Rapport de stage ATIAM (Master 2 Informatique SAR)

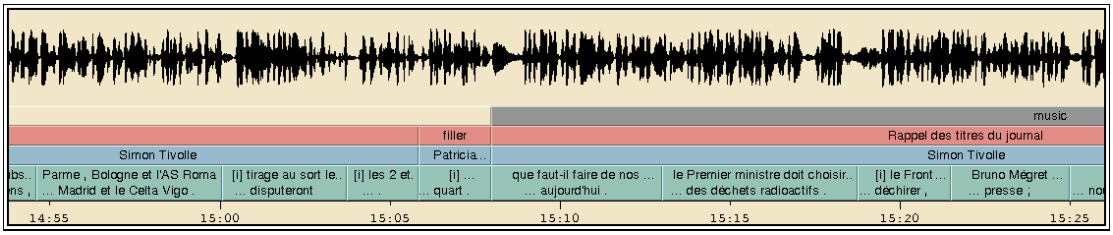
Approches automatiques pour la segmentation parole/musique

(Automatic segmentation of audio streams)

Mathieu RAMONA

Tuteur : Gaël RICHARD

20 mars – 31 juillet 2006



UPMC
4 pl. Jussieu
75252 PARIS CEDEX 05

ENST
37/39 rue Dareau
75014 PARIS

IRCAM
1 pl. Igor Stravinsky
75004 PARIS

Remerciements

Je tenais à remercier mon maître de stage Gaël Richard, qui a su m'accorder tout le temps et la patience nécessaires. J'ai beaucoup apprécié sa spontanéité et la qualité de son encadrement.

Je remercie également Slim Essid, pour sa disponibilité quotidienne pour les diverses questions concernant son travail, sur lequel se base ce stage, et sa patience.

Merci à Bertrand David, qui a également participé à la tutelle de ce stage, et répondu à certaines de mes questions, et à Roland Badeau et David Matignon pour leur aide sur quelques points particuliers.

Merci aux thésards et stagiaires de l'équipe TSI avec qui j'ai pu partager quelques moments de détente, et il en faut parfois pour supporter les caprices de Matlab.

Merci à Fabrice et Sophie-Charlotte, administrateurs réseau, pour contribuer à mon pacte de non-agression avec les machines à puces.

Merci également à Messieurs Marc Peyrade et Yves Grenier, respectivement directeur de l'ENST et responsable du département TSI.

Merci enfin, et surtout, à Yto, pour son soutien inconditionnel, pour sa présence, pour les jours difficiles et les instants de bonheur, merci d'avoir supporté tout ça, merci d'avoir été là.

Table des matières

Chapitre 1

Présentation

1.1 Introduction

Ce stage conclut mon année de Master 2 SAR, rattaché administrativement à l'université Pierre et Marie Curie (Paris VI). J'y ai suivi la filière ATIAM, dispensée par l'IRCAM (Institut de Recherche et Coordination Acoustique/Musique), et qui met en relation la création musique contemporaine et les théories scientifiques traitant du son, de la voix et de la musique. J'ai donc travaillé quatre mois à l'ENST (Ecole Nationale Supérieure de Télécommunications), du 20 mars au 31 juillet 2006, sous la direction de Gaël Richard, sur la segmentation automatique de flux audio.

Devant la quantité croissante de données multimédias disponibles et la facilité accrue des moyens de diffusion, un nouveau défi se pose aux chercheurs en traitement du signal et de l'information. La gestion de l'information multimédia appelle le développement d'un traitement et d'une indexation basées sur le contenu même de ces données.

Dans le domaine audio, les applications sont plétores, de la récupération d'information à partir d'un document sonore à la gestion en temps réel d'un flux audio radiophonique. Celles-ci nécessitent néanmoins, pour une analyse sémantique du contenu, une phase préalable de segmentation, afin d'isoler les segments temporels consécutifs contenant de la parole, de la musique, des jingles radiophoniques, ou encore du chant, et appliquer par la suite un traitement spécifique et adapté à chaque type d'information. En particulier, les systèmes de transcription automatique de la parole, qui drainent actuellement un grand effort de recherche, nécessitent l'identification préalable des segments de paroles dans un flux pouvant contenir d'autres types de contenu.

Nous présentons ici un système de segmentation de flux radiophoniques, basé sur la technique assez récente des Machines à Vecteurs Supports, dans le cadre de la campagne commune d'évaluation ESTER.

1.2 Le département Traitement du Signal et de l'Image (TSI)

Le département TSI de l'ENST, dirigé par M. Yves Grenier, a pour missions l'enseignement, la recherche et la formation par la recherche dans les domaines du traitement du signal et des images et de leurs applications, en particulier pour les

télécommunications. Il se divise en cinq groupes, chacun contribuant à l'ensemble des missions du département :

- **Traitement et Interprétation des Images (TII)**
  Ce groupe conduit des recherches sur la mise en oeuvre de schémas complets de traitement, d'analyse et d'interprétation d'images, en particulier de scènes complexes. Les domaines d'application sont l'imagerie médicale, aérienne, satellitaire, radar ou encore la description d'objets tridimensionnels.
- **Traitements Statistiques et Applications aux Communications (TSAC)**
  Les travaux de ce groupe sont axés sur le signal dans les communications, la séparation de sources ou encore la modélisation statistique des signaux.
- **Perception, Apprentissage et Modélisation (PAM)**
  Ce groupe étudie le rôle des facteurs humains dans l'accès aux divers types d'informations : parole (reconnaissance), image (psychovision), écrit (structuration de documents), fusion des modalités perceptives dans l'appréhension de l'environnement, interfaces multimodales.
- **Codage (COD)**
  Le groupe COD s'intéresse aux techniques de compression de sources et à leur adaptation aux applications audiovisuelles/multimédia (compression audio, codage d'images, transmission audiovisuelle, systèmes temps-réel).
- **Audio, Acoustique et Ondes (AAO)**
  Ce dernier groupe étudie la physique des ondes dans les domaines de l'optique (stockage de l'information) et de l'acoustique (modélisation de la production des sons, perception, antennes acoustiques).

Mon tuteur Gaël Richard appartient au groupe AAO.

1.3 Le projet INFOM@GIC

1.3.1 Origine

Le projet INFOM@GIC, lancé en décembre 2005, s'inscrit dans le cadre du pôle de compétitivité IMVN (Image, Multimédia et Vie Numérique). Ce dernier est porté par l'Agence Régionale du Développement (ARD) d'Ile de France dont il est le troisième pôle. Consacré aux technologies de l'information et de la communication, il implique de grands groupes (TF1, Lagardère Groupe, France Télécom, Eclair, SFP, TSF), des PMEs, des laboratoires et des institutions (LIP6 Paris VI, l'INA, l'IRCAM, Télécom Paris, CNAM, ESIEE, ENSTA, ENS Louis-Lumière, Gobelins, FEMIS).

1.3.2 Objectif

INFOM@GIC vise à mettre en place, sur une période de trois ans, un laboratoire industriel de sélection, de tests, d'intégration et de validation d'applications opérationnelles des meilleures technologies franciliennes dans le domaine de l'ingénierie des connaissances.

Ce laboratoire s'appuie sur une plate-forme commune qui doit couvrir les grands domaines de l'analyse d'information quelles que soient les sources (données structurées, texte, images et sons) :

- la recherche et l'indexation,

– l'extraction de connaissances,
– la fusion d'informations multimédias.

Elle inclut des applications pour les secteurs de la e-Education et de la gestion des patrimoines culturels numériques.

1.3.3 Acteurs

Les partenaires du projet sont répartis en quatre catégories :

– Industriels : Thalès (coordinateur), EADS, Xerox ;
– PMEs : Bertin, Europlace, FIST (CNRS-ANVAR), Intuilab, Odile Jacob, Pertimm, Temis, Vecsys ;
– Etablissements publics : CEA, CNRS, INA, ONERA ;
– Ecoles et universités : GET/ENST-INT, Paris VI (LIP6, LSTA), Paris VIII (LC&U), Paris IX Dauphine (CEREMADE), Paris XIII (LIPN), Paris-Sud Orsay (LIMSI), Université de Marne-la-Vallée (IGM), CNRS/LACAN.

D'autres partenaires sont en instance d'intégration : Canal+ HiTech, Hi-Store, LIRMM, SINEQUA.

1.3.4 Le sous-projet "Patrimoine Numérisé"

L'ENST intervient principalement dans le sous-projet "Patrimoine Numérisé", piloté par l'INA, pour tout ce qui concerne l'acquisition, la description initiale, l'édition et la publication de contenus audiovisuels. La contribution de l'ENST est répartie dans trois domaines : les outils d'analyse des séquences vidéos, la multimodalité voix-image et les outils de traitement de la bande audio. Ce dernier, auquel mon stage est rattaché, inclut les outils de prétraitement de la bande son, la séparation parole/musique ou encore la segmentation en évènements sonores (parole, musique, applaudissements, etc.).

1.4 Etat de l'art

Le problème de la segmentation audio est à l'origine associé au traitement de la voix. Ainsi, dès 1994, Hoyt et Wechsler [Hoyt and Wechsler, 1994] s'intéressent à la détection de voix humaine dans un flux audio, y compris en présence d'un fond sonore, au moyen d'un classificateur binaire basé sur les fonctions à base radiale. De même la détection ou la reconnaissance de locuteur constituent quelques unes des premières problématiques liées à la segmentation. Ainsi, [Sugiyama et al., 1993] propose une approche non-supervisée pour la segmentation des locuteurs. Néanmoins, l'analyse d'un flux radiophonique, qui constitue l'objectif principal de ce genre de recherches, nécessite une segmentation plus générale, permettant de distinguer les segments de parole, non plus entre eux, mais parmi d'autres types de contenu sonore comme la musique, la voix téléphonique, la voix sur fond bruité... le système de transcription de bulletins d'information radiophoniques HTK 1997 a ainsi largment stimulé la recherche aux Etats-Unis dans ce domaine [Woodland et al., 1998, Siegler et al., 1997]. Plus récemment, en 2003, l'AFCP établit avec la campagne ESTER un programme de recherche sur la transcription d'émissions radiophoniques francophones, qui comprend entre autres une tâche de segmentation parole/musique. Nous en suivons les consignes [Gravier et al., 2004] afin de pouvoir comparer nos

résultats à ceux des autres laboratoires [Saunders, 1996] (cf section 5.3.2).

La segmentation nécessite une représentation du signal plus proche de sa réalité physique et perceptive que la forme d'onde. De nombreux descripteurs, associés à des propriétés de natures très différentes, ont été définis ; beaucoup nous proviennent du traitement de la voix. On retrouve par exemple les MFCC (*Mel-Frequency Cepstral Coefficients*), outil de base de l'analyse de la voix, dans la majorité des articles traitant de la segmentation parole/musique ([West and Cox, 2004, Kimber and Wilcox, 1996]). [Scheirer and Slaney, 1997] présente une collection assez hétérogène de descripteurs (modulation à 4Hz, moments spectraux, ZCR...) connus pour leur bonne aptitude dans la discrimination parole/musique. [Saunders, 1996] se focalise sur l'élaboration d'un système temps-réel et exploite des descripteurs très légers en terme de coût, basés sur le calcul du taux de passages par zéro (ZCR - *Zero Crossing Rate*). L'énergie à court terme est également classiquement utilisée pour la détection de silence [Zhang and Kuo, 2001]. [Carey et al., 1999] compare l'efficacité des descripteurs les plus courants pour la segmentation parole/musique.

Certains articles se distinguent par leur proposition de nouveaux attributs, comme [Pinquier et al., 2002] qui introduit des descripteurs statistiques basés sur les résultats d'une segmentation aveugle fine. [El-Maleh et al., 2000] exploite des descripteurs instantanés (estimés sur une fenêtre de 20ms), permettant une discrimination parole/musique en temps réel, basés sur les LSF (*Line Spectral Frequencies*) et le ZCR. [Chou and Gu, 2001] définit le *Coefficient Harmonique*, qui, basé à la fois sur une évaluation spectrale et temporelle de l'autocorrélation, forme un système de décision robuste, combiné à l'énergie de modulation à 4 Hz. On retrouvera une liste quasi exhaustive des descripteurs employés pour l'indexation audio, établie par Peeters, dans [Peeters, 2004]. Notre stage se base sur les descripteurs employés par Essid dans le cadre de sa thèse [Essid, 2005].

La plupart des systèmes présentés exploitent les solutions classiques liées aux problèmes de classification pour la tâche de segmentation ou de discrimination audio, des *plus proches voisins* (kNN - *k Nearest Neighbour*) [Lu et al., 2001a, Kimber and Wilcox, 1996] aux GMM (*Gaussian Mixture Model*), que l'on retrouve dans une énorme majorité des publications [Scheirer and Slaney, 1997, Saunders, 1996], parfois suivis d'une phase de post-traitement servant au lissage des résultats [Chou and Gu, 2001] ou d'un réseau de Markov caché (HMM - *Hidden Markov Model*) [Gravier et al., 2005]. Les GMM sont basés sur l'approche Bayésienne, qui consiste à estimer la densité de propabilité des classes dans l'espace des vecteurs d'attributs. Les Machines à Vecteurs Supports (SVM - *Support Vector Machines*)[Burges, 1998] suivent au contraire une approche dite *discriminative*. La séparation entre deux classes est faite par un hyperplan de décision après projection dans un espace de dimension supérieure. Les SVMs sont employées depuis peu dans de nombreuses tâches de classification, comme la reconnaissance de visages [Osuna et al., 1997] ou récemment la segmentation audio [Lu et al., 2001b]. Cette technique a d'ailleurs également été exploitée pour la reconnaissance d'instruments de musique [Essid, 2005], et présente de très bonnes performances, comparées aux autres modèles de classification.

Dans d'autres cas, la démarche consiste à diviser le signal en segments localement homogènes, sans leur associer de classe. Cette simplification du problème

permet d'utiliser des techniques différentes, généralement métriques. Ainsi, l'usage de distances probabilistes [Siegler et al., 1997, Zhou and Chellappa, 2006] permet de segmenter le signal en se basant sur une distance inter-trames. Ce critère de similarité entre trames peut également être utilisé pour regrouper des segments dans une même classe (de nature inconnue) soit en partant du résultat d'une analyse supervisée [Kimber and Wilcox, 1996], soit à partir de segments arbitraires [Kemp et al., 2000]. D'autres critères basés sur la théorie de l'Information, comme le critère d'information bayésienne (BIC - *Bayesian Information Criterion*), fournissent sur des données gaussiennes des résultats plus robustes que les distances probabilistes classiques [Chen and Gopalakrishnan, 1998], et ont été testés sur la segmentation de flux audio [Zhou and Hansen, 2000]. [Gillet and Richard, 2006] exploite ces techniques pour l'étude de la corrélation entre segmentations audio et video sur un clip musical, ainsi que certains algorithmes de segmentation aveugle basés sur les SVMs [Loosli et al., 2005, Desobry et al., 2005], dont la pertinence pour la détection de nouveauté a été montrée [Schölkopf et al., 1999].

1.5 Objet du stage

Notre travail se distingue par l'application de Machines à Vecteurs Supports, combinées à des algorithmes de sélection automatique d'attributs, pour le problème de la segmentation parole/musique dans un flux audio. Les SVMs ont été exploitées pour des tâches similaires, comme la reconnaissance d'instruments de musique [Essid, 2005], mais la seule application que nous avons constaté pour la segmentation audio [Lu et al., 2001b] reste sommaire car elle exploite un petits nombre d'attributs et ne fournit de résultats que sur une base propre au laboratoire qui ne permet malheureusement pas d'en vérifier la pertinence. Nous espérons ainsi, en nous basant sur la campagne francophone ESTER [Gravier et al., 2004], fournir des résultats plus objectifs, que nous pourrons comparer à ceux obtenus par les autres laboratoires engagés dans le programme [Calmès et al., 2005, Scheffer et al., 2005, Gravier et al., 2005].

En outre, constatant l'efficacité des approches non-supervisées pour la segmentation audio, nous avons étudié l'apport complémentaire de ce type d'approches sur les résultats de la classification par SVM. Cette étude permet la mise en évidence de la très bonne précision temporelle sur les frontières des segments, obtenue par des algorithmes basés sur la détection de nouveauté.

1.6 Organisation du rapport

On commencera par présenter au chapitre 2 l'architecture globale du système mis en place. On traitera ensuite au chapitre 3 de l'extraction des descripteurs fournissant une représentation adéquate du signal ainsi que de la sélection des plus adéquats pour la classification, parmi un panel très large de descripteurs. Les principes de la classification, et plus particulièrement des Machines à Vecteurs Supports, ainsi que des diverses approches non-supervisées exploitées dans ce stage, seront présentés au chapitre 4. Nous exposerons ensuite notre travail dans le chapitre 5, en présentant les corpus utilisés, le protocole expérimental, ainsi que les résultats obtenus par notre système.

Chapitre 2

Architecture du système

Ce chapitre donne une vue d'ensemble du système utilisé durant ce stage. Le code, écrit en matlab, est basé sur le programme développé par Slim Essid pour son travail de thèse [Essid, 2005]. Néanmoins, afin d'adapter ce dernier à nos besoins, de l'optimiser et d'en corriger certaines erreurs, le code a été largement remanié et completé durant la durée de ce stage. La partie concernant la segmentation par méthodes non-supervisées (présentée section 4.2) provient du travail d'Olivier Gillet sur l'étude de la corrélation entre la segmentation des médias visuels et sonores dans des clips musicaux [Gillet and Richard, 2006].

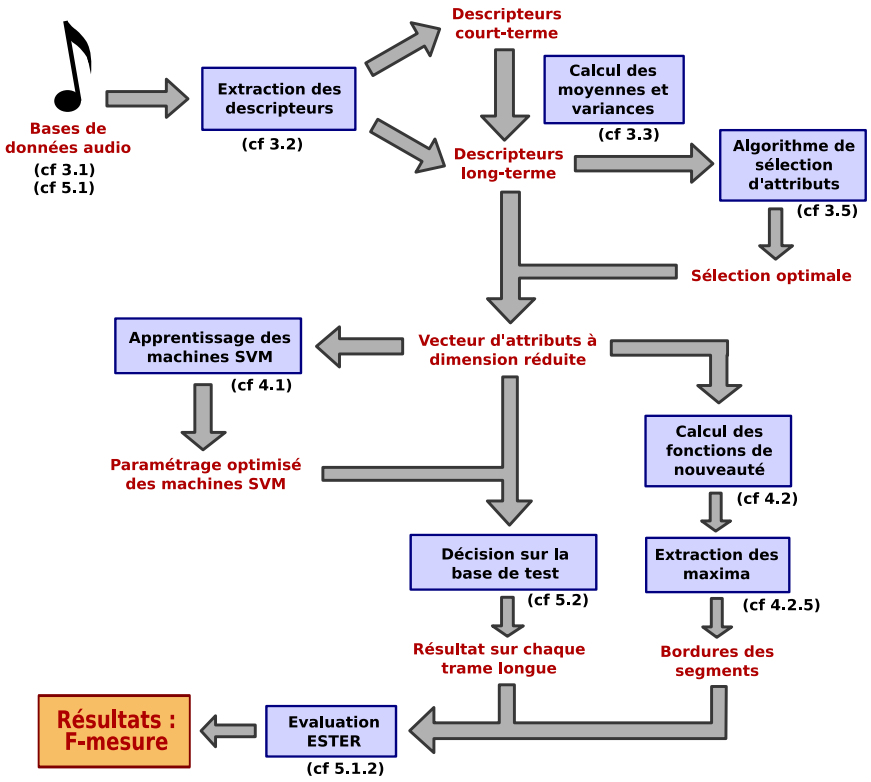


Fig. 2.1 – Architecture globale du système développé et exploité dans le cadre de cette étude sur la segmentation de flux audio. Une référence indique, pour chaque unité de traitement, la section relative dans ce rapport.

Les extraits audio des bases d'apprentissage et de test sont extraits de corpus publics présentés section 5.1. Après pré-traitement et découpage en trames courtes, la détection de trames silencieuses permet de définir le découpage en trames longues (section 3.1). Une large collection de descripteurs, présentée section 3.2, est calculée sur les trames courtes ou longues. Nous expliquerons section 3.3 que les descripteurs sur trames courtes sont ramenés à la représentation de trames longues par le calcul des moyennes et variances sur ces dernières.

Les descripteurs sont alors classés d'après leur aptitude à séparer les classes choisies. Les algorithmes de sélection automatique exploités dans ce stage sont présentés section 3.5. On choisit par la suite de ne retenir que les d premiers descripteurs de ce classement ; la dimension d devra constituer un compromis acceptable entre performances et complexité réduite.

Les vecteurs d'attributs ainsi constitués pour chacune des trames de la base d'apprentissage, serviront au calcul des paramètres optimaux des machines à vecteurs supports, dont les fondements théoriques sont présentés section 4.1.2. C'est avec ces classificateurs paramétrés que l'on estimera ensuite la décision prise pour chacune des trames de la base de test (ou de développement, dans le cas d'une seconde phase d'optimisation).

On exploite par la suite la séquence de ces résultats sur trames pour en déduire les frontières entre segments caractérisés par une classe unique. On peut également introduire une phase de post-traitement intermédiaire permettant de lisser ou de corriger les résultats bruts obtenus par le classificateur. Nous avons d'abord exploité un /emphlissage basique qui consiste à détecter les trames isolées (*outliers*, dont la classe diffère des deux trames voisines) et à en remplacer la classe par celle des trames voisines. Néanmoins, nous verrons que les techniques de segmentation non-supervisée (présentées section 4.2), basées sur la détection de nouveauté, améliorent sensiblement les résultats de notre système en utilisant une approche hybride (cf section 4.3) qui permet de corriger les résultats de classification.

Enfin, à partir des bordures de segments reconnues, on déduit les événements permettant l'évaluation objective des performances du système par le biais d'un outil développé pour la campagne d'évaluation ESTER. La campagne ESTER, et en particulier son corpus et son protocole d'évaluation, sont présentés section 5.1.1. Le résultat final est une *F-mesure*, définie pour la campagne ESTER, que nous pouvons ainsi comparer aux résultats des autres laboratoires participants.

Chapitre 3

Extraction et sélection des descripteurs

Le signal audio est exploité, après prétraitement, sous la forme d'une séquence de trames chevauchantes de durée relativement courtes, afin de traduire les propriétés locales du signal. La classification s'effectue par la prise d'une décision sur chacune de ces trames. Ces dernières sont représentées par une collection de descripteurs de diverses familles censés apporter l'information la plus pertinente pour la séparation des classes choisies. On peut par la suite réduire le nombre de ces descripteurs à l'aide d'algorithmes de sélection d'attributs qui permettent d'estimer les descripteurs les plus aptes à séparer les classes prises en compte.

Après avoir détaillé le traitement préalable des données audio et leur découpage en une séquence de trames, nous énumérerons les descripteurs mis en jeu dans notre expérience, pour finalement présenter les algorithmes de sélection d'attributs utilisés dans ce stage.

3.1 Exploitation des données audio

3.1.1 Prétraitement

Les données audio exploités sont échantillonnés à 16 kHz, sur 16 bits mono, sous la forme de fichier audio non-compressés au format `wav`. On représente la séquence des échantillons par un signal $s(n)$, où n est le temps discrétisé.

Le signal $s(n)$ est tout d'abord normalisé, afin de limiter l'effet des conditions d'enregistrement variables sur la classification. Le signal est centré (soustraction de sa moyenne) et divisé par son maximum absolu (pour obtenir un signal compris dans l'intervalle $[-1; 1]$).

$$\tilde{s}(n) = \frac{\hat{s}(n)}{\max_n |\tilde{s}(n)|} \qquad \text{où } \hat{s} = s(n) - \frac{1}{L}\sum_{n=0}^{L-1} s(n)$$

3.1.2 Fenêtres d'analyse temporelle

Afin de donner une représentation localement homogène du signal nous le divisons en une séquence de fenêtres chevauchantes. La largeur (durée) de ces fenêtres doit constituer un compromis entre les précisions temporelles et fréquentielles, conformément

au principe d'Heisenberg qui précise que le produit des dispersions d'énergie temporelle et fréquentielle est borné inférieurement. Nous avons opté pour une fenêtre de *512* échantillons, soit 32ms à 16kHz, durée de stationnarité locale d'un signal audio. Les fenêtres se recouvrent à *50%* (soit un pas d'avancement de 16ms entre chaque fenêtre). Ces fenêtres sont appelées par la suite *trames courtes*. Aucune pondération (de type Hamming...) n'est appliquée sur la fenêtre pour le calcul des descripteurs temporels.

Certains descripteurs décrivent des phénomènes de durée plus longue que celle de stationnarité, et sont donc calculés sur des *trames longues*. De même, l'intégration des descripteurs en descripteurs de plus haut niveau (moyenne, variance...) se fait sur des trames longues. Elles sont fixées à 2s dans le cadre de ce stage, avec un recouvrement de 50% (soit un pas d'une seconde). La durée de ces fenêtres est sujette à discussion, car encore une fois elle est le fruit d'un compromis, entre précision temporelle et justesse des descripteurs.

D'autres descripteurs se basent sur le spectre et non sur le signal lui-même. On calcule pour cela la puissance spectrale par FFT (*Fast Fourier Transform*) sur chaque trame après application d'une fenêtre de pondération de Hamming. Ceci correspond au calcul d'une Transformée de Fourier à Court Terme (TFCT) définie par Nawab et Quatieri. Néanmoins cette transformée présente une résolution fréquentielle constante (de 31.25 Hz), qui n'est pas forcément adaptée à la représentation de signaux musicaux, où l'on préférerait une meilleure résolution en basses fréquences. C'est ce qui motive le recours à une transformée à résolution variable, dite à facteur de qualité Q constant, où $Q = \frac{f}{\delta f}$, où f représente la fréquence. L'usage d'une telle représentation sera abordé dans la présentation des descripteurs.

3.1.3 Trames silencieuses

Pour finir, un algorithme de *détection de trames silencieuses* est appliqué sur la séquence des trames courtes afin de limiter l'analyse aux trames pertinentes pour la classification. Nous utilisons ici un algorithme très simple, qui montre une bonne efficacité sur les signaux non bruités (comme c'est le cas pour les corpus utilisés). La décision de silence se base sur les critères heuristiques suivants :

- si l'amplitude maximale sur la fenêtre (en valeur absolue) est 30dB en dessous du maximum global sur le signal
- si la fenêtre présente une valeur d'amplitude constante

On parcourt ensuite la séquence des trames non-silencieuses pour déduire les positions (index de trame courte) des trames longues en se basant sur les critères suivants :

- chaque trame longue débute sur une trame courte non silencieuse, ce afin d'améliorer la précision lors de la détermination des limites de segments
- un nombre minimum de 60 trames courtes chevauchantes (correspondant au pas d'une seconde entre trames longues) sépare chaque trame longue de la précédente
- une trame longue contient un taux minimum (fixé à 40%) de trames non silencieuses

On obtient ainsi des mini-séquences de trames longues adjacentes séparées par des plages de silence. C'est sur ces trames que sera effectuée la recherche des limites de segments, ce qui signifie que l'on perd un peu en précision temporelle.

3.2 Descripteurs

Les descripteurs que nous utilisons ont été récupérés ou implémentés par Essid dans le cadre de sa thèse sur la reconnaissance d'instruments de musique ([Essid, 2005]). On trouvera également une liste détaillée et relativement exhaustive des descripteurs couramment employés en indexation audio dans [Peeters, 2004]. Essid exploite également des attributs basés sur la théorie des ondelettes, mais nous avons préféré ne pas les utiliser, en raison d'un temps de calcul trop élevé. Nous utilisons indifféremment les termes *descripteur* ou *attribut* dans la suite de ce rapport.

3.2.1 Descripteurs cepstraux

Les coefficients cepstraux sont les attributs les plus largement utilisés dans le traitement de la voix. Le *cepstre* se base sur une modélisation source-filtre du signal, qui découle de la structure propre de l'instrument vocal. Sa forme réelle s'obtient comme la transformée de Fourier inverse du logarithme du spectre d'amplitude $|X(k)|$. Si l'on exprime le signal comme le produit de convolution d'une source $g(n)$ (glottique) par un filtre $h(n)$ (résonateurs), il est montré que les coefficients ceptraux correspondant aux basses *quéfrences* représentent la contribution du filtre, ce qui explique le succès de ce descripteur pour différentes tâches liées au traitement de la voix. De nombreuses sources sonores musicales (y compris la majorité des instruments) ne répondent pourtant pas à ce modèle, mais la représentation cepstrale reste tout à fait pertinente pour la discrimination parole/musique.

Mel-Frequency Cepstral Coefficients Les MFCC s'obtiennent en considérant, pour le calcul du cepstre, une représentation fréquentielle selon une échelle preceptive (échelle des *fréquences Mel*). Les coefficients sont calculés en utilisant un banc de filtres triangulaires MEL sur lesquels est appliquée une transformée en cosinus discrète inverse (type II) du logarithme du spectre.

Nous utilisons deux bancs de filtres, composés respectivement de 30 et 11 bandes MEL s'étalant jusqu'à 16kHz. Sont également calculées les dérivées temporelles premières et secondes de ces coefficients, en utilisant une approximation polynômiale à l'ordre 2 de la trajectoire spectrale.

Coeffients cepstraux à partir de la CQT Brown propose de remplacer, dans le calcul du cepstre, le spectre Mel par un spectre CQT (*Constant Q Transform*) basé sur une gamme musicale tempérée [Brown, 1999]. Quatre représentations cepstrales sont ainsi calculées en utilisant des résolutions d'une octave, d'une quinte (demi-octave), d'une tierce majeure (tiers d'octave) et d'une tierce mineure (quart d'octave). En considérant une limite inférieure en fréquence de 27.1 Hz (note la plus basse du piano), nous pouvons calculer 9 coefficients pour la résolution d'une octave, et nous gardons les 10 premiers pour les autres résolutions. Sont également calculées les dérivées temporelles premières et secondes de ces coefficients.

3.2.2 Descripteurs spectraux

Moments spectraux Les *moments spectraux* permettent de représenter différentes caractéristiques de la forme spectrale. Soit a_k l'amplitude de la composante spectrale

de fréquence $f_k = \frac{k}{N}$, on définit les descripteurs suivants à partir des moments μ_i définis par :

$$\mu_i = \frac{\sum_{k=0}^{K-1} (f_k)^i a_k}{\sum_{k=0}^{K-1} a_k}$$

- **centroïde spectral :** $S_c = \mu_1$
  décrit le centre de gravité du spectre, qui caractérise la *brillance* du son
- **largeur spectrale :** $S_w = \sqrt{\mu_2 - \mu_1^2}$
  décrit l'étendue du spectre autour de son centroïde
- **assymétrie spectrale :** $S_a = \frac{2(\mu_1)^3 - 3\mu_1\mu_2 + \mu_3}{S_w^3}$
  représente la symétrie du spectre autour de son centroïde
- **platitude spectrale :** $S_k = \frac{-3\mu_1^4 + 6\mu_1\mu_2 - 4\mu_1\mu_3 + \mu_4}{S_w^4} - 3$
  est d'autant plus grande que le spectre est "piqué" autour de son centroïde

Sont également calculées leurs dérivées temporelles premières et secondaires.

Alternatives à la platitude spectrale On peut également exploiter le rapport entre la moyenne géométrique et la moyenne arithmétique de l'amplitude spectrale, désigné par *Amplitude Spectral Flatness* (ASF) dans la panoplie de descripteurs bas-niveau du standard MPEG-7.

Le facteur de crête spectrale (SCF - *Spectral Crest Factor*) [Peeters, 2004] est également un bon descripteur pour représenter la platitude spectrale. Il est défini comme le rapport entre la valeur maximale et la moyenne du spectre d'amplitude.

Coefficients LPC (Linear Prediction Coding) Nous effectuons une analyse Auto-Regressive (AR) à l'ordre 2 du signal, pour utiliser les deux coefficients (après la constante 1) du filtre AR obtenu comme attributs afin de décrire de façon grossière l'enveloppe spectrale du signal (l'assimilation de la réponse fréquentielle du filtre à l'enveloppe spectrale est supposée).

Pente spectrale : La *pente spectrale* est obtenue au moyen d'une regression linéaire du spectre d'amplitude [Peeters, 2004]. Elle permet de mesurer le taux de décroissance spectrale.

$$S_s = \frac{K \sum_{k=1}^{K} f_k a_k - \sum_{k=1}^{K} f_k \sum_{k=1}^{K} a_k}{K \sum_{k=1}^{K} f_k^2 - \left(\sum_{k=1}^{K} a_k\right)^2}$$

Décroissance spectrale : $S_d = \frac{1}{\sum_{k=2}^{K} a_k} \sum_{k=2}^{K} \frac{a_k - a_1}{k-1}$, donnée par [Peeters, 2004].

Flux spectral : défini dans [Scheirer and Slaney, 1997], il permet de caractériser la vitesse de variation du profil spectral par le calcul d'une corrélation normalisée entre spectres correspondant à des fenêtres d'analyse successives.

$$S_v = 1 - \frac{\sum_{k=1}^{K} a_k(t-1) a_k(t)}{\sqrt{\sum_{k=1}^{K} a_k(t-1)^2} \sqrt{\sum_{k=1}^{K} a_k(t)^2}}$$

Fréquence de coupure : définie comme la fréquence en dessous de laquelle 99% de l'énergie spectrale est contenue dans le spectre.

Irrégularité spectrale : $S_i = X(k+1) - X(k)$, pour $0 \leq k \leq 20$
permet de représenter les relations entre les partiels d'un son musical. Nous adoptons l'approche de Brown qui ne demande pas une étape d'estimation des partiels. L'irrégularité S_i est calculée comme la dérivée fréquentielle du module de la CQT $X(k)$ du signal (calculée avec une résolution d'une tierce).

Intensités de sous-bandes en octave Essid propose également un nouveau descripteur dont l'idée est de capturer de façon sommaire la distribution de puissance des différentes harmoniques du son, sans avoir recours à une étape de détection de fréquences fondamentales. Il s'agit seulement de représenter la structure spectrale des sons intrumentaux, sans s'intéresser aux partiels éventuellement présents dans celle-ci. Ainsi, un banc de filtres en octaves, de réponses fréquentielles triangulaires, permet de représenter les particularités des distributions spectrales de chaque instrument. Sont calculées la log-énergie de chaque sous-bande (OBSI - *Octave Band Signal Intensities*) ainsi que le logarithme du rapport d'énergie des couples de sous-bandes voisines (OBSIR - *Octave Band Signal Intensities Ratios*).

3.2.3 Descripteurs temporels

Taux de passage par zéro (*Zero Crossing Rate*) Il s'agit d'une mesure de la fréquence de passage de la forme d'onde temporelle par l'axe d'amplitude nulle [Kedem, 1986]. Le taux de passage par zéro permet de discriminer les signaux bruités (valeurs ZCR élevées) des signaux non bruités (faibles valeurs ZCR). Le ZCR permet ainsi de distinguer les voisés des sons non-voisés, il est donc utile dans la discrimination parole/musique [Scheirer and Slaney, 1997, Saunders, 1996]. Nous calculons ce descripteur sur les fenêtres d'analyse longues et courtes.

Moments statistiques temporels Nous calculons les descripteurs suivants de façon similaire à celle utilisée pour le calcul des moments spectraux sur les fenêtres courtes, longues, ainsi que sur les enveloppes d'amplitude (sur fenêtres longues). Les dérivées temporelles premières et secondaires sont également calculées.

Coefficients d'Autocorrelation (AC) Ce descripteur est obtenu en gardant les 49 premiers coefficients de la tranformée de Fourier inverse du périodogramme du signal (approximant sa *densité spectrale de puissance*). Il peut être vu comme une représentation de l'enveloppe spectrale.

Modulation d'Amplitude (AM) L'intervalle de fréquences 4–8Hz est généralement très caractéristique de la voix car on y observe un pic lié au débit syllabique moyen ; le *trémolo* est également caractérisé par cet intervalle, tandis que des mesures effectuées dans l'intervalle 10–40Hz permettent de décrire la *granularité* (ou *rugosité*) des sons. Nous calculons ainsi sur chacun des deux intervalles :

- **La fréquence AM :** la fréquence du pic d'amplitude maximale dans l'intervalle

- **L'amplitude AM :** la différence entre l'amplitude maximale sur l'intervalle et l'amplitude moyenne sur la totalité du spectre
- **L'amplitude AM heuristique :** la différence entre l'amplitude maximale et l'amplitude moyenne sur l'intervalle

3.2.4 Descripteurs perceptuels

Loudness spécifique relative La *loudness spécifique* (mesure d'intensité sonore perceptive) est définie dans la bande critique bc par $L(bc) = E(bc)^{0.23}$ où $E(bc)$ est l'énergie du signal dans la bande bc. Nous mesurons la *loudness spécifique relative* : $Ld(bc) = \frac{L(bc)}{L_T}$, où $L_T = \sum_{bc} L(bc)$ est la loudness totale, ce qui permet de rendre le descripteur indépendant des conditions d'enregistrement.

Précision (*Sharpness*) $Sh = 0.11 \frac{\sum_{bc} bc\, g(bc)\, Ld(bc)}{L_T}$
La précision représente une version *perceptuelle* du centroïde spectral calculée à partir de la loudness spécifique selon [Peeters, 2004]. Ici $g(bc)$ représente une correction perceptuelle exponentielle qui atténue l'effet des bandes larges.

Largeur perceptuelle (*Spread*) $Sp = \left(\frac{L_T - \max_{bc} Ld(bc)}{L_T}\right)^2$
Il s'agit d'une mesure de l'écart entre la loudness spécifique maximale et la loudness totale.

Nous calculons en outre pour ces descripteurs perceptuels les dérivées temporelles première et seconde qui permettent de prendre en compte leur évolution temporelle.

3.3 Intégration

La valeur de certains attributs, à court ou long terme, est parfois dénuée de toute pertinence car sujette à d'importantes variations elles même porteuses d'informations. Ainsi, par exemple, [West and Cox, 2004] introduit des attributs décrivant le contraste spectral, mais ne prend en considération que les estimations de variance et de moyenne sur une fenêtre d'une seconde. De même, Saunders, qui définit dans [Saunders, 1996] un système de segmentation léger en temps de calcul, utilise de nombreux attributs dérivés du *ZCR* tels que sa moyenne sur 2.4s, la déviation standard de sa dérivée... Pinquier également, dans [Pinquier et al., 2002] et [Pinquier, 2004], se base exclusivement sur des descripteurs déduit d'attributs rudimentaires court-terme (nombre de segments sur 1s après une segmentation aveugle fine, variation de la modulation d'entropie sur 1s...).

Nous avons décidé, en conséquence, de calculer les moyennes et variances des descripteurs *court-terme* (calculés sur trames courtes) sur la durée d'une fenêtre longue, soit 2 secondes. Ces valeurs constituent de nouveaux attributs *long-terme* (calculés sur trames longues) que l'on a finalement choisi de substituer aux attributs court-terme pour deux raisons : d'une part, la mise en concurrence d'attributs long-terme et court-terme parasite les algorithmes de sélection puisque la répétition des attributs long-terme sur les fenêtres courtes provoque une phénomène d'accumulation d'exemples de même valeur qui introduit un biais important et favorise ainsi

les attributs long-terme (constaté en pratique). La seconde raison est une simple considération de complexité et de temps de calcul : en ne gardant que les attributs long-terme, on divise par 60 (dans notre cas) le nombre d'observations, ce qui est indispensable durant la phase de sélection d'attributs, et largement appréciable durant l'apprentissage des SVM.

On a également tenté d'introduire les valeurs minimales et maximales sur la durée d'une fenêtre longue, mais celles-ci n'apportent pas un gain notoire par rapport aux valeurs de moyenne et de variance (alors qu'elles doublent la taille des données).

Une des perpectives de ce projet est également le complément de ces attributs par les coefficients d'un modèle auto-régressif permettant de synthétiser la covariance d'un groupe d'attributs. Néanmoins, cet aspect sort pour l'instant du cadre de notre stage et n'a pas encore fait l'objet d'une étude approfondie.

3.4 Normalisation

Les valeurs des attributs sont issues de descripteurs de natures physiques différentes, présentant souvent des dynamiques très hétérogènes. Ainsi des attributs possédant des valeurs plus grandes risquent d'avoir une influence plus importante sur le comportement des différents traitements à suivre, même si cela ne reflète pas forcément leur pertinence pour les tâches en question.

A cet effet, il est classiquement fait appel à des techniques de normalisation permettant d'uniformiser les dynamiques des différentes variables. La normalisation choisie pour ce stage est la plus classique ; elle est réalisée de façon linéaire en exploitant les estimations empiriques des moyennes et des variances des attributs. Par exemple, du j-ème attribut calculé sur l exemples, on estime :

$$\mu_j = \frac{1}{l}\sum_{k=1}^{l} x_{k,j} \quad \text{et} \quad \sigma_j^2 = \frac{1}{l-1}\sum_{k=1}^{l}(x_{k,j} - \mu_j)^2.$$

La normalisation en question consiste ainsi à substituer à l'attribut $x_{k,j}$ la valeur suivante :

$$\hat{x}_{k,j} = \frac{x_{k,j} - \mu j}{\sigma_j}$$

ce qui a pour effet d'assurer que les attributs normalisés possèdent une moyenne nulle et une variance unitaire. Les valeurs de moyenne et variance sont calculées sur l'ensemble des valeurs de la base d'apprentissage de chaque descripteur.

3.5 Sélection des attributs

La sélection automatique d'attributs est une problématique assez récente qui suscite depuis une dizaine d'années un intérêt croissant dans la communauté de l'apprentissage artificiel. Elle permet de ne plus avoir à choisir un ensemble restreint de descripteurs qui semblent adaptés à un problème particulier mais de sélectionner automatique ces derniers au sein d'une collection parfois très vaste (plusieurs milliers

de descripteurs par exemple dans le domaine de la bioinformatique), en fonction de leur pertinence. La réduction de la dimension de l'espace de travail est essentielle puisqu'elle offre un gain considérable en temps de calcul et en mémoire (pouvant même être essentiel), et permet en outre d'améliorer les performances du classificateur face à des données inconnues en réduisant au mieux les phénomènes de sur-apprentissage. On espère ainsi, en sélectionnant un panel restreint d'attributs, *débruiter* le contenu de la base de descripteurs.

Bien évidemment se pose la question de définir la *pertinence* : dans le cadre d'un problème de classification, parle-t-on d'une aptitude à séparer les exemples des différentes classes, à les représenter le plus synthétiquement possible, ou encore de l'influence sur les performances du classificateur ? Le troisième cas est bien sûr idéaliste car il est impossible d'établir un lien analytique entre la seule donnée des attributs et l'efficacité de la reconnaissance. Néanmoins des solutions existent, basées sur la répartition des classes dans l'espace des attributs, que nous présentons ici succintement.

3.5.1 Principe

Peeters distingue dans [Peeters and Rodet, 2003] trois sortes d'algorithmes de sélection de descripteurs (que nous nommerons *FSA* par la suite, pour Feature Selection Algorithm) :

Embarqués : où le FSA est partie intégrante du processus de classification
Enveloppeurs : où le FSA exploite les résultats de la classification
Filtres : le FSA précède la classification et en est indépendant

Nous nous intéressons au cas de deux algorithmes *filtres* qui présentent l'avantage d'une implémentation plus simple puisqu'elle permet de sélectionner les attributs indépendamment de la phase de classification.

La sélection des descripteurs implique le choix d'un critère, ce qui est loin d'être un problème trivial dans le cas d'une sélection de type filtre. [Molina et al., 2002] aborde en détail la question de l'évaluation comparée de différents algorithmes de sélection d'attributs par rapport à un critère de pertinence défini formellement. Néanmoins les deux algorithmes que nous exploitons sont basés sur des considérations assez intuitives concernant la séparation des classes.

3.5.2 Algorithme de Fisher

L'algorithme de Fisher, s'inspire de l'*Analyse Linéaire Discriminante* (ALD) également appelée Analyse Discriminante de Fisher ([Duda et al., 2001]), qui permet de déterminer les directions utiles à une bonne discrimination des classes.

L'ALD se base sur la détermination de la meilleure projection linéaire des exemples à séparer (que l'on évalue comme un simple produit scalaire $\boldsymbol{w}^t\boldsymbol{x}$). On cherche à cet effet la direction $\boldsymbol{w}$ de la droite qui sépare au mieux les exemples des deux classes.

Cette condition est réalisée en maximisant le rapport

$$r(\boldsymbol{w}) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{\sigma}_1^2 - \tilde{\sigma}_2^2}$$

appelé *discriminant de Fisher*, où μ_q et σ_q^2 sont respectivement la moyenne et la variance empiriques des projections sur $\boldsymbol{w}$ des exemples de la classe q ($q = 1$ ou 2). Cela revient à maximiser le rapport entre la *dispersion inter-classes* et la *dispersion intra-classe*.

En pratique l'algorithme de sélection que nous utilisons est plus simple car il ne consiste plus à évaluer la direction optimale d'un axe de projection mais simplement le descripteur dont les valeurs séparent au mieux les classes. La valeur du descripteur constitue donc la projection sur l'axe du descripteur. On cherchera donc le descripteur qui minimise le discriminant de Fisher moyenné sur l'ensemble des classes du problème.

3.5.3 IRMFSP

Cet algorithme a été proposé et utilisé avec succès pour la reconnaissance automatique des instruments de musique dans [Peeters and Rodet, 2003]. Le principe est également inspiré de l'ALD, mais la sélection se fait par itérations, en cherchant le descripteur optimal sur l'espace *orthogonal* à celui défini par les descripteurs déjà sélectionnés. On entend par *optimal* le descripteur qui maximise le rapport de la dispersion inter-classes (B_k) sur la dispersion intra-classe (R_k) :

$$r_k = \frac{B_k}{R_k} = \frac{\sum_{q=1}^{Q} \frac{l_q}{l} \|\boldsymbol{\mu}_{k,q} - \boldsymbol{\mu}_k\|}{\sum_{q=1}^{Q} \left(\frac{1}{l_q} \sum_{i_q=1}^{l_q} \|\boldsymbol{x}_{k,i_q} - \boldsymbol{\mu}_{k,q}\| \right)}$$

où Q est le nombre de classes, l_q le nombre d'exemples d'apprentissage de la classe q, $l = \sum_{q=1}^{Q} l_q$ le nombre total d'exemples et k le nombre d'attributs sélectionnés (ou le nombre d'itérations). Les $\boldsymbol{x}_{k,i_q}$ sont les vecteurs des attributs sélectionnés à l'itération k pour la classe q, $\boldsymbol{\mu}_{k,q}$ la moyenne sur ces vecteurs et $\boldsymbol{\mu}_k$ la moyenne sur tous les vecteurs à l'itération k.

L'orthogonalisation permet d'introduire une contrainte de non-redondance des attributs sélectionnés, qui vient corriger ce défaut de l'algorithme de Fisher. Néanmoins on doit être attentif au nombre d'attributs sélectionnés car cette contrainte d'orthogonalisation peut conduire l'algorithme à sélectionner des attributs non pertinents qui *bruitent* les étapes suivantes.

3.5.4 Sélection binaire

Essid introduit également [Essid, 2005] le principe de la sélection *binaire* qui consiste simplement à opérer une sélection indépendante pour chaque problème bi-classes et non une sélection unique sur le critère de la séparation optimale de l'ensemble des classes.

Ainsi on extrait pour chaque paire de classes une liste de descripteurs adaptée à la séparation des deux classes. Cette approche est justifiée ici car les Machines à Vecteurs Supports (voir section 4.1.2 sur les SVMs) exploitées pour la classification

sont également basées sur une séparation bi-classe, ce qui permet ainsi d'adapter au mieux les descripteurs de chaque séparateur. On espère ainsi obtenir un gain en terme de performances. En outre, on exploite dans notre problème uniquement 3 classes : parole (notée *SP* pour speech), musique (notée *MU* pour music), et parole sur fond musical (notée *MIX* pour mixed), si bien que le nombre de paires est suffisamment limité pour que cette approche reste réaliste. Néanmoins, pour un nombre de classes plus grand on prendra garde au fait que la complexité est quadratique ($\frac{Q(Q-1)}{2}$ paires pour Q classes).

Chapitre 4

Théories pour la segmentation audio

4.1 Classification par Machines à Vecteurs Supports

4.1.1 Principe de la classification supervisée

La classification supervisée concerne le cas où les données d'entrée sont organisées en Q classes $\{\Omega_q\}_{(1\leq q\leq Q)}$ connues d'avance. Elle peut être *générative* ou *discriminative.*

L'approche *générative*, assez courante, consiste à déduire des observations d'apprentissage une estimation empirique de la *densité de probabilité à postériori* $P(\Omega_q|\boldsymbol{x})$ à partir des données supposées connues des *densités de probabilités conditionnelles* $p(\boldsymbol{x}|\Omega_q)$, décrivant la distribution des vecteurs d'attributs $\boldsymbol{x}$ relative à la classes Ω_q, et des *probabilités à priori* $P(\Omega_q)$ de chaque classe Ω_q.
La régle de décision bayesienne associe $\boldsymbol{x}$ à la classe Ω_{q_0} si et seulement si :

$$q_0 = \arg\max_{1\leq q\leq Q} P(\Omega_q|\boldsymbol{x})$$

on parle alors de décision au sens du *Maximum A Postériori* (MAP), qui garantie une probabilité d'erreur minimale étant donnée l'observation $\boldsymbol{x}$. Néanmoins le problème qui se pose est que les *densités de probabilités conditionnelles* et les *probabilités à priori* sont inconnues et estimées à partir des exemples d'apprentissage, souvent à la suite de certaines hypothèses sur la structure de ces fonctions (par exemple une somme de gaussiennes pour la classification par GMM, *Gaussian Mixture Model*).

L'approche *discriminatives* ne se base pas sur l'estimation de la densité de probabilité des classes. Elle consiste à estimer la *surface de décision* entre leurs domaines dans l'espace des observations. Cette surface est représentée par une *fonction de discrimination* qui associe à une observation $\boldsymbol{x}$ une valeur $f(\boldsymbol{x})$, seuillée pour déterminer la classe estimée.
Ainsi dans le cas le plus simple de la discrimination *linéaire*, la fonction est une combinaison linéaire des composantes de $\boldsymbol{x}$ que l'on peut exprimer comme suit :

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$$

où $\boldsymbol{w}$, le *vecteur poids*, représente la normale de l'hyperplan de décision défini par l'égalité $f(\boldsymbol{x}) = 0$. Cet hyperplan sépare l'espace en deux parties, l'une associant la classe q_1 aux cas $f(\boldsymbol{x}) > 0$, l'autre associant la classe q_2 aux cas $f(\boldsymbol{x}) < 0$.
En exploitant des fonctions de discrimination non linéaires, on peut étendre la famille des surfaces de décision à des surfaces courbes et éventuellement non connexes. La théorie des *Machines à Vecteurs Supports*, que nous présentons dans ce chapitre, se base sur l'approche discriminative.

4.1.2 Principe des Machines à Vecteurs Supports (SVM)

SVM linéaires

Les Machines à Vecteurs Supports sont de puissants classificateurs qui ont prouvé leur efficacité pour diverses tâches de classification parmi lesquelle l'identification de locuteur, la reconnaissance de caractères [Schölkopf et al., 1999], la reconnaissance de visages [Osuna et al., 1997] et récemment la reconnaissance des instruments de musique [Essid et al., 2006, Essid, 2005].

Il s'agit, à partir d'exemples $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_l, y_l)\}$ dans $\mathbb{R} \times \{-1; +1\}$, de déterminer l'*hyperplan optimal* :

$$H_{\boldsymbol{w}_0, b_0} : \boldsymbol{w}_0 \cdot \boldsymbol{x} + b_0 = 0 \quad \boldsymbol{w}_0 \in \mathbb{R}^d, b_0 \in \mathbb{R}$$

qui maximise la distance à l'hyperplan des exemples les plus proches. C'est à dire solution de

$$\max_{\boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}} \min_{\boldsymbol{x}, i} \{ \|\boldsymbol{x} - \boldsymbol{x}_i\| \, ; \boldsymbol{x} \in \mathbb{R}^d, \boldsymbol{w} \cdot \boldsymbol{x} + b = 0 \} \quad \text{avec } i = 1, \ldots, l$$

en supposant que les données sont linéairement séparables, on peut contraindre chaque exemple $\boldsymbol{x}_i$ à satisfaire les conditions :

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1 \geq 0 \quad \forall i \tag{4.1}$$

Etant donné que la surface de décision reste inchangée si l'on factorise $\boldsymbol{w}$ et b par une même constante, on peut supprimer cette redondance et associer chaque surface de décision à un unique couple $(\boldsymbol{w}, b)$ en imposant la contrainte :

$$\min_{i=1,\ldots,l} |\boldsymbol{w} \cdot \boldsymbol{x}_i + b| = 1$$

Les hyperplans satisfaisant cette condition sont appelés *hyperplans canoniques*. Ils présentent l'avantage, toujours dans le cas de données linéairement séparables, d'assurer l'existence d'un exemple $\boldsymbol{x}_1$ de la classe Ω_1 et d'un exemple $\boldsymbol{x}_2$ de la classes Ω_2, situés respectivement sur les hyperplans

$$H_1 : \boldsymbol{w} \cdot \boldsymbol{x}_i + b = +1 \quad \text{et} \quad H_2 : \boldsymbol{w} \cdot \boldsymbol{x}_i + b = -1$$

qui permettent de définir la *marge*. Remarquons que H_1 et H_2 sont parallèles de normale $\boldsymbol{w}$ et qu'il n'existe aucun point $\boldsymbol{x}_i$ entre les deux ; la *marge* est la distance entre H_1 et H_2 qui vaut donc $\frac{2}{\|\boldsymbol{w}\|}$. La figure 4.1 en donne une illustration. Les points qui se trouvent sur les hyperplans H_1 et H_2 sont appelé *Vecteurs supports*. Le problème posé et sa résolution ne dépendent en fait que de ces points particuliers.

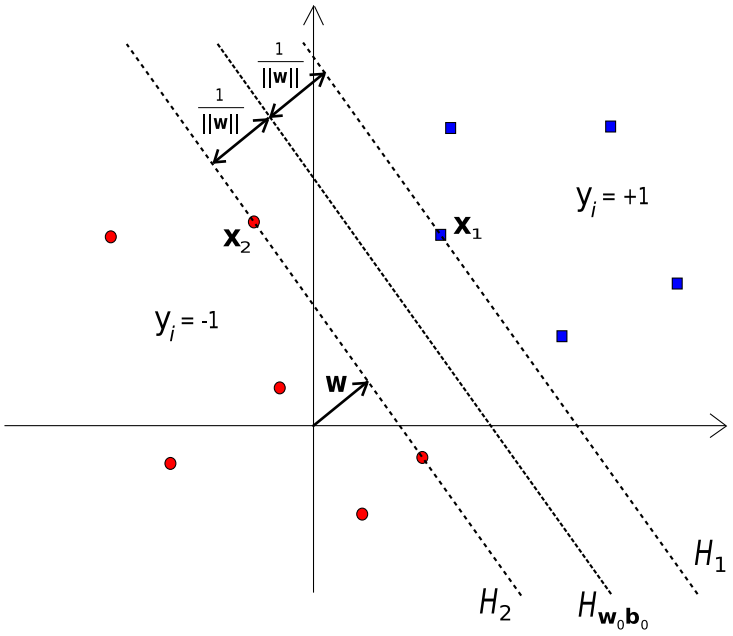


FIG. 4.1 – Hyperplan optimal et marge d'un classificateur SVM. Les ronds représentent des exemples de la classe -1 et les carrés des exemples de la classe +1.

Ainsi l'hyperplan optimal est solution du problème d'optimisation

$$\begin{cases} \text{minimiser} & \tau(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 & \boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R} \\ \text{sous les contraintes} & y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1 \geq 0 & \forall i = 1, \ldots, l. \end{cases}$$

Néanmoins dans des situations plus réalistes, les données sont généralement non séparables par un hyperplan. On introduit une tolérance en assouplissant les contraintes 4.1 par l'introduction de variables d'écart positives ξ_i. La contrainte devient alors

$$\boldsymbol{w} \cdot \boldsymbol{x}_i + b \geq 1 - \xi_i \qquad \text{si } y_i = +1 \tag{4.2}$$

$$\boldsymbol{w} \cdot \boldsymbol{x}_i + b \leq 1 - \xi_i \qquad \text{si } y_i = -1 \tag{4.3}$$

Ce qui implique qu'un exemple d'apprentissage $\boldsymbol{x}_i$ est mal classifié si et seulement si l'écart ξ_i correspondant est supérieur à 1. Par suite, $\sum_i \xi_i$ est une borne supérieure sur le nombre d'erreurs de classification qui peuvent être pénalisées en modifiant la fonction objectif $\tau(\boldsymbol{w})$ par :

$$\tau(\boldsymbol{w}, \boldsymbol{\xi}) = \frac{\|\boldsymbol{w}\|^2}{2} + C(\sum_i \xi_i)$$

où $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_l]^T$ et $C > 0$ est un paramètre permettant de contrôler le compromis entre la maximisation de la marge et la minimisation des erreurs de classification sur l'ensemble d'apprentissage.

L'apprentissage des SVMs (le calcul des paramètres optimaux) se ramène à un problème d'*optimisation sous contraintes*, résolu classiquement par l'approche lagrangienne (introduction d'un *multiplicateur de Lagrange* sur les contraintes) ; nous

n'aborderons pas cette partie de la théorie qui nous semble trop éloignée du cadre de ce stage. L'implémentation pratique des algorithmes de calcul des SVMs nous est déjà fourni et n'a pas constitué un sujet d'étude pour ce travail. On pourra néanmoins consulter [Burges, 1998] et [Osuna et al., 1997] pour une introduction détaillée de la théorie des Machines à Vecteurs Supports.

Introduction de la non-linéarité par le noyau

Il existe une autre réponse au problème des données non linéairement séparables qui mène à l'obtention de surfaces de décision non-linéaires. L'idée est de transformer les données de l'espace des attributs $\mathbb{R}^d$ dans un espace de Hilbert $\mathbb{E}$, de dimension supérieure, dans lequel les données transformées deviennent linéairement séparables. Ainsi, en exploitant une application $\Phi : \mathbb{R}^d \rightarrow \mathbb{E}$, l'algorithme SVM linéaire, appliqué aux données $\Phi(\boldsymbol{x}_i)$ dans l'espace $\mathbb{E}$, produit des surfaces de décision non-planes (éventuellement non-connexes) dans l'espace $\mathbb{R}^d$. De plus du fait que les données apparaissent dans les calculs de l'optimisation sous contraintes uniquement sous la forme de produits scalaires $(\boldsymbol{x}_i \cdot \boldsymbol{x}_j)$, il suffit de trouver une façon efficace de calculer $\Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$. Ainsi l'existence d'une *fonction noyau* k, telle que $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$, nous affranchit de la nécessité de connaître Φ. Le problème reste de savoir quelles conditions une fonction $k(\boldsymbol{x}, \boldsymbol{y})$ symétrique doit remplir pour être associée à un produit interne dans $\mathbb{E}$.

Les *conditions de Mercer* stipulent que $k(\boldsymbol{x}, \boldsymbol{y})$ décrit un produit interne dans $\mathbb{E}$ si et seulement si pour toute fonction $g(\boldsymbol{x})$ sur $\mathbb{R}^d$, de norme $\mathcal{L}_2$ finie, la condition suivante est satisfaite

$$\int_{\boldsymbol{x}} \int_{\boldsymbol{y}} k(\boldsymbol{x}, \boldsymbol{y})\, g(\boldsymbol{x})\, g(\boldsymbol{y})\, d\boldsymbol{x}\, d\boldsymbol{y} \geq 0$$

La condition ne permet pas la construction analytique des noyaux possibles mais nous garantie la validité des noyaux suivants, d'usage courant dans la littérature :

- le noyau *linéaire* $k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x} \cdot \boldsymbol{y}$
- le noyau *polynômial* de degré δ $k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y})^{\delta}$
- le noyau *radial exponentiel* $k(\boldsymbol{x}, \boldsymbol{y}) = exp(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2})$

Nous nous sommes uniquement intéressés dans ce stage au noyau radial exponentiel (ou *noyau gaussien*) qui montre de meilleures aptitudes à la généralisation (réduction des phénomènes de sur-apprentissage) pour des performances similaires à celles des noyaux polynômiaux, d'après les résultats d'Essid [Essid, 2005].

On notera que le choix de la variable σ est crucial car il permet de contrôler la courbure des surfaces de décision. La figure 4.2 (extraite de l'ouvrage [Schölkopf and Smola, 2002]) montre les surfaces de décision correspondant à des valeurs décroissantes de σ. On note qu'il est important de trouver le juste compromis entre la minimisation des erreurs d'apprentissage et le risque de sur-apprentissage. L'image centrale présente pour cet exemple un compromis acceptable.

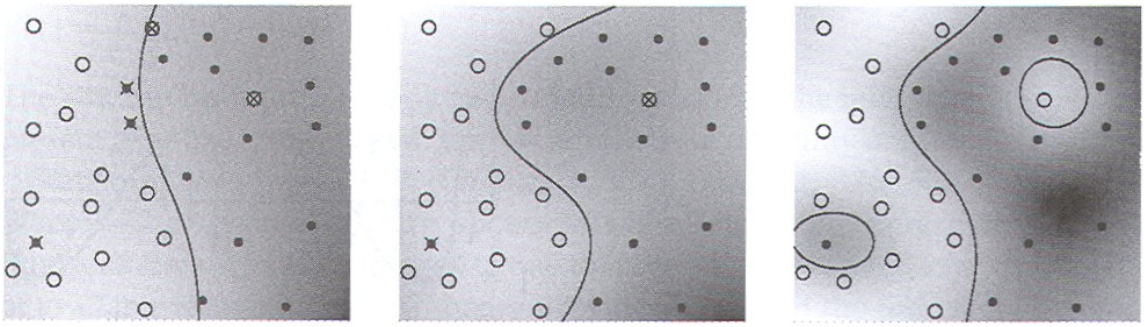


FIG. 4.2 – Effet du paramètre σ sur la courbure de la surface de décision. Ici σ est diminué de gauche à droite.

Performances en généralisation

Les SVM présentent en pratique de très bonne performances en généralisation (c'est à dire sur la classification d'exemples inconnus). La marge joue en cela un rôle important ; on conçoit intuitivement qu'une marge importante sur les exemples d'apprentissage améliore les chances de classifier correctement les nouveaux exemples, ces derniers pouvant demeurer correctement classés même s'ils sont situés dans la marge.

De plus les SVM sont connues pour défier ce que l'on appelle le *"curse of dimensionality"* puisqu'elles sont capables de fournir de bonnes performances de classification à partir d'un nombre réduit d'exemples d'apprentissage tout en agissant dans des espaces de dimension très élevée.

4.1.3 Fusion de décisions binaires

Les Machines à Vecteurs Supports sont des séparateurs bi-classes ("un contre un"). Dans un problème à Q classes, il s'agit donc d'évaluer pour chacune des $\frac{Q(Q-1)}{2}$ paires de classes $\{\Omega_p, \Omega_q\}_{1 \leq p < q \leq Q}$, le classificateur optimal à la discrimination des classes Ω_p et Ω_q. Une *stratégie de fusion* est donc nécessaire afin de ramener la décision à un choix entre les Q classes.

Hastie et Tibshirani proposent une solution efficace au problème de fusion des décisions binaires [Hastie and Tibshirani, 1998] qui permet d'estimer les probabilités $P(\Omega_q|\boldsymbol{x})$. L'algorithme se base sur l'estimation itérative des densités de probabilités $p_q(\boldsymbol{x}) = P(\Omega = \Omega_q|\boldsymbol{x})$ et de $r_{qm}(\boldsymbol{x}) = \frac{p_q(\boldsymbol{x})}{p_q(\boldsymbol{x})+p_m(\boldsymbol{x})}$ la probabilité que la classe correspondant à l'observation $\boldsymbol{x}$ soit Ω_q dans le problème bi-classes (Ω_q vs Ω_m).

A partir de valeurs initiales pour les estimées $\hat{p}_q(\boldsymbol{x})$ et $\hat{r}_{qm}(\boldsymbol{x})$, on actualise $\hat{p}_q(\boldsymbol{x})$ à chaque itération :

$$\hat{p}_q(\boldsymbol{x}) \leftarrow \hat{p}_q(\boldsymbol{x}) \frac{\sum_{q<m} n_{qm} r_{qm}(\boldsymbol{x})}{\sum_{q<m} n_{qm} \hat{r}_{qm}(\boldsymbol{x})}$$

On renormalise ensuite les $\hat{p}_q(\boldsymbol{x})$ afin de respecter la condition $\sum_q \hat{p}_q(\boldsymbol{x}) = 1$, puis on recalcule les probabilités $\hat{r}_{qm}(\boldsymbol{x})$. Hastie et Tibshirani montrent que la distance de Kullback-Leibler entre $r_{qm}(\boldsymbol{x})$ et $\hat{r}_{qm}(\boldsymbol{x})$ décroît à chaque itération, ce qui montre la convergence de l'algorithme, car $\hat{r}_{qm}(\boldsymbol{x})$ est positif.

On obtient alors une estimation de la probabilité à postériori pour chacune des classes Ω_q. On en déduit la décision pour l'observation $\boldsymbol{x}$ de la classe Ω_{q_0}, où $q_0 = \arg\max_q \hat{p}_q(\boldsymbol{x})$.

4.2 Segmentation aveugle

4.2.1 Principe général

Contrairement aux approches supervisées, la segmentation aveugle ne suppose aucune connaissance à priori sur la nature des différentes classes qui composent le signal. On cherche juste à segmenter celui-ci en se basant sur la détection de nouveauté (ou de changement). Ceci suppose donc que le signal est constitué d'une succession de segments localement homogènes aux limites desquels on observe une forte variation des propriétés locales du signal. La détection de nouveauté est un problème ouvert puisque la quantification de la variation d'un signal dépend des propriétés que l'on souhaite prendre en compte.

Notre approche se base sur l'usage d'une fenêtre glissante $W(k_0)$ de $2L+1$ trames, centrée sur la trame k_0. On considère k_0 comme un bon condidat pour une frontière entre segments si le contenu des données "*futures*" $(S_2(k_0) = \boldsymbol{x}(k), k \in [k_0, k_0 + L])$ est *nouveau* relativement au contenu des données "*passées*" $(S_1(k_0) = \boldsymbol{x}(k), k \in [k_0 - L, k_0])$. Elle se base sur l'implémentation de Gillet [Gillet and Richard, 2006] qui a comparé les performances de plusieurs critères de nouveauté que nous détaillons par la suite. Les fenêtres passée et future représentent 5 secondes de signal dans notre cas, soit $L = 5$.

Chacune des fonctions de nouveauté est calculée sur l'ensemble des trames longues du signal, et un algorithme de détection des maxima locaux est appliqué pour en extraire les frontières des segments.

4.2.2 Distances probabilistes

Le critère le plus simple pour la détection de nouveauté est l'évaluation de distances probabilistes entre les deux ensembles d'observations S_1 et S_2, basées sur l'estimation des densités de probabilité $\hat{p}_1(\boldsymbol{x})$ et $\hat{p}_2(\boldsymbol{x})$. On trouve une liste des distances couramment utilisées dans [Zhou and Chellappa, 2006]. Seules deux distances sont exploitées ici (pour leur coût réduit) parmi la liste présentée dans l'article :

Distance de Bhattacharyya : $B(p_1, p_2) = -\log\left(\int_{\boldsymbol{x}} [p_1(\boldsymbol{x}) p_2(\boldsymbol{x})]^{\frac{1}{2}} d\boldsymbol{x}\right)$

Divergence de Kullback-Leibler : $KL(p_1, p_2) = \int_{\boldsymbol{x}} p_1(\boldsymbol{x}) \log\left(\frac{p_1(\boldsymbol{x})}{p_2(\boldsymbol{x})}\right) d\boldsymbol{x}$

basée sur la théorie de l'Information, elle représente le débit additionnel nécessaire pour coder les données de S_2 avec un codage qui est optimal sur les données de S_1. On retrouve son utilisation, sous sa forme symétrisée, pour la segmentation de bulletins d'information radiophoniques en langue anglaise dans [Siegler et al., 1997].

L'évaluation des densités de probabilités de S_1 et S_2 se fait en utilisant un algorithme permettant d'évaluer la distance entre deux ensembles à partir de la définition de la distance entre chacun de leurs points [Zhou and Chellappa, 2006]. Les ensembles

sont représentés par une simple gaussienne, après projection par une fonction noyau (se référer à la section 4.1.2 pour une présentation des fonctions noyaux) dans un espace de Hilbert dit RKHS (pour *Reproducing Kernel Hilbert Space*). Ce passage au niveau ensembliste est nécessaire pour des raisons de complexité algorithmique ; l'algorithme en question est implémenté par Gillet dans le code que nous exploitons.

4.2.3 Bayesian Information Criterion (BIC)

Le critère d'information bayesienne constitue une alternative aux distances probabilistes, d'usage désormais généralisé, y compris dans le domaine de la segmentation audio [Zhou and Hansen, 2000]. Le critère BIC est connu pour être plus robuste que les distances probabilistes courantes [Chen and Gopalakrishnan, 1998], et converge vers la solution optimale, sur un problème défini par deux densités gaussiennes.

Son évaluation se base en effet sur la modélisation gaussienne des densités de probabilités des observations des fenêtres. On cherche à évaluer le modèle le plus vraisemblable entre l'hypothèse H_0 d'un modèle unique $\mathcal{N}(\mu, \Sigma)$ sur l'ensemble de la fenêtre $W(k_0)$ et l'hypothèse H_1 des deux modèles différents $\mathcal{N}(\mu_1, \Sigma_1)$ et $\mathcal{N}(\mu_2, \Sigma_2)$ pour les fenêtres passée (S_1) et future (S_2). La variation du BIC s'exprime par le critère du maximum de vraissemblance, pénalisé par une valeur K ici constante (dépendant de la dimension de l'espace) :

$$\Delta BIC(i) = \frac{1}{2}\left[(2L+1)log|\Sigma| - L \cdot log|\Sigma_1| - L \cdot log|\Sigma_2| - K\right]$$

L'implémentation profite du caractère glissant de la fenêtre $W(k_0)$ pour ne pas recalculer entièrement les matrices de covariances utilisées pour chaque trame [Zhou and Hansen, 2000].

4.2.4 SVMs à une classe

Enfin, l'usage des SVMs est une technique récente mais déjà largement exploitée pour la détection de nouveauté [Schölkopf et al., 1999]. Nous utilisons deux critères se basant sur les SVMs à une classe, qui permettent d'identifier la région de l'espace RKHS (après projection des vecteurs d'observation par la fonction noyau) où se situent la majorité des observations. On identifie pour cela l'hyperplan qui maximise la marge entre les observations et l'origine [Schölkopf and Smola, 2002].

Log-Likelihood Ratio Le premier critère [Loosli et al., 2005] se base sur le test de *rapport de vraissemblance* suivant :

$$R = \frac{\prod_{\boldsymbol{x}\in S_1} P_1(\boldsymbol{x}) \prod_{\boldsymbol{x}\in S_2} P_2(\boldsymbol{x})}{\prod_{\boldsymbol{x}\in W} P_1(\boldsymbol{x})} = \frac{\prod_{\boldsymbol{x}\in S_2} P_2(\boldsymbol{x})}{\prod_{\boldsymbol{x}\in S_2} P_1(\boldsymbol{x})} > t.$$

Les estimations de P_1 et P_2 sont déduites de la solution de l'agorithme SVM.

Kernel Change Detection Le second critère, se base sur la détection de changement du noyau, par une mesure de dissimilarité que l'on peut exprimer comme le rapport de l'étalement inter-classes sur l'étalement intra-classe dans l'espace transformé RKHS.

4.2.5 Extraction des pics

L'ampleur dynamique des fonctions de nouveauté calculées $d(n)$ (figure 4.3a) nous oblige à appliquer un post-traitement pour faciliter la détection de changement.

Une première étape consiste à normaliser $d(n)$ en lui soustrayant le résultat d'un filtrage médian sur une fenêtre glissante de $2W_a + 1$ trames. W_a est fixé à 60 dans notre cas, soit une minute. Ce filtrage associe à chaque trame la moyenne observée sur la fenêtre centrée autour de cette trame.
Les variations locales des amplitudes des pics sont ensuite compensées en divisant la fonction par le résultat d'un filtrage de déviation standard sur une fenêtre glissante de $2W_a + 1$ trames (figure 4.3b), qui associe à chaque trame la variance mesurée sur la fenêtre centrée autour.

La détection de pics est sujette à deux contraintes : les pics doivent être séparés d'un nombre minimal de W_b trames ($W_b = 3$ dans notre cas, ce qui correspond à 3 secondes minimum entre deux pics voisins) et doivent être au dessus d'un *seuil* τ. Une fonction *plateau*, qui associe à chaque trame le maximum observé sur la fenêtre de $2W_b + 1$ trames centrée autour, permet de définir un plateau constant avoisinant les maxima locaux, camouflant les pics voisins parasites.

On extrait alors la valeur maximale de la fonction normalisée observée sur chaque plateau dépassant le seuil τ (figure 4.3b).

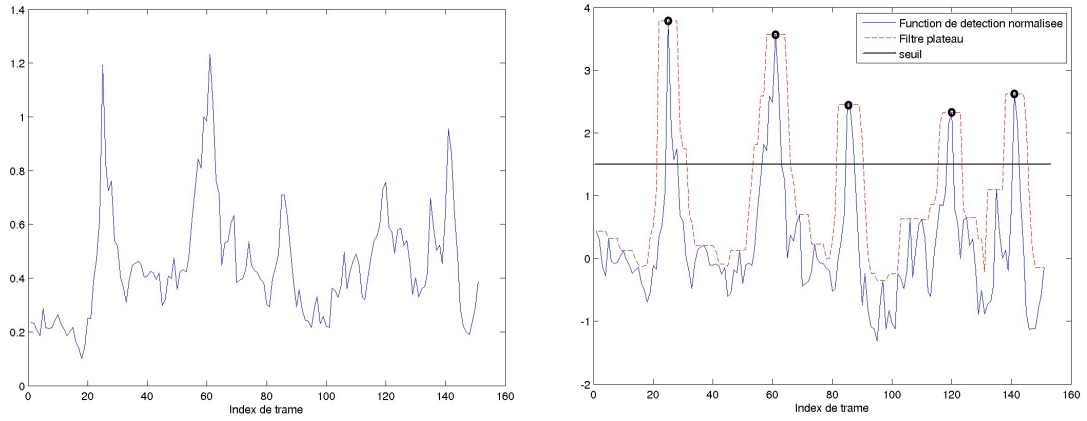


FIG. 4.3 – (a) Fonction de nouveauté avant le post-traitement (b) Extraction des pics après normalisation de la fonction

4.3 Approche hybride

Il apparaît difficile de comparer les deux approches puisque la segmentation aveugle n'apporte aucune information quant à la nature des segments isolées. Néanmoins,

cette dernière peut constituer un apport essentiel à la classification supervisée, qui ne définit les fontières de segments qu'à posteriori à partir de la classe reconnue sur chaque trame.

Pinquier tire profit des informations liées à une segmentation aveugle [Pinquier et al., 2002] en construisant des descripteurs pour la segmentation parole/musique représentant le nombre moyen de segments dans une seconde et leur durée moyenne sur une seconde, après une segmentation fine par l'algorithme "*Forward-Backward Divergence*" basé sur les propriétés statistiques du signal.

Nous utilisons les résultats de la segmentation aveugle comme post-traitement des résultats de la segmentation par classification. En utilisant pour la segmentation aveugle la sélection d'attributs calculée sur les classes souhaitées, nous assimilons l'homogénéité locale du signal sur chaque segment à une uniformité du résultat de classification. On se contentera donc de sommer les résultats probabilistes pour chaque classe sur l'ensemble des trames de chaque segment, pour lui associer la classe majoritaire. Nous étudierons l'impact de cet apport complémentaire sur les performances générales de la classification.

Cependant cette approche a l'inconvénient de ne pas exploiter l'homogénéité locale du segment lors du test de classification. Une approche réellement hybride consiste à exploiter l'ensemble des données de chaque segment pour un résultat de classification unique, et constitue une des perspectives majeure de notre stage, que nous n'avons pas encore eu le temps d'aborder, pour des raisons de complexité d'implémentation.

On notera également que la segmentation non-supervisée peut également être exploitée pour regrouper (d'après un critère de similarité) des segments découpées arbitraitrement, pour ensuite associer une classe à chaque groupe de segments, basée sur les descripteurs de l'ensemble des trames [Kemp et al., 2000]. Le regroupement des segments par similarité constitue également une des perspectives futures liée à la segmentation non-supervisée, sur ce projet.

Chapitre 5

Expériences et résultats

5.1 Corpus d'évaluation

L'évaluation objective des performances est un aspect essentiel mais néanmoins complexe de la recherche en segmentation audio. Le corpus de test doit généralement être annoté manuellement (ce qui représente une lourde charge pour le laboratoire) et devrait être commun à la plupart des publications dans le domaine, pour pouvoir proposer une évaluation type à tous les laboratoires et permettre ainsi la comparaison des résultats. Trop souvent peut-on lire en conclusion de certains articles des pourcentages très optimistes dont la portée se limite malheureusement à une base constituée sur mesure et décrite en quelques lignes.

Afin d'apporter une réponse claire à ces considérations nous avons choisi de nous focaliser dans le cadre de ce stage sur le corpus de la campagne ESTER.

5.1.1 Présentation de la campagne ESTER

ESTER, pour Evaluation des Systèmes de Transcription enrichie d'Emission Radiophoniques, est un projet né en 2003 de la réunion d'intérêts communs à plusieurs laboratoires de recherche dans le domaine de la transcription automatique de parole, et proposé par l'AFCP (*Association Francophone de la Communication Parlée*). Celle-ci défini un cadre commun pour les différents laboratoires en concurrence, dont les systèmes sont évalués par un acteur extérieur impartial, représenté par le Centre d'Expertise Parisien de la Délégation Générale pour l'Armement (DGA/CEP).

La majorité des tâches définies concerne la transcription et l'indexation de parole, et couvre toute la chaîne qui permet, à partir du signal audio, et en passant par la reconnaissance de locuteurs et la transcription de la parole, d'obtenir une base textuelle organisée autour de la catégorisation automatique du contenu et des mots clés du message sonore (recherche d'"*entités nommées*"). Néanmoins la première de ces tâches concerne en toute logique précisemment la localisation des segments de parole et de musique. Si la segmentation et la transcription peuvent être traitées de manière tout à fait indépendantes, on comprend aisément en quoi la première est indispensable à la seconde pour un signal complétement inconnu.

La campagne s'est déroulée en deux phases. La première [Gravier et al., 2004]

a posé en 2003 les bases protocolaires des différentes tâches de la campagne. Elle présente également la première partie des ressources d'apprentissage et de test mises à la disposition des participants. La présentation de la phase II [Galliano et al., 2004], publiée en 2005, présente en détail les protocoles d'évaluation ainsi que les performances comparées des systèmes des laboratoires participants. On compte parmi ceux-ci France Télécom R&D [Blouch and Collen, 2005], l'ENST/IRISA [Gravier et al., 2005] (la publication s'intéresse surtout au problème de la transcription mais propose également un système de segmentation assez simple), l'IRIT à Toulouse [Calmès et al., 2005] et le LIA de l'Université d'Avignon [Scheffer et al., 2005]. La phase II s'accompagne également d'un corpus additionnel d'apprentissage.

Le corpus contient un certain nombre d'heures d'enregistrements d'informations radiophoniques annotés ou non ainsi que des transcriptions textuelles de journaux. Nous n'exploitons bien sûr que les enregistrements annotés dans le cadre de ce stage. Les documents sonores proviennent des radios France Inter, France Info, RFI, RTM (Radio Télévision Marocaine), ainsi le tableau 5.1 résume la durée des enregistrements de chaque radio disponibles dans les corpus (*appr* pour le corpus d'apprentissage, *test* pour le corpus de test) des deux phases.

source	phase I	phase II	
	appr	appr	test
France Inter	25h	10h	2h
France Info	-	10h	2h
RFI	15h	10h	2h
RTM	-	20h	2h
Culture + Musique	-	-	2h
total	40h	50h	10h
année	1998-2000	2003	2004

TAB. 5.1 – Contenu des corpus d'apprentissage et de test de la campagne ESTER

Nous avons réuni pour notre étude les deux corpus d'apprentissage en un seul et conservé le corpus de test de la phase II comme corpus d'évaluation, ce qui représente donc 90h d'apprentissage (et de développement) et 10h d'évaluation.

Les extraits sont échantillonnés à 16kHz, en 16 bits sur un seul canal, dans un fichier wav non compressé. Les annotations sont fournies sous la forme d'un fichier XML contenant la structure hiérarchique de l'enregistrement ainsi que la transcription textuelle de son contenu.

La tache SES de segmentation est en fait la réunion de deux segmentations indépendantes : parole/non parole et musique/non musique. Le chevauchement de segments de parole et de musique représente la parole sur fond musical. Néanmoins, ces passages particuliers correspondent en fait pour la majorité à la présentation des titres d'un journal radio sur un léger fond musical, et diffèrent largement des segments de musique seule (généralement des chansons ou des jingles). C'est pourquoi nous avons préféré baser notre système sur une segmentation à trois classes (parole, musique et parole sur fond musical), que nous jugeons plus pertinente. Les résultats de cette segmentation sont traduits au final en une collection de segments de parole ou de musique.

La base étant essentiellement destinée à la transcription de parole, la répartition des 3 classes est largement disproportionnée puisque que l'on trouve, sur la base d'apprentissage de 90h, 77h30 de parole seule, 11h45 de parole sur fond musical et seulement 40 minutes de musique seule. Soit un ratio de 116 sur 1 entre parole et musique et 6.6 sur 1 entre parole et parole sur fond musical. Cette répartition introduit un biais désastreux si l'on ne prend pas soin de fournir une volume à peu près égale des trois classes lors de la sélection d'attributs et de l'apprentissage, dont on expliquera les conséquences par la suite. Ceci implique en particulier que nous avons dû sélectionner un sous-ensemble très restreint de la base d'apprentissage pour la classe parole (autant que de musique) ainsi que pour la parole sur fond musical, pour les phases de sélection d'attributs et d'apprentissage.

5.1.2 Protocole d'évaluation

Outils d'évaluation

L'évaluation confronte les résultats de segmentation obtenus aux informations contenues dans les fichiers d'annotation. Les fichiers `trs` au format XML se convertissent à l'aide d'un utilitaire fourni par ESTER, en fichiers texte (fichier `etf` pour *Event Tracking Format*) détaillant les seules informations d'événements. Chaque ligne signale ainsi la présence ou l'absence de musique ou de texte dans un segment dont l'instant de début et la durée sont précisées. Ainsi en évaluant l'intersection des segments de musique et de parole on obtient les segments de parole sur fond musical.

Notre système produit donc pour chaque fichier sonore (wav) analysé un fichier `etf`, décrivant les événements reconnus, qui est confronté au fichier `etf` décrivant le contenu réel du fichier, à l'aide de l'utilitaire `trackeval`, également fourni par ESTER, qui calcule automatiquement les résultats obtenus sur le fichier en question, et garantie ainsi la neutralité de l'évaluation.

Rappel, précision, F mesure

Nous avons expliqué que le format `etf` décrit des *événements*, que l'on peut assimiler aux limites entre segments consécutifs. Ces événements sont mesurés en secondes, avec une tolérance de 0.25s. Si l'on indexe par i l'ensemble des événements (réel ou détectés) sur le signal, on définit les trois valeurs suivantes :

- $t(c_i; c_i)$ vaut 1 si l'événement i est correctement détecté, 0 sinon
- $t(\bar{c}_i; c_i)$ vaut 1 si l'événement i est manqué , 0 sinon
- $t(c_i; \bar{c}_i)$ vaut 1 pour tout fausse alerte, 0 sinon

Ainsi, $\sum_i t(c_i; c_i)$ représente le nombre d'événements correctement détectés, $\sum_i t(\bar{c}_i; c_i)$ le nombre d'événements manqués et $\sum_i t(c_i; \bar{c}_i)$ le nombre de fausses alertes sur l'ensemble du signal.

On déduit de ces valeurs les mesures de *rappel* (R) et de *précision* (P) définies comme suit :

$$R = \frac{\sum_i t(c_i; c_i)}{\sum_i t(c_i; c_i) + t(\bar{c}_i; c_i)} \text{ et } P = \frac{\sum_i t(c_i; c_i)}{\sum_i t(c_i; c_i) + t(c_i; \bar{c}_i)}$$

Intuitivement, le *rappel* est le taux d'événements réels détectés et la *précision* est le taux d'événements corrects parmi les événements détectés. $R = 1$ représente un

système qui détecte tous les événements réels tandis que $P = 1$ représente un système qui ne commet aucune erreur sur ses prédictions. La *F-mesure* réunit ces deux mesures et est utilisée pour l'évaluation des systèmes dans le cadre de la campagne ESTER. Elle est définie par :

$$F = \frac{2RP}{R+P},$$

sorte de compromis entre une moyenne arithmétique et géométrique.

Nous exploitons également le taux de trames correctement classifiées (et non plus les événements) afin d'évaluer les performances du système sur les trois classes (en effet après la conversion des résultats en événements on n'a plus d'information sur les classes puisque le problème est redéfini en deux problèmes "parole/non parole" et "musique/non musique").

5.1.3 Corpus additionnel RWC

Pour pallier à l'insuffisance de la classe musique dans le corpus d'apprentissage ESTER, nous avons tenté d'ajouter des extraits musicaux qui permettraient de renforcer l'apprentissage de cette classe.

Les extraits choisis proviennent de la base de genres musicaux du corpus RWC [Goto et al., 2003]. Le corpus RWC (*Real World Computing*) est la première base de donnée musicale spécialement dédiée à la recherche. Les droits sont détenus par l'Institut National de l'AIST au Japon, ce qui leur permet de vendre le corpus a bas prix aux laboratoires de recherche. Elle est constituée de plusieurs bases contenant plusieurs dizaines de titres de musique populaire, classique, jazz ou d'intruments de musique seuls... la base qui nous intéresse est constituée de 9 CDs audio offrant un large panel de 100 titres couvrant pratiquement tous les genres musicaux dans le monde, ceci afin de représenter au mieux ce que l'on pourra indexer comme "musique", dans toute sa diversité. Les enregistrements sont en qualité CD (44,1 kHz, 16 bits) sur un seul canal, ce qui signifie que nous avons dû sous-échantillonner les extraits à 16 kHz pour pouvoir constituer un corpus commun avec les extraits ESTER.

Nous avons donc comparé les performances de notre système en complétant soit avec l'ensemble de la base en question, soit avec un résumé de la même taille que le volume de musique dans ESTER. Il n'est pas dit que l'apport des extraits de RWC soit significatif, car ils pourraient très bien perturber la bonne classification des jingles radiophoniques, qui sont assez différents de la musique "générale".

5.2 Protocole expérimental

Nous avons commencé par réunir l'ensemble des données des corpus ESTER (100h de bulletins d'information radiophoniques) et RWC (10h30 de musique de tous les genres), afin d'en extraire tous les descripteurs présentés dans la section 3.2, sur des trames courtes de 32 ms et des trames longues de 2 s.
Nous constituons à partir de ces données les 5 bases détaillées ci-dessous.

Bases préliminaires Les deux premières bases servent d'étude préliminaire à cette expérience, afin d'évaluer l'efficacité trame à trame de notre système sur des bases

de volume restreint (40 minutes pour chaque classes, ce qui garantie une confiance suffisante sur des résultats à une décimale), constituées en collant bout à bout des extraits de chaque classe afin de construire des fichiers ne contenant qu'une seule classe.

`DB_ART` : la première base est destinée à évaluer l'efficacité trame à trame de notre système sur des classes clairement identifiables. En effet les extraits de la classe parole proviennent d'ESTER, ceux de la classe musique proviennent de RWC, et les extraits de parole sur fond musical sont des fichiers générés en mixant à parts égales des extraits des deux bases.

`DB_ESTER` : les trois classes ne proviennent que du corpus ESTER (les extraits sont localisées d'après les fichiers d'annotations). On espère ainsi estimer l'efficacité trame à trame sur la base ESTER.

Chacune de ces bases est séparée en deux ensembles : une *base d'apprentissage* et une *base de test*, de tailles égales (soit 1h en tout pour chaque base, 20 min pour chaque classe).

Bases d'étude Les bases sur lesquelles porte le fond de notre étude ne sont plus des collections de fichiers d'une seule classe mais les séquences de la base ESTER, dont on extrait la segmentation correcte à partir des fichiers ETF (voir 5.1.2). On a conservé l'ensemble de test de la phase II comme *base de test* pour notre expérience (soit 10h d'enregistrement), le reste étant réparti entre une *base d'apprentissage* et une *base de développement* (90h en tout). La base de développement permet d'opérer une première évaluation de notre système sur des exemples distincts des exemples d'apprentissage, afin d'affiner les divers paramètres pour accroître les performances du système. En effet, l'affinage heuristique des paramètres sur la base de test introduit une connaissance de la base de test dans la constitution du système, ce qui est tout à fait contraire à une démarche valide d'apprentissage. On a extrait pour la base de développement un volume à peu près égal à celui de la base de test. On suppose une corrélation relative entre les performances obtenues sur les deux ensembles. La base d'apprentissage contient donc environ 80h d'enregistrement, et la base de développement 10h.

Ici la répartition des volumes des trois classes est naturellement inégale, ce que nous avons cherché à compenser en introduisant des extraits de la base RWC. Ceux-ci sont uniquement ajoutés à la base d'apprentissage. Etant donné que la mesure sur la base de test ne prend en compte que les séquences de la base ESTER, nous n'avons pas introduit d'extraits musicaux de RWC dans la base de développement.

On a donc constitué les trois bases suivantes :

`SEQ_ESTER` : contient toutes et uniquement les séquences sonores de la base ESTER.

`SEQ_ESTER++` : on a ajouté à la base d'apprentissage de `SEQ_ESTER` le contenu entier du corpus RWC (soit 10h30), ce afin d'évaluer le gain éventuel d'un apport conséquent dans la classe musique.

`SEQ_ESTER+` : devant la disportion de l'apport provenant de la base RWC, par rapport au contenu en musique d'ESTER, dans la base `SEQ_ESTER++` (10h30 provenant de RWC pour seulement 40 minutes provenant d'ESTER), nous avons jugé que l'impact pouvait être plutôt néfaste que bénéfique. La base `SEQ_ESTER+` est

constituée en n'ajoutant de la base RWC qu'une quantité égale en musique (soit 40 minutes) à celle de la base ESTER.

Sélection d'attributs Pour chacune de ces bases on a appliqué les deux algorithmes de sélection d'attributs (Fisher et IRMFSP), en sélection binaire ou sur toutes les classes, en se limitant à un nombre restreint d'attributs sélectionnés. On commentera les résultats de cette sélection plus loin. Etant donné que le résultat est un classement de descripteurs, on pourra tronquer celui-ci pour fournir aux machines à vecteurs supports des observations de dimension inférieure.

Apprentissage L'apprentissage des SVMs est effectué (sur la base d'apprentissage) pour chacune de ces 4 sélections, en faisant varier la dimension (nombre d'attributs). L'essai de ces machines sur la base de développement nous permet ainsi de sélectionner la solution optimale quand au choix de l'algorithme de sélection d'attributs, l'usage ou non de la sélection binaire, et la dimension optimale pour l'apprentissage.

Classes équivalentes La répartition égale du volume attribué à chaque classe dans la base d'apprentissage est fondamentale. En effet, la prédominance disproportionnée des exemples d'une classe (comme c'est le cas pour la classe parole dans la base ESTER) induit un comportement monoclasse sur les SVMs ou même, au pire, complétement erronné. Si l'on évalue le taux de trames correctes on croit alors obtenir de très bons résultats du fait de la répartition inégale des classes. Nous avons pour cela adapté le programme afin de ne fournir qu'un volume égal sur chacune des classes pour les algorithmes de sélection d'attributs et l'apprentissage des SVMs. Les exemples sont choisis de manière aléatoire dans la classe majoritaire.

Optimisation des SVMs On cherche ensuite à affiner l'apprentissage de la machine à vecteur support choisie. Celle-ci est déterminée par trois paramètres (dont l'influence est égalemente étudiée par Essid [Essid, 2005]) :

Le paramètre C : pondère la pénalisation du nombre d'erreurs de classification dans le problème d'optimisation de l'hyperplan de décision. Nous avons expliqué section 4.1.2 que celui-ci permet de contrôler le compromis entre la maximisation de la marge et la minimisation des erreurs de classification sur l'ensemble d'apprentissage. Nous avons testé sur l'une des bases une grande variété de valeurs pour C (de 0.2 à 100) mais on obtient systématiquement de meilleurs résultats en utilisant la valeur par défaut fixée de façon adaptive à partir des exemples d'apprentissage. Cette valeur, notée C_{dat} est obtenue comme l'inverse de la longueur moyenne des l exemples d'apprentissage transformés $\Phi(\boldsymbol{x}_i)$, en prenant :

$$C_{dat} = \frac{1}{\frac{1}{l}\sum_{i=1}^{l} k(\boldsymbol{x}_i, \boldsymbol{x}_i)}$$

La taille θ des sous-ensembles de travail : qui permet d'affiner la décomposition de l'ensemble des exemples d'apprentissage en sous-problèmes de taille moindre.

Nous verrons que l'impact est pratiquement nul sur les performances du classificateur. Nous avons conservé la valeur optimale de $\theta = 20$ par défaut déterminée dans [Essid, 2005].

Le paramètre σ : qui influence la complexité des surfaces de décision. Les choix intéressant de σ se situent dans l'intervalle $[0; 1]$, du fait d'une mise à l'échelle de la fonction noyau obtenue en la factorisant par $\frac{1}{d}$ (où d est la dimension des vecteurs d'observation) :

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

La valeur par défaut (avant affinage du paramètre) est de $\sigma = 1$.

Nous pourrions également évaluer les différents types de noyaux, mais nous nous sommes limités dans ce stage au noyau gaussien. L'optimisation du classificateur portera essentiellement sur la détermination du paramètre σ qui maximise les performances sur la base de développement.

Post-traitement par segmentation aveugle On conclut l'expérience en calculant les 5 fonctions de nouveauté présentées dans la section 4.2 :

BAT : distance probabiliste de Bhattacharyya

DIV : divergence de Kullback-Leibler

BIC : Critère d'Information Bayesien

KCD : Kernel Change Detection : évaluation de distance entre noyaux obtenus sur des SVMs à une classe.

LLR : Log Likelihood Ratio, basé sur le rapport entre les estimations sur les fenêtres passée et future par les SVMs à une classe.

On teste ensuite les performances obtenues sur la base de développement en appliquant différentes valeurs de seuil τ pour la détermination des pics de -0,5 à 3. On évalue ainsi la valeur maximisante pour chaque fonction de nouveauté.

Estimation des performances On conclut l'expérience en appliquant notre classificateur optimal sur la base de test sans post-traitement, avec un post-traitement basique de lissage (élimination des intrus), et en utilisant le post-traitement par segmentation aveugle, pour la fonction la plus efficace. ESTER pose pour condition de ne faire qu'un seul calcul sur la base de test, sans intervention humaine, pour pouvoir valider les résultats du système. Nous présenterons donc le résultat obtenu par le système optimisé sur la base de développement.

Quand nous parlons d'évaluation des performances, il est important de préciser qu'il s'agit du calcul de la *F-mesure* sur la segmentation prédite par le système. Néanmoins nous verrons que parfois la F-mesure sur les seuls segments de musique ou de parole (également calculée par l'outil `trackeval` d'ESTER) est également significative et permet de lever certaines ambiguités.

5.3 Résultats

5.3.1 Premiers résultats sur bases artificielles

Base composite DB_ART

Notre première étude porte donc sur une base composée à partir d'extraits de parole d'ESTER, de musique de RWC, et de parole sur fond musical construite par mixage des deux. Notons que nous appellerons par la suite ces trois classes respectivement SP (*speech*), MU (*music*) et MIX (*mixed*), afin d'alléger l'écriture. Notre premier constat porte sur la détermination de la méthode de sélection d'attributs la plus efficace. La figure 5.1 nous montre que la sélection par IRMFSP semble être plus efficace que Fisher, y compris en sélection binaire, qui empire les résultats plutôt que de les améliorer, comme on pouvait l'espérer. On note également que la tendance observée entre les divers algorithmes pour 20 descripteurs est à peu près confirmée avec 50 descripteurs. Nous emploierons dans la suite la combinaison la plus efficace ici, à savoir la classification sur 50 descripteurs sélectionnés par IRMFSP non-binaire.

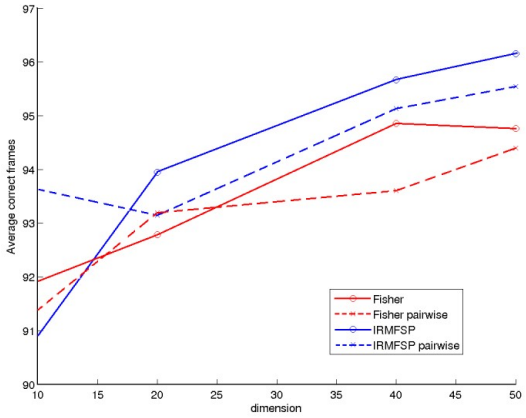


FIG. 5.1 – Efficacité des deux algorithmes de sélection d'attributs en binaire (*pairwise*) ou non, avec 10, 20, 40 ou 50 attributs.

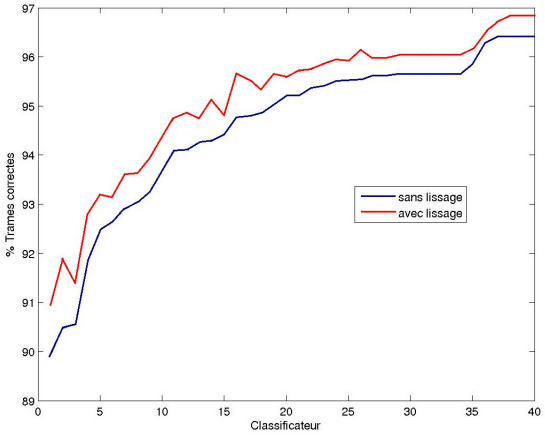


FIG. 5.2 – Comparaison des résultats obtenus avec et sans lissage, sur l'ensemble des expériences.

Une courte étude sur les paramètres C, θ et σ des machines SVMs, que nous ne détaillerons pas ici, ne montre pas d'influence notable dans le sens positif lorsque l'on fait varier ces derniers hors de leur valeur par défaut. En particulier le paramètre σ maximise les performances pour la valeur par défaut $\sigma = 1$. Par contre, l'application d'un post-traitement lissant, consistant à extraire les *outliers*, est systématiquement bénéfique, comme l'atteste la figure 5.2 qui montre le pourcentage de trames correctement reconnues pour chaque expérience réalisée, avec et sans lissage. Le gain moyen sur toutes les expériences est de 0.5% et peut atteindre jusqu'à 1.4%.

On obtient donc, avec lissage des résultats, un taux de trames exactes de 96.16 %. Les performances sont d'ailleurs bien meilleures sur la seule classe de parole puisqu'on obtient 98.86 % de bonnes reconnaissance. La classe MIX est la moins bien reconnue (91.93%), en raison d'une confusion à 6% avec la classe MU. On note néanmoins que la confusion est absolument nulle entre parole et musique, ce qui

traduit le fait que ces deux classes proviennent de bases différentes. Ces résultats sont synthétisés dans la matrice de confusion tableau 5.2.

Classe	SP	MU	MIX
SP	98.86%	0.00%	1.14%
MU	0.00%	97.69%	2.31%
MIX	2.02%	6.05%	91.93%

TAB. 5.2 – Matrice de confusion obtenue avec classification sur 50 attributs sélectionnés par IRMFSP non binaire sur la base `DB_ART`, après lissage des résultats

Les performances obtenues sur cette base d'étude sont très encourageantes, car nous obtenons un taux d'erreur moyen de 3.84%, ce qui est inférieur au meilleur résultat obtenu par Scheirer (5.3%) dans son article de référence sur la segmentation parole/musique [Scheirer and Slaney, 1997]. On reste également très proche des résultats obtenus par Saunders [Saunders, 1996] qui obtient un taux moyen de classification correcte entre 95% et 96%.

Cependant, ces résultats sont à prendre avec précaution, car les classes proviennent de sources très différentes et sont en définitive assez facilement reconnaissable. Nous verrons par la suite que la classification est moins aisée lorsque l'on teste sur le seul contenu de la base ESTER.

Base condensée `DB_ESTER`

Comme nous l'avons expliqué, cette base est constituée exclusivement d'extraits des séquences ESTER regroupés en parts égales de 40 minutes pour chaque classe.

Nous avons également estimé l'efficacité des classificateurs pour chacun des algorithmes de sélection d'attributs (binaire ou simple), en fournissant les 10, 20 ou 40 premiers descripteurs au classificateur. Nous avons par ailleurs étendu la variété des descripteurs en nous basant non plus seulement sur les moyennes et variances des descripteurs sur trames longues, mais également sur leurs valeurs minimales et maximales. Néanmoins, en comparant les résultats obtenus avec et sans les descripteurs minimaux et maximaux nous ne pouvons conclure sur un effet systématique, positif ou négatif, de cet apport. Comme le montre la figure 5.3, l'effet sur chaque expérience est assez aléatoire. Etant donné que cet apport double le nombre de descripteurs, et se révèle ainsi très coûteux lors de la sélection des attributs, nous avons préféré nous cantonner aux moyennes et variances des descripteurs dans la suite de notre étude.
L'étude des performances en fonction de la dimension du vecteur d'attributs nous confirme que la hausse de la dimensionnalité améliore les performances du système, mais néanmoins il reste difficile de conclure sans ambiguité quand à l'efficacité manifeste de l'un des algorithmes de sélection par rapport aux autres, comme le montre le tableau 5.3. Nous conservons l'algorithme de sélection le plus efficace (Fisher non-binaire) pour l'affinage des paramètres des SVM, avec une dimension de 40 attributs. Le lissage des résultats par suppression des *outliers* se révèle encore bénéfique, comme le montre le tableau 5.4 décrivant les résultats obtenus sur les mêmes sélections que précédemment. On constate cette fois-ci un gain moyen de 1.8% du taux de trames correct sur l'ensemble des expériences, et un gain maxi-

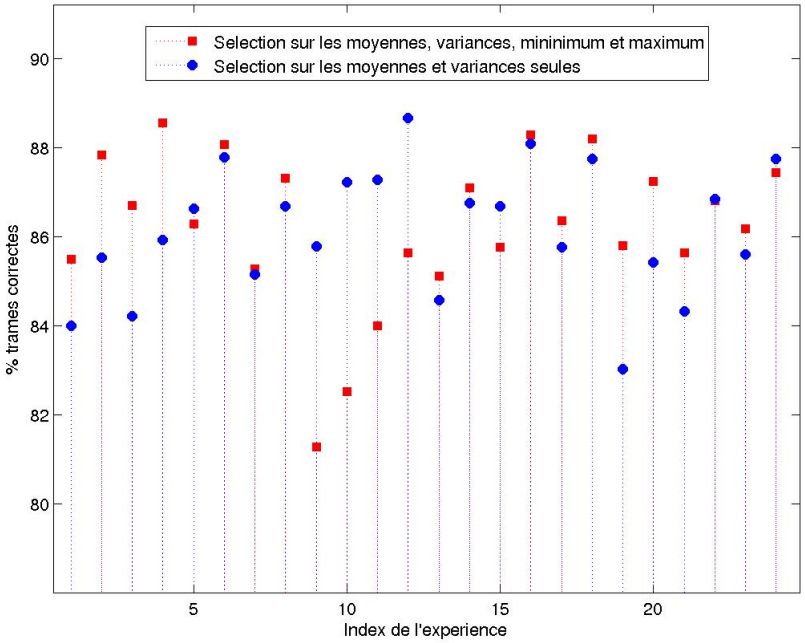


FIG. 5.3 – Efficacité comparée sur chaque expérience, avec ou sans l'inclusion des valeurs minimales et maximales dans l'ensemble des descripteurs sélectionnés.

mum de 2.7%. Ces résultats confirment d'ailleurs notre choix pour la sélection de 40 attributs par l'algorithme Fisher non-binaire.

Algorithme / Dimension	**10**	**20**	**40**
IRMFSP	83.99	84.20	86.61
Fisher	85.15	85.78	***87.27***
IRMFSP binaire	84.57	86.68	85.76
Fisher binaire	83.02	84.33	85.60
Moyenne	84.18	85.24	86.31

TAB. 5.3 – Taux de trames correctes pour chaque algorithme de sélection d'attributs, pour une dimension de 10, 20 ou 40 attributs, sans post-traitement

La taille des sous-ensembles θ et le facteur de pénalité C n'ont aucune influence sur les performances des machines, mais par contre le paramètre σ se rélève ici assez important. En effet, on observe sur la courbe, figure 5.4, la présence d'un pic mis en évidence de manière heuristique (par affinages successifs du paramètre) autour de la valeur $\sigma = 0.65$. Nous avons vu plus haut que réduire la valeur de σ revient à accentuer la courbure de la surface de décision. Le fait que le paramètre maximise ici les performances pour une valeur inférieure à celle estimée pour la base `DB_ART` laisse donc penser que les classes sont plus difficilement séparables, ce qui se traduit par une complexité accrue de la surface de décision. Ceci confirme donc notre intuition, puisque les classes de la base `DB_ART`, provenant de corpus différents, étaient à priori plus facilement décidables. Si l'on examine la matrice de confusion tableau 5.5, on constate en outre que la baisse de performances (90.21% de trames corrects sur

Algorithme / Dimension	10	20	40
IRMFSP	85.53	85.92	87.781
Fisher	86.68	87.22	***88.66***
IRMFSP binaire	86.74	88.09	87.74
Fisher binaire	85.42	86.85	87.74
Moyenne	86.09	87.02	87.98

TAB. 5.4 – Taux de trames correctes pour chaque algorithme de sélection d'attributs, pour une dimension de 10, 20 ou 40 attributs, avec suppression des *outliers*

DB_ESTER contre 96.16% sur DB_ART) s'explique par une plus grande confusion entre les classes SP et MIX (15.26% et 11.15% dans les deux sens). On retrouve ce constat si l'on s'intéresse au nombre de vecteurs supports de la machine de décision SP/MIX, qui traduit également la complexité de la surface de décision. Alors que l'on n'avait que 167 vecteurs supports (c'est à dire d'exemples situés sur les hyperplans aux frontières de la marge) et aucun exemple de la base d'apprentissage mal classifié pour la base DB_ART, on trouve 545 vecteurs supports pour la base DB_ESTER, et 91 vecteurs mal classifiés pour un nombre égal d'exemples dans les deux bases. On confirme ainsi l'hypothèse d'une surface de décision plus complexe, mise en évidence par le paramètre σ.

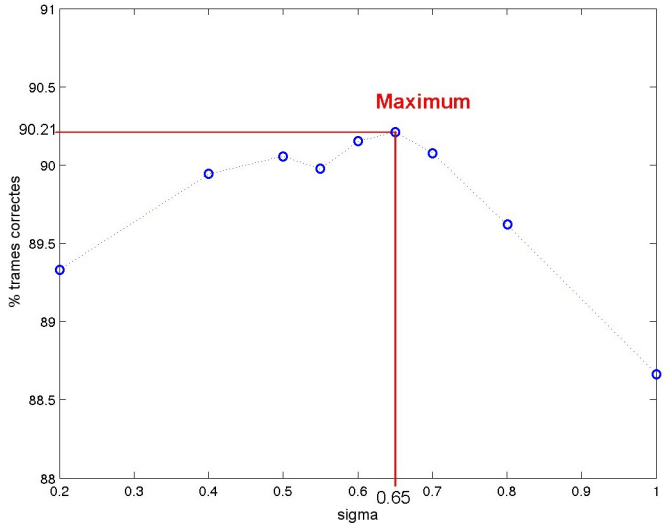


FIG. 5.4 – Mise en évidence heuristique d'un maximum global des performances à 90.21% de trames correctes pour la valeur $\sigma = 0.65$.

Classe	SP	MU	MIX
SP	84.74%	0.00%	15.26%
MU	0.17%	98.34%	1.50%
MIX	11.15%	1.29%	87.56%

TAB. 5.5 – Matrice de confusion obtenue avec classification sur 40 attributs sélectionnés par Fisher non binaire sur la base DB_ESTER

Ce constat s'explique très simplement si l'on prend la peine d'écouter les extraits

sonores de la classe MIX (c'est à dire de parole sur fond musical). Alors que dans la base `DB_ART` ceux-ci étaient construits en mixant à part égale de la parole d'ESTER sur de la musique de RWC, dans la base ESTER les extraits de classe MIX sont des sommaires de bulletins d'informations où une voix présente les titres sur un fond musical répétitif très discret, procédé classique en radio et en télévision. On conçoit aisément que ce type d'extraits soient très proches de la parole seule et ainsi beaucoup plus difficile à discerner pour le classificateur. Cette étude préliminaire nous aura donc permis de mettre en évidence le caractère très particulier des segments annotés lors de la campagne ESTER, et la complexité de la tâche par rapport à une simple étude de segmentation parole/musique, comme dans la plupart des publications.

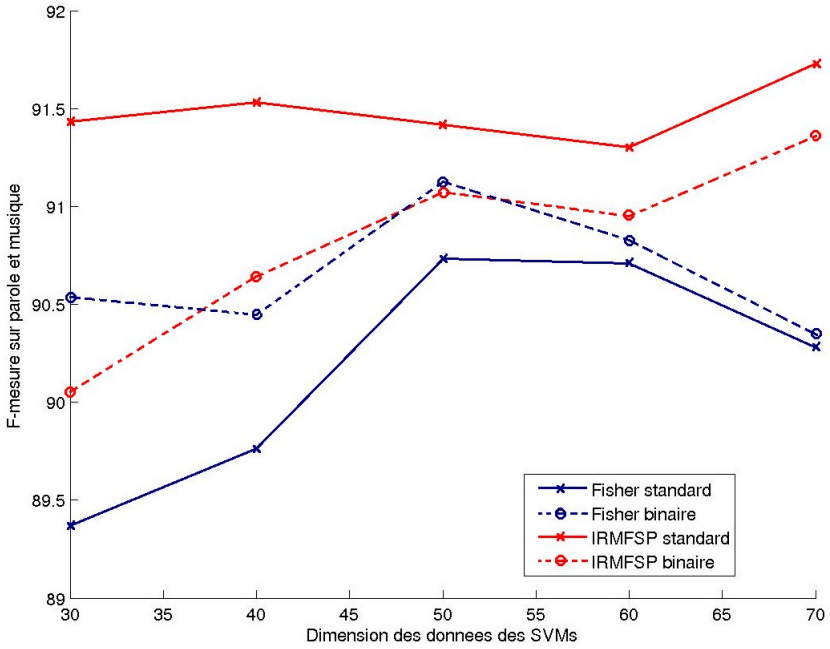
5.3.2 Résultats et situation au sein de la campagne ESTER

Nous présentons ici notre étude sur le corpus ESTER. Nous commencerons par détailler les résultats obtenus par affinages successifs des divers paramètres sur notre base `SEQ_ESTER`, uniquement constituée des séquences sonores d'ESTER. Nous étudierons le gain de performances résultant de l'usage complémentaire des techniques de segmentation aveugles, puis nous présenterons les résultats comparés sur les bases `SEQ_ESTER+` et `SEQ_ESTER++` auxquelles des extraits musicaux du corpus RWC ont été ajoutés. Nous finirons par la comparaison des performances de notre système final avec celles obtenues par les autres laboratoires participant à la campagne ESTER. La comparaison des performances se base désormais sur la F-mesure calculée sur les résultats de la segmentation par l'utilitaire `trackeval`. Celle-ci est dans un premier temps calculée sur chacun des fichiers de la base de test, puis moyennée en pondérant par la taille cumulée des segments de chaque fichier.

Sélection d'attributs

Nous avons comparé les deux algorithmes de sélection d'attributs en binaire ou sur toutes les classes, pour un nombre d'attributs variant entre 30 et 70 (par pas de 10). On peut constater, figure 5.5, qu'ici la différence entre les deux algorithmes est bien plus claire et évidente en termes de performances. Hormis quelques cas, l'algorithme IRMFSP, qu'il soit en sélection binaire ou non, obtient de meilleures performances que l'algorithme de Fisher. On constate par ailleurs une dégradation des performances à haute dimension (au delà de 50 attributs) pour l'algorithme de Fisher, qui chute de 0.45% pour l'algorithme standard et de 0.77% pour la sélection binaire ; degrédation qui n'est pas constatée avec IRMFSP. Nous constatons également que la sélection binaire profite systématiquement à l'algorithme de Fisher mais détériore les résultats obtenus avec IRMFSP, ce que nous n'expliquons pas.

Si l'on s'intéresse à la F-mesure calculée sur les seuls segments de parole ou de musique, on constate que les différences de performances proviennent avant tout d'une nette amélioration sur les segments de musique : 3.5% de différence sur la F-mesure entre les résultats à 70 attributs par IRMFSP simple et Fisher binaire (figure 5.6). La différence mesurée sur les segments de parole n'est que de 0.17% entre ces mêmes algorithmes sur 70 attributs (figure 5.7). Si l'on pondère ces gains par la durée totale des segments de chaque classe (comme c'est le cas pour le calcul de la F-mesure global), soit 29488s de parole et 4487s de musique, le gain obtenu sur les segments de musique contribue trois fois plus à la hausse globale que le gain sur les segments



Algorithme	30	40	50	60	70
Fisher	89.36	89.76	90.73	90.71	90.28
Fisher binaire	90.53	90.45	91.12	90.82	90.34
IRMFSP	91.43	91.53	91.41	91.30	***91.73***
IRMFSP binaire	90.05	90.64	91.07	90.95	91.36

FIG. 5.5 – F-mesure mesurée en conservant de 30 à 70 attributs sélectionnés par chacun des algorithmes FSA sur la base `SEQ_ESTER`

de parole. Il est bien évident qu'étant donné les fortes performances mesurées sur les segments de paroles (autour de 99%), l'amélioration des performances sur les segments de musique est essentielle à l'optimisation du système.
Nous conservons donc les 70 attributs sélectionnés par IRMFSP simple pour la suite de notre étude.

Affinage des SVMs

Comme nous l'avons constaté sur la base `DB_ESTER`, les paramètres C et θ n'ont ici aucune influence notable sur les performances des machines à vecteurs supports. Le paramètre σ reste influent sur la base `SEQ_ESTER`, comme le montre la figure 5.8, sur laquelle on peut observer un pic heuristique des performances pour la valeur $\sigma = 0.5$ que nous conserverons par la suite. Néanmoins le gain est très limité puisqu'il ne représente que 0.22% sur une F-mesure avoisinant les 91.7%. Nous préciserons tout de même que l'on constate des effets opposés si l'on considère dans le tableau 5.6 l'influence du paramètre σ sur la F-mesure de parole et sur la F-mesure de musique. On observe une baisse, très légère mais monotone, de la première lorsque l'on augmente la valeur de σ tandis que la seconde augmente de manière plus significative.

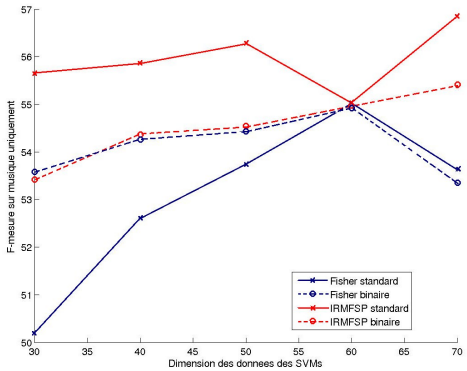


FIG. 5.6 – F-mesure sur les segments de musique de la base SEQ_ESTER.

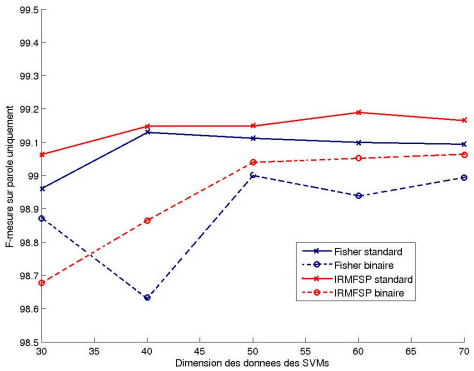


FIG. 5.7 – F-mesure sur les segments de parole de la base SEQ_ESTER.

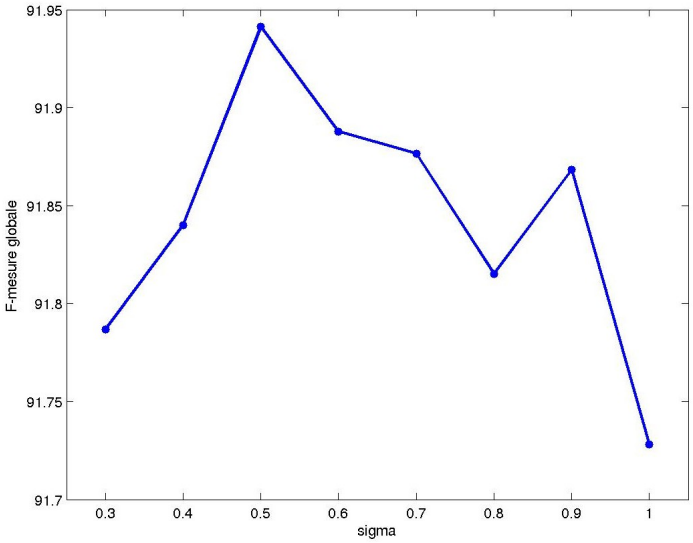


FIG. 5.8 – Influence du paramètre σ sur la F-mesure globale du système

Segmentation aveugle

Nous avons par la suite calculé les 5 fonctions de nouveauté présentées section 5.2. Celles-ci sont calculées sur des fenêtres passées et futures de 5s. Estimant que la durée minimale d'un segment était de l'ordre de quelques secondes, nous avons pensé que cette taille de fenêtre permettait de supposer une certaine homogénéité locale des descripteurs long-terme. Le calcul de ces fonctions est très coûteux en temps de calcul, et le temps nous a manqué pour pouvoir effectuer une estimation empirique de la taille de fenêtre optimale, qui diffère à priori d'une fonction à l'autre.

La figure 5.9 nous montre que le choix du seuil τ dans l'algorithme d'extraction des pics (maxima locaux) à une influence notable sur les performances du système. Certains comportements communs à toutes les fonctions de nouveauté se dégagent à la lecture de cette figure.

Avant tout on constate que pour des seuils très bas ($\tau < 0.5$), la F-mesure globale semble converger vers une valeur limite stable qui reste supérieure aux résultats obtenus sans post-traitement. Ceci n'est pas étonnant si l'on se souvient que l'algorithme de détection des pics impose une distance minimale de 3s entre deux maxima

σ	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
F-mesure parole	99.42	99.40	99.36	99.28	99.23	99.20	99.21	99.16
F-mesure musique	54.87	55.32	55.54	55.57	55.48	55.59	55.75	56.84
F-mesure globale	91.79	91.84	91.94	91.89	91.88	91.81	91.87	91.73

TAB. 5.6 – Comparaison de l'influence du paramètre σ sur les F-mesures globale, de parole et de musique

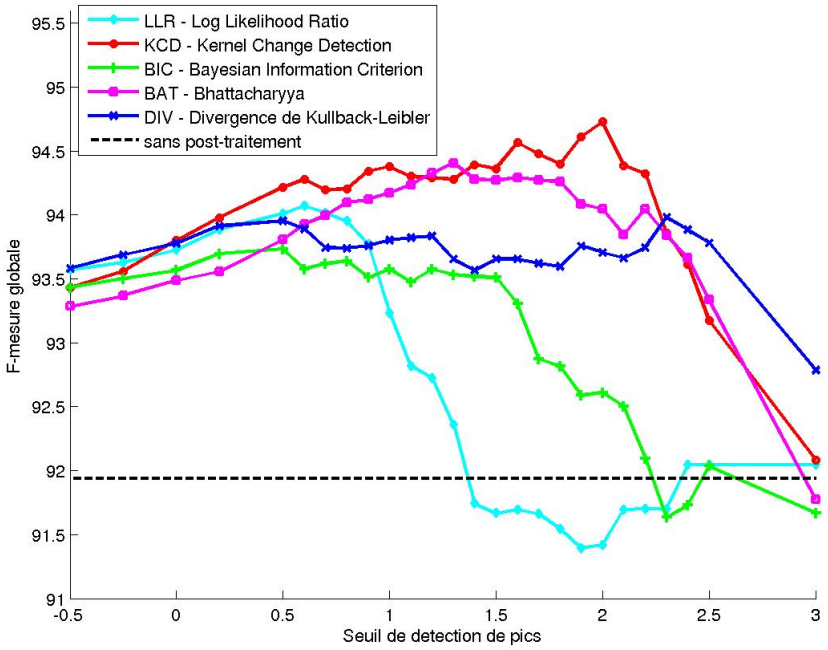


FIG. 5.9 – F-mesure globale du système après post-traitement par segmentation aveugle, pour des valeurs de seuil s'échelonnant entre -0.5 et 3, sur la base `SEQ_ESTER`.

voisins. Cependant on constate que pour toutes les fonctions, cette valeur limite reste environ 1.5% supérieure aux résultats obtenus sans post-traitement, ce qui montre que même utilisée grossièrement, cette correction reste tout à fait pertinente.

De même, si l'on accroît τ au dela de 2.5, on constate une chute des performances avec toutes les fonctions, qui converge vers 0. En effet, puisqu'aucun maxima n'est détecté si le seuil est trop élevé, aucun segment ne peut être reconnu.

La valeur τ_{max} maximisant les performances varie selon la fonction de nouveauté utilisée. Ainsi on constate, tableau 5.7, que le LLR et le BIC maximisent les résultats autour de $\tau_{max} = 0.5$, pour des gains respectifs de 2.13% et 1.79% sur la F-mesure globale, mais restent moins performantes, que les distances probabilistes de Bhattacharyya et de Kullback-Leibler qui apportent respectivement des gains de 2.47% à $\tau_{max} = 1.3$ et 2.04% à $\tau_{max} = 2.3$. Ceci s'explique par le fait que le critère BIC est calculée sur les vecteurs d'attributs, en supposant leur répartition gaussienne, tandis que, dans le cas des distances probabilistes, l'hypothèse gaussienne n'est faite qu'après projection non-linéaire dans l'espace RKHS de dimension supérieure.

Les meilleures performances sont obtenues avec le KCD, qui apporte un gain

Fonction de nouveauté	τ_{max}	F-mesure (gain)		
		globale	parole	musique
Log Likelihood Ratio	0.6	94.07 (2.13)	99.38 (0.02)	61.55 (6.01)
Kernel Change Detection	2	94.73 (2.79)	99.56 (0.20)	66.20 (10.66)
Bayesian Information Criterion	0.5	93.73 (1.79)	99.29 (-0.07)	59.94 (4.40)
Distance de Bhattacharyya	1.3	94.40 (2.47)	99.32 (-0.04)	63.52 (7.98)
Divergence de Kullback-Leibler	2.3	93.98 (2.04)	99.62 (0.26)	56.60 (1.06)

TAB. 5.7 – F-mesure calculée sur les différentes classes de segments après application du post-traitement par segmentation aveugle sur la base `SEQ_ESTER`, pour chacune des fonctions de nouveauté.

de 2.79% sur la F-mesure globale à $\tau_{max} = 2$. On constate par ailleurs que le gain provient essentiellement d'une meilleure détection des segments de musique puisque le gain observé sur la F-mesure de musique s'élève jusqu'à 10.66% avec le KCD contre seulement 0.2% sur les segments de parole, ce qui, après pondération par les durées des segments, représente une contribution dix fois supérieure sur les performances globale. On observe d'ailleurs que pour certaines fonctions, le post-traitement déteriore légèrement les résultats sur les segments de parole (de l'ordre de quelques centièmes de pourcent).

Cette étape de post-traitement représente donc un apport majeur à notre système de segmentation puisqu'il corrige fortement les performances sur la partie la plus sensible de l'évaluation. Notons, à titre de comparaison, que l'on obtient une F-mesure globale de 92.97% après application du post-traitement rudimentaire de suppression des outliers présenté plus haut, soit un gain de 1% environ, et un gain de 3% sur la F-mesure de musique.

Apport de la base RWC

Nous avons suivi le même protocole expérimental sur les deux bases `SEQ_ESTER+` et `SEQ_ESTER++`, présentées précédemment. Le tableau 5.8 présente les résultats obtenus sur la base de développement par le système optimisé sur chacune des bases `SEQ_ESTER`, `SEQ_ESTER+` et `SEQ_ESTER++`. L'apport d'une quantité égale de musique à celle présente dans ESTER, au travers de la base `SEQ_ESTER+`, semble plus profitable que l'apport de la totalité de la base RWC, si l'on considère la F-mesure globale obtenue sur chacune des bases. On constate en effet un gain de 0.54% sur `SEQ_ESTER+` alors que l'on perd 0.47% sur `SEQ_ESTER++`. On pourra s'étonner de constater que la F-mesure est pourtant accrue aussi bien sur les segments de parole que sur les segments de musique sur la base `SEQ_ESTER++`, néanmoins le résultat global mitigé s'explique par le déséquilibre important sur les segments de musique entre le gain très fort observé sur la mesure de rappel (R) et la très forte déterioration de la précision (P), ce qui correspond à un système détectant la plupart des événements réels, mais en ajoute également beaucoup par erreur. La détection erronnée d'un nombre trop important de frontières de segments musicaux signifie que l'apprentissage de la classe se généralise mal à des exemples nouveaux et colle trop précisisemment aux données d'apprentissage.

L'optimisation du système sur la base `SEQ_ESTER++` avait d'ailleurs conduit au choix de la valeur $\sigma = 0.3$ durant l'affinage des SVMs, ce qui laisse supposer une surface de décision plus complexe que sur les autres bases (on trouve $\sigma = 1$ sur

Segments	Tous			Parole			Musique		
Critère	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**
SEQ_ESTER	94.73	94.39	95.70	99.56	99.66	99.45	66.20	74.55	71.09
SEQ_ESTER+	95.27	95.74	95.33	99.40	99.52	99.30	67.11	78.83	69.25
SEQ_ESTER++	94.26	90.84	98.68	99.73	99.72	99.77	68.89	61.34	90.20

TAB. 5.8 – Comparaison des résultats de F-mesure, rappel (R) et précision (P) sur les trois bases SEQ_ESTER, SEQ_ESTER+ et SEQ_ESTER++

.

SEQ_ESTER+). On ne retrouve pas cette baisse de précision sur la base SEQ_ESTER+ pour les segments de musique, sur lesquels la F-mesure observe un gain de 0.91% ; mais par contre la F-mesure calculée sur les segments de parole est légèrement en deça des performances obtenues sur la base d'origine SEQ_ESTER (chute de 0.16%).

En définitive nous avons jugé les résultats bien trop ambigus et difficilement interprétables pour pouvoir conclure sur les conséquences positives d'un ajout d'extraits musicaux provenant de la base RWC. On observe certes une hausse de la F-mesure globale mais les performances sur les segments de parole ou de musique sont loin d'être aussi claires.

Par ailleurs, le taux de trames correctes pour le système optimisé sur la base SEQ_ESTER est de 89.93%, alors qu'il n'est que de 88.83% sur la base SEQ_ESTER+ et 86.09% sur la base SEQ_ESTER++, ce qui accentue nos doutes quand à la pertinence d'un tel ajout. Une simple écoute des extraits musicaux de la base ESTER nous fournis d'ailleurs quelques pistes sur cette question. En effet nous avons pu constater qu'environ la moitié des extraits étaient des chansons tandis que l'autre moitié était constituée de jingles radiophoniques. Ceux-ci sont généralement d'une sonorité très particulière et nous pensons que le système gagnerait beaucoup à considérer une classe spécifique pour les jingles. Cette distinction permettrait de lever l'ambiguïté de la classe musique, qui pourrait dès lors être constituée d'une majorité d'extraits de RWC.

Résultats sur la base de test

Si l'on conserve le système optimisé sur la base SEQ_ESTER (sélection de 70 attributs par IRMFSP non-binaire, puis apprentissage des SVMs avec $\sigma = 0.5$, $\theta = 20$ et $C = C_{dat}$) on trouve sans post-traitement une F-mesure de **95.95%** sur l'ensemble des segments (respectivement 99.28% et 77.72% pour les segments de parole et de musique), et avec le post-traitement optimisé (KCD avec $\tau = 2$) une F-mesure de **96.50%** sur l'ensemble des segments (respectivement 98.92% et 79.30% pour les segments de parole et de musique). On note qu'ici le gain apporté par le post-traitement est beaucoup moins franc que sur la base de développement mais tout de même significatif.

Nous sommes surpris d'obtenir de meilleurs résultats sur notre base de test que sur la base de développement après optimisation, mais nous pensons pouvoir expliquer en partie ce résultats en considérant les matrices de confusion calculées sur les taux de trames correctes de chaque classe sur les bases de développement (tableau 5.9) et de test (tableau 5.10). On remarque en effet que les trames des classes SP

et MIX sont mieux reconnues sur la base de test, mais que par contre les trames de la classe MU sont très mal reconnues sur la base de test (47.74% de bonne reconnaissance contre 74.43% sur la base de développement). Néanmoins, comme les trames de classe MU (musique seule) sont largement minoritaires, et que la majorité (les 9 dixièmes) des segments de musique sont confondus avec des segments de parole (trames de classe MIX), il est logique que l'on obtienne au final une meilleure segmentation des segments de musique, puisque l'on y retrouve les trames de classe MIX. De plus la hausse notable de reconnaissance de trames SP explique en grande partie la moindre efficacité du post-traitement par segmentation aveugle, puisque l'on a à priori beaucoup moins de trames incorrectes avant post-traitement.

Classe	**SP**	**MU**	**MIX**
SP	91.98%	0.20%	7.82%
MU	10.50%	74.43%	15.07%
MIX	30.00%	2.64%	67.36%

TAB. 5.9 – Matrice de confusion du taux de trames correctes pour chaque classe sur la base de développement de `SEQ_ESTER`.

Classe	**SP**	**MU**	**MIX**
SP	97.58%	0.08%	2.34%
MU	16.12%	47.74%	36.14%
MIX	24.97%	4.86%	70.17%

TAB. 5.10 – Matrice de confusion du taux de trames correctes pour chaque classe sur la base de test de `SEQ_ESTER`.

Résultats de la campagne ESTER

Nous présentons pour finir le tableau 5.11 résumant les performances des participants à la campagne ESTER, extrait de [Galliano et al., 2004], auquel nous avons ajoutés nos résultats, qui dépassent de 2.3% ceux de tous les autres participants sur la F-mesure globale.

La plupart des participants se sont focalisés sur d'autres tâches, principalement celles de transcription ou de suivi de locuteur, ce qui explique que pour beaucoup la segmentation parole/musique soit restée un détail traité avec des techniques très classiques. On constate d'ailleurs de bons résultats pour la reconnaissance de parole, sur laquelle plusieurs laboratoires obtiennent de meilleurs résultats que nous, mais beaucoup plus mitigés sur les segments de musique, sur lesquels nous avons porté notre attention. Il est bien évident que, disposant de la base de test et des résultats

	globale			**parole**			**musique**		
Participant	F	%fa	%fr	F	%fa	%fr	F	%fa	%fr
Stage ENST	96.5	4.8	4.1	98.9	43.5	2.1	79.3	5.0	8.8
IRIT	94.2	2.1	9.5	98.8	30.1	1.5	52.7	1.2	61.7
IRISA	93.1	1.3	12.1	98.9	9.7	1.9	33.7	1.0	78.5
LIA	92.7	11.7	5.7	99.2	36.6	0.7	54.8	10.9	38.7
LIUM	90.7	1.3	16.2	97.4	8.0	4.9	17.8	1.1	89.6
SIS	83.7	11.5	20.9	93.4	82.2	10.4	12.7	10.4	89.2
UOB	88.2	3.9	18.6	95.1	20.1	8.9	26.2	3.4	82.0
FT R&D	—	—	—	99.1	25.5	1.1	—	—	—
LORIA	—	—	—	97.5	34.2	4.0	—	—	—

TAB. 5.11 – Performances des participants à la campagne ESTER pour la tâche SES de segmentation

et publications des autres participants, nous ne pouvons prétendre nous comparer objectivement à leur travail dans le cadre de la campagne ESTER.

Néanmoins, nous espérons montrer que de nombreux progrès peuvent être accomplis quant à la segmentation des segments de musique, et particulièrement des segments mixtes. La F-mesure que nous obtenons sur les segments de musique est largement supérieure à celle des autres participants (54.8% et 52.7% pour les deux meilleurs résultats dans le classement ESTER) et, nous l'espérons, encore sujette à certaines améliorations que nous avons évoqué dans ce rapport. La F-mesure sur les segments de parole reste également comparable aux meilleurs résultats de la campagne, malgré un taux de fausse alarme assez élevé (43.5%).

Chapitre 6

Conclusion et perspectives

Nous avons pu montrer la pertinence des Machines à Vecteurs Supports, une technique de classification très récente, et encore peu exploitée dans le domaine de l'indexation audio, pour une tâche de segmentation d'un signal audio. L'usage complémentaire de techniques de sélection automatique d'attributs se montre également crucial car il permet d'opérer une classification à dimension réduite (nous n'avons délibéremment pas atteint les 100 attributs) à partir d'un panel très varié de descripteurs. En particulier l'algorithme IRMFSP a montré de meilleures performances que l'algorithme plus classique de Fisher.

En outre, nous avons constaté le gain notable apporté par une phase de post-traitement basée sur les résultats d'une segmentation aveugle. Celle-ci reste d'ailleurs sujette à un certain nombre d'améliorations que nous avons présentées dans ce rapport.

Les performances de notre système ont par ailleurs pu être comparées à celles d'autres laboratoires évalués selon les mêmes critères et sur le même matériel, ce qui donne un crédit supplémentaire aux résultats obtenus dans le cadre de cette étude. Néanmoins, nous avons pu observer à plusieurs reprises dans ce rapport que la représentation des segments de musique dans la base ESTER est loin d'être idéale, car elle mélange des extraits sonores de natures très différentes (jingles, chansons, parole sur léger fond musical) difficilement reclassifiables manuellement, étant donné le volume important de données. De plus la proportion très largement majoritaire de segments de paroles, adaptée à une campagne de transcription de parole, se révele plutôt handicapante pour la seule tâche SES.

Nous sommes très satisfait de ce stage car il nous a permis de nous familiariser avec bon nombre de points théoriques associés à la segmentation et l'indexation audio, préparant ainsi notre futur travail de thèse au sein d'une grande radio, sur un travail similaire de segmentation audio. Nous avons en outre eu l'occasion de nous pencher sur l'essentiel du code de notre système, que nous avons corrigé et complété, ce qui nous a permis d'avoir une connaissance complémentaire, plus concrête, de ces outils théoriques.

Nous avons suggéré dans ce rapport plusieurs lignes de poursuite du projet qui permettrait d'en améliorer les performances ou d'en élargir le champ d'application.

Précision temporelle Nous n'avons pu mener à bien nos essais basés sur un pas entre trames longues réduit à 0.25 secondes, ce qui correspond à la tolérance admise sur un événement pour le calcul de la F-mesure. Au delà du cadre de la

campagne ESTER, il peut être pertinent, pour tout média prenant en charge un flux audio, d'obtenir une segmentation en temps réel, qui nécessiterait par conséquent un temps de latence bien moindre.

Nouveaux attributs Nous n'avons pas proposé de nouveaux descripteurs dans le cadre de stage, si ce n'est l'adjonction des moyennes et variances des descripteurs estimées sur les trames longues. Certains attributs assez récents ont prouvés leur efficacité dans la littérature, pour des tâches liées à l'indexation audio. En particulier les attributs définis par Pinquier [Pinquier et al., 2002] basés sur une statistique calculée à partir du résultat d'une segmentation aveugle fine, ou encore les coefficients de Fepstre, présentés par Tyagi et Wellekens [Tyagi and Wellekens, 2005]. L'usage des coefficients d'un modèle Auto Regressif des attributs semble également prometteur.

Alternative pour la classification Nous nous sommes cantonnés à l'usage des SVMs dans ce stage, d'après le constat de leur efficacité dans [Essid, 2005], mais également pour leur relative nouveauté dans ce contexte. Néanmoins d'autres méthodes (GMM, avec usage éventuel des modèles de Markov cachés, k plus proches voisins...) restent d'usage très courant dans la littérature, si bien que l'évaluation de leurs performances nous paraît pertinente dans un second temps.

Extension des classes Le problème posé par ce stage ne concernait que la segmentation parole/musique, qui reste un problème crucial pour l'indexation audio. Mais une classification plus fine serait largement profitable pour une utilisation radiophonique. Ainsi, la localisation des jingles, publicités, voix téléphonique permettrait d'affiner la connaissance tirée de l'analyse du signal. On pourra même se pencher par la suite sur le problème de la reconnaissance de chant (qui est généralement très différent de la parole), et la séparation de locuteurs, qui fait d'ailleurs partie des tâches de la campagne ESTER. Les techniques employées dans ce stage resteraient pertinentes pour ce genre de segmentation. Néanmoins, une segmentation plus fine nécessiterait avant tout une forte confiance dans les résultats de la segmentation plus grossière.

Regroupement de segments Les techniques de segmentation aveugle présentées ici, basées sur des critères de similarité entre trames, sont également couramment utilisées dans des sytèmes basés sur le regroupement de segments générés de manière plus ou moins grossière ou aléatoire. Ce regroupement permet ainsi de réunir au sein d'une même classe un ensemble de segments d'un signal, afin d'appliquer la classification sur l'ensemble des informations réunies dans cette classe [Kemp et al., 2000]. Nous n'avons pu tester ce type d'algorithmes, qui nous paraissent prometteurs.

Bibliographie

[Blouch and Collen, 2005] Blouch, O. L. and Collen, P. (2005). Méthode de segmentation parole/non-parole. In *Rencontre Jeunes Chercheurs en Parole '05.*

[Brown, 1999] Brown, J. C. (1999). Computer identification of musical instruments using pattern recognition with cepstral coefficients as features. *Journal of the Acoustical Society of America*, vol. 105 :1933–1941.

[Burges, 1998] Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, vol. 2(2) :121–167.

[Calmès et al., 2005] Calmès, M. D., Farinas, J., Ferrané, I., and Pinquier, J. (2005). Campagne ESTER : une première version d'un système complet de transcription automatique de la parole grand vocabulaire. In *Atelier ESTER, Avignon.*

[Carey et al., 1999] Carey, M. J., Parris, E. S., and Thomas, H. L. (1999). A comparison of features for speech, music discrimination. In *Proc. ICASSP '99*, pages 149–152.

[Chen and Gopalakrishnan, 1998] Chen, S. S. and Gopalakrishnan, P. S. (1998). Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *Proc. DARPA Speech Recognition Workshop.*

[Chou and Gu, 2001] Chou, W. and Gu, L. (2001). Robust singing detection in speech/music discriminator design. In *Proc. ICASSP '01*, pages 865–868.

[Desobry et al., 2005] Desobry, F., Davy, M., and Doncarli, C. (2005). An online kernel change detection algorithm. *IEEE Trans. Sig. Proc.*, vol. 53(8) :2961–2974.

[Duda et al., 2001] Duda, R., Hart, P., and Stork, D. (2001). *Pattern Classification.* Wiley-Interscience, second edition.

[El-Maleh et al., 2000] El-Maleh, K., Klein, M., Petrucci, G., and Kabal, P. (2000). Speech/music discrimination for multimedia applications. In *Proc. ICASSP '00, Istanbul, Turkey*, pages 2445–2448.

[Essid, 2005] Essid, S. (2005). *Classification automatique des signaux audiofréquences : reconnaissance des instruments de musique.* PhD thesis, ENST.

[Essid et al., 2006] Essid, S., Richard, G., and David, B. (2006). Instrument recognition in polyphonic music based on automatic taxonomies. *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14(1) :68–80.

[Galliano et al., 2004] Galliano, S., Geoffrois, E., Mostefa, D., Choukri, K., Bonastre, J.-F., and Gravier, G. (2004). The ESTER Phase II evaluation campaign for the rich transcription of french broadcast news. In *Proc. of the European Conf. on Speech Communication and Technology.*

[Gillet and Richard, 2006] Gillet, O. and Richard, G. (2006). On the correlation of automatic audio and visual segmentations of music videos. *IEEE Trans. Circuits and Systems for Video Technology*, Vol X(XX).

[Goto et al., 2003] Goto, M., Hashiguchi, H., Nishimura, T., and Oka, R. (2003). RWC music database : Music genre database and musical instrument sound database. In *Proc. ISMIR '03*, pages 229–230.

[Gravier et al., 2004] Gravier, G., Bonastre, J.-F., Geoffrois, E., Galliano, S., Tait, K. M., and Choukri, K. (2004). The ESTER evaluation campaign of rich transcription of french broadcast news. In *Proc. Language Evaluation and Resources Conference*.

[Gravier et al., 2005] Gravier, G., Yvon, F., and Ben, M. (2005). IRENE, le système commun IRISA - ENST d'indexation d'émissions radiophoniques. In *Atelier ESTER, Avignon*.

[Hastie and Tibshirani, 1998] Hastie, T. and Tibshirani, R. (1998). Classification by pairwise coupling. In Jordan, M. I., Kearns, M. J., and Solla, S. A., editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press.

[Hoyt and Wechsler, 1994] Hoyt, J. D. and Wechsler, H. (1994). Détection of human speech using hybrid recognition models. In *Proc. ICASSP '94*, volume vol. 1, pages 330–333.

[Kedem, 1986] Kedem, B. (1986). Spectral analysis and discrimination by zero-crossings. In *Proc. IEEE*, volume vol. 74 no. 11, pages 1477–1493.

[Kemp et al., 2000] Kemp, T., Schmidt, M., Westphal, M., and Waibel, A. (2000). Strategies for automatic segmentation of audio data. In *Proc. ICASSP '00*, volume vol. 3, pages 1423–1426.

[Kimber and Wilcox, 1996] Kimber, D. and Wilcox, L. (1996). Acoustic segmentation for audio browsers. In *Interface Conference, Sydney, Australia*.

[Loosli et al., 2005] Loosli, G., Lee, S.-G., and Canu, S. (2005). Context changes detection by one class SVMs. In *Workshop on Machine Learning for User Modeling*.

[Lu et al., 2001a] Lu, L., Jiang, H., and Zhang, H. (2001a). A robust audio classification and segmentation method. In *ACM Multimedia*, pages 203–211.

[Lu et al., 2001b] Lu, L., Li, S., and Zhang, J. (2001b). Content-based audio segmentation using support vector machines. In *Proc. ICME '01, Tokyo, Japan*, pages 956–959.

[Molina et al., 2002] Molina, L., Belanche, L., and Nebot, A. (2002). Feature selection algorithms : A survey and experimental evaluation. In *Proc. ICDM '02*.

[Osuna et al., 1997] Osuna, E., Freund, R., and Girosi, F. (1997). Support Vector Machines : Training and applications. Technical Report AIM-1602.

[Peeters, 2004] Peeters, G. (2004). A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, IRCAM.

[Peeters and Rodet, 2003] Peeters, G. and Rodet, X. (2003). Hierarchical gaussian tree with inertia ratio maximization for the classification of large musical instrument database. In *Proc. DAFX '03*.

[Pinquier, 2004] Pinquier, J. (2004). *Indexation sonore : recherche de composantes primaires pour une structuration audiovisuelle.* PhD thesis, Université Paul Sabatier (Toulouse III).

[Pinquier et al., 2002] Pinquier, J., Rouas, J.-L., and André-Obrecht, R. (2002). Robust speech / music classification in audio documents. In *International Conference on Spoken Language Processing (ICSLP'2002), Denver, Etats-Unis*, volume vol. 3, pages 2005–2008. Causal Productions Pty Ltd.

[Saunders, 1996] Saunders, J. (1996). Real-time discrimination of broadcast speech/music. In *Proc. ICASSP '96*, pages 993–996.

[Scheffer et al., 2005] Scheffer, N., Istrate, D., Fredouille, C., and Bonastre, J.-F. (2005). Les systèmes du LIA pour les tâches de segmentation et de suivi : SES, SRL, SVL. In *Atelier ESTER, Avignon.*

[Scheirer and Slaney, 1997] Scheirer, E. and Slaney, M. (1997). Construction and evaluation of a robust multifeature speech/music discriminator. In *Proc. ICASSP '97*, pages 1331–1334, Munich, Germany.

[Schölkopf and Smola, 2002] Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels.* The MIT Press.

[Schölkopf et al., 1999] Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., and Platt, J. C. (1999). Support vector method for novelty detection. In Solla, S. A., Leen, T. K., and Müller, K.-R., editors, *NIPS*, pages 582–588. The MIT Press.

[Siegler et al., 1997] Siegler, M., Jain, U., Raj, B., and Stern, R. (1997). Automatic segmentation, classification and clustering of broadcast news audio. In *Proc. DARPA Speech Recognition Workshop.*

[Sugiyama et al., 1993] Sugiyama, M., Murakami, J., and Watanabe, H. (1993). Speech segmentation and clustering based on speaker features. In *Proc. ICASSP*, volume vol. 2, pages 395–398.

[Tyagi and Wellekens, 2005] Tyagi, V. and Wellekens, C. (2005). Fepstrum representation of speech signal. In *ASRU 2005, IEEE Automatic Speech Recognition and Understanding Workshop, Cancun, Mexico*, pages 11–16.

[West and Cox, 2004] West, K. and Cox, S. (2004). Features and classifiers for the automatic classification of musical audio signals. In *Proc. ISMIR '04*, pages 531–536.

[Woodland et al., 1998] Woodland, P. C., Hain, T., Johnson, S. E., Niesler, T. R., Tuerk, A., Whittaker, E. W. D., and Young, S. J. (1998). The 1997 HTK broadcast news transcription system. In *Proc. 1998 DARPA Broadcast News Transcription and Understanding Workshop*, pages 41–48.

[Zhang and Kuo, 2001] Zhang, T. and Kuo, J. (2001). Audio content analysis for online audiovisual data segmentation and classification. *IEEE Trans. on Speech and Audio Processing*, vol. 9(4) :441–457.

[Zhou and Hansen, 2000] Zhou, B. and Hansen, J. (2000). Unsupervised audio stream segmentation and clustering via the bayesian information criterion. In *Proc. ICASSP '00, Beijing.*

[Zhou and Chellappa, 2006] Zhou, S. K. and Chellappa, R. (2006). From sample similarity to ensemble similarity : Probabilistic distance measures in reproducing kernel hilbert space. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28(6) :917–929.