40
RFI	68h00	1h20	1h10
Africa 1	4h50	2h15	1h30
TVME (ex RTM)	-	1h00	1h00
Total	99h30	7h15	7h20

TABLE 10.3 – Contenu des sous-ensembles de la campagne ESTER 2.

Cette seconde édition a vu l'essor des recherches sur le sujet de la reconnaissance d'entités nommées. Toutefois, un soin supplémentaire a été apporté à l'annotation de la tâche SES, et le contenu du corpus s'est diversifié pour mieux prendre en considération les problèmes de la détection de la musique et des enregistrements bruités. On constate ainsi dans le tableau 10.4 que les parts de musique et de parole sur musique (mix) sont rehaussées en terme de durée totale (4 heures de plus de mix et 2 heures de plus de musique). Néanmoins la parole demeurent forteme majoritaire dans le corpus.

Ensemble	**Mix**	**Musique**	**Parole**	**Total**
Apprentissage	12h42	3h32	82h36	98h51
	12.8%	*3.6%*	*83.6%*	
Développement	0h22	0h08	5h34	6h04
	6.2%	*2.2%*	*91.6%*	
Test	0h22	0h26	6h12	7h01
	5.3%	*6.2%*	*88.5%*	
Total	13h27	4h06	94h23	111h57

TABLE 10.4 – Répartition des classes dans les sous-ensembles du corpus ESTER2.

10.1.3 Corpus de Scheirer

Le corpus que Scheirer a constitué en 1996 pour l'évaluation de ses travaux sur la classification parole/musique [207] est diffusé par l'auteur, et a été repris par la suite dans plusieurs publications de la communauté [195][43][5][13], dont deux publications de Ellis et de ses coauteurs [247][24] qui ont complété l'annotation originale du corpus.

Celui-ci est constitué d'un ensemble de 160 extraits de 15 secondes collectés au hasard à la radio, la moitié étant des extraits de parole pure et l'autre moitié de musique pure. Il ne contient donc pas d'extraits de parole sur fond musical.

La répartition en fichiers de classes homogèness a un impact non négligeable sur l'évaluation des résultats puisque l'absence de transition entre classes (qui constituent des zones plus difficilement caractérisables) facilite beaucoup la tâche de classification. Nous verrons ainsi que les

post-traitements les plus simples (cumul des résultats sur des fenêtres de décision longues) améliorent facilement les résultats.

La taille réduite du corpus et l'absence de test sur la classe mix limitent l'importance des résultats sur ce dernier, mais la base nous permettra avant tout de comparer nos résultats à ceux des auteurs l'ayant exploité, sur le problème de classification parole/musique.

10.1.4 Corpus Jamendo pour la détection de chant

La recherche sur le problème de la détection de chant est plus récente que la classification parole/musique et le sujet est beaucoup moins traité par la littérature. Il existe donc peu de corpora publics couvrant ce sujet. On peut citer le corpus de Holzapfel et Stylianou, constitué pour l'identification de chanteur [105], et qui sera par la suite exploité pour la détection de chant [148], qui cumule 3h12 de musique, mais se limite au genre particulier du Rembetiko (musique traditionnelle grecque).

Nous avons donc constitué un corpus [3], introduit dans [192], qui pourra, nous l'espérons, servir de base commune à la communauté pour l'évaluation de la détection de chant. Afin de pouvoir diffuser les données audio, nous avons réuni un ensemble de titres musicaux téléchargés depuis le site Jamendo [1], un site communautaire de partage de musique sous licence *Creative Commons* (c'est-à-dire libre de droits). Le corpus, d'une durée totale de 6 heures de musique, est constitué de 93 titres répartis entre les sous-ensembles d'apprentissage (61 titres), d'évaluation (16 titres) et de test (16 titres), et est constitué d'exemples de musique pop ou rock, qui constitue le genre majoritaire sur les radios généralistes. Les chansons ont été annotées avec une précision de l'ordre d'un dixième de seconde sur les frontières de segments. L'annotation du chant demeure néanmoins complexe car il existe en réalité énormément d'interruptions de la voix durant une même phrase, et certaines consonnes prolongées sont parfois très ambigües.

Le tableau 10.5 résume la répartion des classes dans les sous-ensembles du corpus. À nouveau la classe musique représente les segments sans voix chantée. Comme sur le corpus de Scheirer, on constate que les deux classes sont à peu près équilibrées sur tous les sous-ensembles, ce qui semble être un constat assez général sur la musique populaire.

Ensemble	**Chant**	**Musique**	**Total**
Apprentissage	2h05 *52.6%*	1h53 *47.4%*	3h58
Développement	0h31 *51.4%*	0h29 *48.6%*	1h00
Test	0h32 *49.6%*	0h33 *50.4%*	1h06
Total	3h09	2h55	6h05

TABLE 10.5 – Répartition des classes dans les sous-ensembles du corpus Jamendo. À nouveau Chant désigne la présence de voix chantée, sur fond musical éventuel, tandis que Musique désigne la présence de musique instrumentale, sans voix chantée.

10.2 Protocole d'évaluation

Le découpage du signal audio en une séquence de trames étant la méthode presque unanimement employée dans le domaine de la classification audio, c'est généralement le taux moyen d'erreur de classification par trames qui est employé pour évaluer les performances des algorithmes, parfois associé à la matrice de confusion, qui permet de distinguer le taux d'erreur sur les classes considérées.

3. Le corpus Jamendo est disponible à l'adresse suivante :
`http://www.telecom-paristech.fr/~ramona/icassp08/`.

Toutefois, la discrétisation de l'annotation induite par le découpage en trames suppose lors de l'évaluation, l'homogénéité en termes de classe sur chaque trame. L'impact est généralement minime mais peut devenir non-négligeable lorsque les trames de décision atteignent une taille de l'ordre de plusieurs secondes, puisque certains segments peuvent alors être ignorés lors de l'évaluation. De plus, l'usage d'un tel critère pour une évaluation comparative implique d'imposer le même pas d'avancement à tous les systèmes.

Le protocole des campagnes d'évaluation ESTER 1 et 2 exploite une alternative qui permet en plus d'évaluer le cas des classes se chevauchant. En effet, bien que nous ayons justifié la pertinence pour la classification d'un problème à trois classes, dont l'une est la superposition des deux autres (voir la section 2.2), le protocole d'évaluation ESTER, que nous suivons dans ce document, se base sur un problème à deux classes (parole et musique) pouvant se chevaucher. Ce chevauchement constitue ce que nous avons défini comme la classe mix.

Ainsi, comme le montre la figure 10.1, on réunit dans un premier temps les trames successives en segments, puis l'on sépare ces derniers en autant de séquences de segments que de classes, en tenant compte des classes superposées.

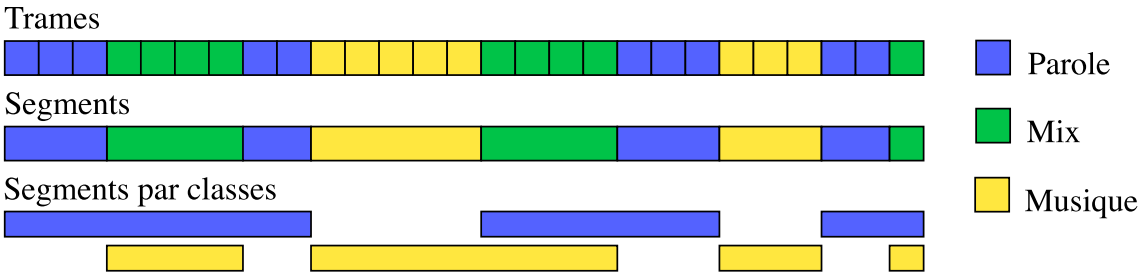


FIGURE 10.1 – Conversion des résultats à 3 classes sur les trames vers les problèmes sur segments pour chaque classe.

Par la suite le protocole suivant est appliqué sur chaque classe (voir la figure 10.2) :

- On inclut d'abord entre chaque paire de segments consécutifs de la classe un segment de « non-classe » qui représente l'absence de cette classe.
- On applique ensuite une tolérance aux frontières qui permet d'ignorer lors de l'évaluation les décalages par rapport à la frontière réelle d'une durée inférieure à un seuil τ_{tol} qui est fixé à 0.25 s dans les campagnes ESTER. Concrètement on exclut de l'évaluation les τ_{tol} secondes qui précèdent et suivent chaque frontière réelle.
- On reporte sur les segments estimés les zones ignorées par la tolérance, et l'on introduit également la « non-classe » sur les zones non ignorées qui ne sont pas estimées dans la classe.
- On extrait de la comparaison des segments pré-traités réels et estimés, les segments de bonne classification, de fausse alerte (faux positif) et de détection manquée (faux négatif).

On calcule ainsi les grandeurs d_{OK}, d_{FA} et d_{DM} respectivement définies comme les durées cumulées des segments corrects, de fausse alerte et de détection manquée. On définit également les durées cumulées de segments estimés $d_{\mathrm{EST}} = d_{\mathrm{OK}} + d_{\mathrm{FA}}$ et de segments réels $d_{\mathrm{REEL}} = d_{\mathrm{OK}} + d_{\mathrm{DM}}$,

Les valeurs de rappel R et de précision P sont alors calculées de la manière suivante :

$$R = \frac{d_{\mathrm{OK}}}{d_{\mathrm{OK}} + d_{\mathrm{DM}}}, \qquad P = \frac{d_{\mathrm{OK}}}{d_{\mathrm{OK}} + d_{\mathrm{FA}}},$$

ce qui revient à définir le rappel et la précision comme le temps cumulé de détection correction sur le temps où, respectivement, la classe est réellement présente et où la classe est détectée.

La F-mesure, qui représente le critère global d'évaluation, est définie comme la moyenne harmonique des deux précédentes mesures :

$$F = \frac{2RP}{R+P}.$$

Elle constitue ainsi un compromis entre les deux, dont l'effet est beaucoup plus pénalisant que la moyenne arithmétique si l'une des valeurs est particulièrement faible (puisque $F = 0$ si $R = 0$ ou

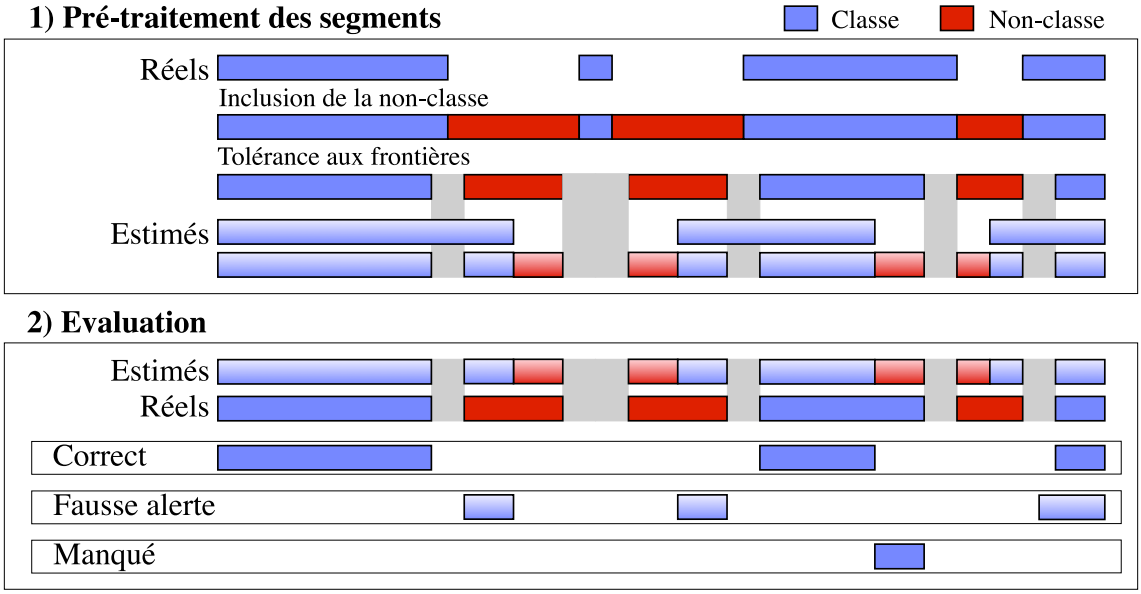


FIGURE 10.2 – Post-traitement des segments pour la prise en compte de la tolérance aux frontières et extraction des segments corrects, de fausses alertes, et manqués, pour le calcul des critères d'évaluation.

$P = 0$).

Les résultats de la campagne ESTER mentionnent également, comme mesures d'erreur, les classiques taux de fausse alerte FA $= \frac{d_{\text{FA}}}{d_{\text{EST}}}$ (ou faux positif) et de faux rejet MD $= \frac{d_{\text{DM}}}{d_{\text{REEL}}}$ (ou faux négatif).

10.3 Expérience 1 : comparaison des taxonomies

Nous comparons, dans cette première expérience, différentes taxonomies de classification multi-classes pour le problème de la classification parole/musique, synthétisées dans la figure 10.3. Nous étudions par ailleurs l'influence de la prise en compte des trames de chant sur les performances.

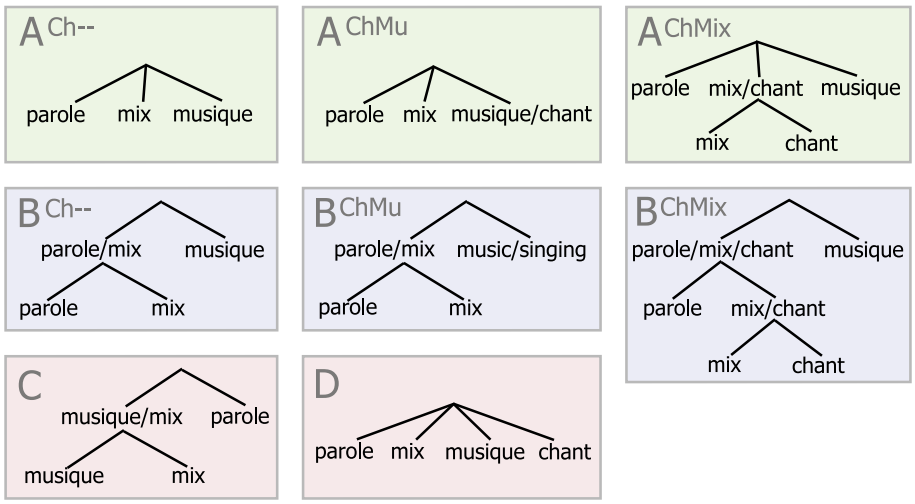


FIGURE 10.3 – Représentation hiérarchique des huit taxonomies multi-classes comparées dans l'expérience 1.

Nous proposons ici deux taxonomies principales dont l'une (A) est basée sur un cadre non-

hiérarchique de combinaison *one-vs-one*, et l'autre (B) sur une approche hiérarchique d'arbre binaire de classification, qui consiste d'abord à séparer la musique des exemples contenant de la parole, puis à distinguer parmi ces derniers les exemples de parole pure et de mix.

Les suffixes *Ch–*, *ChMu* et *ChMix* désignent des variantes qui différent par leur prise en compte des exemples de chant. Dans les taxonomies *Ch–*, les exemples de chant ne sont pas pris en compte lors de l'apprentissage ; la classe musique ne contient alors que de la musique instrumentale sans voix chantée. Cette première approche est basée sur l'hypothèse que les exemples chant sur fond musical constituent une source de confusion possible avec la classe mix (en supposant que le chant est proche de la voix parlée). Dans la variante *ChMu*, on conserve, sans modification, les exemples contenant de la voix chantée dans l'ensemble des exemples de musique. Le chant n'est bien sûr pas considéré comme une classe supplémentaire lors de la phase d'évaluation. Enfin dans l'approche *ChMix*, partant de la proximité supposée du chant avec le mix, on associe dans un premier temps ces deux classes pour les séparer par la suite (les exemples de chant seront ensuite associés à la classe musique lors de l'évaluation).

La taxonomie C est une variante de l'approche hiérarchique qui consiste d'abord à séparer la parole pure de tout signal contenant de la musique. Celle-ci est toutefois beaucoup moins intuitive que la B parce que la musique est généralement en retrait par rapport à la parole dans les exemples de mix.

Enfin, la taxonomie D se base sur un paradigme *one-vs-one* incluant également la classe de chant (dont les exemples détectés sont par la suite associés à la classe de musique).

Les nœuds binaires des arbres représentés impliquent l'application simple d'un SVM discriminatif sur les classes des fils. Nous avions de plus mentionné dans la section 5.1.5.3 la possibilité de définir un cadre hybride en introduisant des noeuds non-binaires dans un arbre de classification, traités par une approche *one-vs-one*, que nous retrouvons dans la taxonomie A *ChMix*. Les labels « classe1/classe2 » désigne une classe formée pour l'apprentissage par l'union des exemples des deux classes.

Cette étude est basée sur le corpus ESTER 1 (section 10.1.1), sur lequel nous avons annoté les occurrences de voix chantée. Nous utilisons les descripteurs présentés dans la section 6.5, dont nous sélectionnons par l'algorithme IRMFSP (présenté dans la section 7.3.2) les d plus pertinents. Les SVM exploitent un noyau RBF gaussien, dont nous discutons l'affinage ci-dessous, et l'apprentissage est effectué sur un maximum de 20000 exemples par classe. L'application des taxonomies multi-classes se base sur l'algorithme d'estimation pondérée des probabilités a posteriori, proposé en section 5.1.5.4, lesquelles sont lissées par un filtrage médian (voir section 8.2).

10.3.1 Affinage du noyau par la mesure d'Alignement

Nous commençons par comparer les procédures de recherche par maillage (section 4.2.1) et d'optimisation par maximisation du critère d'Alignement, introduit en section 4.4.1 pour l'affinage du paramètre σ du noyau. Dans cette partie, le nombre de descripteurs sélectionnés est fixé arbitrairement à $d = 20$. La recherche par maillage est effectuée sur un ensemble de 12 valeurs logarithmiquement réparties entre 0.2 et 15. Les SVM impliquées dans les taxonomies sont donc apprises pour chacune de ces valeurs ; la valeur de σ maximisant la F-mesure globale sur l'ensemble de validation est choisie.

La figure 10.4 montre, pour chaque taxonomie, les F-mesures calculées après affinage du noyau sur l'ensemble de test pour les deux méthodes (la recherche par maillage est en teintes foncées et l'optimisation sur l'alignement en couleurs claires), avec et sans post-traitement par lissage médian (respectivement en vert et en bleu). Le dernier sous-histogramme indique la moyenne sur toutes les taxonomies.

Il est clair qu'en l'absence de post-traitement, la recherche par maillage se montre plus efficace que l'alignement pour l'affinage du noyau. Toutefois, le constat s'inverse lorsque l'on applique le filtrage médian. Ceci s'explique par le fait que l'alignement apporte un meilleur affinage pour quelques-uns des discriminateurs impliqués dans la taxonomie. Le filtrage médian corrige alors efficacement les erreurs accidentelles (sur une ou deux trames adjacentes) et compense ainsi le léger désavantage (de l'ordre de quelques dizièmes de pourcent sur la F-mesure) des SVM affinés par l'alignement.

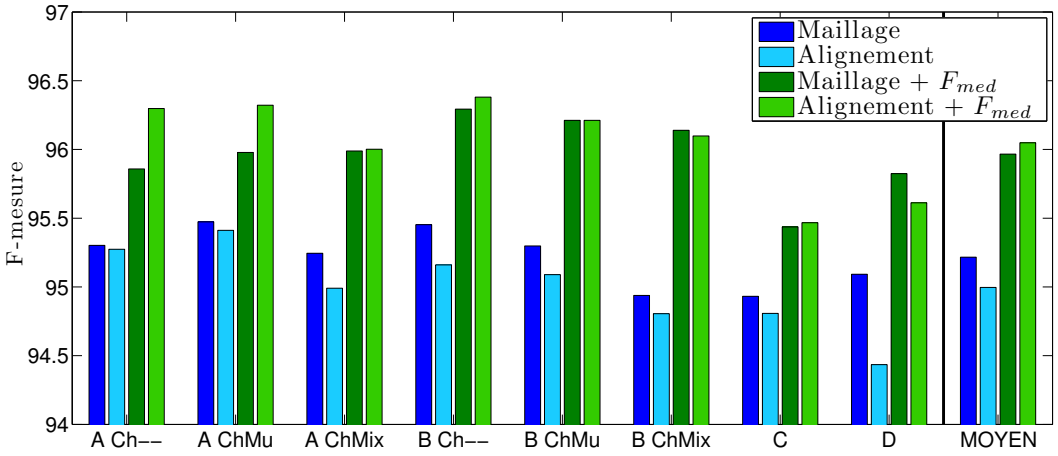


FIGURE 10.4 – Comparaison des résultats (en F-mesure) sur les différentes taxonomies après affinage des noyaux par recherche par maillage ou par optimisation de l'alignement, avec et sans filtrage médian.

On confirme donc que l'affinage du paramètre par la maximisation de l'alignement permet d'obtenir des performances comparables à la recherche par maillage, à un coup fortement réduit, puisqu'une seule opération d'apprentissage de SVM est nécessaire. Ce résultat est pour nous essentiel, parce qu'il permet la mise en place d'un système de classification audio dont l'apprentissage est entièrement automatisé et ne nécessite pas de corpus de validation.

10.3.2 Résultats sur le corpus ESTER 1

La figure 10.5 montre l'évolution de la F-mesure avec le nombre d de descripteurs sélectionnés, pour chacune des huit taxonomies présentées précédemment.

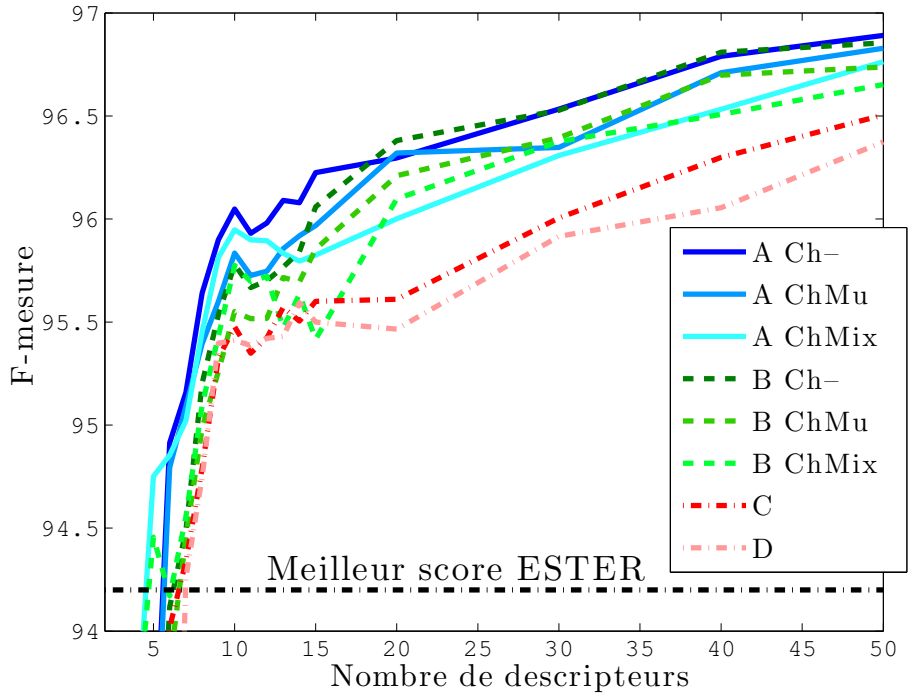


FIGURE 10.5 – Evolution de la F-mesure pour les 8 taxonomies multi-classes en fonction de la dimension d du vecteur de descripteurs, et comparaison au meilleur résultat de la campagne ESTER 1.

	globale			**parole**			**musique**		
Participant	F	%fa	%fr	F	%fa	%fr	F	%fa	%fr
$d = 50$	96.9	2.0	4.5	99.4	13.0	0.5	78.8	1.5	29.6
$d = 10$	95.9	3.3	5.4	99.1	19.4	1.0	73.8	2.5	33.2
$d = 2$	93.3	11.9	4.1	98.9	16.2	1.5	64.8	11.6	20.3
1er ESTER	94.2	2.1	9.5	98.8	30.1	1.5	52.7	1.2	61.7
2e ESTER	93.1	1.3	12.1	98.9	9.7	1.9	33.7	1.0	78.5
3e ESTER	92.7	11.7	5.7	99.2	36.6	0.7	54.8	10.9	38.7
4e ESTER	90.7	1.3	16.2	97.4	8.0	4.9	17.8	1.1	89.6

TABLE 10.6 – Performances des participants à la campagne ESTER 1 pour la tâche SES de segmentation.

On remarque en premier lieu que les configurations A et B (en traits pleins, respectivement bleus et verts) sont sensiblement plus efficaces que les configurations C et D (en pointillés rouges et roses), de manière presque uniforme sur l'ensemble des dimensions d. En effet, la taxonomie C est pénalisée par l'union de deux classes trop éloignées acoustiquement (mix et musique), tandis que dans la D, la classe de chant, trop peu fournie en exemples, est trop faiblement caractérisée et réduit ainsi les performances globales. L'écart avec les approches A et B reste limité à moins de 1% mais, sur un score de 96%, l'avantage de ces dernières représente une réduction relative de 25% sur l'erreur, ce qui confirme l'importance du choix de la meilleure taxonomie.

En revanche, on ne remarque pas d'écart notable entre les taxonomies A et B, à part à basses dimensions ($d < 20$) où l'approche *one-vs-one* (A) se montre plus efficace que l'approche hiérarchique. De plus, l'influence de la classe de chant est très similaire sur les deux cas. A haute dimension ($d > 15$), l'absence des exemples de chant est la plus profitable, tandis que leur inclusion dans une classe impliquant les exemples de mix se révèle moins efficace que l'union plus naturelle chant/musique. Toutefois, à dimension très basse ($d < 7$), la première union (chant/mix) devient plus efficace, probablement parce que la diversité apportée par les exemples de chant compense en partie les défauts de caractérisation dus au faible nombre de descripteurs.

La F-mesure augmente avec la dimension du vecteur de descripteurs, approchant asymptotiquement les 97%. Toutes les taxonomies testées dépassent, pour $d > 7$, le meilleur score obtenu durant la campagne ESTER 1 (indiqué par la ligne noire en pointillés, et égal à 94.2%), ce qui montre l'efficacité du système proposé pour cette tâche. On note même que certaines taxonomies (A *ChMix* et B *ChMix*) demeurent d'ailleurs efficaces à très basse dimension ($d = 5$), avec une F-mesure autour de 94.5%. Ainsi, pour une complexité raisonnable ($d = 10$), la meilleur et la pire taxonomie apportent respectivement un gain absolu de 2% et 1.3% sur le meilleur résultat d'ESTER. Tous les systèmes proposés dans le cadre de la campagne étaient basés sur les descripteurs MFCC et leurs dérivées premières et secondes, pour une dimension entre 33 et 40. L'usage de techniques de sélection de descripteurs apporte donc ici un réel avantage en termes de performances et de complexité.

Le tableau 10.6 détaille les résultats des trois meilleurs participants à la campagne ESTER 1, et les confronte à ceux de la taxonomie la plus efficace (A *Ch–*) sur différentes dimensions d. On remarque que même à très faible dimension ($d = 2$), la taxonomie choisie surpasse le deuxième participant, ce qui confirme à nouveau la pertinence de la sélection de descripteurs adjointe à l'emploi des SVM. L'amélioration la plus notable concerne la détection de la musique, sur laquelle le système proposé apporte un gain absolu de 12 à 26%, dû principalement à une forte réduction du taux de fausse alerte (colonne %fa). Ceci s'explique par le fait que sur la plupart des autres systèmes, l'accent a été mis sur la bonne reconnaissance des regions de parole (en raison de l'importance des autres tâches sur la parole dans la campagne).

10.4 Expérience 2 : post-traitements

Nous poursuivons l'étude précédente sur le corpus ESTER 1 pour montrer les effets des algorithmes de post-traitement dynamiques présentés dans la partie III.

Le système exploité ici est identique au système correspondant aux résultats de la seconde ligne ($d = 10$) du tableau 10.6, basé sur la taxonomie A *Ch–*, soit une approche *one-vs-one* sur les trois

classes de parole, de mix et de musique, les exemples de chant étant exclus de cette dernière classe.

Nous comparons ici les gains en performances apportés respectivement par le filtrage médian (qui est appliqué dans l'expérience précédente), le post-traitement par HMM proposé dans la section 8.3.2.2, ainsi que les 5 algorithmes hybrides basés sur un principe de détection de rupture, présentés dans le chapitre 9.

Nous avons vu dans la section 9.6, que la détermination des frontières de segments avec ces dernières approches, se base en définitive sur l'application d'un seuil empirique τ (voir section 9.6) qui fixe l'habituel compromis entre frontières fausses et manquées (faux positifs et faux négatifs). La valeur optimale du seuil est ainsi déterminée en recherchant le maximum de la F-mesure globale calculée après application du post-traitement sur l'ensemble de validation. La figure 10.6 montre l'évolution de la F-mesure en fonction du seuil τ (échelonné entre -1 et 3), pour les cinq métriques proposés, à savoir le critère BIC (*Bayesian Information Criterion*) en vert, les mesures LLR (*Log Likelihood Ratio*) et KCD (*Kernel Change Detection*), toutes deux basées sur les SVM à une classe, respectivement en rose et rouge, et enfin les mesures DIV (Divergence de Kullback-Leibler) et BAT (Distance probabiliste de Bhattacharyya) en bleu clair et foncé, toutes deux calculées dans l'espace RKHS, espace image de la transformation implicite appliquée par le noyau. Toutes les mesures de détection de rupture sont calculées sur deux fenêtres glissantes de 9 trames longues (qui correspondent chacune à 5 secondes de signal).

Les résultats mesurés sans post-traitement et avec le filtrage médian sont respectivement indiqués par les lignes pointillées noire et grise.

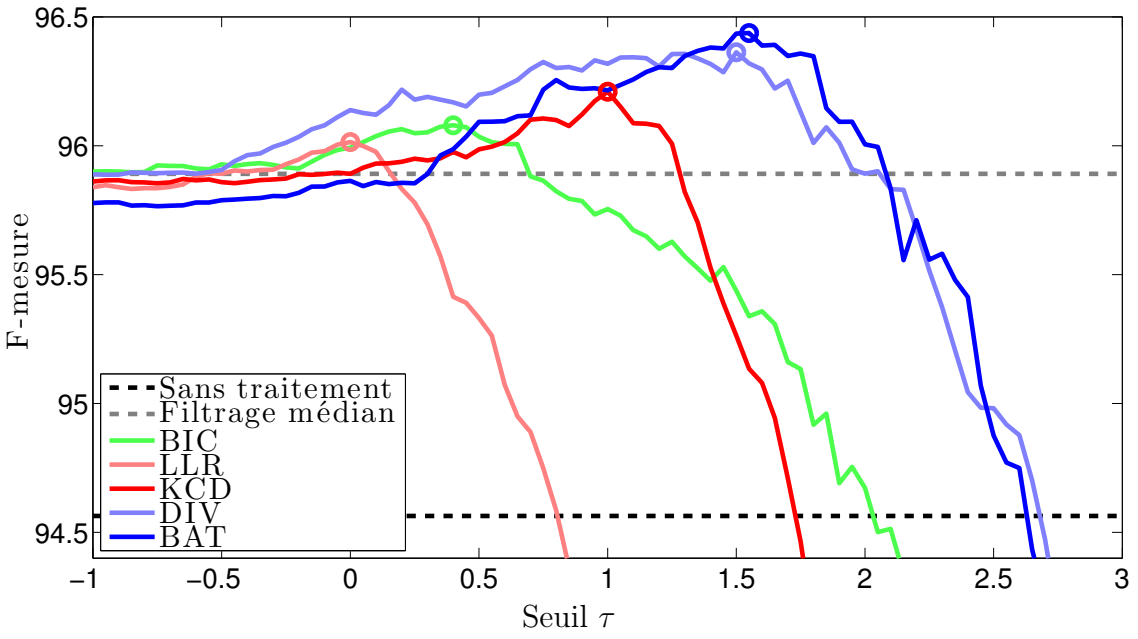


FIGURE 10.6 – Évolution de la F-mesure globale sur l'ensemble de validation par rapport au seuil τ, pour chacune des mesures de détection de rupture.

On remarque en premier lieu que les cinq courbes suivent toutes un profil similaire. Lorsque le seuil est très faible ($\tau < 0$), les métriques ont à peu près les mêmes performances, par ailleurs proches de celles du filtrage médian. En effet, le nombre de maxima dans les courbes de détection de rupture étant limité (du fait de la procédure de recherche, présentée dans la section 9.6) en deçà d'un certain seuil, tous les pics sont retenus et l'algorithme n'évolue plus. On est alors dans une situation de « sur-segmentation », où le traitement est appliqué sur une série de segments très courts, dont l'échelle avoisine celle de la fenêtre de filtrage médian, ce qui explique la proximité observée entre les différentes approches.

Lorsque le seuil τ augmente, les performances augmentent de manière quasi monotone pour atteindre un maximum (indiqué sur la figure par un cercle, pour chaque métrique), au delà duquel celles-ci chutent brutalement. En effet, plus on augmente le seuil, plus le nombre de maxima est

restreint et plus les segments sont larges. Ainsi lorsque l'on dépasse le seuil optimal, on fusionne alors les classes de segments de plus en plus importants, ce qui explique que les effets s'amplifient rapidement. Bien sûr si l'on pousse le seuil à l'extrême on ne détecte plus aucune frontière, ce qui revient à assigner la même classe à l'ensemble du signal audio, à priori la classe de parole, puisque celle-ci est majoritaire. On n'aura donc pas un résultat nul, mais très pénalisé.

La figure montre sans équivoque la supériorité des métriques basées sur les distances probabilistes dans l'espace RKHS, par rapport aux autres métriques. Nous rappelons que les fenêtres glissantes ont une largeur de 9 trames, ce qui signifie que les SVM à une classe impliqués dans les métriques KCD et LLR, effectuent l'apprentissage de la classe sur ces seules 9 trames. Ainsi le critère LLR, qui n'implique en réalité qu'un seul SVM (appris sur la fenêtre passée et évalué sur la fenêtre future), montre sa faiblesse par rapport aux autres métriques, du fait d'une caractérisation si réduite. Le critère KCD, qui implique bien deux SVM à une classe, fournit de meilleurs résultats, mais on peut supposer que les 9 exemples sont insuffisants pour caractériser assez précisément l'axe des hyperplans délimitant les classes, qui définissent le critère lui-même. Ainsi, les métriques RKHS, qui n'impliquent que la modélisation gaussienne des exemples dans l'espace RKHS se comportent beaucoup mieux par rapport au faible nombre d'exemples.

Il est important de préciser que le choix des longueurs de fenêtre n'est pas arbitraire et résulte d'une détermination empirique sur l'ensemble de validation, que nous ne détaillons pas ici, qui traduit le compromis entre précision du modèle et précision temporelle des fenêtres modélisées. En effet, l'augmentation de la largeur des fenêtres implique que celles-ci ont plus de chance d'inclure des changements de classes, qui viennent compenser l'effet bénéfique sur la modélisation.

	globale			**parole**			**musique**		
Système	**F**	**%fa**	**%fr**	**F**	**%fa**	**%fr**	**F**	**%fa**	**%fr**
Sans traitement	94.56	5.68	6.15	98.73	15.70	1.81	67.84	5.16	33.59
Filtre médian	95.89	3.34	5.38	99.06	19.44	0.98	73.80	2.51	33.17
HMM à 2 gauss.	95.45	5.03	4.96	99.01	10.90	1.46	72.98	4.72	27.09
HMM à 5 gauss.	96.04	4.07	4.55	99.08	**8.31**	1.44	76.67	3.85	**24.21**
HMM à 10 gauss.	96.15	3.18	5.00	99.12	9.88	1.29	76.26	2.83	28.43
Hybride BIC	96.08	3.40	4.97	99.16	23.49	0.59	74.48	2.37	32.66
Hybride LLR	96.01	3.55	4.99	99.11	22.83	0.73	74.54	2.56	31.93
Hybride KCD	96.21	3.15	4.92	99.16	22.61	0.64	75.51	2.15	31.93
Hybride DIV	96.36	3.34	**4.48**	99.10	29.92	**0.43**	**77.19**	1.97	30.09
Hybride BAT	**96.44**	**2.93**	4.65	**99.22**	22.68	0.50	76.81	**1.91**	30.86

TABLE 10.7 – Comparaison des résultats obtenus sur le corpus ESTER avec les différents paradigmes de post-traitement présentés (filtrage médian, HMM et segmentation aveugle).

Le tableau 10.7 compare les résultats sans post-traitement et après application d'un filtrage médian, d'un lissage HMM, ou des métriques de segmentation aveugle proposées. Les critères de F-mesure, fausse alerte (%fa) et faux rejet (%fr) sont indiqués pour les deux classes parole et musique, ainsi que pour l'ensemble des segments (classe « globale »).

On constate en premier lieu que les différences en terme de F-mesure globale sont assez réduites (le gain maximal absolu est de 1.88% avec l'approche hybride Bhattacharrya, soit tout de même une réduction relative de l'erreur d'environ 35%), ce qui montre sans surprise que les effets du post-traitement ne peuvent se substituer au soin à apporter à la mise en place et à l'affinage du système de classification. Les erreurs accidentelles (de l'ordre d'une ou quelques trames consécutives) sont facilement corrigées en considérant les résultats proches dans le temps, mais les erreurs structurelles (une classe mal apprise ...) sont trop étalées dans le temps pour satisfaire le formalisme des méthodes de post-traitement. Cependant, bien que la F-mesure constitue le critère global d'évaluation, les différences sont plus marquées et plus facilement interprétables sur les autres critères présentés.

On constate que les deux méthodes de post-traitements proposées dans cette thèse (HMM et hybride par segmentation) améliorent les performances par rapport au filtrage médian. L'augmentation de la complexité des modèles de probabilités des HMM (c'est-à-dire le nombre de gaussiennes

des modèles GMM) favorise, sans surprise, les résultats avec cette méthode ; toutefois, au delà de 10 gaussiennes, on ne note pas d'amélioration notable.

Bien que les écarts entre les trois approches sont ténus, ils restent cependant significatifs au regard du volume de la base de test, et se traduisent surtout dans les critères de fausse alerte et de faux rejet. Ainsi on remarque que les HMM réduisent fortement les taux de fausse alerte sur la parole et de faux rejet sur la musique, ce qui signifie que la part de trames de musique prises pour de la parole est fortement réduite par ce post-traitement. Les approches hybrides par segmentation aveugle sont caractérisées par la correction opposée, à savoir à la réduction du nombre de trames de parole prises pour de la musique, et donc des taux de fausse alerte en musique et de faux rejet en parole. La parole étant largement majoritaire dans le corpus, l'effet des approches par segmentation est donc plus efficace que celui du post-traitement par HMM, qui prend pourtant en compte la proportion des classes dans l'apprentissage de la matrice de transition $\boldsymbol{A}$ (voir section 8.3.1).

Confirmant les observations sur la figure 10.6, les métriques probabilistes dans l'espace RKHS (DIV et BAT) sont les plus efficaces et apportent un gain absolu de 0.5% environ sur le filtrage médian, soit une réduction relative de l'erreur de 12%.

10.5 Résultats à ESTER 2 et sur le corpus de Scheirer

Campagne ESTER 2

Nous avons également participé à la campagne d'évaluation ESTER 2, pour laquelle nous avons appliqué le système présenté dans la première expérience (sur ESTER 1) en exploitant la taxonomie A *Ch–*, c'est-à-dire un paradigme *one-vs-one* sur les trois classes de parole, musique et mix. Nous avons également exclu les exemples de chant des données d'apprentissage de la classe musique. Le nombre de descripteurs sélectionnés est ici fixé à $d = 50$, et la sélection exploite l'algorithme IRMFSP, nos recherches sur les algorithmes de sélection de descripteurs n'ayant pas encore abouti à l'époque de la campagne de test d'ESTER 2. Les résultats sur la tâche SES sont résumés dans le tableau 10.8, et proviennent en partie de la publication de clôture de la campagne [82].

Malheureusement dans les résultats publics de la campagne, seuls les taux d'erreur, de détection manquée (md), de fausse alerte (fa) et de F-mesure pour chaque classe ont été publiés. Les mesures de rappel et de précision ne sont pas disponibles, et la F-mesure globale, qui servait de critère de référence pour la campagne ESTER 1, et qui évaluait le compromis entre les détections de parole et de musique, ne fait plus partie du protocole d'évaluation, principalement parce que certains participants n'ont publié des résultats que sur la détection de parole (en raison de sa proéminence dans le corpus, et sur les autres tâches). Ainsi, l'IRIT est notre seul concurrent sur la tâche de détection de la musique. On remarquera enfin dans le tableau que les taux d'erreur ont également été renseignés sur les données propres à chaque station de radio du corpus.

Système	**Erreur(%)**					**md(%)**	**fa(%)**	**F**
	Africa	**Inter**	**RFI**	**TVME**	**Globale**			
Classe de parole								
IRISA	1,65	1,42	0,58	2,44	1,49	**0.37**	16.42	99,20
IRIT	2,05	0,85	0,65	2,47	1,31	0.72	9.28	99,29
LIMSI	2,55	**0,52**	**0,26**	**1,71**	**1,08**	0.80	**4.91**	**99,42**
RTL	**1,40**	1,10	0,61	2,07	1,23	0.50	11.01	99,34
Classe de musique								
IRIT	**6,63**	5,17	5,93	4,63	5,51	43.13	**0.77**	69,80
TPT/RTL	12,40	**2,95**	**3,92**	**4,10**	**5,25**	**12.56**	4.33	**78,85**

TABLE 10.8 – Comparaison des résultats des participants à la tâche SES de la campagne ESTER 2. Notre système est désigné par TPT/RTL (pour TELECOM ParisTech / RTL).

Les résultats que nous obtenons sur le corpus ESTER 2 (participant TPT/RTL pour TELECOM ParisTech / RTL) sont très proches de ceux mesurés sur ESTER 1, ce qui souligne la proximité entre les deux corpora. Le meilleur résultat sur la détection de la parole est obtenu

par le système du LIMSI, tant en termes de F-mesure que de taux d'erreur et de fausse alerte, ainsi que sur la majorité des stations de radio du corpus. On notera toutefois que les F-mesures des quatre participants sont assez proches (les écarts mutuels étant de l'ordre de 0.1%), et notre système obtient le résultat le plus proche du meilleur participant. Le taux de détections manquées est d'ailleurs inférieur, même si notre taux de fausse alerte est significativement plus élevé. Il est difficile cependant de comparer les deux systèmes, en raison de notre participation conjointe à la tâche de détection de musique. En effet, l'optimisation du système résulte d'un compromis entre trois classes, en comptant la classe mix, tandis que l'optimisation des résultats de détection de parole n'implique que deux classes (parole et non-parole) et simplifie ainsi le problème.

Les écarts de performances sur la tâche de détection de musique sont d'ailleurs beaucoup plus marqués que dans le cas précédent, et notre système marque sur le second participant une avance nette en termes de F-mesure (+9% en absolu, et 30% de réduction relative d'erreur) et de taux de détection manquée (65% de réduction relative). En contrepartie, l'IRIT présente un taux de fausse alerte largement réduit par rapport au notre (82% de réduction relative), ce qui semble indiquer que notre système modélise une classe de musique plus étendue que celle modélisée par l'IRIT. L'inversion de ce constat sur la classe de parole confirme cette intuition.

En définitive, il est difficile d'analyser plus en profondeur les résultats sur la campagne ESTER 2, en raison du manque de participants à la tâche de détection de musique.

Corpus de Scheirer

S'il existe peu de corpora publics pour l'évaluation de la classification parole/musique (à l'exception des campagnes d'évaluation), le corpus de Scheirer, que nous avons introduit dans la section 10.1.3, est l'un des plus cités et des plus repris dans la littérature. Ainsi, il constituera une troisième opportunité d'évaluer notre système, et par la même occasion, de le comparer aux publications internationales, contrairement à la portée des campagnes ESTER, qui reste nationale.

Toutefois il est difficile de comparer directement les résultats des auteurs en raison de divergences dans les protocoles d'évaluation suivis. La démarche fixée à l'origine par Scheirer et Slaney [207] consiste à diviser le corpus audio en un ensemble de test contenant 10% des 160 fichiers (soit 16 fichiers) et un ensemble d'apprentissage contenant les 90% restant. Les fichiers ne sont pas découpés entre les deux bases de manière à ne pas bénéficier des similarités au sein d'un même extrait. Le taux d'erreur est évalué en reproduisant un grand nombre de fois ce processus, le découpage étant aléatoire à chaque itération. Les auteurs publient principalement les résultats obtenus sur les trames d'une seconde (qui correspondent à notre trame long-terme), en mentionnant en plus les résultats calculés en étendant la fenêtre de décision à 2.5 s, de manière à reproduire le protocole qu'applique Saunders dans son article [205]. Williams et Ellis [247] poussent cette dernière idée à l'extrême en publiant les résultats obtenus en étendant la fenêtre de décision à l'ensemble de l'extrait de 15 s, mais ne publient malheureusement pas de résultats sur les trames d'une seconde, contrairement à Casagrande et al. [43], qui se limitent justement à cette échelle.

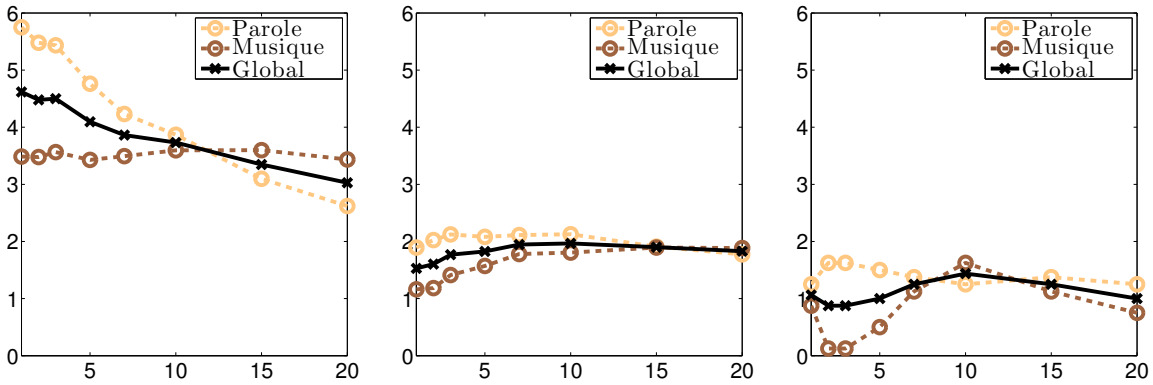
Fenêtre de décision			**Trame longue (1 s)**			**2.5 s**	**Tout 15s**
Système	**Dim**	**Ref**	**Parole**	**Musique**	**Global**	**Global**	**Global**
Saunders	-	[205]	-	-	-	2.0	-
Scheirer & Slaney	3	[207]	6.7±1.9	4.9±3.7	5.8±2.1	1.4	-
	8		6.2±2.2	7.3±6.1	6.7±3.3	-	-
Casagrande et al.	-	[43]	-	-	6.7	-	-
Williams & Ellis	3	[247]	-	-	-	1.3	0.0
	4		-	-	-	1.7	0.0
Ramona	1		5.7±3.9	3.5±2.5	4.6±2.3	1.5±2.2	1.0±2.3
	20		2.6±3.7	3.4±3.1	3.0±2.2	1.8±2.5	1.0±2.3

TABLE 10.9 – Comparaison des résultats de la littérature sur le corpus de Scheirer (à l'exception de Saunders qui est donné ici à titre indicatif), pour différentes longueurs de fenêtres de décision. L'écart type des résultats sur les itérations est précédé du symbole ±.

Le tableau 10.9 synthétise l'ensemble des résultats des auteurs mentionnés. Nous indiquons le résultat de Saunders, mesuré sur des fenêtres de décision de 2.5 s, bien que celui-ci ne soit pas

calculé sur le même corpus. Scheirer et Slaney, ainsi que Williams et Ellis, publient des résultats pour différentes combinaisons de descripteurs, en faisant varier le nombre de descripteurs sélectionnés. Nous indiquons ici les résultats les plus significatifs des deux auteurs. En revanche, dans la contribution de Casagrande et al. la sélection des descripteurs pertinents fait partie du processus de classification et ne constitue pas une variable du protocole expérimental. Le système que nous appliquons sur ce corpus est constitué d'un unique classifieur SVM, dont le noyau est sélectionné par minimisation du critère d'alignement, appris sur un sous-ensemble des descripteurs proposés, sélectionnés par l'algorithme SAS (minimisation de l'alignement sur noyau pondéré) que nous avons présenté dans la section 7.5.1. La taille restreinte du corpus est ici propice à l'application de cette méthode qui, bien qu'elle soit plus légère que ses concurrentes, reste très coûteuse sur un grand nombre d'exemples. Nous mentionnons dans le tableau les résultats obtenus avec le meilleur descripteur et avec les 20 meilleurs descripteurs. Nos résultats sont estimés sur 100 itérations.

Les résultats sur les trames d'une seconde sont les plus significatifs puisqu'ils valident sans ambiguïté la pertinence du système de classification employé ainsi que l'algorithme de sélection de descripteurs. En effet, avec un unique descripteur, notre système réduit l'erreur à 4.6%, ce qui représente un gain relatif de 20% sur l'erreur minimale de 5.8% de Scheirer de Slaney ; le système de Casagrande et al. ne présente qu'une erreur minimale de 6.7 %. L'erreur est d'ailleurs encore réduite si l'on augmente le nombre de descripteurs et descend jusqu'à 3% pour $d = 20$. La figure 10.7(a) montre l'évolution de l'erreur globale ainsi que pour chaque classe, par rapport au nombre de descripteurs sélectionnés. On observe que la décroissance de l'erreur globale est due à la décroissance de l'erreur sur la parole, tandis que la détection de la musique semble être essentiellement due au premier descripteur sélectionné, et évolue peu lorsque l'on augmente la dimension.



(a) 1 seconde (b) 2.5 secondes (c) 15 secondes

FIGURE 10.7 – Évolution de l'erreur globale et des erreurs de classes par rapport au nombre de descripteurs sélectionnés, pour notre système. Les trois figures correspondent aux différentes longueurs de fenêtres de décision impliquées dans le protocole expérimental.

Les figures 10.7(b) et 10.7(c) montrent également l'évolution des erreurs pour les fenêtres de décisions respectives de 2.5 s et de 15 s (c'est-à-dire la totalité des extraits). Si l'élargissement de la fenêtre de décision réduit sans surprise l'erreur, on constate par contre avec étonnement que l'augmentation de la dimension devient un handicap pour la classification. Ainsi on obtient de meilleurs résultats avec un seul descripteur (1.5%) sur les fenêtres de 2.5 s, qu'avec 20 descripteurs (1.8%), le premier résultat étant d'ailleurs légèrement inférieur mais très proche de ceux de Scheirer et Slaney, et de Williams et Ellis. L'homogénéisation de la décision sur l'ensemble de l'extrait réduit encore l'erreur mais l'évolution par rapport à la variation du nombre de descripteurs est encore moins intuitive. Williams et Ellis affichent une erreur nulle sur cette échelle, mais nous restons très sceptiques sur ce résultat, car il semble que celui-ci soit calculé sur une unique itération. Or, nous rencontrons de nombreuses itérations dans notre expérience où l'erreur est également nulle sur la fenêtre de 15 s.

10.6 Expérience 4 : Détection de voix chantée

Nous concluons cette partie expérimentale en appliquant le système décrit sur le problème auxiliaire de la détection de chant, afin de montrer que l'architecture en question peut s'adapter à différents problèmes de classification. Cependant, parce que les variations de classes sont beaucoup plus fréquentes sur ce problème (chaque pause dans le phrasé du chanteur induit une transition de musique instrumentale), nous nous restreignons ici à l'échelle temporelle courte (soit des trames de 32 ms, avec un pas d'avancement de 16 ms). Ceci limite donc considérablement le nombre de descripteurs mis en jeu puisque les résultats d'intégration temporelle ne sont pas exploitables ici. Nous employons tout de même les descripteurs long-termes en répétant leur valeur sur l'ensemble des trames courtes couvertes par chaque trame longue. On obtient ainsi 116 composantes pour caractériser les classes de chant et de musique instrumentale, parmi lesquelles on sélectionne les d plus pertinentes par l'algorithme IRMFSP. La classification se fait par une unique machine SVM apprise sur une base contenant 20000 exemples de chaque classe.

L'emploi d'une fenêtre de décision très courte implique nécessairement une plus forte variabilité de la sortie des SVM, puisque beaucoup moins d'information est exploitée pour le calcul de chaque descripteur. Ainsi, on peut constater sur la figure 10.8(a), que la probabilité a posteriori de la classe de chant (en bleu) est assez bruitée et oscille autour du seuil de décision de 0.5 (indiqué par une ligne grise). Ceci se traduit par une séquence estimée de classes (les points rouges) très instable, qui contient de très fréquentes erreurs accidentelles (l'annotation réelle est représentée en noir, immédiatement au-dessus et en dessous des estimations en rouge). Nous avons donc profité ici des techniques de post-traitement que nous avons introduites précédemment. La figure 10.8(b) montre le résultat du filtrage médian sur les probabilité a posteriori, et l'estimation déduite, sur laquelle on peut constater que quelques transitions accidentelles subsistent. Nous verrons qu'on obtient de meilleurs résultats encore en appliquant le post-traitement par HMM que nous avons proposé, et dont le résultat sur l'exemple précédent est illustré sur la figure 10.8(c). On remarque que même si les frontières des segments ne correspondent pas exactement à l'annotation réelle, la séquence de classes est beaucoup plus stable et les transitions correspondent mieux (à un décalage près) à la vérité terrain.

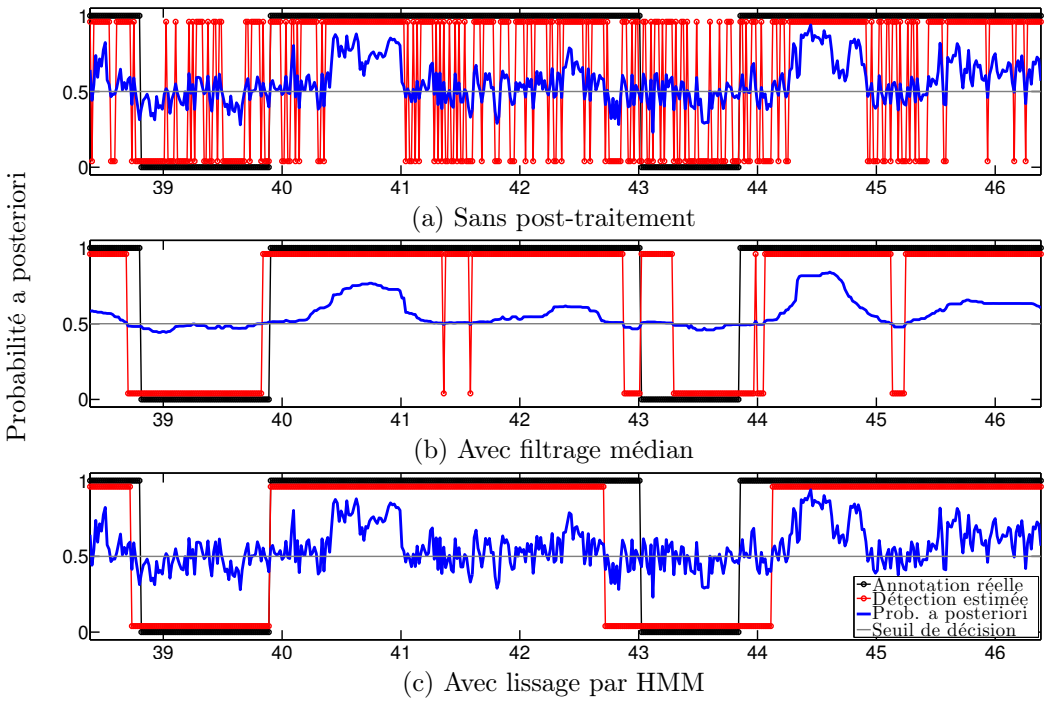


FIGURE 10.8 – Illustration de l'effet des post-traitements sur la probabilité a posteriori de la classe de chant. On constate que le filtrage médian (b) et le lissage par HMM (c) réduisent considérablement le nombre de transitions erronées par rapport à l'estimation de base (a).

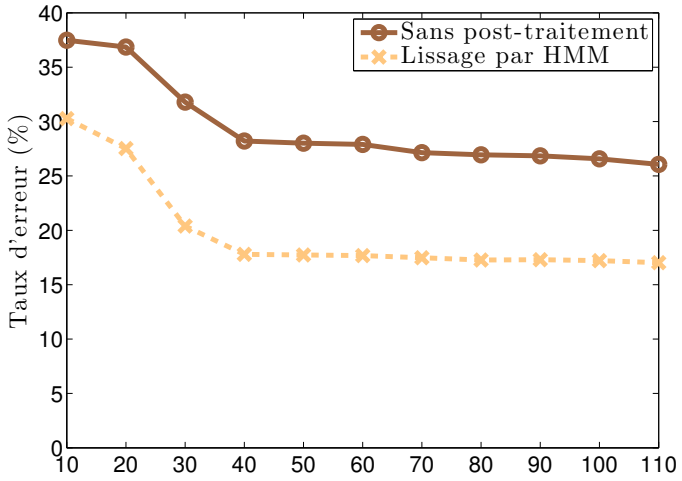


FIGURE 10.9 – Comparaison de l'évolution de la F-mesure sur le corpus Jamendo, avec ou sans post-traitement par HMM, en fonction du nombre de descripteurs d.

La figure 10.9 montre l'évolution de l'erreur de classification (calculée sur les trames) en fonction du nombre de descripteurs sélectionnés d, évoluant entre 10 et 110, en confrontant les résultats sans post-traitement (en ligne pleine marron) et après lissage par HMM (en pointillés jaunes). Le système est appris et testé sur le corpus Jamendo que nous avons introduit dans la section 10.1.4.

On constate que l'erreur décroît avec l'augmentation du nombre de composantes, mais reste assez stable au délà de $d = 40$, ce qui laisse supposer que l'essentiel de l'information discriminante est contenue dans les premiers descripteurs sélectionnés. Ainsi le taux de bonne classification de 82.2% pour $d = 40$ augmente légèrement jusqu'à 83.0% à $d = 110$, mais le premier cas constitue un meilleur compromis entre coût et performances ; nous fixons donc le nombre de descripteurs sélectionnés à $d = 40$ dans le reste de cette expérience.

Classe	**Chant**		**Musique**		**Global**	
Critère	**fr%**	**F%**	**fr%**	**F%**	**fr%**	**F%**
Sans post-traitement	74.8	72.6	68.5	70.4	71.8	71.6
Avec filtrage médian	84.6	81.6	76.4	80.5	80.7	81.1
Segmentation	88.0	84.8	74.0	80.3	81.3	82.7
Lissage par HMM	80.9	84.4	84.0	82.0	82.2	83.2
Vembu & Baumann [237]	87.7	70.5	35.8	47.4	62.7	62.4
Regnier & Peeters [196]	-	-	-	-	-	76.8

TABLE 10.10 – Comparaison du taux de bonne classification (fr%) et de la F-mesure (F) sur l'ensemble de test du corpus Jamendo, avec les différents algorithmes de post-traitement proposés, et confrontés à ceux de deux publications.

Le tableau 10.10 décrit plus en détail les résultats obtenus en termes de taux de bonne classification et de F-mesure, ainsi que sur chaque classe. Les performances sans pré-traitement et avec filtrage médian ou lissage par HMM, sont comparées. Nous avons également testé un post-traitement suivant l'approche hybride par segmentation aveugle, inspiré de celui proposé par Tsai et Wang [203], où les segments sont délimités par une procédure de détection d'attaques introduite par Duxbury et al. [67]. Nous avons également implémenté, à titre de comparaison, une autre approche par SVM tirée de la littérature [237], basée sur un vecteur de descripteurs de 38 composantes groupant les MFCC, les PLP et les LFPC. Nous avons conservé pour les paramètres du noyau RBF gaussien, les valeurs que l'auteur suggère. Les décisions de ce système sont basées sur des fenêtres de 190 ms, sur lesquelles seules les moyennes des valeurs des descripteurs sont considérées. Nous indiquons enfin le résultat en F-mesure globale de Regnier et Peeters [196], dont le système repose sur le seuillage empirique d'un unique descripteur basé sur une mesure combinée de vibrato et de trémolo, que nous avions évoqué dans la section 2.4.2 de l'état de l'art.

Notre système atteint 71.8% de bonne classification sans post-traitement, ce qui est largement supérieur aux résultats de l'approche de Vembu et Baumann, pourtant calculés sur des fenêtres de décision beaucoup plus conséquentes ; ceci montre la pertinence des descripteurs employés, ainsi que celle du processus de sélection, par rapport à un ensemble plus classique de descripteurs tirés de l'analyse de la parole. Le lissage par HMM offre ici les meilleurs résultats, par rapport aux deux autres post-traitements évalués, avec un taux de bonne classification de 82.2% et une F-mesure de 83.2%. On remarque toutefois que le traitement par segmentation est beaucoup plus efficace sur la classe de chant (88% contre seulement 80.9% pour les HMM), sans doute parce que celle-ci contient des attaques plus prononcées que le fond musical, qui favorisent donc le processus de segmentation. Les HMM, parce qu'ils appliquent un traitement distinguant les deux classes (chacune étant modélisé par un modèle GMM et des probabilités de transition propres), contrairement aux deux autres approches, appliquent en définitive le meilleur compromis entre les deux classes. On remarque d'ailleurs que l'on retrouve ce biais vers la classe de parole dans l'approche de Vembu de Baumann. Nous constatons que notre système se révèle plus efficace que celui proposé par Regnier et Peeters (qui post-traite également les décisions), même si celui-ci se distingue par l'étonnante efficacité de son unique descripteur.

Pour finir, le tableau 10.11 détaille les résultats sur quelques-uns des 16 fichiers audio du corpus de test, avec lissage par HMM. Bien que le système montre un léger avantage pour l'identification de la musique instrumentale, par rapport au chant, on remarque que c'est généralement la mauvaise identification de la musique qui fait chuter les résultats sur les pires fichiers (comme par exemple les numéros 4, 5, 7 et surtout 8). Après écoute des fichiers en question on constate que l'erreur provient d'un instrument particulier dont le timbre est proche de la voix. Ceci montre les limites de notre caractérisation de cette classe, dont l'extrême diversité peut impliquer des manifestations acoustiques très proches de la voix chantée.

Classe	**Chant**		**Musique**		**Global**	
Fichier audio	**fr%**	**F%**	**fr%**	**F%**	**fr%**	**F%**
1. À Poings Fermés.wav	84.5	92.2	98.7	95.7	93.7	94.6
2. 16 ans.wav	69.3	84.8	99.6	94.7	91.5	92.7
3. Une charogne.wav	87.1	91.7	79.4	72.3	85.3	86.8
4. Castaway.wav	94.4	87.3	55.1	73.4	79.0	82.8
5. Believe.wav	94.4	88.5	52.3	65.2	80.0	82.3
6. Si Dieu.wav	66.1	80.7	97.4	72.6	76.4	77.3
7. Elles disent.wav	76.1	78.7	63.6	59.9	71.8	72.1
8. L'Irlandaise.wav	93.8	64.2	33.4	47.5	57.7	57.1
Tous	80.9	84.3	84.0	81.8	82.2	83.2

TABLE 10.11 – Résultats détaillés de la détection de chant sur quelques-uns des fichiers de l'ensemble de test du corpus Jamendo.

Chapitre 11

Implémentation

Cette thèse CIFRE est avant tout le fruit d'une collaboration entre un institut de recherche et une entreprise. Ainsi, le résultat le plus tangible pour RTL reste la livraison du système de classification audio en C++.

Comme dans beaucoup de domaines liés à l'indexation, l'apprentissage des systèmes de classification constitue la part la plus importante du code développé. Cette partie du code comprend le découpage en trames du flux audio, l'importation des annotations de classes, le calcul des descripteurs, leur sélection automatique, la paramétrisation des noyaux, ainsi que l'apprentissage des machines SVM impliquées dans la taxonomie de classification définie. Tout le processus d'apprentissage a été livré à l'entreprise sous la forme de code Matlab. Il est le fruit de nos travaux de recherches et offre ainsi une collection fournie d'algorithmes, propre au travail de l'expérimentateur.

Toutefois, l'entreprise doit pouvoir disposer d'un programme « clé en main » qui ne nécessite pas la maîtrise technique de ses divers rouages. Ainsi, le processus de classification sur des données audio inconnues constitue la part la plus importante de notre travail pour RTL. C'est cette seconde partie qui a été développée en C++ et livrée à l'entreprise. De manière à simplifier son usage, nous avons réduit le champ des possibilités pour offrir un système fiable fonctionnant en ligne sur un flux audio. Tous les paramètres du système appris sont synthétisés dans un ensemble de fichiers textes réunis dans un même dossier, et automatiquement générés par le système d'apprentissage en Matlab.

Ainsi le système de classification inclut les phases suivantes :

- **Découpage du flux audio** : le système peut travailler sur un fichier audio ou sur le flux sortant de l'interface *Jack* (`http://jackaudio.org/`). Cette dernière permet, sur tout type de plateforme, d'interconnecter n'importe quelles applications audio. Ainsi le système peut directement analyser le flux audio sortant d'une application de *streaming* audio comme VLC. Le découpage en trames suit le processus que nous avons décrit dans la section 6.2. Les paramètres N, R, N_{mul} et R_{mul} sont fournis au système via un fichier de configuration.
- **Extraction des descripteurs** : pour chaque classifieur SVM nécessaire, un fichier de configuration précise la liste des descripteurs nécessaires, parmi ceux que nous avons présentés dans la section 6.5. Le système réunit l'ensemble de ces informations et calcule ainsi dans un processus commun les descripteurs nécessaires à l'ensemble des classifieurs, puis réordonne ceux-ci afin de fournir à chaque classifieur la liste associée.
- **Classification par SVM** : les modèles SVM, appris par l'implémentation *SVMlight* de Joachims [114] et transmis dans les fichiers de configuration, sont exploités pour appliquer le processus de classification sur les vecteurs de descripteurs.
- **Calcul des probabilités a posteriori** : un fichier de configuration décrit la manière dont les différents classifieurs sont agencés dans la taxonomie de classification. Ainsi le système applique l'algorithme proposé dans la section 5.1.5.4 pour le calcul des probabilités a posteriori à partir des résultats des SVM d'un arbre de classification hiérarchique.
- **Filtrage médian** : de manière à répondre à la contrainte de classification en ligne (c'est-à-

dire avec un temps de réponse contrôlé), seul le post-traitement par filtrage médian (présenté dans la section 8.2) est implémenté dans le système en C++. Le système renvoie finalement sur la sortie standard, pour chaque trame, le temps associé, la classe estimée, ainsi que le vecteur des probabilités lissées, sous la forme suivante :

```
CLASSES : "mix music speech "
TIMESTAMP: 0 - CLASS: "speech" - PROB: (0.0267974, 0.0811253, 0.892077)
TIMESTAMP: 0.48 - CLASS: "speech" - PROB: (0.0758417, 0.0811253, 0.892077)
TIMESTAMP: 0.96 - CLASS: "speech" - PROB: (0.0413623, 0.0799202, 0.892077)
TIMESTAMP: 1.44 - CLASS: "speech" - PROB: (0.0316471, 0.0259504, 0.892077)
TIMESTAMP: 1.92 - CLASS: "speech" - PROB: (0.0316471, 0.0259504, 0.844238)
TIMESTAMP: 2.4 - CLASS: "speech" - PROB: (0.0332106, 0.0259504, 0.805994)
TIMESTAMP: 2.88 - CLASS: "speech" - PROB: (0.260817, 0.0259504, 0.559638)
TIMESTAMP: 3.36 - CLASS: "mix" - PROB: (0.156032, 0.0259504, 0.00116314)
TIMESTAMP: 3.84 - CLASS: "music" - PROB: (0.156032, 0.842805, 0.00116314)
TIMESTAMP: 4.32 - CLASS: "music" - PROB: (0.156032, 0.933296, 0.00116314)
TIMESTAMP: 4.8 - CLASS: "music" - PROB: (0.156032, 0.848773, 0.00116314)
TIMESTAMP: 5.28 - CLASS: "music" - PROB: (0.154462, 0.848773, 0.00116314)
```

La sortie du système est volontairement minimaliste et exclusivement basée sur la sortie texte, de manière à pouvoir aisément appliquer tout type d'interface graphique, indépendamment de la plateforme et du langage de programmation employé.

Plusieurs modèles appris en Matlab ont été livrés à l'entreprise (sous forme de fichiers de configuration) pour appliquer en situation réelle le système de classification, dont un dédié à la tâche de classification parole/musique/parole+musique définie dans la campagne ESTER, et un autre pour la détection de voix chantée dans un titre musical.

Nous ferons la démonstration de ce système lors de la soutenance de thèse.

Chapitre 12

Conclusion

Bilan de cette thèse

Nous avons traité dans cette thèse la question de la classification audio en nous concentrant sur l'application des machines à vecteurs de support et en tentant de répondre à des questions peu abordées dans le domaine de l'indexation audio. En effet, l'approche par apprentissage statistique suppose la conjonction de nombreux éléments (méthode de classification, descripteurs audio, sélection de descripteurs et post-traitements dynamiques) dont nous avons vu que la plupart sont étroitement liés au fonctionnement des SVM.

Nous avons pu constater, en présentant la théorie des SVM, que ces derniers sont équivalents à l'application d'une simple séparation linéaire, après une transformation sur les exemples, entièrement décrite par la fonction noyau. Ainsi, la matrice de Gram, qui décrit exhaustivement l'effet de la fonction noyau sur les exemples d'apprentissage, permet de caractériser très précisément le noyau. Nous avons donc introduit, dans le chapitre 4, plusieurs critères d'évaluation du noyau basés sur la matrice de Gram, dont nous avons montré que les performances sont comparables à l'état de l'art, pour un moindre coût en temps de calcul. Nous avons de plus proposé une amélioration au critère d'Alignement en introduisant la composante d'ajustement des matrices de Gram pour la prise en compte du facteur d'erreur C.

La sélection automatique de descripteurs est également gouvernée par le choix de la méthode de classification. Ainsi nous avons vu que de nombreuses méthodes ont été présentées dans la littérature, qui prennent en compte l'effet de la fonction noyau. Nous avons donc proposé dans le chapitre 7 plusieurs algorithmes de sélection de descripteurs exploitant les critères d'évaluation du noyau précédemment introduits, pour déterminer, à travers la matrice de Gram, la pertinence conjointe des descripteurs impliqués. Une expérience sur des données réelles a confirmé la pertinence des méthodes introduites par rapport aux méthodes existantes, tant du point de vue des performances de classification que du temps de calcul.

L'emploi de méthodes automatiques de sélection se justifie par le fait que la notion de pertinence est très complexe et ne peut être jugée indépendamment sur chaque descripteur. Alors que la plupart des auteurs comparent directement les performances du classifieur sur chaque groupe de descripteurs, l'automatisation du processus nous permet d'introduire une large collection de descripteurs, dont la plupart sont populaires dans la littérature pour leurs propriétés discriminantes entre parole et musique.

Le formalisme des SVM nous a donc permis de déployer des critères d'évaluation, dont l'expression dans l'espace transformé est fondée sur de simples critères de séparabilité. Ceci n'est pas possible avec les méthodes génératives, tels les modèles de mélanges de gaussiennes, et justifie ainsi notre choix de nous concentrer dans cette thèse sur les SVM. Cependant, les machines à vecteurs de support sont basées sur un paradigme essentiellement discriminatif et nous ont donc obligé à étudier les méthodes d'extension sur des problèmes à plus de deux classes. La comparaison des approches de la littérature nous a permis de définir, dans le chapitre 5, un cadre hybride combinant les méthodes *one-vs-one* et hiérarchique, sur lequel nous avons proposé une méthode d'estimation des probabilités a posteriori, basée sur la pondération itérative des sorties de SVM,

après transformation sigmoïdale.

Ce résultat est essentiel puisqu'il nous a permis d'introduire l'application de techniques de post-traitement sur les résultats de classification. Nous avons en ce sens proposé deux nouvelles approches qui corrigent les erreurs accidentelles de classification, en prenant en compte les résultats des trames voisines. La première, introduite dans le chapitre 8, est basée sur un modèle HMM dont les observations sont les probabilités a posteriori et les états sont les classes du problème. Nous avons de plus proposé une manière d'exploiter les HSMM comme post-traitement, afin de mieux modéliser la durée des segments. Toutefois cet apport théorique n'a pas apporté en pratique les résultats escomptés. La seconde approche exploite le résultat d'une segmentation aveugle par détection de rupture, pour homogénéiser la classe détectée sur chaque segment délimité. Plusieurs métriques récentes de la littérature ont été exploitées pour la détection de rupture, essentiellement basées sur les SVM et les méthodes à noyaux.

Ces dernières contributions ont été validées dans le chapitre 10, sur le corpus de la campagne d'évaluation nationale ESTER, pour la tâche de classification parole/musique. Nous avons dans un premier temps comparé différentes taxonomies de classification pour mettre en avant l'avantage de la méthode *one-vs-one* sur les approches hiérarchiques évaluées ; nous avons également montré que les exemples de chant présents dans la classe de musique pénalisent les résultats de classification. Nous avons ensuite confirmé la pertinence des algorithmes de post-traitement proposés en comparant les résultats sur le corpus ESTER, par rapport au simple filtrage médian. Ces résultats ont fait apparaître la supériorité des métriques de détection de rupture basées sur les distances probabilistes dans l'espace RKHS. Par ailleurs, la comparaison aux résultats des participants de la campagne ESTER a montré que le système proposé produit les meilleurs résultats sur cette tâche, sur un nombre de descripteurs plus restreint.

Toutefois, l'antériorité de la campagne ESTER par rapport à nos travaux limite la portée de ces résultats. Ainsi, notre participation durant la thèse à la campagne d'évaluation ESTER 2 apporte la confirmation de l'efficacité du système proposé, comme le montrent les résultats que nous présentons dans la section 10.5 du chapitre précédent. Ainsi notre approche apporte un gain significatif par rapport aux autres participants sur la détection de musique, sans pour autant perdre en efficacité sur la reconnaissance de parole. Une dernière expérience sur le corpus public de Scheirer, étend la portée de nos résultats au delà du cadre des participants français aux campagnes ESTER.

Le soin apporté à la mise en place d'un système d'apprentissage statistique fiable et efficace est en grande partie indépendant de la tâche d'application. Nous avons ainsi eu l'occasion d'étendre le cadre proposé en proposant également une expérience sur la détection du chant dans la musique. Nous avons pour cela constitué et annoté un corpus public suffisamment conséquent pour le besoins de l'expérience. Une étude comparative à d'autres algorithmes de la littérature a montré la supériorité de notre système pour la tâche de détection du chant.

Enfin, le chapitre 11 nous a permis d'introduire l'implémentation pratique en C++ du système de classification audio que nous avons livré à RTL à la fin de cette thèse.

Perspectives

Nous dégageons de nos travaux de thèse plusieurs perspectives poursuivant certains aspects de nos recherches que nous avons présentés, ou bien étendant le domaine d'application du problème posé.

Nous avons dégagé de l'étude des critères d'alignement et de séparabilité des classes plusieurs algorithmes de sélection de descripteurs, qui constituent des alternatives fiables et moins coûteuses aux approches de l'état de l'art. Pourtant, leur application sur des bases conséquentes restent prohibitive, en raison de la complexité quadratique par rapport au nombre d'exemple, et de la taille des structures mises en jeu, qui oblige à recalculer de nombreuses valeurs ne pouvant être conservées en mémoire. Un progrès sur ce point représenterait donc une avancée essentielle qui permettrait la généralisation de l'emploi de ces techniques sur des problèmes complexes.

De plus, nous avons constaté que l'optimisation basée sur le critère KCS (mesurant la séparabilité des classes dans l'espace transformé) est problématique puisqu'elle converge vers une solution

triviale qui n'est pas pertinente en pratique. La procédure de régularisation que nous avons employée corrige en partie ce défaut mais ne suffit malheureusement pas à stabiliser la procédure d'optimisation de manière fiable. La régularisation des critères d'optimisation est un domaine largement couvert par la littérature que nous n'avons pas eu le temps d'explorer et qui apporterait sans doute des réponses à ce problème.

Un aspect plus technique à développer concerne l'exploration plus en profondeur des variantes des SVM. En effet, le domaine est actuellement en plein essor et en progrès constants. Par exemple il existe actuellement des techniques d'apprentissage extrêmement rapides (en complexité linéaire par rapport au nombre d'exemples) pour les SVM à noyaux linéaires. Le gain radical en complexité permet ainsi d'apprendre les systèmes sont des bases beaucoup plus conséquentes, qui peuvent compenser les moindres performances des noyaux linéaires par rapport au noyau RBF gaussien par exemple. L'apprentissage semi-supervisé, qui consiste à introduire en plus dans l'apprentissage un grand nombre d'exemples dont la classe est inconnue, permet d'introduire une quantité virtuellement infinie d'information supplémentaire dans la modélisation, puisque l'annotation n'est plus nécessaire pour exploiter un corpus audio.

Nous avons également mentionné le problème des classes mixtes, qui reste la source d'erreur principale sur la tâche de classification audio. Nous avons entrepris des recherches pour le développement d'un nouveau descripteur basé sur la détection du pitch prédominant afin d'estimer la puissance du spectre résiduel (après soustraction des partiels du pitch). Ainsi on détecte sur les zones de parole voisée dominante la présence d'une autre source acoustique. Néanmoins, cette approche est très coûteuse et ne se révèle efficace que dans les cas où la musique en fond est limitée en fréquences. En effet, si celle-ci s'étale sur un spectre plus large que la parole, on détecte facilement sa présence par l'estimation de la puissance sur les bandes de hautes fréquences. De plus, la définition de la classe de parole sur fond musical reste très hétérogène dans les corpus disponibles puisque les rapports signaux à bruit des deux sources ne sont pas considérés. Ainsi on réunit des exemples où la musique est prédominante à des exemples où elle est quasiment inaudible. Un contrôle rigoureux de ce paramètre apporterait une clarification nécessaire sur le problème étudié. Enfin, outre le fond musical, le bruit de fond constitue également une source de confusion significative dans l'identification de la parole. La réduction de bruit est un domaine à part entière du traitement de la parole, et son application préalable sur les signaux audio pourrait épurer les régions de parole. Ainsi, l'apprentissage et le test s'appuieraient sur des classes mieux définies, et donc plus clairement identifiables.

Nos recherches sur les techniques de post-traitement sont également sujettes à de possibles améliorations. En particulier, les HSMM sont supposés améliorer la modélisation de la séquence d'observations, par rapport aux HMM. Nos expériences n'ont pour l'instant pas dégagé de gain significatif lié à l'emploi de ce modèle, mais nous espérons néanmoins pouvoir en tirer parti dans des expériences futures.

La prise en compte de la dimension temporelle ouvre d'ailleurs d'autres perspectives qui dépassent largement le cadre exploré dans ce document. En premier lieu nous avons constaté par exemple, à l'écoute des extraits du corpus ESTER, que la structure sémantique des bulletins d'information (*jingle* d'introduction - présentation des titres - détail des sujets - *jingle* de transition - etc.) est très caractéristique et pratiquement figée. Ainsi l'apprentissage et la détection de cette structure apporterait une information supplémentaire sur la probabilité d'irruption d'une classe donnée. Sous un angle méthodologique différent, la détection d'événements clés connus dans une base (comme les *jingles*, les génériques, ou les publicités) permet également de structurer le signal d'un point de vue sémantique et d'en tirer une information pertinente. Par exemple la détection d'un générique particulier indique qu'une émission a débuté, qui impliquerait des modèles adaptés spécifiquement à son contenu. La détection de rires ou d'applaudissements peut également resserrer le champ des émissions possibles, et donc des modèles à employer. Cependant une adaptation du système aux spécificités locales d'une grille de programme nécessiterait un corpus autrement plus conséquent que ceux que nous avons considérés jusque-là.

Annexes

Annexe A

Estimation du rayon R

Nous avons vu que le rayon minimal R de la sphère incluant tous les exemples dans l'espace transformé est une donnée essentielle des machines à vecteurs de support, en premier lieu parce qu'il détermine directement la dimension VC de l'espace transformé (équation 3.19), dont dépend le risque structurel (équation 3.17). Le critère rayon-marge, présenté dans la section 4.3.6, et qui a pour expression $R^2\|\boldsymbol{w}\|^2$, constitue donc une borne naturelle à l'erreur de généralisation.

Nous avons par ailleurs considéré, dans la section 4.5.1, la valeur proposée par Joachims pour estimer la valeur optimale du facteur d'erreur C (équation 4.20) qui dépend directement du rayon R.

Nous présentons dans cette annexe les méthodes proposées par les auteurs de ces deux critères pour estimer le rayon R.

A.1 Minimisation de Vapnik

Weston et al. emploient pour calculer la borne rayon-marge [245] l'algorithme proposé par Vapnik [230], qui consiste à résoudre le problème de minimisation suivant :

$$R^2 = \max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \beta_i k(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i,j=1}^{n} \beta_i \beta_j k(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{A.1}$$

sous les contraintes

$$\begin{aligned} \sum_i \beta_i &= 1 \\ \beta_i &\geq 0, \qquad i = 1, \ldots, n \end{aligned}$$

où l'on a introduit les n variables β_i (avec $\boldsymbol{\beta} = [\beta_1, \ldots, n]$).

La procédure de minimisation fait appel à des techniques d'optimisation classiques que nous n'abordons pas ici. Elle est toutefois très coûteuse et peut avantageusement être remplacée par l'approche suivante, proposée par les auteurs dans la toolbox *Spider* [2].

A.2 Approximation moyenne

On approxime le rayon par la valeur suivante :

$$R^2 = \frac{1}{n} \sum_{i=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_i) - \frac{1}{n^2} \sum_{i,j=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_j),$$

qui est basée sur la norme moyenne des exemples dans l'espace transformé et le produit scalaire moyen entre tous les couples d'exemples, et revient à employer dans la formule A.1 les valeurs $\beta_i = \frac{1}{n}$.

A.3 Approximation de Joachims

La formule de Joachims (qu'il n'a pas publié à notre connaissance) est destinée à estimer le rayon $\bar{R}$ moyen entre les exemples et l'origine dans l'espace transformé. Etant donné qu'il est impossible d'exprimer analytiquement l'origine de l'espace transformé, celui-ci est approximé par l'image de l'origine de l'espace d'entrée, soit :

$$\begin{aligned} \bar{R} &= \sum_{i=1}^{n} ||\phi(\boldsymbol{x}_i) - \phi(\mathbf{0})|| \\ &= \sum_{i=1}^{n} \sqrt{k(\boldsymbol{x},\boldsymbol{x}) - 2\,k(\boldsymbol{x},\mathbf{0}) + k(\mathbf{0},\mathbf{0})} \end{aligned}$$

Annexe B

Bases de données pour l'évaluation

Cette annexe décrit les bases de données secondaires utilisées pour la validation d'algorithmes liés à la classification, en particulier pour la comparaison pratique des critères d'évaluation de noyaux présentée dans la section 4.6, et l'évaluation des algorithmes de sélection de descripteurs proposés par rapport aux méthodes existantes, dans la section 7.7. Nous avons exploité à cet effet quatre bases qui dépassent le contexte de la classification audio en introduisant d'autres sources de données et qui reposent sur un cadre strict de « sac d'exemples » absolument indépendants entre eux.

B.1 Spambase & Ionosphere

Les bases *Spambase* et *Ionosphere* proviennent toutes deux du dépôt public UCI [16], destiné à offrir à la communauté de l'apprentissage statistique des bases de données libres de droits et décrivant de multiples modalités expérimentales. La notoriété de ce dépôt permet en outre de comparer objectivement les résultats des divers auteurs sur une tâche commune. On trouvera une description détaillée sur le dépôt UCI à l'adresse suivante : `http://archive.ics.uci.edu/ml/`. Nous nous sommes ici cantonnéd à deux bases parmi les bases à deux classes sur des descripteurs à valeurs réelles.

La base *Spambase* décrit un problème de détection d'e-mails de *spam* (ou pourriel, en employant le vocable québécois). Les e-mails « sains » sont tirés de messages personnels. Chacun des 4601 exemples est caractérisé par 57 descripteurs numériques réels ou entiers décrivant des fréquences d'occurrences de certaines mots, caractères ou acronymes.

La base *Ionosphere* décrit un problème de classification entre de « bonnes » images où apparaît quelque indice de la présence d'un certain type de structure dans l'ionosphère, et de « mauvaises » images où cette structure n'est pas observable, parmi un ensemble de 351 exemples. 24 attributs réels d'autocorrélation ont été calculés sur des pulsations d'antennes à hautes fréquences (chaque valeur complexe est décomposée en deux paramètres réels).

B.2 Lymphoma

L'analyse de données de puces à ADN [1] est un domaine particulier de la bioinformatique qui se distingue par l'exploitation de bases très réduites en nombre d'exemples (puisque ceux-ci proviennent de cas cliniques) caractérisées par un très grand nombre de descripteurs basées sur le code génétique. La nécessité d'identifier parmi cette collection de gènes, ceux qui ont une influence sur le phénomène observé est d'ailleurs en grande partie à l'origine de l'essor des techniques de sélection automatique de descripteurs.

Le problème décrit par la base contient 96 exemples caractérisés par 4026 descripteurs exprimant le code génétique, et répartis entre les cas sains et les cas malins (au sens médical) manifestant la présence d'un lymphome des cellules B.

1. *Microarray* en anglais, on trouve également le terme de « microréseau d'ADN » en français.

Cette base, dans le contexte de la sélection de descripteurs, permet d'évaluer la fiabilité des algorithmes en présence de nombreux descripteurs fortement redondants. Elle nous permet en outre de reproduire l'expérience décrite par Weston et al. dans [244], qui ont mis la base à disposition sur leur site.

B.3 Parole / musique

Nous avons également inclus une quatrième base construite à partir des données que nous exploitons en classification audio. Celle-ci traduit un simple problème de discrimination entre parole pure et musique pure. Les exemples ont été calculés à partir de trames tirées aléatoirement dans le corpus de la campagne d'évaluation ESTER 1 , que nous présentons en détail dans la section 10.1.1.

321 descripteurs ont été extraits à partir de la collection présentée en détail dans l'annexe B pour caractériser les trames audio. Cette dernière base se distingue des autres par la quantité très conséquente d'exemples disponibles (en effet le nombre de trames d'une seconde dans un corpus de 90 heures est de l'ordre de 324 000), ce qui nous permet de construire une base contenant 10000 exemples pour chaque classe, de manière à couvrir le plus précisemment possible la complexité des distributions en jeu.

Nous avons également mis cette base à disposition publique à l'adresse suivante : `http://perso.telecom-paristech.fr/~ramona/kfs/`.

Annexe C

Descripteurs audio pour la classification

Cette annexe présente plus en détail la collection de descripteurs brièvement introduits dans la section 6.5.

C.1 Descripteurs spectraux

C.1.1 Moments statistiques spectraux (SM)

Cette série de descripteurs est basée sur le barycentre et les moments d'ordre i d'un modèle probabiliste du spectre, définis par :

$$\begin{aligned} \mu &= \int x^T p(x)\, dx \\ \mu_i &= \int (x-\mu)^i p(x)\, dx \quad \forall\, i > 1, \end{aligned}$$

où x représente les données observées (les fréquences f_k), et $p(x)$ la probabilité d'observer x (les amplitudes spectrales normalisées $\bar{a_k} = a_k / \sum_{k=0}^{K-1} a_k$). On peut donc estimer numériquement ces grandeurs par :

$$\begin{aligned} \mu &= \sum_{k=0}^{K-1} f_k \bar{a_k} \\ \mu_i &= \sum_{k=0}^{K-1} (f_k - \mu)^i \bar{a_k} \quad \forall\, i > 1. \end{aligned}$$

Les quatre premiers moments spectraux sont exploités pour caractériser le spectre :

Centroïde spectral

Le centroïde spectral permet de mesurer l'équilibre entre fréquences basses et hautes, et de juger ainsi de la brillance d'un spectre, un signal musical étant en général beaucoup plus chargé en hautes fréquences qu'un signal de parole.

$$S_c = \mu.$$

Largeur spectrale

La largeur spectrale peut être assimilée à une variance spectrale, elle décrit l'étalement du spectre autour de son barycentre. Cette statistique du second ordre permet d'estimer la largeur de

bande du spectre, le spectre musical couvrant une bande de fréquence beaucoup plus large que le spectre de parole.

$$S_w = \sqrt{\mu_2}.$$

Asymétrie spectrale

L'asymétrie spectrale (ou *Skewness*) est basée sur le moment de 3^e ordre. Elle est négative lorsque le spectre présente plus d'énergie dans les fréquences supérieures à son barycentre, positive dans le cas inverse, et nulle lorsque le spectre est symétrique.

$$S_a = \frac{\mu_3}{S_w^3}.$$

Platitude spectrale (Kurtosis)

La platitude spectrale (ou *Kurtosis*), statistique du 4^e ordre, quantifie la platitude du spectre. Le spectre est quasiment plat lorsque celle-ci tend vers 0, et très « piqué » autour de sa moyenne lorsque $S_a \to \infty$. La valeur $S_a = 3$ correspond à un profil spectral quasi-gaussien.

$$S_a = \frac{\mu_4}{S_w^4}.$$

C.1.2 Descripteurs MPEG-7

La norme MPEG-7 fournit un standard [3] pour la spécification de méta-données associées aux contenus multimédias (texte, image, audio ou vidéo...). Bien que le standard ne spécifie en théorie que l'encodage et la structuration de l'information, de nombreux descripteurs sont proposés et normalisés pour chacune des modalités. Deux d'entre eux sont exploités dans le cadre de cette étude, que nous présentons ci-dessous.

Rapport spectral

On trouve dans les spécifications MPEG-7 une alternative à la mesure de platitude spectrale présentée précédemment. Celle-ci est basée sur le rapport entre les moyennes géométrique et arithmétique des amplitudes spectrales, ce rapport étant compris entre 0 et 1 puisque la moyenne géométrique de termes positifs est toujours inférieure à la moyenne arithmétique :

$$S_r = \frac{\prod_k a_k^{1/K}}{\frac{1}{K}\sum_k a_k}.$$

Le rapport spectral tend vers 1 lorsque le spectre est plat. Àl'inverse, lorsque le spectre est harmonique (c'est-à-dire lorsqu'il présente de forts pics), le rapport spectral tend vers 0.

Platitude d'Amplitude Spectrale (ASF)

La spécification MPEG-7 conseille également d'affiner ce descripteur en calculant le rapport spectral sur des sous-bandes fréquencielles d'une tierce de largeur. Le descripteur ASF (*Amplitude Spectral Flatness*) regroupe donc 23 composantes associées à chacune des sous-bandes. Soit :

$$ASF(B) = \frac{\prod_{k\in B} a_k^{1/K}}{\frac{1}{K}\sum_{k\in B} a_k}.$$

Facteur de Crête Spectrale (SCF)

Peeters propose également [178] une mesure de platitude spectrale, basée sur le *Facteur de Crête Spectrale* (*Spectral Crest Factor*). Ce descripteur ne fait pas partie du standard MPEG-7, mais sa définition reprend la répartition en 23 sous-bandes fréquentielles introduite pour l'ASF. Le SCF est défini, sur la sous-bande de fréquences B, comme le rapport entre la valeur maximale et la moyenne des amplitudes :

$$SCF(B) = \frac{\max_{k\in B} a_k}{\frac{1}{K}\sum_{k\in B} a_k}.$$

C.1.3 Pente spectrale

La pente spectrale S_s représente le taux de décroissance spectrale. Elle est estimée par régression linéaire sur les amplitudes spectrales : $\hat{a}(k) = S_s \cdot f(k) + c$. En appliquant la méthode des moindres carrés, on déduit l'expression suivante :

$$S_s = \frac{K \sum_k f_k a_k - \sum_k f_k \sum_k a_k}{K \sum_k f_k^2 - \left(\sum_k f_k\right)^2}.$$

C.1.4 Décroissance spectrale

Ce descripteur mesure la décroissance des amplitudes spectrales ; il est basé sur des études perceptives et traduit donc la perception humaine. La mesure de décroissance spectrale apporte une information essentielle pour la reconnaissance du signal de parole qui est connu pour présenter une pente décroissante de 12 dB par octave.

Il a pour expression :

$$S_d = \frac{1}{\sum_{k=1}^{K-1} a_k} \sum_{k=1}^{K-1} \frac{a_k - a_0}{k}.$$

C.1.5 Fréquence de coupure

La fréquence de coupure F_c est définie comme la fréquence en deçà de laquelle 95% de l'énergie du spectre est contenue. Tout comme la mesure de platitude spectrale, elle est directement liée à l'allure harmonique du spectre. De par la décroissance naturelle des harmoniques hautes, un spectre plat aura sa fréquence coupure plus élevée qu'un spectre harmonique. On la définit donc indirectement comme suit :

$$\sum_{f=0}^{F_c} a^2(f) = 0.95 \sum_{f=0}^{f_s/2} a^2(f).$$

C.1.6 Flux spectral

Scheirer et Slaney proposent [207] un descripteur de mesure de variation spectrale entre deux trames consécutives, appelé flux spectral, basé sur le constat que le spectre de musique varie plus rapidement qu'un spectre de parole. On retrouve ce descripteur dans l'article de Peeters [178] sous la forme reprise ici, basée sur une mesure de corrélation entre les amplitudes des trames d'indice $t-1$ et t :

$$S_v = 1 - \frac{\sum_k a_k(t-1) a_k}{\sqrt{\sum_k a_k(t-1)^2} \sqrt{\sum_k a_k(t)^2}},$$

où $a_k(t)$ est l'amplitude du $k^{\text{ème}}$ *bin* de fréquence de la trame d'indice temporel t.

C.1.7 Linear Prediction Coding (LPC)

L'analyse LPC est un outil courant dans le domaine du codage audio. Elle consiste à modéliser le signal audio par un filtre linéaire à réponse impulsionnelle finie (filtre RIF), soit, pour un filtre à P composantes :

$$x(n) = \sum_{p=1}^{P} a_p \, x(n-p).$$

Nous ne détaillons pas ici l'algorithme classique de détermination des coefficients du filtre. On interprète généralement ces derniers dans le domaine spectral comme déterminant l'enveloppe du spectre, modulant un train d'impulsion en entrée du filtre lors de la synthèse.

Nous exploitons les deux coefficients LPC d'un modèle à deux composantes comme descripteurs pour caractériser l'allure du spectre. On sait néanmoins que ces coefficients sont très instables et ne varient pas de façon continue avec le spectre.

C.1.8 Sous-bandes en octaves

Essid propose pour la reconnaissance d'instruments de musique [72] deux nouveaux descripteurs destinés à capturer la structure spectrale de sons instrumentaux. Ceux-ci sont basés sur un banc de filtres triangulaires d'une octave de largeur avec recouvrement d'une demi-octave. La largeur d'une octave est destinée à offrir une modélisation adaptée à la structure des partiels harmoniques. Le banc comporte 10 filtres s'échelonnant de la note de piano la plus basse (La0, 27.5 Hz) à la fréquence de Nyquist.

Intensité de signaux de sous-bandes en octaves

Le premier descripteur, nommé OBSI (*Octave Band Signal Intensities*), mesure les log-énergies de chaque sous-bande du banc décrit.

Rapports d'intensité de signaux de sous-bandes en octaves

Le second descripteur, nommé OBSIR (*Octave Band Signal Intensities Ratios*), mesure la différence des valeurs OBSI de filtres consécutifs :

$$OBSIR_k = OBSI_{k+1} - OBSI_k.$$

Il décrit donc le logarithme du rapport d'énergie de chaque paire de sous-bandes consécutives.

C.1.9 Modulation d'Amplitude (AM)

Une série de descripteurs ont été proposés par Martin [149] puis repris par Eronen [70] pour la tâche de reconnaissance d'instruments de musique. Ils reposent sur la caractérisation des phénomènes de *trémolo* et de *rugosité*, qui se traduisent respectivement par une modulation de l'amplitude spectrale observée dans les bandes de fréquences entre 4 et 8 Hz et entre 10 et 40 Hz. L'amplitude spectrale est calculée sur l'enveloppe d'amplitude du signal (définie précédemment) sur les trames longues. Les auteurs définissent trois critères auxquels s'ajoute un critère proposé par Essid [72] :

- **Fréquence AM** : fréquence du pic d'amplitude maximale
- **Amplitude AM** : différence entre l'amplitude maximale et l'amplitude moyenne globale du spectre
- **Amplitude AM heuristique** : différence entre l'amplitude maximale et l'amplitude moyenne sur la bande de fréquences
- **Produit AM** : produit de la fréquence AM et de l'amplitude AM

Chacun de ces critères est calculé sur les deux bandes de fréquences citées, correspondant au trémolo et à la rugosité.

C.2 Descripteurs temporels

Les descripteurs suivants sont basés exclusivement sur la forme d'onde du signal audio dans une trame courte (ou longue si précisée) et ne font pas intervenir le spectre. On considérera que le calcul porte à chaque fois sur les échantillons $[x(1), \ldots, x(N)]$ d'une trame donnée (également notés $[x_1, \ldots, x_N]$ pour alléger les notations).

C.2.1 Taux de passage par zéro (ZCR)

Le taux de passage par zéro comptabilise le nombre de changements de signe sur une portion de signal. Autrement dit, si l'on définit le signal binaire $Z(t) = H(x(t))$ indiquant la positivité de l'échantillon $x(t)$, où H est la fonction de Heaviside, le taux de passage par zéro prend l'expression suivante :

$$ZCR = \sum_{n=2}^{N} \left[Z(t) - Z(t-1) \right]^2 .$$

Kedem [118] donne à ce dernier le nom alternatif de *fréquence dominante*, démontrant que le ZCR est statistiquement corrélé au centroïde spectral et donc à une éventuelle fréquence fortement dominante dans le spectre.

Le ZCR constitue donc une alternative très peu coûteuse au calcul du centroïde spectral, construite sur une modalité temporelle.

C.2.2 Taux de passage par zéro long-terme (LZCR)

On calcule également le taux de passage par zéro sur les trames long-terme.

C.2.3 Moments statistiques temporels

Court-terme (TM)

On applique les moments statistiques définis sur les amplitudes spectrales dans la section C.1.1, en remplaçant les amplitudes a_k par les échantillons x_k, pour calculer de manière similaire le centroïde temporel, la largeur temporelle, l'asymétrie temporelle et la platitude temporelle. Le terme temporel a peu de sens ici mais traduit la dualité par rapport aux descripteurs spectraux.

Long-terme (LTM)

On applique également les moments statistiques sur les trames longues. Les définitions restent inchangées.

Sur l'enveloppe (ETM)

On calcule enfin les moments statistiques sur une estimée de l'enveloppe d'amplitude à partir des trames longues. La méthode d'estimation de l'enveloppe est basée sur la transformée de Hilbert $\Psi(t)$ du signal $x(t)$ défini comme le produit de convolution avec la fonction $h(t) = \frac{1}{\pi t}$:

$$\Psi(t) = (h * s)(t) = \frac{1}{\pi} \lim_{\epsilon \to 0} \left\{ \int_{-\infty}^{t-\epsilon} \frac{x(\tau)}{t-\tau} d\tau + \int_{t+\epsilon}^{\infty} \frac{x(\tau)}{t-\tau} d\tau \right\}.$$

Le calcul pratique de la transformée de Hilbert discrète n'est pas détaillé ici. On estime l'enveloppe d'amplitude $e(n)$ comme le produit de convolution du module du *signal analytique* $y(n) = x(n) + i \cdot \Psi(n)$ avec une fenêtre de Hanning $h(n)$ de 50 ms, utilisée comme filtre passe-bas :

$$e(n) = |y(n)| * h(n).$$

C.2.4 Coefficients d'autocorrélation (AC)

On définit les coefficients d'autocorrélation sur un signal aléatoire centré $X(n)$ par

$$R(k) = E\left[X(n)X(n+k)\right].$$

En pratique on estime généralement ces derniers sur une réalisation finie de X au moyen du *périodogramme* $\frac{1}{N}\left|\mathcal{F}\left\{X\right\}\right|^2$ estimant la densité spectrale de puissance du signal. En effet on peut montrer que (si $\mathcal{F}$ est la transformée de Fourier et $\mathcal{F}^{-1}$ la transformée inverse),

$$R = \mathcal{F}^{-1}\left\{\frac{1}{N}\left|\mathcal{F}\left\{X\right\}\right|^2\right\}.$$

Le signal d'autocorrélation $R(k)$ fait apparaître des maxima aux valeurs pour lesquelles le signal présente une quasi-périodicité. Il apporte donc une information sur la présence de structures harmoniques dans le signal. On utilise donc les 49 premiers coefficients d'autocorrélation comme descripteurs audio pour couvrir les fréquences supérieures à 326 Hz.

C.3 Descripteurs cepstraux

C.3.1 Mel-Frequency Cepstral Coefficients (MFCC)

Les MFCC [57] substituent au spectre classique une représentation fréquentielle basée sur une échelle perceptive, appelée échelle *mel*[1], reproduisant une perception linéaire des distances entre hauteurs. On utilise généralement la formule suivante liant les mel aux fréquences en Hertz :

$$m(f) = 2595 \log_{10}\left(\frac{f}{700} + 1\right).$$

Le spectre est réparti dans un nombre fini de M valeurs de l'échelle mel au moyen d'un banc de filtres triangulaires centrées sur les M fréquences correspondantes aux valeurs mel. Les amplitudes associées à chaque filtre d'indice k sont notées $\bar{a}_k$.

Après l'application du logarithme sur les amplitudes mel, le processus de calcul des MFCC diffère du cepstre classique par l'application d'une transformée en cosinus discrète (DCT) à la place de la transformée de Fourier. Les coefficients MFCC ont donc l'expression suivante :

$$MFCC_i = \sum_{k=1}^{M} \log\left(\bar{a}_k\right) \cos\left[i\left(k - \frac{1}{2}\right)\frac{\pi}{M}\right].$$

Ils sont très couramment utilisés dans le domaine du traitement de la parole. On utilise ici un banc de filtres à $M = 13$ coefficient, comme dans la plupart des auteurs exploitant ces coefficients. La fréquence de Nyquist correspond à la valeur mel maximale $m(\frac{f_s}{2}) = m(8000) \approx 2840$. Les filtres ont donc pour largeur environ 227 mels.

C.3.2 Coefficients Cepstraux à Q constant

La transformée à Q constant est proposée par Brown [38] pour corriger la mauvaise répartition des *bins* fréquentiels de la TFD par rapport à la répartition géométrique des hauteurs musicales dans la gamme chromatique tempérée (où les demi-tons sont tous égaux). La transformation est définie comme un banc de filtres répartis logarithmiquement dans l'échelle des fréquences. Soit f_k et δf_k respectivement le centre et la largeur de la bande d'indice k, la proposition de Brown repose sur l'uniformité du facteur de qualité Q sur l'ensemble des bandes, défini par

$$Q = \frac{f_k}{\delta f_k}.$$

On en déduit la longueur N_k des fenêtres associées à chaque filtre :

$$N_k = \frac{f_s}{\delta f_k} = \frac{f_s}{f_k} Q.$$

On montre que la contrainte introduite mène à la transformation suivante :

$$X\left[k\right] = \frac{1}{N_k} \sum_{n=0}^{N_k - 1} w_k(n)\, x(n)\, e^{-\frac{j2\pi Q n}{N_k}},$$

où w_k est la fonction fenêtre adaptée à la longueur N_k du filtre k. Par exemple, si l'on se base sur la fenêtre de Hamming :

$$w_k(n) = \alpha + (1 - \alpha)\cos\left(\frac{2\pi n}{N_k}\right) \quad \alpha = \frac{25}{46} \text{ et } 0 \leq n \leq N_k - 1.$$

Ainsi si l'on répartit les bandes de fréquences entre une fréquence minimale f_{min} (dans notre cas, fixée à 100 Hz, d'après l'ambitus de la parole) et la fréquence de Nyquist $\frac{f_s}{2}$ et si l'on fixe un

1. Le nom *mel* est la contraction du mot mélodie et fait référence au fait que l'échelle est basée sur la comparaison de hauteurs musicales.

intervalle I_f (c'est-à-dire un rapport de fréquences) entre deux fréquences successives, on peut dénombrer $\lfloor \frac{\log(f_s/2f_{min})}{\log(I_f)} \rfloor$ filtres à Q constants ayant pour centres :

$$f_k = (I_f)^k f_{min}.$$

On exploite donc cette représentation fréquentielle pour calculer les coefficients cepstraux introduits précédemment, pour des intervalles de demi-octave ($I_f = \sqrt{2}$, soit 12 coefficients) et de tierce majeure (tiers d'octave : $I_f = \sqrt[3]{2}$, soit 18 coefficients).

C.4 Descripteurs perceptifs

C.4.1 Mesures liées au *pitch*

Le *pitch* désigne généralement dans la littérature la mesure de la hauteur perçue par l'oreille humaine. Si celle-ci suit grossièrement une échelle logarithmique sur les fréquences, la correspondance n'est pas aussi simple, et elle justifie la différentiation entre le pitch et la fréquence fondamentale. De plus la complexité des timbres de certains instruments (par exemple les percussions) permet la perception d'une hauteur musicale dans un spectre inharmonique. Cependant, la plupart des algorithmes se basent en pratique sur la mesure de la fréquence fondamentale, généralement considérée comme la fréquence la plus corrélée avec les partiels observés.

Il existe de nombreuses méthodes d'estimation de la fréquence fondamentale, basées sur la mesure d'autocorrélation, le filtrage par peigne, ou l'analyse cepstrale. Nous exploitons ici la méthode YIN proposée par Cheveigné et Kawahara [60], de faible complexité et très fiable sur des signaux monophoniques. Celle-ci se base sur la méthode par autocorrélation en y ajoutant plusieurs étapes destinées à en corriger les biais, dont nous présentons brièvement les plus importantes ci-dessous.

Fréquence fondamentale (F_0)

Le méthode classique par autocorrélation consiste à déterminer le premier pic de période τ non nulle de la fonction d'autocorrélation en un instant t sur une fenêtre de longueur W :

$$R_t(\tau) = \sum_{j=t+1}^{t+W} x_j x_{j+\tau}. \tag{C.1}$$

Les auteurs montrent que le résultat est différent et moins biaisé lorsque l'on cherche la période non-nulle τ minimisant la fonction de différence suivante :

$$\begin{aligned} d_t(\tau) &= \sum_{j=t+1}^{t+W} (x_j - x_{j+\tau})^2 && \text{(C.2)} \\ &= r_t(0) + r_{t+\tau}(0) - 2r_t(\tau). && \text{(C.3)} \end{aligned}$$

Cette différence est due à la présence du terme $r_{t+\tau}(0)$ dont la valeur dépend de τ.
La période τ minimisant la fonction de différence $d_t(\tau)$ permet de déterminer la fréquence fondamentale estimée $\hat{f}_0 = \frac{1}{\tau}$ qui est convertie logarithmiquement en hauteur musicale pour former le descripteur de pitch.

Mesure de périodicité (PM)

Une autre étape de l'algorithme YIN vise à corriger les éventuels problèmes d'« erreur d'octave » dus à la détection d'une période multiple de la vraie période, dans l'hypothèse où la valeur de la fonction de différence est inférieure pour celle-ci. Les auteurs proposent de sélectionner la plus petite période parmi celles correspondant à un minimum inférieur à un seuil donné. Ils étayent leur proposition en se basant sur la relation suivante :

$$2\left(x_t^2 + x_{t+\tau}^2\right) = \left(x_t + x_{t+\tau}\right)^2 + \left(x_t - x_{t+\tau}\right)^2.$$

Ainsi on obtient, en sommant les termes sur une trame :

$$2\sum_{j=t+1}^{t+W}\left(x_t^2+x_{t+\tau}^2\right)=\sum_{j=t+1}^{t+W}\left(x_t+x_{t+\tau}\right)^2+d_t(\tau).$$

Le terme de gauche peut être assimilé à une mesure de la puissance du signal, qui est répartie entre les deux termes de droite. On peut donc interpréter le terme $d_t(\tau)$ (nul dans le cas d'une période τ parfaite) comme une mesure de puissance d'apériodicité du signal, apportant une information essentielle sur le caractère périodique du signal considéré. On exploite donc cette mesure comme descripteur, en complément du descripteur F_0.

C.4.2 Mesures liées à l'intensité perceptive

La mesure d'intensité perceptive exploitée ici est fixée par Moore et al [162], sous le nom de *loudness*, terme que l'on préférera conserver ici. Celle-ci, de même que la perception de la hauteur, suit une échelle logarithmique par rapport à l'énergie du signal. On définit la *loudness* spécifique, comme la mesure de *loudness* sur chacune des bandes de fréquences de l'échelle bark [2]. Elle a pour expression, pour la bande de k barks :

$$L(k)=E(k)^{0.23}.$$

Celle-ci est évaluée sur le spectre de trames courtes.

Loudness spécifique relative

On utilise comme descripteurs les 24 coefficients de la *loudness* spécifique relative, définie comme le rapport de la loudness spécifique sur la *loudness* totale ($L_T=\sum_k L(k)$, somme de la *loudness* de toutes les bandes) :

$$L_r(k)=\frac{L(k)}{L_T}.$$

La normalisation introduite dans la *loudness* relative introduit simplement une invariance par rapport au volume sonore, différant largement selon les conditions de prises de son. Il s'agit donc d'indicateurs de répartition de la puissance dans le spectre, prenant en compte l'aspect perceptif. On exploite également deux autres descripteurs basés sur la mesure de *loudness*, décrit dans les paragraphes suivants.

Acuité perceptive

L'acuité perceptive (originellement *sharpness*, proposé par Peeters [178]) est l'équivalent du centroïde spectral sur des amplitudes perceptives (les mesures de *loudness*) :

$$AP=0.11\frac{\sum_k k\,g(k)\,L(k)}{L_T},$$

où $g(k)$ est une fonction compensant la largeur des bandes bark :

$$g(k)=\begin{cases}1 & \text{si } k<15\\ 0.066\,\exp(0.171\,k) & \text{si } k\geq 15\end{cases}.$$

Etalement perceptif

L'étalement perceptif (originellement *spread*, également proposé par Peeters) mesure la distance normalisée entre la *loudness* spécifique la plus élevée et la *loudness* totale :

$$EP=\left(\frac{L_T-\max_k L(k)}{L_t}\right)^2.$$

2. Echelle empirique de 24 bandes de fréquences modélisant le comportement des bandes critiques de l'audition.

Publications personnelles

Actes de conférences

- Gaël RICHARD, Mathieu RAMONA et Slim ESSID : Combined supervised and unsupervised approaches for automatic segmentation of radiophonic audio streams. *In Proc. ICASSP 2007.*
- Mathieu RAMONA, Gaël RICHARD et Bertrand DAVID : Vocal detection in music with Support Vector Machines. *In Proc. ICASSP 2008.*
- Mathieu RAMONA et Gaël RICHARD : Segmentation parole/musique par Machines à Vecteurs de Support. *Journées d'Étude sur la Parole 2008.*
- Mathieu RAMONA et Gaël RICHARD : Comparison of different strategies for a SVM-based audio segmentation. *In Proc. EUSIPCO 2009.*

Articles de journaux

- Mathieu RAMONA, Gaël RICHARD et Bertrand DAVID : Feature Selection with Kernel Gram-matrix based criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, soumission en 2010.

Rapport de Master

- Mathieu RAMONA : Approches automatiques pour la segmentation parole/musique. *TELECOM ParisTech*, septembre 2006. Rapport de stage de Master 2 ATIAM.

Bibliographie

[1] *Jamendo, open your ears*, `http://www.jamendo.com/en/`. pages 149

[2] *The Spider, General Purpose Machine Learning Toolbox in Matlab*, `http://www.kyb.mpg.de/bs/people/spider/main.html`. pages 171

[3] ISO/IEC information technology - multimedia content description interface - part 4 : Audio. International Standard ISO/IEC FDIS 15938-4, juin 2001. pages 25, 88, 176

[4] Ahmad R. Abu-El-Quran et Rafik A. Goubran : Pitch-based feature extraction for audio classification. *In Proc. IEEE Workshop on Haptic, Audio and Visual Environments and Their Applications*, pages 43–47, 20-21 septembre 2003. pages 25

[5] Ahmad R. Abu-El-Quran, Rafik A. Goubran et A. D. C. Chan : Adaptive feature selection for speech music classification. *In Proc. IEEE Workshop on Multimedia Signal Processing*, pages 212–216, octobre 2006. pages 18, 22, 148

[6] André G. Adami, Sachin S. Kajarekar et Hynek Hermansky : A new speaker change detection method for two-speaker segmentation. *In Proc. ICASSP '02*, volume 4, pages 3908–3911, 2002. pages 129

[7] Peter Ahrendt, Anders Meng et Jan Larsen : Decision time horizon for music genre classification using short-time features. *In Proc. EUSIPCO '04*, pages 1293–1296, 2004. pages 86

[8] M. Aizerman, E. Braverman et L. Rozonoer : Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964. pages 40

[9] Jitendra Ajmera, Iain McCowan et Hervé Bourlard : Robust hmm-based speech/music segmentation. *In Proc. ICASSP '02*, volume 1, pages 297–300, 13-17 mai 2002. pages 21, 22, 24, 124

[10] Jitendra Ajmera, Iain McCowan et Hervé Bourlard : Robust speaker change detection. *IEEE Signal Processing Letters*, 11(8):649–651, août 2004. pages 129, 133, 134

[11] Hirotugu Akaike : A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974. pages 133

[12] Selim Aksoy et Robert M. Haralick : Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern Recognition Letters*, 22(5):563–582, avril 2001. pages 87

[13] Enrique Alexandre-Cortizo, Manuel Rosa-Zurera et Francisco Lopez-Ferreras : Application of fisher linear discriminant analysis to speech music classification. *In Proc. EUROCON '05*, volume 2, pages 1666–1669, 22-24 novembre 2005. pages 21, 22, 24, 25, 148

[14] Erin L. Allwein, Robert E. Schapire et Yoram Singer : Reducing multiclass to binary : A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000. pages 73

[15] N. Aronszajn : Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68, 1950. pages 34, 38

[16] Arthur Asuncion et D.J. Newman : UCI machine learning repository, `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 2007. pages 60, 107, 173

[17] Raimo Bakis, Scott Chen, Ponani S. Gopalakrishnan, Ramesh Gopinath, Stéphane Maes, Lararos Polymenakos et Martin Franz : Transcription of broadcast news shows with the ibm large vocabulary speech recognition system. *In Proc. DARPA Speech Recognition Workshop*, pages 67–72, 1997. pages 131

[18] Mark A. Bartsch et Gregory H. Wakefield : Singing voice identification using spectral envelope estimation. *IEEE Trans. on Speech and Audio Processing*, 12(2):100–109, mars 2004. pages 23

[19] G. Baudat et F. Anouar : Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, octobre 2000. pages 42

[20] Richard Ernest Bellman : *Adaptive Control Processes : A Guided Tour*. Princeton University Press, 1st édition, 1961. pages 23

[21] Khalid Benabdeslem et Younès Bennani : Dendogram-based SVM for multi-class classification. *Journal of Computing and Information Technology (CIT)*, 14(4):283–289, 2006. pages 74

[22] Kristin P. Bennett et Colin Campbell : Support vector machines : Hype or hallelujah. *ACM SIGKDD Explorations Newsletter*, 2(2):1–13, décembre 2000. pages 42

[23] Kristin P. Bennett et Olvi L. Mangasarian : Robust linear programmating discrimination of two linearly inseparable sets. *Optimization Method and Software*, 1:23–34, 1992. pages 41, 97

[24] Adam Berenzweig et Daniel P. W. Ellis : Locating singing voice segments within music signals. *In Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 119–122, 2001. pages 23, 27, 148

[25] Adam Berenzweig, Daniel P. W. Ellis et Steve Lawrence : Using voice segments to improve artist classification of music. *In AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, pages 1–8, mai 2002. pages 16, 23, 27

[26] Maïtine Bergounioux : *Optimisation et contrôle des systèmes linéaires*. Dunod, 2001. pages 49

[27] A. Bhattacharrya : On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943. pages 134

[28] Jacek Biesiada et Wlodzislaw Duch : Feature selection for high-dimensional data : A kolmogorov-smirnov correlation-based filter. *Advances in Soft Computing*, 30:95–103, 2005. pages 97

[29] Christopher M. Bishop : *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995. pages 95

[30] Olivier Le Blouch et Patrice Collen : Méthode de segmentation parole non-parole. *In Rencontres Jeunes Chercheurs Parole*, 27-28 septembre 2005. pages 86

[31] Avrim L. Blum et Pat Langley : Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, décembre 1997. pages 93, 94

[32] Tobias Bocklet, Andreas Maier, Josef G. Bauer, Felix Burkhardt et Elmar Nöth : Age and gender recognition for telephone applications based on gmm supervectors and support vector machines. *In Proc. ICASSP '08*, pages 1605–1608, 31 mars-4 avril 2008. pages 16, 22

[33] Bruce P. Bogert, M. J. R. Healy et John W. Tukey : The quefrency alanysis of time series for echoes : Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking. *In* M. Rosenblatt, éditeur : *Proceedings of the Symposium on Time Series Analysis*, pages 209–243, New-York, 1963. Wiley. pages 88

[34] Jean-François Bonastre, Perrine Delacourt, Corinne Fredouille, Teva Merlin et Christian Wellekens : A speaker tracking system based on speaker turn detection for nist evaluation. *In Proc. ICASSP '00, Beijing*, 2000. pages 129, 131

[35] Paul S. Bradley et Olvi L. Mangasarian : Feature selection via concave minimization and support vector machines. *In Proc. International Conf. on Machine Learning*, pages 82–90, 1998. pages 97

[36] Paul S. Bradley et Olvi L. Mangasarian : Feature selection via mathematical programming. *INFORMS Journal on Computing*, 10(2):209–217, février 1998. pages 97

[37] Erin J. Bredensteiner et Kristin P. Bennett : Multicategory classification by support vector machines. *Computational Optimization and Applications*, 12(1-3):53–79, janvier 1999. pages 77

[38] Judith C. Brown : Calculation of a constant Q spectral transform. *Journal of the Acoustical Society of America*, 89(1):425–434, janvier 1991. pages 180

[39] Christopher J.C. Burges : A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, juin 1998. pages 37, 40

[40] Christopher J.C. Burges et David J. Crisp : Uniqueness of the SVM solution. *Advances in Neural Information Processing Systems*, 12, 2000. pages 43

[41] Juan José Burred et Alexander Lerch : Hierarchical automatic audio signal classification. *Journal of the Audio Engineering Society*, 52(7/8):724–739, juillet 2004. pages 16, 17, 24, 25, 85, 89

[42] Michael J. Carey, Eluned S. Parris et Harvey Lloyd-Thomas : A comparison of features for speech music discrimination. *In Proc. ICASSP '99*, volume 1, pages 149–152, 15-19 mars 1999. pages 17, 21, 25, 28

[43] Norman Casagrande, Douglas Eck et Balazs Kégl : Frame-level speech music discrimination using adaboost. *In Proc. ISMIR '05*, pages 345–350, 11-15 septembre 2005. pages 22, 148, 158

[44] Mauro Cettolo et Marcello Federico : Model selection criteria for acoustic segmentation. *In Proc. ISCA ASR 2000*, pages 221–227, 18-20 septembre 2000. pages 134

[45] Mauro Cettolo et Michele Vescovi : Efficient audio segmentation algorithms based on the bic. *In Proc. ICASSP '03*, 6-10 avril 2003. pages 131, 134

[46] Olivier Chapelle et Vladimir Vapnik : Model selection for support vector machines. *In Advances in Neural Information Processing Systems*, volume 12, pages 230–236. MIT Press, 2000. pages 52

[47] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet et Sayan Mukherjee : Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002. pages 50, 53, 99, 106

[48] Lei Chen, Sule Gündüz et Tamer Özsu : Mixed type audio classification with support vector machines. *In IEEE International Conference on Multimedia and Expo (ICME)*, pages 781–784, 9-12 juillet 2006. pages 22, 25, 86

[49] Scott Shaobing Chen et Ponani S. Gopalakrishnan : Speaker, environment and channel change detection and clustering via the barysian information criterion. *In Proc. DARPA Speech Recognition Workshop*, 1998. pages 131, 133, 134

[50] Shi-Sian Cheng et Hsin min Wang : METRIC-SEQDAC : A hybrid approach for audio segmentation. *In Proc. INTERSPEECH '04 International Conference on Skoen Language Processing*, pages 1617–1620, 4-8 octobre 2004. pages 131, 134

[51] Shi-Sian Cheng et Hsin-Min Wang : A sequential metric-based audio segmentation method via the bayesian information criterion. *In Proc. Eurospeech 2003*, pages 945–948, septembre 2003. pages 131

[52] Wu Chou et Liang Gu : Robust singing detection in speech music discriminator design. *In Proc. ICASSP '01*, volume 2, pages 865–868, 7-11 mai 2001. pages 17, 21, 23, 26, 119

[53] Perry Raymond Cook : *Identification of control parameters in an articulatory vocal tract model, with applications to the synthesis of singing.* Thèse de doctorat, Stanford University, Stanford, CA, 1990. pages 26

[54] Corinna Cortes et Vladimir Vapnik : Support vector networks. *Machine Learning*, 20(3): 273–297, septembre 1995. pages 41

[55] Koby Crammer et Yoram Singer : On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, mars 2001. pages 21, 77

[56] Nello Cristianini, Jaz Kandola, Andre Elisseeff et John Shawe-Taylor : On kernel target alignment. *Journal of Machine Learning Research*, 1, 2002. pages 54, 55

[57] Steven B. Davis et Paul Mermelstein : Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 28(4):357–366, 1980. pages 180

[58] Manuel Davy, Frédéric Desobry, Arthur Gretton et Christian Doncarli : An online support vector machine for abnormal events detection. *Signal Processing*, 86(8):2009–2025, août 2005. pages 140

[59] Manuel Davy et Simon Godsill : Audio information retrieval : A bibliography study. Rapport technique, CUED/F-INFENG/TR.429, février 2002. pages 24, 25

[60] Alain de Cheveigné et Hideki Kawahara : Yin, a fundamental frequency estimator for speech and music. *Acoustical Society of America Journal*, 111(4):1917–1930, avril 2002. pages 89, 181

[61] Perrine Delacourt et Christian Wellekens : Distbic : a speaker-based segmentation for audio data indexing. *Special Issue of Speech Communication on Accessing Information in Spoken Audio*, 32(1-2):111–126, septembre 2000. pages 131, 132, 134

[62] A. P. Dempster, N. M. Laird et D. R. Rubin : Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977. pages 20, 124

[63] Frédéric Desobry, Manuel Davy et Christian Doncarli : An online kernel change detection algorithm. *IEEE Trans. Acoustics, Speech and Signal Processing*, 53(8):2961–2974, août 2005. pages 138, 139

[64] Christopher P. Diehl : Approximate leave-one-out error estimation for learning with smooth, strictly convex margin loss functions. *In Proc. IEEE Workshop on Machine Learning for Signal Processing*, pages 63–72, 29 septembre-1 octobre 2004. pages 51

[65] Thomas G. Dietterich et Ghulum Bakiri : Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, janvier 1995. pages 73

[66] Kaibo Duan, S. Sathiya Keerthi et Aun Neow Poo : Evaluation of simple performance peasures for tuning SVM hyperparameters. *Neurocomputing*, 51:41–59, 2003. pages 50

[67] Chris Duxbury, Juan Pablo Bello, Mike Davies et Mark Sandler : Complex domain onset detection for musical signals. *In Proc. DAFX '03*, 2003. pages 161

[68] Khaled El-Maleh, Mark Klein, Grace Petrucci et Peter Kabal : Speech/music discrimination for multimedia applications. *In Proc. ICASSP '00*, volume 6, pages 2445–2448, 5-9 juin 2000. pages 16, 17, 21, 25, 28

[69] Daniel P. W. Ellis et Keansub Lee : Features for segmenting and classifying long-duration recordings of personal audio. *In Workshop on Statistical and Perceptual Audio Processing SAPA '04*, 3 octobre 2004. pages 16

[70] Antti Eronen : Automatic musical instrument recognition. Mémoire de D.E.A., Tempere University of Technology, avril 2001. pages 178

[71] S. Esmaili, S. Krishnan et K. Raahemifar : Content based audio classification and retrival using joint time-frequency analysis. *In Proc. ICASSP '04*, volume 5, pages 665–668, 17-21 mai 2004. pages 21, 25

[72] Slim Essid : *Classification automatique des signaux audio-frequences - Reconnaissance des instruments de musique.* Thèse de doctorat, TELECOM ParisTech, décembre 2005. pages 88, 178

[73] Slim Essid, Gaël Richard et Bertrand David : Instrument recognition in polyphonic music based on automatic taxonomies. *IEEE Trans. on Audio, Speech and Language Processing*, 14(1):68–80, janvier 2006. pages 16, 18

[74] Belkacem Fergani, Manuel Davy et Amrane Houacine : Unsupervised speaker indexing using one-class support vector machines. *In Proc. EUSIPCO '06*, 4-8 septembre 2006. pages 129, 138

[75] J. D. Ferguson : Variation duration models for speech. *In Proc. of the Symposium on the Application of Hidden Markov Hidden to Text and Speech*, pages 143–179, octobre 1981. pages 125

[76] Ronald A. Fisher : The use of multiple measurements in taxonomic problems. *Annals Eugen.*, 7:179–188, 1936. pages 34

[77] Jonathan Foote : A similarity measure for automatic audio classification. *In Proc. AAAI Spring Symposium on Intelligent Integration and Use of Text, Image, Video and Audio Corpora*, mars 1997. pages 24

[78] G.D. Forney : The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, mars 1973. pages 123

[79] Yoav Freund et Robert E. Schapire : A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, août 1997. pages 22

[80] Yoav Freund et Robert E. Schapire : A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, septembre 1999. pages 22

[81] Jerome H. Friedman : Another approach to polychotomous classification. Rapport technique, Department of Statistics, Stanford University, 1996. pages 71

[82] Sylvain Galliano, Guillaume Gravier et Laura Chaubard : The ESTER 2 evaluation campaign for the rich transcription of french radio broadcasts. *In Proc. INTERSPEECH '09*, pages 2583–2586, 6-10 septembre 2009. pages 148, 157

[83] Sylvrain Galliano, Edouard Geoffrois, Djamel Mostefa, Khalid Choukri, Jean-François Bonastre et Guillaume Gravier : The ESTER Phase II evaluation campaign for the rich transcription of french broadcast news. *In Proc. Interspeech '05*, 2005. pages 146

[84] Nicolas Garcia-Pedrajas et Domingo Ortiz-Boyer : Improving multiclass pattern recognition by the combination of two strategies. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(6):1001–1006, juin 2006. pages 72

[85] Jean-Luc Gauvain, Lori Lamel et Gilles Adda : Audio partitioning and transcription for broadcast data indexation. *Multimedia Tools and Applications*, 14(2):187–200, 2001. pages 17, 18, 19, 24

[86] Shahrokh Ghaemmaghami : Audio segmentation and classification based on a selective analysis scheme. *In Proc. Multimedia Modelling Conference MMM '04*, pages 42–48, 5-7 janvier 2004. pages 17, 18, 24

[87] T. Giannakopoulos, A. Pikrakis et S. Theodoridis : A speech/music discriminator for radio recordings using bayesian networks. *In Proc. ICASSP '06*, volume 5, pages 809–812, 2006. pages 22, 25, 28

[88] Olivier Gillet, Slim Essid et Gaël Richard : On the correlation of automatic audio and visual segmentations of music videos. *IEEE Trans. Circuits and Systems for Video Technology*, 17(3):347–355, mars 2007. pages 140

[89] Herbert Gish et Michael Schmidt : Text-independent speaker identification. *IEEE Signal Processing Magazine*, 11(4):18–32, octobre 1994. pages 131, 132

[90] Carl Gold et Peter Sollich : Model selection for support vector machine classification. *Neurocomputing*, 55:221–249, 15 mars 2002. pages 50

[91] Michael M. Goodwin et Jean Laroche : A dynamic programming approach to audio segmentation and speech music segmentation. *In Proc. ICASSP '04*, volume 4, pages 309–312, 17-21 mai 2004. pages 17, 21, 22

[92] Mihaela Gordan, Constantine Kotropoulos et Ioannis Pitas : Application of support vector machines classifiers to visual speech recognition. *Proc. International Conference on Image Processing (ICIP)*, 3:129–132, 2002. pages 124

[93] Guillaume Gravier, Jean-François Bonastre, Edouard Geoffrois, Sylvrain Galliano, Kevin Mc Tait et Khalid Choukri : The ester evaluation campaign of rich transcription of french broadcast news. *In Proc. Language Evaluation and Resources Conference*, 2004. pages 146

[94] Arthur Gretton et Frédéric Desobry : An online support vector machine for abnormal events detection. *In Proc. ICASSP '03*, volume 2, pages 709–712, 6-10 avril 2003. pages 140

[95] Yann Guermeur : *SVM Multiclasses, Théorie et Applications*. Hdr, Université Henri Poincaré - Nancy I, novembre 2007. pages 69, 76, 77

[96] Guodong Guo et Stan Z. Li : Content-based audio classification and retrieval by support vector machines. *IEEE Trans. on Neural Networks*, 14(1):209–215, janvier 2003. pages 22

[97] Isabelle Guyon et André Elisseeff : An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, mars 2003. pages 93, 94

[98] Isabelle Guyon et David G. Stork : Linear discriminant and support vector classifiers. *In* A.J. Smola, P.L. Bartlett, B. Schölkopf et D. Schuurmans, éditeurs : *Advances in Large Margin Classifiers*, pages 147–169. MIT Press, 2000. pages 43

[99] Isabelle Guyon, Jason Weston, Stephen Barnhill et Vladimir Vapnik : Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002. pages 99, 107

[100] T. Hain, S.E. Johnson, A. Tuerk, P.C. Woodland et S.J. Young : Segment generation and clustering in the htk broadcast news transcription system. *In Proc. 1998 DARPA Broadcast News Transcription and Understanding Workshop*, pages 133–137, 1998. pages 17, 18, 21, 24

[101] Tomoyuki Hamamura, Hiroyuki Mizutani et Bunpei Irie : A multiclass classification method based on multiple pairwise classifiers. *In Proc. International Conf. on Document Analysis and Recognition (ICDAR '03)*, pages 809–813, 3-6 août 2003. pages 72

[102] Hadi Harb, Liming Chen et Jean-Yves Auloge : Mixture of experts for audio classification an application to male female classification and musical genre recognition. *In Proc. ICME Multimedia and Expo '04*, volume 2, pages 1351–1354, 27-30 juin 2004. pages 16, 22

[103] Trevor Hastie et Robert Tibshirani : Classification by pairwise coupling. *In* Michael I. Jordan, Michael J. Kearns et Sara A. Solla, éditeurs : *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998. pages 70, 72

[104] John Hertz, Anders Krogh et Richard G. Palmer : *Introduction to the theory of neural computation*. Addison-Wesley, Redwood City, California, 1991. pages 23

[105] André Holzapfel et Yannis Stylianou : Singer identification in rembetiko music. *In Proc. SMC '07 Conference on Sound and Music Computing*, Lefkada, Greece, 11-13 juillet 2007. pages 23, 26, 149

[106] John D. Hoyt et Harry Wechsler : Detection of human speech using hybrid recognition models. *In Proc. ICPR Int Conf on Pattern Recognition*, volume 2, pages 330–333, 9-13 octobre 1994. pages 22, 24

[107] Chih-Wei Hsu et Chih-Jen Lin : A comparison of methods for multiclass support vector machines. *IEEE Trans. on Neural Networks*, 13(2):415–425, mars 2002. pages 69, 77, 78

[108] Chaug-Ching Huang, Jhing-Fa Wang et Dian-Jia Wu : Automatic scene change detection for composed speech and music sound under low snr noisy environment. *IEEE Trans. on Speech and Audio Processing*, 13(5):689–699, septembre 2005. pages 134

[109] Jincheng Huang, Zhu Liu et Yao Wang : Joint scene classification and segmentation based on hidden markov model. *IEEE Trans. on Multimedia*, 7(3):538–550, juin 2005. pages 21, 90

[110] Tommi S. Jaakkola et David Haussler : Probabilistic kernel regression models. *In Proceedings of the 1999 Conference on AI and Statistics*. Morgan Kaufmann, 1999. pages 51

[111] Roman Jarina, Noel O'Connor, Sean Marlow et Noel Murphy : Rhythm detection for speech-music discrimination in mpeg compressed domain. *In Proc. International Conf. on Digital Signal Processing DSP*, volume 1, pages 129–132, 2002. pages 21, 25, 89

[112] Hongchen Jiang, Junmei Bai, Shuwu Zhang et Bo Xu : SVM-based audio scene classification. *In Proc. of the IEEE Int. Conf. on Natual Language Processing and Knowledge Engineering (NLP-KE)*, pages 131–136, 30 octobre-1 novembre 2005. pages 18

[113] Thorsten Joachims : SVMlight, `http://svmlight.joachims.org/`. pages 60, 63

[114] Thorsten Joachims : *Making Large-Scale SVM Learning Practical*, pages 169–184. MIT Press, Cambridge, MA, 1999. pages 111, 163

[115] Thorsten Joachims : Estimating the generalization performance of an SVM efficiently. *In Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 431–438. Morgan Kaufmann Publishers, San Francisco, US, 2000. pages 52

[116] Cyril Joder, Slim Essid et Gaël Richard : Temporal integration for audio classification with application to musical instrument classification. *IEEE Trans. on Audio, Speech and Language Processing*, 17(1):174–186, janvier 2009. pages 86

[117] George H. John, Ron Kohavi et Karl Pfleger : Irrelevant features and the subset selection problem. *In Internation Conference on Machine Learning*, pages 121–129, 1994. pages 93, 94

[118] Benjamin Kedem : Spectral analysis and discrimination by zero-crossings. *Proc. IEEE*, 74(11):1477–1493, novembre 1986. pages 88, 179

[119] S. Sathiya Keerthi, Chong Jin Ong et Martin M.S. Lee : Two efficient methods for computing leave-one-out error in SVM algorithms. Rapport technique, National University of Singapore, 23 novembre 2000. pages 51

[120] Thomas Kemp, Michael Schmidt, Martin Westphal et Alex Waibel : Strategies for automatic segmentation of audio data. *In Proc. ICASSP '00*, volume 3, pages 1423–1426, 2000. pages 131

[121] Ji-Soo Keum et Hyon-Soo Lee : Speech music discrimination based on spectral peak analysis and multi-layer perceptron. *In International Conference on Hybrid Information Technology*, volume 2, pages 56–61, novembre 2006. pages 17, 22, 89, 119

[122] John Maynard Keynes : *A Treatise on Probability*, chapitre XVII, page 201. Dover Publications, 1921. pages 122

[123] Hyunsoo Kim, Barry L. Drake et Haesun Park : Relationships between support vector classifiers and generalized linear discriminant analysis on support vectors. Rapport technique GT-CSE-06-16, Georgia Institute of Technology, 2006. pages 43

[124] Youngmoo E. Kim et Brian Whitman : Singer identification in popular music recordings using voice coding features. *In Proc. ISMIR '02*, 13-17 octobre 2002. pages 16, 23, 27

[125] Don Kimber et Lynn Wilcox : Acoustic segmentation for audio browsers. *In Proc. Interface Conference*, juillet 1996. pages 16, 21, 124

[126] S. Knerr, L. Personnaz et G. Dreyfus : Single-layer learning revisited : A stepwise procedure for building and training a neural network. *Neurocomputing : Algorithms, Architectures and Applications*, 68:41–50, 1990. pages 71

[127] Ulrich H.-G. Kressel : Pairwise classification and support vector machines. *In* Bernhard Schölkopf, Christopher J.C. Burges et Alexander J. Smola, éditeurs : *Advances in Kernel Methods*, pages 255–268. The MIT Press, Cambridge, MA, 1999. pages 72

[128] Soonil Kwon et Shrikanth Narayanan : Unsupervised speaker indexing using generic models. *IEEE Trans. on Speech and Audio Processing*, 13(5):1004–1013, septembre 2005. pages 129, 131

[129] Hélène Lachambre, Régine André-Obrecht et Julien Pinquier : Singing voice characterization for audio indexing. *In Proc. EUSIPCO '07*, 3-7 septembre 2007. pages 23, 27, 90

[130] Hansheng Lei et Venu Govindaraju : Half-against-half multi-class support vector machines. *In Multiple Classifier Systems, 6th International Workshop (MCS 2005)*, pages 157–164, juin 2005. pages 74

[131] Tat-Wan Leung, Chong-Wah Ngo et Ron W.H. Lau : Ica-fx features for classification of singing voice and instrumental sound. *In Proc. ICPR Int Conf on Pattern Recognition*, volume 2, pages 367–370, 23-26 août 2004. pages 17, 23, 27

[132] Stan Z. Li : Content-based classification and retrieval of audio using the nearest feature line method. *IEEE Trans. on Speech and Audio Processing*, 8(5):619–625, septembre 2000. pages 16, 24, 25

[133] Ying Li et Chitra Dorai : SVM-based audio classification for instructional video analysis. *In Proc. ICASSP '04*, volume 5, pages 897–900, 17-21 mai 2004. pages 22

[134] Yipeng Li et DeLiang Wang : Detecting pitch of singing voice in polyphonic audio. *In Proc. ICASSP '05*, volume 3, pages 17–20, 18-23 mars 2005. pages 26

[135] Yipeng Li et DeLiang Wang : Singing voice separation from monaural recordings. *In Proc. ISMIR '06*, octobre 2006. pages 17, 23, 27

[136] Zeyu Li, Shiwei Tang et Shuicheng Yan : Multi-class SVM classifier based on pairwise coupling. *In Proc. of the First International Workshop on Pattern Recognition with Support Vector Machines*, pages 321–333, 10 août 2002. pages 72

[137] Gaëlle Loosli, Sang-Goog Lee et Stéphane Canu : Context changes detection by one class SVMs. *UM 2005 - Workshop on Machine Learning for User Modeling : Challenges*, 2005. pages 137

[138] Guojun Lu et Templar Hankinson : An investigation of automatic audio classification and segmentation. *In Proc. ISCP 2000*, volume 2, pages 776–781, 21-25 août 2000. pages 25, 90

[139] Lie Lu, Hao Jiang et HongJiang Zhang : A robust audio classification and segmentation method. *In Proc. ACM International Multimedia Conference*, volume 9, pages 203–211, 2001. pages 18, 19, 21, 22, 25, 119

[140] Lie Lu, Stan Z. Li et Hong-Jiang Zhang : Content-based audio segmentation using support vector machines. *In Proc. ICME Multimedia and Expo '01*, pages 749–752, 22-25 août 2001. pages 18, 19, 22, 25, 86

[141] Hanna Lukashevich, Matthias Gruhne et Christian Dittmar : Effective singing voice detection in popular music using arma filtering. *In Proc. DAFX '07*, septembre 2007. pages 17, 27

[142] A. Luntz et V. Brailovsky : On estimation of characters obtained in statistical procedure of recognition. *Tchnicheskaya Kibernetica*, 3, 1969. pages 51

[143] Namunu C. Maddage, Kongwah Wan, Changsheng Xu et Ye Wang : Singing voice detection using twice-iterated composite fourier transform. *In Proc. ICME Multimedia and Expo '04*, volume 2, pages 1347–1350, 27-30 juin 2004. pages 21, 23, 27

[144] Namunu C. Maddage, Changsheng Xu et Ye Wang : Singer identification based on vocal and instrumental models. *In Proc. ICPR Int Conf on Pattern Recognition*, volume 2, pages 375–378, 23-26 août 2004. pages 23

[145] Michael I. Mandel et Daniel P.W. Ellis : Song-level features and support vector machines for music classification. *In Proc. ISMIR '05*, 2005. pages 23

[146] Michael I. Mandel, Graham E. Poliner et Daniel P.W. Ellis : SVM active learning for music retrieval. *Multimedia Systems*, 12(1), août 2006. pages 16

[147] O. L. Mangasarian : Linear and nonlinear separation of patterns by linear programming. *Operations Research*, 13(3):444–452, mai-juin 1965. pages 97

[148] Maria Markaki, Andre Holzapfel et Yannis Stylianou : Singing voice detection using modulation frequency features. *In Proc. SAPA*, pages 7–10, Brisbane, Australia, 21 septembre 2008. pages 149

[149] Keith Dana Martin : *Sound-Source Recognition : A Theory and Computational Model.* Thèse de doctorat, MIT, juin 1999. pages 178

[150] Julie Mauclair et Julien Pinquier : Fusion de parametres en classification parole/musique. *In Journés d'étude sur la Parole 2004*, avril 2004. pages 18, 22

[151] Martin McKinney et Jeroen Breebaart : Features for audio and music classification. *In Proc. ISMIR '03*, pages 151–158, 2003. pages 16, 21, 25, 26

[152] Marina Meila : Data centering in feature space. Rapport technique 421, University of Washington, 2002. pages 58

[153] Hugo Meinedo et Joao Neto : Audio segmentation, classification and clustering in a broadcast news task. *In Proc. ICASSP '03*, volume 2, pages 5–8, 6-10 avril 2003. pages 131

[154] Anders MENG, Peter AHRENDT et Jan LARSEN : Improving music genre classification by short-time feature integration. *In Proc. ICASSP '05*, volume 5, pages 497–500, 18-23 mars 2005. pages 86

[155] Anders MENG et John SHAWE-TAYLOR : An investigation of feature models for music genre classification using the support vector classifier. *In Proc. ISMIR '05*, 2005. pages 16, 85, 86

[156] James MERCER : Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London*, A 209:415–446, 3 novembre 1909. pages 39

[157] Nima MESGARANI, Malcolm SLANEY et Shihab A. SHAMMA : Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations. *IEEE Transactions on audio, speech and language processing*, 14(3):920–930, mai 2006. pages 26, 89

[158] Sebastian MIKA, Gunnar RÄTSCH, Jason WESTON, Bernhard SCHÖLKOPF et Klaud-Robert MÜLLER : Fisher discriminant analysis with kernels. *In Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop*, pages 41–48, août 1999. pages 42, 103

[159] Jonathan MILGRAM, Robert SABOURIN et Mohamed CHERIET : Two-stage classification system combining model-based and discriminative approaches. *In Proc. ICPR Int Conf on Pattern Recognition*, volume 1, pages 152–155, 23-26 août 2004. pages 17, 22

[160] Luis Carlos MOLINA, Lluis BELANCHE et Angela NEBOT : Feature selection algorithms : A survey and experimental evaluation. *In Proc. ICDM '02*, pages 306–313, décembre 2002. pages 93

[161] Michinari MOMMA et Kristin P. BENNETT : A pattern search method for model selection of support vector regression. *In Proceedings of the SIAM International Conference on Data Mining*, 2002. pages 50

[162] Brian C. J. MOORE, Brian R. GLASBERG et Thomas BEAR : A model for the prediction of thresholds, loudness, and partial loudness. *Journal of Audio Engineering Society*, 45(4):224–240, avril 1997. pages 89, 182

[163] Klaus-Robert MÜLLER, Sebastian MIKA, Gunnar RÄTSCH, Koji TSUDA et Bernhard SCHÖLKOPF : An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural Networks*, 12(2):181–201, mars 2001. pages 42

[164] José Enrique MUÑOZ-EXPOSITO, Sebastian GARCIA-GALAN, Nicolas RUIZ-REYES, Pedro VERA-CANDEAS et Fernando RIVAS-PEÑA : Speech music discrimination using a single warped lpc-based feature. *In Proc. ISMIR '05*, 2005. pages 16, 25, 85

[165] Kevin P. MURPHY : Hidden semi-markov models (HSMMs). Rapport technique, MIT, 20 novembre 2002. pages 125

[166] Julia NEUMANN, Christoph SCHÖRR et Gabriele STEIDL : Combined SVM-based feature selection and classification. *Machine Learning*, 61(1-3):129–150, novembre 2005. pages 101

[167] Canh Hao NGUYEN et Tu Bao HO : An efficient kernel matrix evaluation measure. *Pattern Recognition*, 41(11):3366–3372, novembre 2008. pages 58

[168] Andreas B. NIELSEN, Lars K. HANSEN et Ulrik KJEMS : Pitch based sound classification. *In Proc. ICASSP '06*, volume 3, pages 788–791, 2006. pages 17, 25

[169] Naoki NITANDA, Miki HASEYAMA et Hideo KITAJIMA : Accurate audio-segment classification using feature extraction matrix. *In Proc. ICASSP '05*, volume 3, pages 261–264, 18-23 mars 2005. pages 17, 18, 19, 25

[170] Tin Lay NWE et Haizhou LI : Broadcast news segmentation by audio type analysis. *In Proc. ICASSP '05*, volume 2, pages 1065–1068, 18-23 mars 2005. pages 17, 25

[171] Tin Lay NWE et Haizhou LI : Singing voice detection using perceptually motivated features. *In Proc. International Conf. on Multimedia*, pages 309–312, 2007. pages 27

[172] Tin Lay NWE, Arun SHENOY et Ye WANG : Singing voice detection in popular music. *In Proc. ACM International Multimedia Conference*, pages 324–327, 10-16 octobre 2004. pages 23, 27

[173] Manfred Opper et Ole Winther : Gaussian processes and SVM : Mean field results and leave-one-out. *In Advances in Large Margin Classifiers*, pages 43–65, 19 mars 2000. pages 51

[174] M. Ostendorf, V. Digalakis et O.A. Kimball : From hmms to segment models : a unified view of stochastic modelingfor speech recognition. *IEEE Trans. on Speech and Audio Processing*, 4(5):360–378, septembre 1996. pages 125, 126

[175] Edgar E. Osuna, Robert Freund et Federico Girosi : Support Vector Machines : Training and applications. Rapport technique AIM-1602, MIT, mars 1997. pages 37

[176] Costas Panagiotakis et George Tziritas : A speech music discriminator based on rms and zero-crossings. *IEEE Multimedia*, 7(1), février 2005. pages 21, 25, 28

[177] Emanuel Parzen : On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962. pages 55

[178] Geoffroy Peeters : A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Rapport technique, IRCAM, 2004. pages 88, 89, 176, 177, 182

[179] Geoffroy Peeters : A generic system for audio indexing application to speech music segmentation and music genre recognition. *In Proc. DAFX '07*, pages 205–212, 10-15 septembre 2007. pages 17, 28

[180] Geoffroy Peeters : A generic system for audio indexing : Application to speech/music segmentation and music genre recognition. *In Proc. DAFX '07*, 2007. pages 16, 21

[181] Geoffroy Peeters et Xavier Rodet : Hierarchical gaussian tree with inertia ratio maximization for the classification of large musical instrument database. *In Proc. DAFX '03*, 8-11 septembre 2003. pages 28, 96

[182] Vesa Peltonen, Juha Tuomi, Anssi Klapuri, Jyri Huopaniemi et Timo Sorsa : Computational auditory scene recognition. *In Proc. ICASSP '02*, volume 2, pages 1941–1944, 13-17 mai 2002. pages 16, 22, 24, 25, 86

[183] Luis Perez-Freire et Carmen Garcia-Mateo : A multimedia approach for audio segmentation in tv broadcast news. *In Proc. ICASSP '04*, volume 1, pages 369–372, 17-21 mai 2004. pages 133

[184] David Perrot et Robert O. Gjerdingen : Scanning the dial : An exploration of factors in identification of musical style. *In Proc. of the Society for Music Perception and Cognition*, page 88, 1999. pages 84, 86

[185] John C. Platt : Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *In Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA, 1999. pages 70

[186] John C. Platt, Nello Cristianini et John Shawe-Taylor : Large margin dags for multiclass classification. *In Advances in Neural Information Processing Systems*, volume 12, pages 547–553. MIT Press, 2000. pages 73

[187] Tomaso Poggio et Federico Girosi : Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247(4945):978–982, février 1990. pages 40

[188] Jean-Baptiste Pothin et Cédric Richard : Optimal feature representation for kernel machines using kernel-target alignment criterion. *In Proc. ICASSP '07*, volume 3, pages 1065–1068, 15-20 avril 2007. pages 58

[189] Jean-Baptiste Pothin et Cédric Richard : Optimizing kernel alignment by data translation in feature space. *In Proc. ICASSP '08*, pages 3345–3348, 2008. pages 58

[190] Lawrence Rabiner : A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, février 1989. pages 21, 123

[191] Mathieu Ramona : Approches automatiques pour la segmentation parole/musique. Rapport de stage de master 2, TELECOM ParisTech, Septembre 2006. pages 86

[192] Mathieu Ramona, Gaël Richard et Bertrand David : Vocal detection in music with support vector machines. *In Proc. ICASSP '08*, pages 1885–1888, 31 mars-4 avril 2008. pages 149

[193] Vishweshwara Rao, S. Ramakrishnan et Preeti Rao : Singing voice detection in north indian classical music. *In Proc. of the National Conference on Communications (NCC)*, 2008. pages 17

[194] Sourabh Ravindran et David V. Anderson : Audio classification and scene recognition for hearing aids. *In Proc. Circuits and Systems ISCAS '05*, volume 2, pages 860–863, 23-26 mai 2005. pages 16, 17, 18, 22, 26, 89

[195] Josph Razik, Dominique Fohr, Odile Mella et Parlangeau-Valles : Segmentation parole musique pour la transcription automatique. *In Journées d'étude sur la Parole*, pages 417–420, 2004. pages 24, 28, 124, 148

[196] Lise Regnier et Geoffroy Peeters : Singing voice detection in music tracks using direct voice vibrato detection. *In Proc. ICASSP '09*, 19-24 avril 2009. pages 21, 23, 26, 27, 161

[197] Ryan Rifkin et Aldebaro Klautau : In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, décembre 2004. pages 69, 76, 78

[198] Jorma Rissanen : *Stochastic complexity in statistical inquiry*. World Scientific, Singapore, 1989. pages 133

[199] Martin Rocamora et Perfecto Herrara : Comparing audio descriptors for singing voice detection in music audio files. *In Brazilian Symposium on Computer Music*, septembre 2007. pages 17, 23, 26, 28

[200] Frank Rosenblatt : The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, novembre 1958. pages 22

[201] Stéphane Rossignol : *Segmentation et indexation des signaux sonores musicaux*. Thèse de doctorat, Université Pierre et Marie Curie - Paris VI, juillet 2000. pages 25

[202] Stéphane Rossignol, Xavier Rodet, Joël Soumagne, Jean-Louis Collette et Philippe Depalle : Automatic characterisation of musical signals feature extration and temporal segmentation. *Journal of New Music Research*, 28(4):281–295, décembre 1999. pages 17, 22, 25, 28

[203] Volker Roth et Volker Steinhage : Nonlinear discriminant analysis using kernel functions. *In Advances in Neural Information Processing Systems*, pages 568–574. MIT Press, 1999. pages 42, 161

[204] M. De Santo, G. Percannella, C. Sansone et M. Vento : Classifying audio of movies by a multi-expert system. *In Proc. International Conf. on Image Analysis and Processing*, pages 386–391, 26-28 septembre 2001. pages 16, 22

[205] John Saunders : Real-time discrimination of broadcast speech music. *In Proc. ICASSP '96*, volume 2, pages 993–996, 7-10 mai 1996. pages 17, 20, 24, 158

[206] Robert E. Schapire : The boosting approach to machine learning : An overview. *MSRI Workshop on Nonlinear Estimation and Classification*, 2003. pages 22

[207] Eric Scheirer et Malcolm Slaney : Construction and evaluation of a robust multifeature speech/music discriminator. *In Proc. ICASSP '97*, volume 2, pages 1331–1334, Munich, Germany, 21-24 avril 1997. pages 17, 20, 21, 24, 25, 28, 88, 90, 148, 158, 177

[208] Bernhard Schölkopf, Chris Burges et Vladimir Vapnik : Extracting support data for a given task. *In Proceedings ot the First International Conference on Knowledge Discovery & Data Mining*, pages 252–257. AAAI Press, 1995. pages 36, 40, 43, 71

[209] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola et Robert C. Williamson : Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001. pages 42, 136, 137

[210] Bernhard Schölkopf, Alex J. Smola et Klaus-Robert Müller : Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, juillet 1998. pages 42

[211] Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson et Peter L. Bartlett : New support vector algorithms. *Neural Computation*, 12(5):1207–1245, mai 2000. pages 42, 137

[212] Bernhard SCHÖLKOPF et Alexander J. SMOLA : *Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 1st édition, 15 décembre 2001. pages 38, 39, 40, 136, 137

[213] Bernhard SCHÖLKOPF, Kah-Kay SUNG, Christopher J.C. BURGES, Federico GIROSI, Partha NIYOGI, Tomaso POGGIO et Vladimir VAPNIK : Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Trans. on Signal Processing*, 45(11): 2758–2765, novembre 1997. pages 43

[214] Bernhard SCHÖLKOPF, Robert C. WILLIAMSON, Alex J. SMOLA, John SHAWE-TAYLOR et John C. PLATT : Support vector method for novelty detection. *In Proc. Advances in Neural Information Processing Systems*, volume 12, 2000. pages 42, 137

[215] Gideon SCHWARZ : Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978. pages 133

[216] Amon SHASHUA : On the relationship between the support vector machine for classification and sparsified fisher's linear discriminant. *Neural Processing Letters*, 9(2):129–139, avril 1999. pages 43, 112

[217] Matthew A. SIEGLER, Uday JAIN, Bhiksha RAJ et Richard M. STERN : Automatic segmentation, classification and clustering of broadcast news audio. *In Proc. DARPA Speech Recognition Workshop*, 1997. pages 131

[218] P. SIVAKUMARAN, J. FORTUNA et M. ARIYAEEINIA : On the use of the bayesian information criterion in multiple speaker detection. *In Proc. Eurospeech 2001*, pages 795–798, 3-7 septembre 2001. pages 134

[219] Fred W. SMITH : Pattern classifier design by linear programming. *IEEE Trans. on Computers*, 17(4):367–372, avril 1968. pages 41

[220] Jan STADERMANN et Gerhard RIGOLL : A hybrid SVM hmm acoustic modeling approach to automatic speech recognition. *In Proc. INTERSPEECH '04 International Conference on Spoken Language Processing*, pages 661–664, 4-8 octobre 2004. pages 124

[221] Ingo STEINWART : On the generalization ability of support vector machines. Rapport technique, University of Jena, 2001. pages 60

[222] Andreas STOLCKE, Sachin KAJAREKAR et Luciana FERRER : Nonparametric feature normalization for SVM-based speaker verification. *In Proc. ICASSP '08*, pages 1577–1580, 1-4 avril 2008. pages 87

[223] Johan SUNDBERG : *The acoustics of the singing voice*. Scientific American, mars 1977. pages 26

[224] Pham Dinh TAO et Le Thi Hoai AN : A d. c. optimization algorithm for solving the trust-region subproblem. *SIAM Journal on Optimization*, 8(2):476–505, 1998. pages 101

[225] Sergios THEODORIDIS et Konstantinos KOUTROUMBAS : *Pattern Recognition*. Academic Press, 4ème édition, novembre 2008. pages 86

[226] Godfried T. TOUSSAIN : Note on optimal selection of independent binary-valued features for pattern recognition. *IEEE Trans. on Information Theory*, 17(5):618, septembre 1971. pages 93

[227] Alain TRITSCHLER et Ramesh GOPINATH : Improved speaker segmentation and segments clustering using the bayesian information criterion. *In Proc. Eurospeech '99, Budapest, Hungary*, volume 2, pages 679–682, 1999. pages 129, 133, 134

[228] Wei-Ho TSAI et Hsin-Min WANG : Automatic singer recognition of popular music recordings via estimation and modeling of solo vocal signals. *IEEE Trans. on Audio, Speech and Language Processing*, 14(1):330–341, janvier 2006. pages 16, 23

[229] George TZANETAKIS et Perry COOK : Musical genre classification of audio signals. *IEEE Trans. on Speech and Audio Processing*, 10(5):293–302, juillet 2002. pages 16, 17, 25, 85, 86

[230] Vladimir VAPNIK : *Statistical Learning Theory*. Wiley Interscience, 16 septembre 1998. pages 40, 52, 171

[231] Vladimir VAPNIK et Olivier CHAPELLE : Bounds on error expectation for support vector machines. *Neural Computation*, 12(9):2013–2036, septembre 2000. pages 52, 53

[232] Vladimir Vapnik et Alexey Chervonenkis : On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971. pages 37

[233] Vladimir Vapnik et Alexey Chervonenkis : The necessary and sufficient conditions for consistency in the empirical risk minimization method. *Pattern Recognition and Image Analysis*, 1(3):283–305, 1991. pages 37

[234] Vladimir Vapnik, Steven E. Golowich et Alex J. Smola : Support vector method for function approximation, regression estimation and signal processing. *In Advances in Neural Information Processing Systems*, volume 9, pages 281–287. MIT Press, 1996. pages 42

[235] Vladimir Vapnik et A. Lerner : Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24, 1963. pages 34

[236] Vladimir N. Vapnik : *The Nature of Statistical Learning Theory*. Information Science and Statistics. Springer Verlag, 2nd édition, 2000. pages 23, 36, 37, 40, 42, 43, 71

[237] Shankar Vembu et Stephan Baumann : Separation of vocals from polyphonic audio recordings. *In Proc. ISMIR '05*, pages 337–344, 11-15 septembre 2005. pages 23, 27, 161

[238] Andrew J. Viterbi : Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. on Information Theory*, 13(2):260–269, avril 1967. pages 123

[239] Lei Wang : Feature selection with kernel class separability. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(9):1534–1546, septembre 2008. pages 93, 103

[240] Lei Wang et Kap Luk Chan : Learning kernel parameters by using class separability measure. *In NIPS Kernel Workshop*, 2002. pages 58

[241] Andrew Webb : *Statistical Pattern Recognition*. Wiley, 2nd edition édition, 15 octobre 2002. pages 93

[242] Kris West et Stephen Cox : Features and classifiers for the automatic classification of musical audio signals. *In Proc. ISMIR '04*, pages 531–536, 2004. pages 21, 25

[243] Kris West et Stephen Cox : Finding an optimal segmentation for audio genre classification. *In Proc. ISMIR '05*, 2005. pages 25, 84

[244] Jason Weston, André Elisseeff, Bernhard Schölkopf et Mike Tipping : Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1491, mars 2003. pages 98, 104, 107, 109, 174

[245] Jason Weston, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio et Vladimir Vapnik : Feature selection for SVMs. *In Advances in Neural Information Processing Systems*, volume 13, pages 668–674. MIT Press, Cambridge, MA, 2000. pages 99, 106, 171

[246] Jason Weston et Chris Watkins : Support vector machines for multi-class pattern recognition. *In ESANN*, 1999. pages 77, 78

[247] Gethin Williams et Daniel P. W. Ellis : Speech/music discrimination based on posterior probability features. *In Proc. Eurospeech '99*, pages 687–690, 5-9 septembre 1999. pages 17, 22, 27, 148, 158

[248] Erling Wold, Thomas Blum, Douglas Keislar et James Wheaton : Content-based classification, search, and retrieval of audio. *IEEE Multimedia*, 3(3):27–36, 1996. pages 25

[249] P. C. Woodland, Thomas Hain, S. E. Johnson, T. R. Niesler, A. Tuerk, E. W. D. Whittaker et S. J. Young : The 1997 HTK broadcast news transcription system. *In Proc. 1998 DARPA Broadcast News Transcription and Understanding Workshop*, pages 41–48, 1998. pages 131

[250] Chung-Hsien Wu et Chia-Hsin Hsieh : Multiple change-point audio segmentation and classification using an mdl-based gaussian model. *IEEE Trans. on Audio, Speech and Language Processing*, 14(2):647–657, mars 2006. pages 131

[251] Kuo-Ping Wu et Sheng-De Wang : Choosing the kernel parameters of support vector machines according to the inter-cluster distance. *In Proc. International Joint Conference on Neural Networks IJCNN '06*, pages 1205–1211, 16-21 juillet 2006. pages 55

[252] Huilin Xiong, M. N. S. Swamy et M. Omair Ahmad : Optimizing the kernel in the empirical feature space. *IEEE Trans. on Neural Networks*, 16(2):460–474, mars 2005. pages 58

[253] Lei Yu et Huan Liu : Feature selection for high-dimensional data : A fast correlation-based filter solution. *In Proc. ICML*, pages 856–863, 2003. pages 97

[254] Shun-Zheng Yu et Hisashi Kobayashi : An efficient forward-backward algorithm for an explicit-duration hidden markov model. *IEEE Signal Processing Letters*, 10(1):11–14, janvier 2003. pages 126

[255] Marco Zaffalon et Marcus Hutter : Robust feature selection by mutual information distributions. *In Proceedings of the 18th International Conference on Uncertainty in Artificial Intelligence (UAI-2002)*, pages 577–584. Morgan Kaufmann, 3 juin 2002. pages 97

[256] Heiga Zen, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi et Tadashi Kitamura : Hidden semi-markov model based speech synthesis system. pages 126

[257] Shilei Zhang, Shuwu Zhang et Bo Xu : A two-level method for unsupervised speaker-based audio segmentation. *In International Conference on Pattern Recognition ICPR' 06*, volume 4, pages 298–301, août 2006. pages 131

[258] Tong Zhang : Automatic singer identification. *In Proc. ICME Multimedia and Expo '03*, volume 1, pages 33–36, 6-9 juillet 2003. pages 23, 27

[259] Tong Zhang et C.-C. Jay Kuo : Hierarchical system for content-based audio classification and retrieval. *In Proc. SPIE Multimedia Storage and Archiving Systems III*, volume 3527, pages 398–409, 1998. pages 21

[260] Tong Zhang et C.-C. Jay Kuo : Heuristic approach for generic audio data segmentation and annotation. *In Proc. ACM International Multimedia Conference*, volume 1, pages 67–76, 1999. pages 119

[261] Tong Zhang et Jay Kuo : Audio content analysis for online audiovisual data segmentation and classification. *IEEE Trans. on Speech and Audio Processing*, 9(4):441–457, mai 2001. pages 16, 18, 19, 21, 25, 119

[262] Yibin Zhang et Jie Zhou : Audio segmentation based on multi-scale audio classification. *In Proc. ICASSP '04*, volume 4, pages 349–352, 17-21 mai 2004. pages 120

[263] Bowen Zhou et John H. L. Hansen : Efficient audio stream segmentation via the combined T2 statistic and bayesian information criterion. *IEEE Trans. on Speech and Audio Processing*, 13(4):467–474, juillet 2005. pages 20, 131, 134

[264] Bowen Zhou et John H.L. Hansen : Unsupervised audio stream segmentation and clustering via the bayesian information criterion. *In Proc. ICASSP '00, Beijing*, 2000. pages 134

[265] Shaohua Kevin Zhou et Rama Chellappa : From sample similarity to ensemble similarity : probabilistic distance measures in reproducing kernel hilbert space. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(6):917–929, juin 2006. pages 134, 135