FORMATION DOCTORALE EN INFORMATIQUE
UNIVERSITE PAUL SABATIER
DEA INFORMATIQUE de L'IMAGE et du LANGAGE (IIL)
Responsable R. CAUBET, Professeur

Année 2003/2004

Laboratoire IRIT (**I**nstitut de **R**echerche en **I**nformatique de **T**oulouse)
Equipe SAMoVA (**S**tructuration, **A**nalyse et **Mo**délisation de la **V**idéo et de l'**A**udio)

Méthodes à vecteurs de support et Indexation sonore

par José Anibal ARIAS AGUILAR

Directeur de recherche : Régine André-Obrecht
Encadrant scientifique : Jérôme Farinas, Julien Pinquier

Mots-clé : Méthodes à Vecteurs de Support, Indexation sonore, Modèles de Mélanges de lois Gaussiennes (MMG).

Résumé : Ce document est une étude comparative des performances de deux systèmes d'indexation sonore. Différents paramètres et méthodes de classification sont recensés. Une modélisation classe/non classe est utilisée pour l'indexation des sons de milieu ambiant. Sa mise en œuvre est faite à partir de systèmes de classification basés sur les Modèles de Mélanges de lois Gaussiennes et les Méthodes à Vecteurs de Support. L'évaluation est faite sur des émissions de télévision.

Abstract : This document deals with indexing techniques for environment sounds. Various features and two classification methods are proposed. A class/non class modeling approach is carried out using Gaussian Mixture Modeling and Support Vector Machines. A TV broadcasting-based evaluation is then conducted.

Table des Matières

Table des figures

Table des tableaux

Introduction

De nos jours, il est aisé et courant de manipuler des documents de type texte : le stockage, la manipulation, la recherche d'information sont des opérations abordables par le grand public. Or, la société de l'information actuelle est très friande d'autre types de media : les nouveaux moyens de communications (téléphones portables UMTS, accès Internet à haut débit, télévision numérique) tendent à vulgariser l'usage de documents audiovisuels auprès d'un nombre d'usagers en constante augmentation. Peu développé jusqu'alors, et souvent cantonné à un monde exclusivement professionnel, le besoin de traitement de ce type de documents apparaît assez critique. En effet, il n'est pas aisé de stocker, manipuler et exploiter des documents audiovisuels. Il n'est pas encore possible d'utiliser un moteur de recherche sur une grande collection d'émissions de télévisions de la même façon que l'on recherche des mots clés sur l'ensemble des pages Internet disponibles sur le réseau. Vouloir rechercher la vidéo contenant la première poignée de main entre le président François Mitterrand et Helmut Kohl dans les archives de la télévision est pour l'instant irréalisable si l'on ne traite que des documents multimédia. Cela nécessiterait l'usage de techniques permettant d'extraite du sens de la vidéo, de l'audio, et conjointement de l'audio et de la vidéo. S'il est de nos jours possible d'extraire quelques mots corrects de la bande sonore, l'indexation fiable et systématique de collections de documents disponibles nécessite encore beaucoup de travail de recherche.

Les problématiques d'extraction d'information de documents audiovisuels sont au cœur des préoccupations de notre équipe de recherche (Structuration Analyse Modélisation de la Vidéo et de l'Audio). Si l'on ne considère que les documents audio, on se retrouve principalement devant des problèmes de classification ou de reconnaissance de formes. Les systèmes utilisés ont alors pour but de segmenter et/ou d'identifier les différentes composantes du flux sonore. C'est ainsi que sont apparus des systèmes de détection de la parole et de la musique, des systèmes d'identification de la langue, des systèmes de détection de locuteurs, des systèmes de transcription automatique des zones de parole…

Les contenus sonores les plus variables dans une émission télévisée sont issus de sons provenant du milieu ambiant. Sons d'animaux, rires, voitures, avions, cloches, cris, explosions sont autant d'exemples qui peuvent aider à la détermination du contenu sémantique du document. Or, ils ont été jusqu'ici fort peu étudiés, car ils sont en général difficiles à appréhender. Au cours de ce stage, nous nous sommes intéressés à deux types de sons : les rires et les applaudissements. Ces sons sont très présents dans des émissions de télévision dites de « plateau » (Grand Échiquier, jeu…), et leur détection révèle la présence d'un événement remarquable.

Mon stage de DEA a pour but de comparer les performances de deux systèmes de classification de sons (rires, applaudissements) : d'une part, un système de référence est mis en œuvre à partir de modèles de mélanges de lois gaussiennes, d'autre part, un système est développé à partir de la théorie des méthodes à vecteurs de support. Le travail demandé est d'une part d'adapter le système de classification Parole/Musique existant [Pinquier02] à la classification Rires/Applaudissements et d'autre part de

s'approprier la théorie et de mettre en œuvre un système basé sur les méthodes à vecteurs de support.

Tout d'abord, je vais présenter un état de l'art qui décrit les paramètres couramment employés pour caractériser les signaux audio ainsi que les approches les plus utilisées pour la classification. Ensuite, la seconde partie décrit les méthodes les plus utilisées pour la classification, et aussi la théorie des méthodes à vecteurs de support. Dans la troisième partie, il s'agit de la modélisation et la mise en œuvre de deux systèmes d'indexation. Finalement, les résultats obtenus sur le corpus utilisé dans le cadre du projet RIAM FERIA (Framework pour l'Expérimentation et la Réalisation Industrielle d'Applications Multimédias) sont analysés afin d'établir des comparaisons et des conclusions sur les deux systèmes.

Chapitre 1 État de l'art

Ce chapitre débute par une caractérisation de la composante sonore, qui est ici la source d'information utilisée par le système d'indexation. Ensuite, différentes techniques de traitement et de représentation du flux audio sont présentées, et enfin quelques méthodes de classification et de reconnaissance des formes classiquement appliquées au signal sonore sont détaillées suivi d'une présentation des méthodes à vecteur support.

1.1 Indexation audio

1.1.1 L'analyse du contenu d'un document audio

On parle d'analyse automatique du contenu d'un document audio quand un processus logiciel permet d'extraire sa signification sémantique [Wang00]. Cela implique la segmentation du document en unités de signification, sa classification entre certaines catégories prédéfinies, son indexation pour le rendre disponible à la recherche et à la navigation.

A partir de la compréhension du contenu du signal audio, on peut dire par exemple, si une émission radiophonique est un journal d'informations, une publicité ou un match. Dans le cas d'émissions audiovisuelles, les résultats de l'analyse audio peuvent être combinés avec ceux issus de traitements vidéo pour améliorer la qualité de l'indexation.

1.1.2 Le signal audio

Le signal acoustique est un signal complexe. Il s'agit d'une perturbation d'air qui ne peut pas être comparée directement avec une autre dans l'échelle du temps : sa variabilité en amplitude et phase rend deux signaux différents alors qu'ils peuvent porter la même information. Cette variation de pression d'air peut avoir pour origine plusieurs sources. Il peut s'agir de l'appareil phonique humain, d'un instrument musical ou de l'environnement naturel.

La parole est le résultat d'une «phonation» (source) et d'une «articulation» (filtre) selon un modèle simple de la théorie acoustique de la production de la parole [Mariani02]. La source est un signal composé d'une partie périodique (vibration des cordes vocales) et d'une partie bruitée, utilisée séparément ou conjointement. Le conduit vocal est une cavité acoustique de forme complexe. Sa fonction est de transformer le signal de source par des phénomènes de résonance et anti-résonance. La parole est donc une alternance de sons voisés (quasi-périodiques) et de sons non voisés (bruit).

La musique traditionnelle est caractérisée par une multiplicité de tons, avec une unique distribution harmonique pour chaque instrument. Les variations de la longueur des tons n'ont aucune corrélation avec le processus d'articulation [Saunders96]. La bande passante de la production musicale est étendue jusqu'à la limite supérieure de réponse de l'ouïe (env. 20 kHz), alors que pour la parole les limites sont autour de 8 kHz.

Quand on parle de « sons du milieu ambiant », on fait référence à tout événement qui produit une variation (périodique ou bruitée) de la pression de l'air perçue par l'ouïe humaine. Dans cette catégorie se trouvent les sons produits par les animaux, par la nature et par les objets qui nous entourent.

La figure 1 montre quatre signaux dans le domaine temporel. Le signal de parole est non stationnaire tandis qu'il est plus stable pour la musique et les applaudissements. Le rire est un signal fortement bruité et difficile à modéliser.

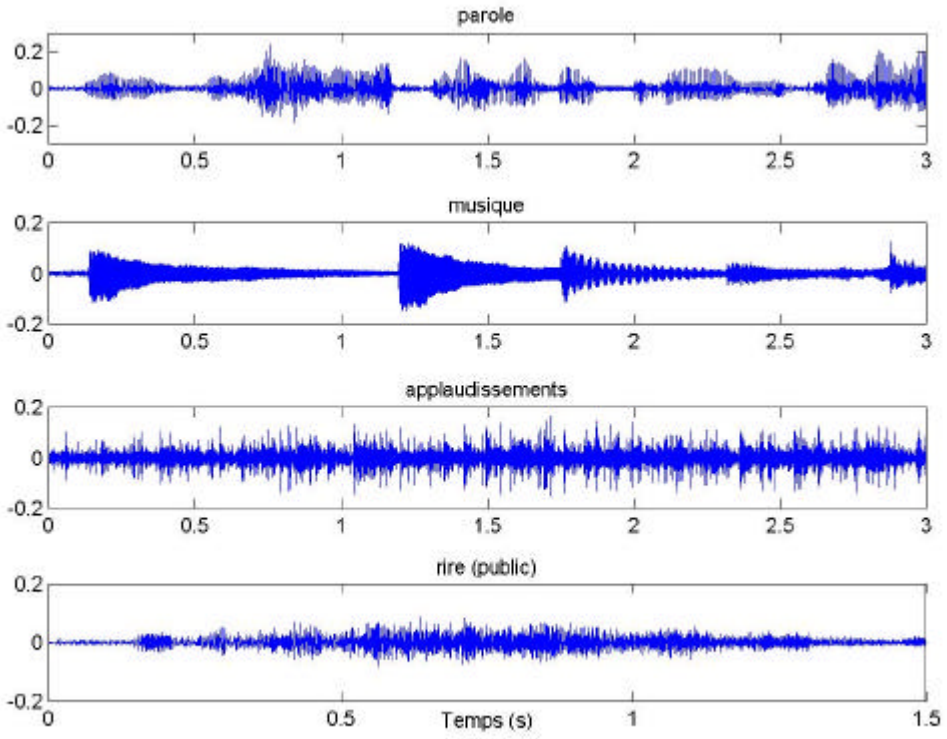


Figure 1: Divers types de sons. Le signal de parole montre des changements subis tandis que des caractéristiques plus régulières sont observables sur la musique et les applaudissements. Le rire du public dans une émission est un signal de bruit.

1.2 Paramètres audio

Plusieurs paramètres peuvent être utilisés pour caractériser les signaux audio et ainsi réduire la quantité d'information à traiter.

Dans une première étape de traitement, le signal est découpé en fenêtres d'analyse appelées trames[1] : une trame est définie comme une succession d'échantillons de 10 à 40 ms de longueur, où l'on considère le signal comme stationnaire. Néanmoins, pour trouver une signification sémantique, l'information doit être extraite sur des segments de durées d'une à plusieurs secondes.

1.2.1 L'énergie

L'énergie du signal, ou volume, calculée sur une trame, est un indicateur pour détecter le silence et ensuite préciser la frontière d'un segment. C'est aussi un critère pour différencier un signal de parole (composé de pics d'énergie) d'un signal de musique (plus stable).

Le volume dépend du gain du dispositif d'enregistrement et numérisation. Pour éliminer cette dépendance il est courant de normaliser sa valeur en fonction du volume maximum des trames adjacentes (1 à 2 s).

L'énergie d'une trame *n* est calculée comme la puissance quadratique moyenne de l'amplitude du signal.

$$e(n) = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} x_n^2(i)}$$

avec $x_n(i)$ = ième échantillon de la trame *n*. N est le nombre d'échantillons dans la trame. Dans une échelle plus proche de la perception de l'oreille humaine, on obtient :

$$e_{db}(n) = 10 \cdot \log_{10} \sum_{i=0}^{N-1} x_n^2(i)$$

Dans la parole, le rythme syllabique se situe autour de 4Hz [Scheirer97]. On peut observer le comportement de l'énergie du signal dans cette zone de fréquence, afin de caractériser la parole.

1.2.2 Le taux de passage à zéro (ZCR)

Le ZCR d'une trame est déduit du nombre de fois où le signal audio change de signe.

$$z(n) = \frac{1}{2N}\left(\sum_{i=1}^{N} \left| sign\left(x_n(i)\right) - sign\left(x_n(i-1)\right) \right|\right)$$

Le ZCR est une des mesures les plus appropriées pour distinguer d'une part la parole voisée de la fricative (qui présente une augmentation subite du ZCR) et d'autre

[1] La première étape complète de traitement consiste en une accentuation des aigus, un découpage avec une fenêtre de Hamming et un recouvrement sur la moitié de la trame [Rabiner93].

part, après une analyse plus étendue dans le temps, la parole de la musique. En effet, la musique ne montre pas de changements brusques du ZCR [Saunders96].

Selon certaines expérimentations [Lu02], la variation du ZCR est plus significative que sa valeur exacte, alors on peut en déduire la mesure « haute proportion du ZCR (HZCRR) », c'est-à-dire le nombre de trames dont le ZCR est supérieur a 1,5 fois la valeur moyenne du ZCR. Cette mesure est calculée sur plusieurs trames, ce qui se ramène à un temps d'analyse d'environ une seconde.

$$HZCRR = \frac{1}{2K}\left(\sum_{n=1}^{K}\left(sign\left(ZCR(n) - 1.5avZCR\right)+1\right)\right)$$

Ce paramètre est à manipuler avec précaution car il est très sensible au bruit ambiant.

1.2.3 Centroïde spectral

Le centroïde spectral est le « centre de gravité » du spectre pour une trame donnée [Scheirer97].

$$C(n) = \frac{\sum_{i=1}^{M} w_i \cdot S_n(w_i)}{\sum_{i=1}^{M} S_n(w_i)}$$

où $S_n(w_i)$ représente la composante spectrale de la trame n à la fréquence w_i.

Certains types de sons du milieu ambiant et de la musique possèdent des composantes de haute fréquence, ce qui entraîne une élévation de la moyenne spectrale. La parole se caractérise par un centroïde plus bas qui présente des variations importantes entre les zones voisées et non voisées. D'après [Wang 2000], $C(n)$ est associé a la sensation humaine de brillance d'un son.

Le spectre d'une trame peut être interprété comme l'énergie de sortie d'un banc de filtres, mais il permet aussi une interprétation statistique : du fait qu'il correspond à une fonction positive de la fréquence, il peut être considéré comme une fonction de densité de probabilité (pdf) non normalisée de la fréquence. Cette formalisation permet d'utiliser certains concepts appliqués aux variables aléatoires [Davy02]. Par exemple, avec une interprétation probabiliste, le centroïde spectral est le moment statistique de premier ordre (moyenne) du spectre au temps n.

1.2.4 Largeur de bande

A partir de l'interprétation du spectre comme une fonction de densité de probabilité pour un temps donné, la statistique de deuxième ordre (variance) du spectre est :

$$BndW(n) = \frac{\sum_{i=1}^{M} (w_i - C(n))^2 \cdot S_n(w_i)}{\sum_{i=1}^{M} S_n(w_i)}$$

Cette valeur est également connue comme la largueur de bande de la trame courante. Cette mesure caractérise la richesse en fréquences d'un son, c'est donc un indicatif de sa complexité.

1.2.5 Flux spectral

Le flux spectral est défini comme la variation du spectre entre deux trames successives :

$$SF = \sum_{i=1}^{M} \left(\frac{S_n(w_i)}{\|S_n\|} - \frac{S_{n-1}(w_i)}{\|S_{n-1}\|} \right)^2$$

Selon [Lu02], œtte valeur, calculée sur les trames de 1 s de durée, est plus élevée pour la parole que pour la musique. Les sons d'ambiance ont une valeur de SF très importante avec de fortes variations. Le SF se montre être un bon outil pour distinguer parole/musique/son ambiant.

1.2.6 Fréquence fondamentale

La fréquence fondamentale f_0 est très importante dans l'analyse et la synthèse de la parole. A priori seules la parole voisée et la musique harmonique ont une fréquence fondamentale bien définie, associée à la fréquence de vibration des cordes vocales et à la hauteur du son respectivement. Les plages de variation typiques du f_0 pour un humain sont comprises entre [100-250 Hz] pour un homme, [200-300 Hz] pour une femme, alors que pour la musique elles sont beaucoup plus étendues.

Ce paramètre n'est pas facile à estimer. Les méthodes les plus couramment utilisées sont de nature :

1. Temporelle ; elles mettent en œuvre la fonction de corrélation.
2. Spectrale ; elles sont fondées sur une analyse de la structure périodique de la transformée de Fourier. Par exemple, on peut chercher le plus grand commun diviseur de tous les maxima locaux de l'amplitude du spectre.

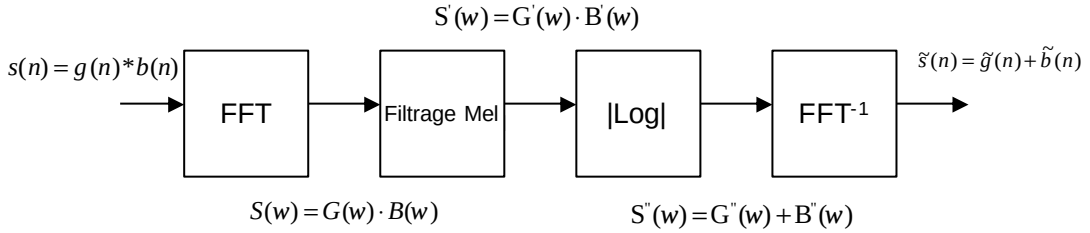
1.2.7 Analyse cepstrale

L'analyse cepstrale est un procédé particulièrement bien adapté au signal de parole [Mariani02]. Le cepstre complexe pour une trame *x(n)* est calculé en prenant la transformée de Fourier inverse du logarithme naturel de la transformée de Fourier de *x(n)*.

$$\hat{x}(n) = \frac{1}{2p}\int_{-p}^{p} \log(X(e^{iw}))e^{iwn}dw$$

Le cepstre peut être utilisé pour la séparation source-filtre (figure 2 et 3) du modèle de production de la parole. Si les deux composantes dans le domaine cepstral occupent des domaines distincts, on peut espérer les séparer. Dans le cepstre de la parole, en première approximation, les premiers coefficients correspondent à l'information relative au seul conduit vocal, sans interaction avec la période fondamentale. Cette paramétrisation est une des plus utilisée actuellement.

Transformation homomorphique $\quad s(n) \rightarrow \tilde{s}(n)$



$s(n)$ *signal acoustique*

$g(n)$ *source / exitation* (*cordes vocales*)

$b(n)$ *filtre* (*conduit vocal*)

$\tilde{g}(n)$ *suite d'impulsions aux inst. multiples de* n_0

$\tilde{b}(n)$ *décroit en* $1/n$

Figure 2: Création de coefficients cepstraux. Plusieurs opérations sont appliquées au signal *s(n)* : d'abord une transformée rapide de Fourier (FFT), ensuite un filtrage non linéaire calculé à partir de l'échelle perceptive de Mel [Rabiner93], une transformation logarithmique et finalement une FFT inverse. Les composantes $\tilde{g}(n)$ et $\tilde{b}(n)$ sont décorrélées dans le domaine temporel.

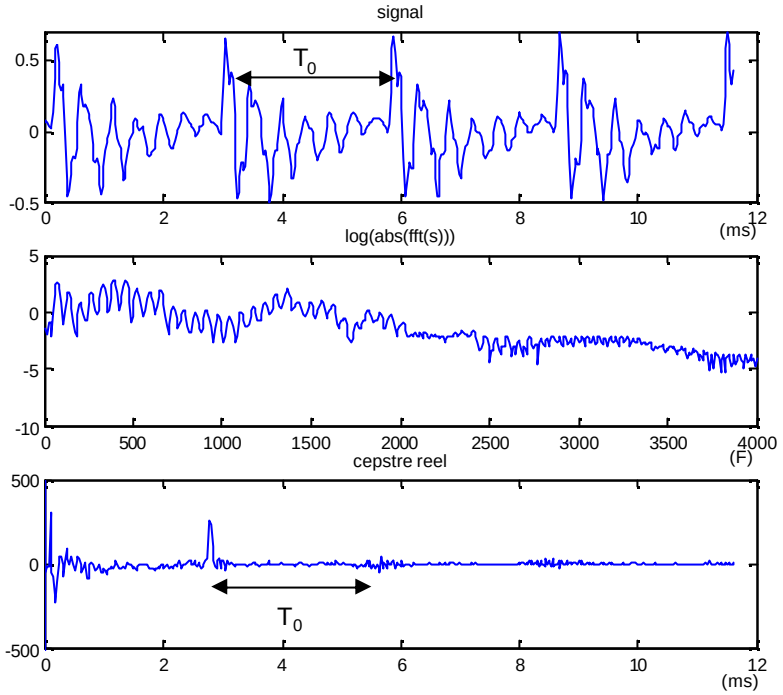


Figure 3: Signaux issus de la transformation homomorphique. Sur le cepstre apparaissent les multiples de T_0.

1.3 Classification

Après la première étape de paramétrisation du signal, chaque trame est représentée par un vecteur dans un espace vectoriel, muni d'un produit scalaire.

L'étape de classification évalue l'information contenue dans le vecteur pour prendre une décision sur la classe ou la région de l'espace à laquelle il appartient.

Dans cette section, on analyse différentes méthodes de classification. Chacune d'elles diffère selon les hypothèses prises lors de la modélisation des frontières entre les classes et la manière d'estimer une règle de décision pour toute observation possible.

1.3.1 Perceptron

Le premier algorithme itératif pour réaliser les classifications linéaires fut le perceptron de Frank Rosenblatt [Cristianini00]. Le perceptron est capable de séparer des données en deux classes. De façon simple, il est constitué d'un vecteur d'entrée, d'une fonction d'activation et d'une sortie. C'est l'élément basique d'un réseau de neurones.

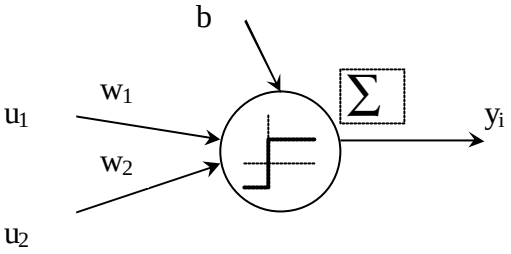


Figure 4: Schéma d'un perceptron.

La figure 4 montre un perceptron qui possède deux entrées et une sortie. Les données **u** à classifier sont multipliées par un poids **w** pour être additionnées à un compensateur b. Une fonction d'activation définit et limite sa sortie. La fonction d'activation produit une sortie égale à +1 ou -1 correspondant respectivement aux classes y_1 et y_2.

Le vecteur **w** est défini par apprentissage supervisé, ce qui nécessite un ensemble de données x_k (k = 1, 2, 3 … K) d'apprentissage, chacun avec la définition de la classe y_i à laquelle il appartient.

Règle d'apprentissage du perceptron

Si les deux classes sont linéairement séparables, il existe un vecteur de poids **w** avec les propriétés :

$\langle w \cdot u \rangle + b \geq 0$ pour chaque vecteur **u** qui appartient a la classe $y_{i=+1}$

$\langle w \cdot u \rangle + b < 0$ pour chaque vecteur **u** qui appartient a la classe $y_{i=-1}$

où $\langle w \cdot u \rangle = \sum_{i=1}^{n} w_i u_i = |w| \cdot |u| \cos q$ désigne le produit scalaire.

L'équation $\langle w \cdot u \rangle + b = 0$ définit un hyperplan qui sépare les deux classes. **w** correspond au vecteur normal à l'hyperplan.

L'algorithme de calcul de **w** est un algorithme itératif (figure 5).

A la k-ieme étape :

1). Si le k-ième vecteur d'apprentissage x_n est correctement classé, aucune modification n'est faite sur la valeur de **w**.

$w_{k+1} = w_k$

$b_{k+1} = b_k$

2). Sinon,

$w_{k+1} = w_k + n y_i x_k$

$b_{k+1} = b_k + n y_i$

Où *n* est le taux d'adaptation. Une valeur de *n* très élevée génère des basculements importants de l'hyperplan alors qu'une valeur trop petite demande plusieurs itérations d'actualisation.

La convergence de la règle d'apprentissage du perceptron dans un nombre fini d'itérations est assurée si la solution existe.

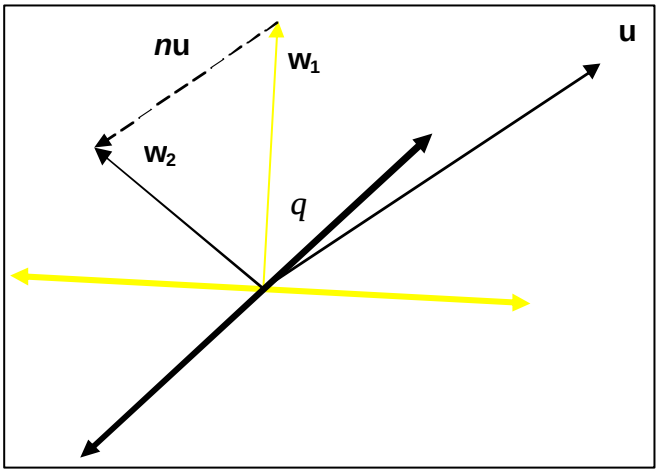


Figure 5: Règle d'apprentissage du perceptron. D'après la valeur initiale de **w** (w_1), le produit scalaire $|w_1| \cdot |u| \cos q$ est positif. La définition de la classe de **u** ($y_{i=-1}$) demande un changement en **w** donc un ajustement -*n***u** est effectué. Après une itération, le produit scalaire $|w_2| \cdot |u| \cos q$ devient négatif car q > 90°. Le vecteur **u** est désormais bien classé.

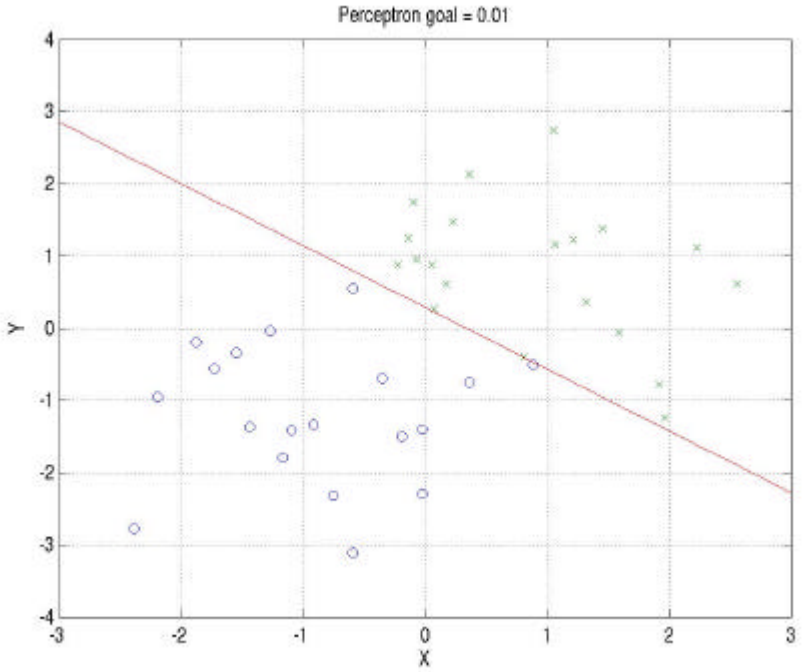


Figure 6: Exemple de classification de données 2d avec un perceptron. Les deux classes sont séparées par une droite. Le taux d'adaptation *n* = 0.01.

1.3.2 K plus proches voisins

Règle du plus proche voisin

L'idée de base de ce classificateur est d'attribuer à un vecteur **u** d'entrée la classe y_i du plus proche vecteur de caractéristiques disponible, selon un certain critère de distance (Euclidienne[2], Mahalanobis[3]). En cas d'égalité, une procédure spéciale doit être utilisée pour prendre une décision. Les frontières entre classes définies avec ce critère sont linéaires par segments et décrivent un polygone complexe.

En deux dimensions, l'algorithme du plus proche voisin divise l'espace de décision en cellules de Voronoi ; chaque cellule est étiquetée suivant la catégorie des vecteurs qu'elle contient.

Règle des K plus proches voisins

Le nombre de k vecteurs plus proches peut déterminer aussi le choix de la classification, car la méthode du plus proche voisin présente une sensibilité élevée aux frontières entre classes. A fin d'équilibrer cet effet, la classe assignée au vecteur **u** peut être celle qui est la plus représentée parmi les k plus proches vecteurs disponibles.

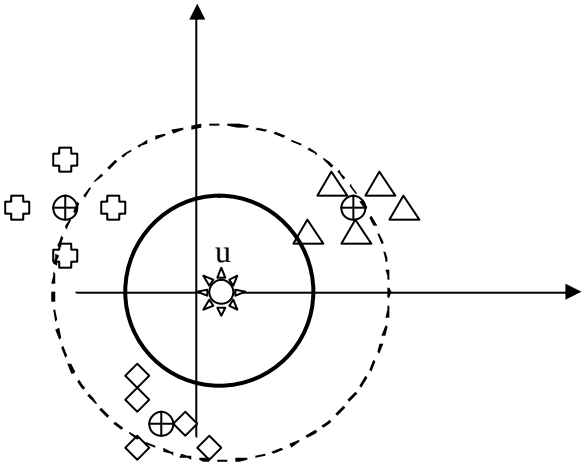
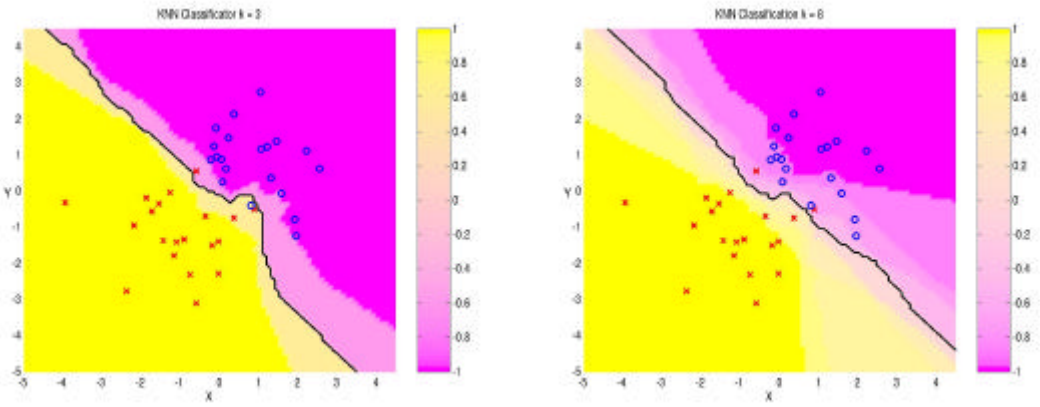


Figure 7: Un nouveau vecteur est classé avec la méthode des k plus proches voisins. Si k=1, la classe triangle est assignée à **u**. Néanmoins, si k=4, **u** appartient à la classe représentée par les carrés. Aucune notion d'apprentissage n'est nécessaire.

---

[2] $D(a,b) = \left(\sum_{k=1}^{d} (a_k - b_k)^2 \right)^{1/2}$

[3] $r^2 = (x-m)^t \Sigma^{-1} (x-m)$, distance d'un point x à la moyenne m d'une distribution de données. Σ^{-1} est l'inverse de la matrice de covariance de la distribution.



(a) (b)

Figure 8: Classification de deux classes de données avec la méthode de k plus proches voisins. Les frontières changent entre le critère (a) k = 3 et (b) k = 8 pour la même distribution de données.

1.3.3 Approche statistique et lois gaussiennes.

La théorie de la décision bayesienne quantifie les compromis entre les décisions de classification et les coûts qu'impliquent de telles décisions. Elle suppose que la nature du problème de décision est probabiliste et que l'on connaît toutes les valeurs de probabilité [Duda01].

Dans le cadre de la théorie bayesienne, le vecteur **u** que nous voulons classer est supposé appartenir à une classe y_i parmi plusieurs possibles, toutes caractérisées par une loi de probabilité *a priori* $P(y_i)$. Les probabilités a priori reflètent notre hypothèse initiale du problème, c'est à dire, comment le vecteur **u** sera classé sans le connaître davantage.

Les vecteurs **u** d'observation sont considérés être les réalisations de variables aléatoires dont les distributions sont différentes pour chaque classe y_i, exprimées comme une *densité de probabilité conditionnelle* $p(u|y_i)$ [Ash93].

La probabilité que le vecteur **u** appartienne à la classe y_i peut s'exprimer avec le théorème de Bayes :

$$P(y_i \mid u) = \frac{p(u \mid y_i)P(y_i)}{p(u)}$$

$$avec : \quad p(u) = \sum_{j=1}^{k} p(u \mid y_i)P(y_i)$$

Ce théorème montre que d'après la valeur de **u**, on peut transformer la probabilité a priori $P(y_i)$ en une probabilité a posteriori $P(y_i|u)$. On appelle $p(u|y_i)$ la similarité de y_i par rapport à **u** et p(u) le facteur d'échelle qui garantit que la somme de toutes les probabilités a posteriori est égale à un.

Pour un vecteur donné, les probabilités a posteriori de toutes les classes possibles sont calculées et la classe choisie est celle qui minimise le risque d'erreur. Par exemple, pour deux classes :

$$u \in y_1 \; si \; P(y_1 \mid u) > P(y_2 \mid u)$$

La probabilité $P(y_i)$ et la densité de probabilité $p(u \mid y_i)$ sont obtenues pendant une étape d'apprentissage à partir d'un ensemble de vecteurs $\mathbf{x}_k$. Ces vecteurs permettent d'obtenir des informations sur la « forme » de chaque classe et ainsi de déterminer les paramètres qui vont modéliser ces formes.

Parmi plusieurs fonctions de densité de probabilité, la fonction normale gausienne est en général privilégiée [Duda01]. D'un coté, les paramètres de cette loi sont faciles à estimer et d'un autre, elle permet de modéliser le cas des vecteurs $\mathbf{x}$ appartenant à une classe y_i comme s'ils étaient des versions d'un vecteur prototype m_i.

L'expression de la loi normale en d dimensions est :

$$N(m,\Sigma) = \frac{1}{(2p)^{d/2}\left|\Sigma\right|^{1/2}} e^{-\frac{1}{2}(x-m)^t \Sigma^{-1}(x-m)}$$

où :

m = vecteur moyenne

Σ = matrice de covariance

$\left|\Sigma\right|$ = déterminant de Σ

Σ^{-1} = inverse de Σ

Le problème de la modélisation de classes avec de lois gaussiennes est que l'on les utilise comme si elles étaient exactes, alors que très souvent on dispose de données d'apprentissage en quantité limitée.

Dans le cas de deux classes correspondant à deux distributions normales de données, N_1 et N_2, la fonction de décision pour classer un vecteur $\mathbf{u}$ est :

$$f(u) = sign\left[\frac{1}{2}(u-m_1)^t \Sigma_1^{-1}(u-m_1) - \frac{1}{2}(u-m_2)^t \Sigma_2^{-1}(u-m_2) - ln\frac{\Sigma_2}{\Sigma_1}\right]$$

(on mesure la distance de Mahalanobis de $\mathbf{u}$ à chaque vecteur moyenne et on assigne $\mathbf{u}$ à la classe plus proche).

L'estimation des paramètres des lois normales peut se faire de plusieurs manières. Les plus utilisées sont la méthode de maximum de vraisemblance (où les paramètres sont considérés comme fixes) et l'estimation bayesienne (où les paramètres sont considérés comme des variables aléatoires) [Tomasi04].

La méthode la plus simple est celle du maximum de vraisemblance. Elle précise chaque loi gaussienne avec :

$$m = \frac{1}{n}\sum_{k=1}^{n} x_k$$

$$\Sigma = \frac{1}{n}\sum_{k=1}^{n} (x_k - m)(x_k - m)^t$$

où $\mathbf{x}_k$ sont les vecteurs d'apprentissage de la loi gaussienne et Σ la matrice de covariance.

1.3.4 Machines à vecteurs de support [4]

La méthode des machines à vecteurs de support (SVM) est une alternative récente pour la classification. Puisque le problème est un problème de classification à deux classes, là encore, ce type de classification nécessite un ensemble d'apprentissage pour apprendre les paramètres du modèle.

Cette méthode repose sur l'existence d'un hyperplan séparateur dans un espace approprié.

1.3.4.1 Cas linéairement séparable

Le modèle le plus simple de SVM est celui appelé linéaire de marge maximale. Il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables. Ce SVM cherche à séparer les deux classes de données par un hyperplan qui est équidistant des « frontières » de chaque classe.

Maximalisation de la marge

Reprenons la terminologie utilisée dans le cadre du perceptron, avec l'équation $\langle w \cdot x \rangle + b = 0$ qui définit un hyperplan séparant deux classes. **w** correspond au vecteur normal à l'hyperplan.

La normale de l'hyperplan séparateur est normalisée. Par conséquent, les vecteurs les plus proches de **w**, notés $\mathbf{x}_+$ et $\mathbf{x}_-$, restent sur des hyperplans parallèles « canoniques » $\langle w \cdot x_+ \rangle + b = 1$ et $\langle w \cdot x_- \rangle + b = -1$ (figure 9). On peut regarder $\mathbf{x}_+$ et $\mathbf{x}_-$ comme projetés sur le vecteur unité $w/\|w\|$ (figure 10) [Schölkopf02].

[4] Les travaux de [Burges98], [Cortes95] et [Vapnik99] présentent des discussions intéressantes sur les principes de base des SVM.

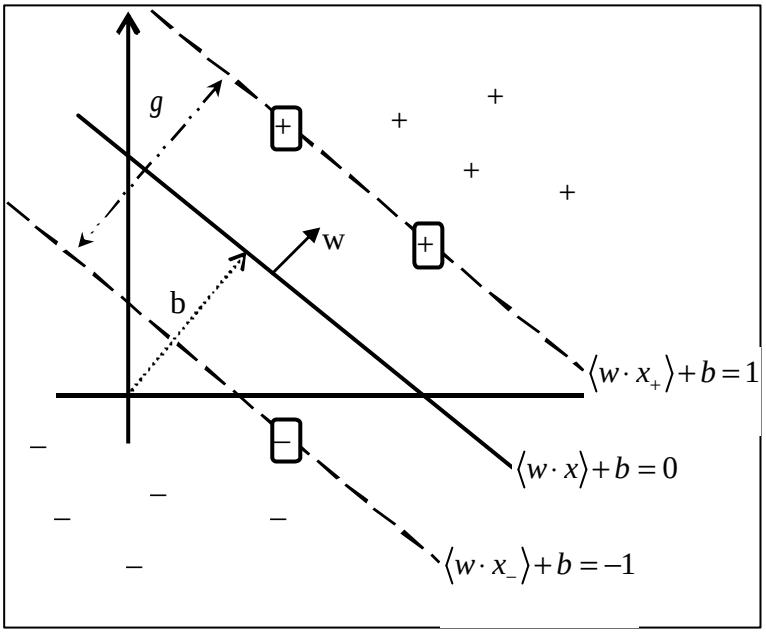


Figure 9: Hyperplan séparateur de deux classes (+) et (-). Il est défini comme de « marge g maximal », et situé au milieu des frontières entre classes.

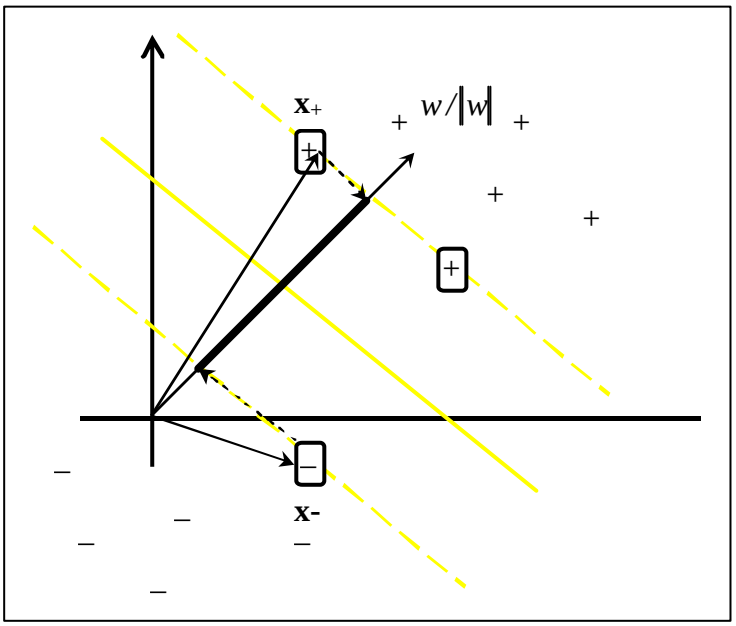


Figure 10: La marge est calculée à partir du produit scalaire entre les vecteurs situés à la frontière de chaque classe et le vecteur unitaire normal de l'hyperplan séparateur[5].

De cette manière les deux classes sont définies comme suit :

[5] [Bennet98] construit le polygone convexe de chaque classe et considère la marge comme la distance la plus courte entre les polygones.

$$\begin{cases} w \cdot x_i + b \geq +1 & si \quad y_i = +1 \\ w \cdot x_i + b \leq -1 & si \quad y_i = -1 \end{cases}$$

ce qui s'écrit :

$$y_i (w \cdot x_i + b) - 1 \geq 0 \quad \forall i \quad (1)$$

et la marge g est exprimée en fonction de $\mathbf{x}_+ - \mathbf{x}_-$:

$$g = \left(\left\langle \frac{w}{\|w\|} \cdot x_+ \right\rangle - \left\langle \frac{w}{\|w\|} \cdot x_- \right\rangle \right)$$

$$g = \frac{1}{\|w\|} \left(\langle w \cdot x_+ \rangle - \langle w \cdot x_- \rangle \right)$$

$$g = \frac{2}{\|w\|} \quad (2)$$

Pour maximiser la marge, il faut donc minimiser $\|w\|$. Par la suite, nous chercherons en fait à minimiser $(1/2) \cdot \|w\|^2$, pour simplifier les calculs. Les contraintes (1) à respecter indiquent qu'on ne permet aucune donnée d'apprentissage $\mathbf{x}_i$ dans la marge g. Un problème de minimisation d'une fonction sujet aux contraintes peut être formulé en utilisant la théorie des multiplicateurs de Lagrange.

Soit $S = ((x_1, y_1), \ldots, (x_l, y_l))$ *un ensemble de données d'apprentissage*

A. Minimiser $\frac{1}{2}\|w\|^2$ *avec* $y_i (\langle w \cdot x_i \rangle + b) - 1 \geq 0 \qquad i = 1, \ldots, l$

B. $L = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} a_i \left[y_i (\langle w \cdot x_i \rangle + b) - 1 \right]$

*C*1. $\frac{\partial L}{\partial w} = w - \sum_{i=1}^{l} a_i y_i x_i = 0$

$$w = \sum_{i=1}^{l} a_i y_i x_i$$

*C*2. $\frac{\partial L}{\partial b} = \sum_{i=1}^{l} a_i y_i = 0$

*Substitution de C*1 *en B :*

D. $L = \frac{1}{2}\left[\sum a_i y_i x_i \right]^2 - \left[\sum a_i y_i x_i \right] \cdot \left[\sum a_i y_i x_i \right] - b \sum a_i y_i + \sum a_i$

$$L_D = \sum_{i=1}^{l} a_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} a_i a_j y_i y_j \langle x_i \cdot x_j \rangle \quad (3)$$

Maintenant on doit résoudre (3) et trouver les paramètres a_i. On passe de la expression du Lagrangien L en termes de **w** et b à son expression par rapport à a_i (formulation duelle L_D) sous les contraintes C2 à respecter.

Il s'agit d'un problème d'optimisation quadratique convexe sujet à des contraintes linéaires (C2) qui limitent l'espace de recherche des solutions. Ce type de problème a été largement étudié et les solutions sont applicables aux SVMs [Schölkopf99].

La plupart des stratégies numériques suivent la procédure consistant à faire d'abord une estimation arbitraire de la solution et ensuite incrémenter la valeur de L_D sans dépasser la zone de la solution jusqu'à satisfaire un critère d'arrêt.

Quand les paramètres a_i ont été trouvés, le vecteur $w^* = \sum_{i=1}^{l} a_i y_i x_i$ représente le vecteur normal de l'hyperplan de « marge maximale ».

Vecteurs support

Sous les conditions de Karush-Kuhn-Tucker (KKT) pour un problème d'optimisation convexe (4) [Cristianini00], deux valeurs a_i sont possibles pour les vecteurs d'apprentissage : soit ils sont à l'extérieur de la contrainte ($y_i(\langle w \cdot x_i \rangle + b) - 1 > 0$) soit ils se trouvent dans la frontière ($y_i(\langle w \cdot x_i \rangle + b) - 1 = 0$). Les paramètres sont $a_i = 0$ pour le premier cas et $a_i > 0$ dans le deuxième. On en déduit la valeur de w, puis celle de b.

$$a_i \left[y_i(\langle w \cdot x_i \rangle + b) - 1 \right] = 0 \qquad i = 1,\ldots l \quad (4)$$

C'est un résultat où une condition implique que seuls les vecteurs $\mathbf{x}_i$ à une distance «1 » de l'hyperplan de marge maximale possédant un paramètre $a_i > 0$ sont pris en compte pour le calcul de **w**. Ils sont appelés «vecteurs de support (sv) ». Les vecteurs qui ne sont pas de support n'ont aucune influence dans la solution.

Une conséquence importante des conditions KKT est la définition de la marge en termes des a_i.

$$\langle w^* \cdot w^* \rangle = \sum_{j \in sv} y_j a_j \langle x_j \cdot w^* \rangle$$

$$\langle w^* \cdot w^* \rangle = \sum_{j \in sv} a_j y_j \sum_{i \in sv} y_i a_i \langle x_i \cdot x_j \rangle$$

$$\langle w^* \cdot w^* \rangle = \sum_{j \in sv} a_j (1 - y_j b)$$

$$comme \sum_{j \in sv} a_i y_i = 0,$$

$$\langle w^* \cdot w^* \rangle = \sum_{i \in sv} a_i$$

$$g = \frac{1}{\|w^*\|} = \left(\sum_{i \in sv} a_i \right)^{-1/2}$$

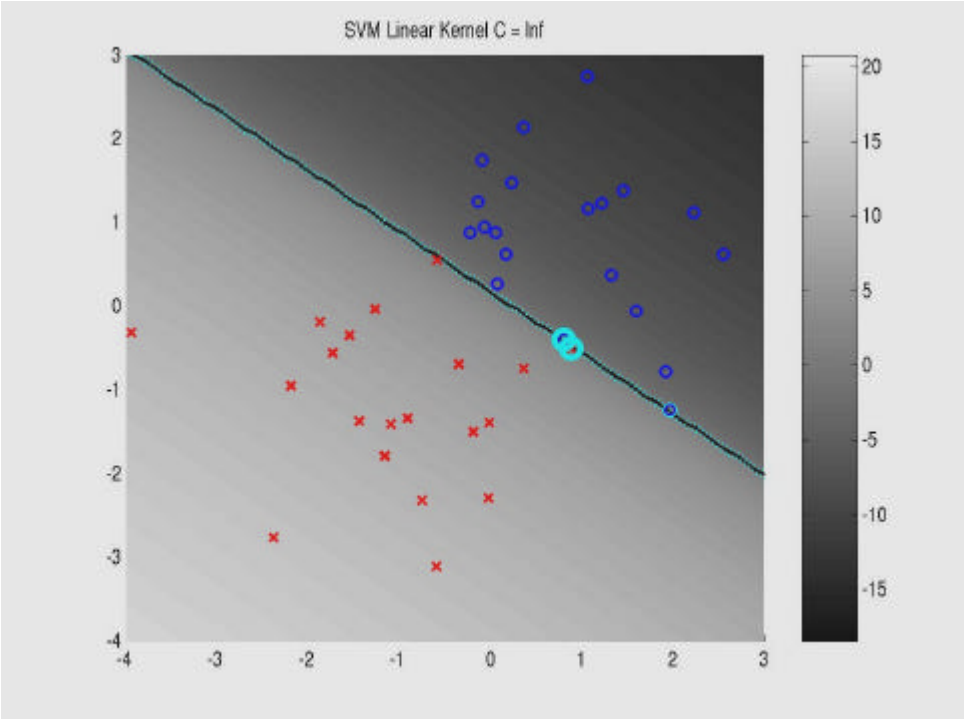


Figure 11: Classification de deux classes de données avec un SVM linéaire. Les vecteurs support ont été encerclés.

La fonction de décision pour la classification de vecteurs inconnus **u** est donnée par :

$$f(u) = sign\left(\sum_{i=1}^{m} a_i y_i \langle x_i \cdot u \rangle + b\right)$$

où m est le nombre de vecteurs de support.

1.3.4.2 Cas linéairement non séparable

Le classificateur de marge maximale ne peut pas être utilisé dans la plupart des problèmes réels : si les données ont été affectées par le bruit, il n'y a pas de séparation linéaire entre elles. Dans ce cas, le problème d'optimisation ne peut pas être résolu.

Pour surmonter ces inconvénients, de nouvelles mesures de la marge ont été proposées. Ces mesures tolèrent le bruit et prennent en compte les données d'apprentissage en plus de celles qui sont dans les frontières de la classe.

Le problème d'optimisation initial était :

$$\textit{Minimiser} \quad \frac{1}{2}\|w\|^2 \quad \textit{avec} \quad y_i(\langle w \cdot x_i \rangle + b) - 1 \geq 0 \qquad i = 1,\ldots,l$$

Il s'agit dans ce nouveau cas (dit de « marges douces ») de relâcher les contraintes de la marge. On introduit alors des variables d'écart (normalisées par rapport à **w**) $z_i \geq 0,\ i = 1,\ldots l$ dans la définition des contraintes :

$$\begin{cases} w\cdot x_i + b \geq +1 - z_i & si \quad y_i = +1 \\ w\cdot x_i + b \leq -1 + z_i & si \quad y_i = -1 \end{cases}$$

ce qui s'écrit :

$$y_i(w\cdot x_i + b) \geq 1 - z_i \qquad \forall i = 1\ldots l$$

Quand une erreur de classification intervient, la variable z_i a une valeur plus grande que 1, donc $\sum_i z_i$ est une borne supérieure du nombre d'erreurs à l'apprentissage. De là, un moyen naturel pour pénaliser les erreurs est de remplacer la fonction précédente à minimiser par $\frac{1}{2}\|w\|^2 + C\sum_i x_i$. D'où, le fait de choisir une valeur pour le paramètre C revient à définir une valeur pour $\|w\|$ en minimisant x pour cette valeur de **w**.

$$L = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} x_i - \sum_{i=1}^{l} a_i\left[y_i(\langle w\cdot x_i\rangle + b) - 1 + x_i\right] - \sum_{i=1}^{l} r_i x_i$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{l} a_i y_i x_i = 0$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{l} a_i y_i = 0$$

$$\frac{\partial L}{\partial x_i} = C - a_i - r_i = 0$$

$$L_D = \sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i,j=1}^{l} a_i a_j y_i y_j \langle x_i \cdot x_j\rangle$$

La seule différence avec le cas linéairement séparable est que :

$$C - a_i - r_i = 0$$
$$r_i \geq 0$$
$$a_i \leq C \quad \forall i = 1\ldots l \quad (5)$$

Le problème d'optimisation est équivalent au cas de la marge maximale, avec une contrainte additionnelle (5). Cette formulation est connue comme «contrainte de boîte », car chaque valeur a_i est limitée par 0 d'un côté et par C de l'autre. C s'est révélé être un compromis entre la précision et la régularisation (le contrôle de l'erreur).

Les conditions KKT sont désormais :

$$a_i\left[y_i(\langle w\cdot x_i\rangle + b) - 1 + x_i\right] = 0 \qquad i = 1,\ldots l$$
$$x_i(a_i - C) = 0 \qquad i = 1,\ldots l$$

Ces conditions impliquent que les variables d'écart soient différentes de zéro quand $a_i = C$, c'est à dire, quand leur marge est moins de $1/|w|$. Les vecteurs pour lesquels $0 < a_i < C$, sont considérés comme vecteurs support.

La fonction de décision pour la classification de vecteurs inconnus **u** reste :

$$f(u) = sign\left(\sum_{i=1}^{m} a_i y_i \langle x_i \cdot u \rangle + b\right)$$

La définition de la marge en termes des a_i est définie comme suit :

$$\langle w^* \cdot w^* \rangle = \sum_{j \in sv} y_j a_j \langle x_j \cdot w^* \rangle$$

$$\langle w^* \cdot w^* \rangle = \sum_{j \in sv} a_j y_j \sum_{i \in sv} y_i a_i \langle x_i \cdot x_j \rangle$$

$$\langle w^* \cdot w^* \rangle = \sum_{j \in sv} a_j (1 - x_j - y_j b)$$

$$\langle w^* \cdot w^* \rangle = \sum_{i \in sv} a_i - \sum_{i \in sv} a_i x_i$$

$$g = \frac{1}{\|w^*\|} = \left(\sum_{i \in sv} a_i - \frac{1}{C}\langle a \cdot a \rangle\right)^{-1/2}$$

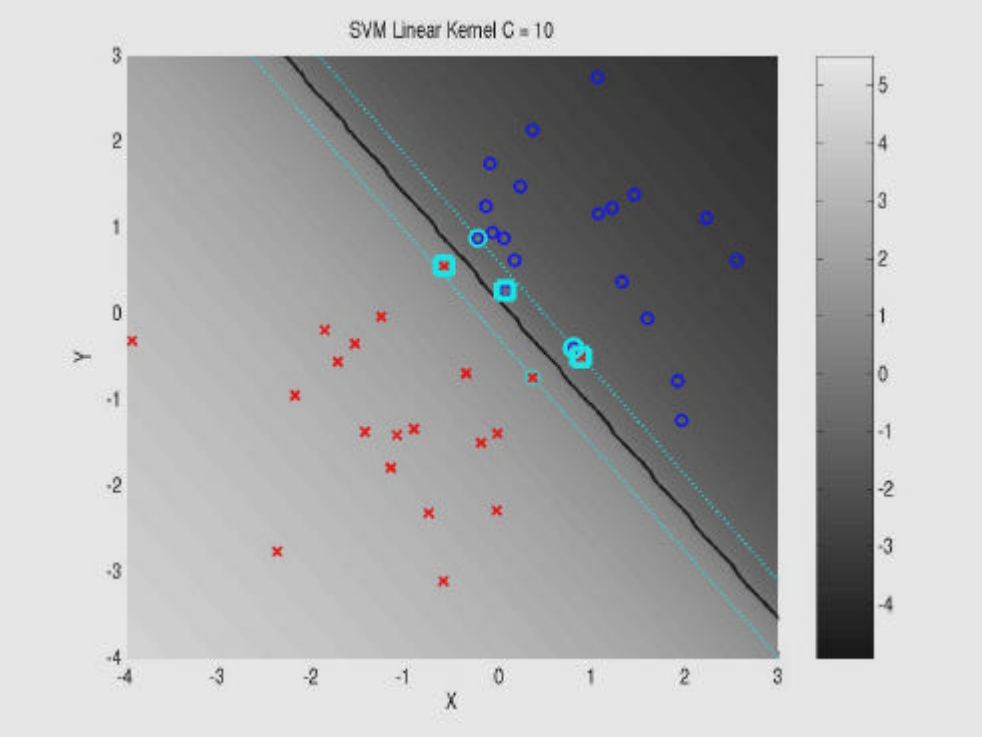


Figure 12: Classification de deux classes de données par une SVM linéaire de « marges douces ». La valeur de C = 10. Des erreurs de classification et de vecteurs à l'intérieur de la marge (points encadrés) sont permises. Les points encerclés sont les vecteurs de support.

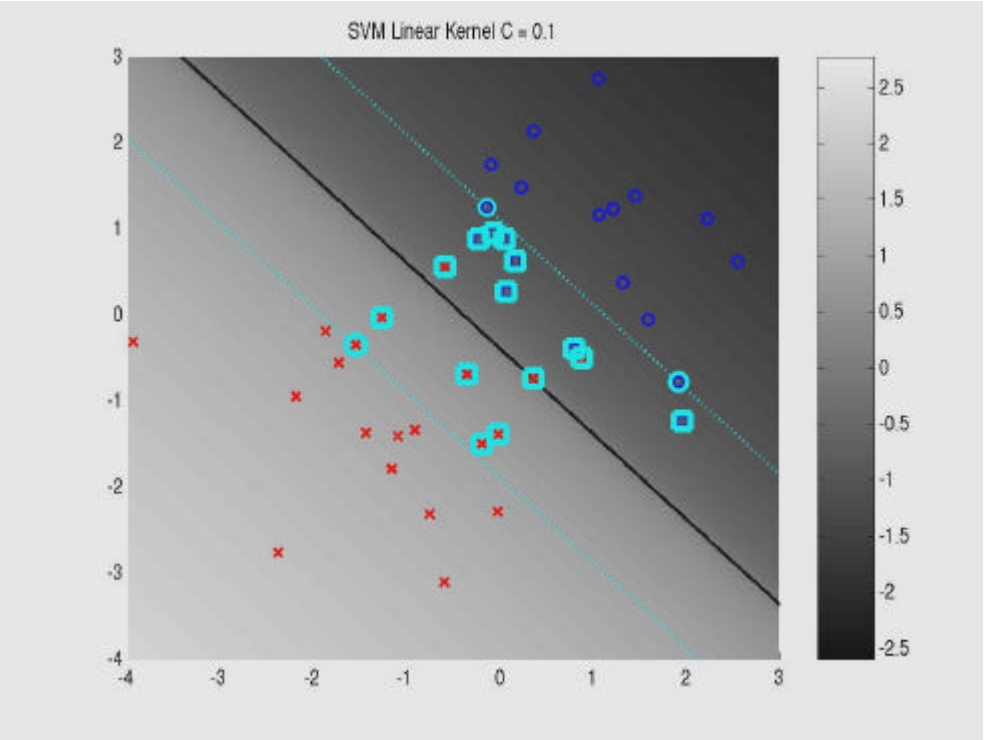


Figure 13: Même classification en prenant C=0,1. Avec une valeur de pénalisation plus petite la marge devient encore plus tolérante aux erreurs.

1.3.4.3 SVM non linéaires

En général, la plupart des applications ont besoin de fonctions plus complexes que les fonctions linéaires pour faire de la classification. Une stratégie de pré-traitement peut être utilisée pour simplifier la tâche [Schölkopf02]. Il s'agit de changer l'espace original d' « attributs » en un nouvel espace appelé de « caractéristiques », ce qui implique de trouver une fonction : $f \quad \Re^n \to \Re^N$

$$x = (x_1, \ldots, x_n) \mapsto f(x) = (f_1(x), \ldots f_N(x))$$

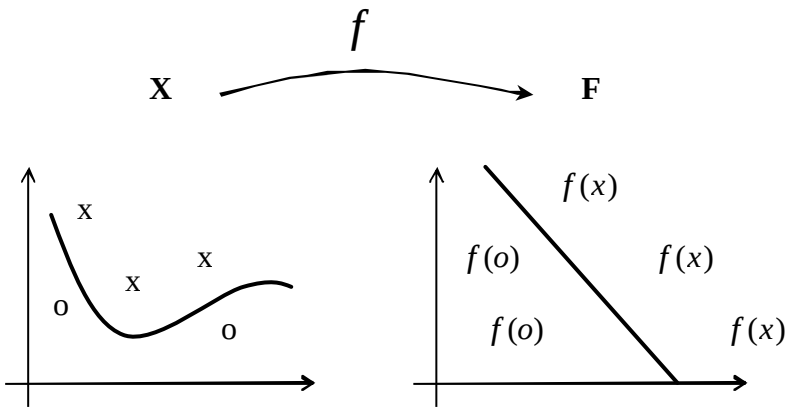


Figure 14: Un changement de représentation peut simplifier la classification.

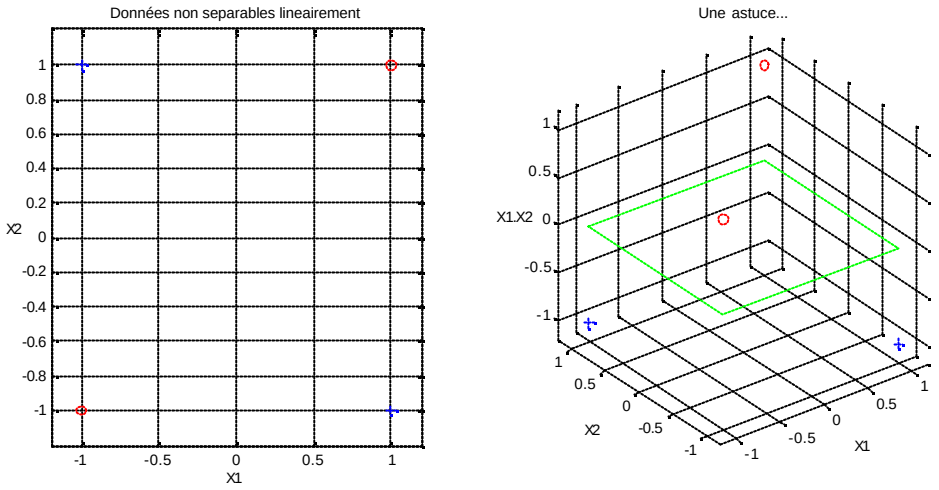


Figure 15: Un espace d'attributs 2d (x1, x2) peut être transformé en un espace 3d (x1, x2, x1.x2) qui rend explicite l'information pertinente.

Avec cette logique, deux étapes se dégagent pour construire un SVM non linéaire : (1) une transformation non linéaire pour placer les données dans l'espace de caractéristiques et (2) un SVM linéaire pour classifier les vecteurs.

La normale de l'hyperplan séparateur dans le nouvel espace devient :

$$w = \sum_{i=1}^{l} a_i y_i f(x_i)$$

Les a_i doivent être calculés dans l'espace de caractéristiques. La classification d'un vecteur inconnu se fait par la fonction :

$$f(u) = sign\left(w \cdot f(u) + b\right)$$

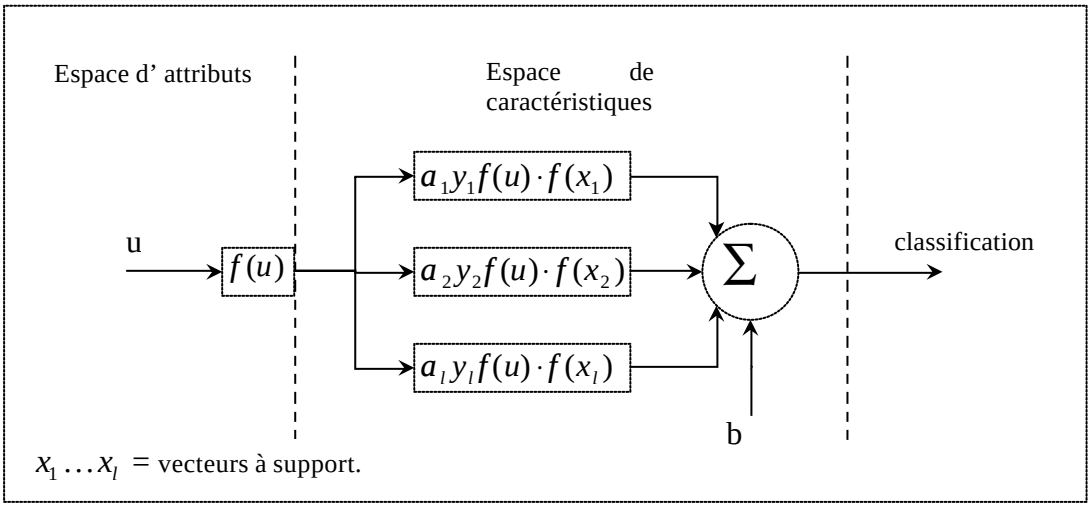


Figure 16: Classification du vecteur **u**. Le vecteur est transformé dans un espace de caractéristiques (de dimension plus haute). Un hyperplan séparateur construit dans le nouvel espace détermine la sortie de la classification.

Néanmoins, l'ordre des opérations pour construire la fonction de décision peut être changé [Cortes95] : au lieu de faire d'abord la transformation non linéaire et après le produit scalaire avec le vecteur normal de l'hyperplan, on peut en premier lieu comparer les vecteurs **u** et **w** dans l'espace d'attributs et faire la transformation non linéaire implicitement dans l'espace de caractéristiques.

Théorème de Mercer

Dans la littérature des SVM on se réfère à l'espace de caractéristiques comme un espace de Hilbert. L'espace de Hilbert est une généralisation de l'espace euclidien qui peut avoir un nombre infini de dimensions. Il est un espace linéaire où la transformation :

$$K(u,v)=\sum_i f_i(u)\cdot f_i(v)$$

est valable si et seulement si, pour toute fonction *g(u)* telle que :

$\int g^2(u)du$ est finie

alors,

$$\iint K(u,v)g(u)g(v)dudv>0 \qquad (6)$$

Le théorème nous indique si une fonction représente un produit scalaire dans un espace de Hilbert. Dans certains cas, il n'est pas évident de vérifier si le théorème de Mercer (6) est satisfait, néanmoins, dans [Burges98], on démontre que le théorème de Mercer est satisfait pour les puissances entières du produit scalaire.

$$K(u,v)=(\langle u\cdot v\rangle)^d$$

La fonction de décision d'un SVM devient :

$$f(u)=sign(w\cdot f(u)+b)=sign\left(\sum_{i=1}^{l} a_i y_i\langle f(u)\cdot f(x_i)\rangle+b\right)=sign\left(\sum_{i=1}^{l} a_i y_i K(u,x_i)+b\right)$$

La fonction *K(u,v)* est appelée « noyau » (kernel). Grâce à elle il est possible de transformer les vecteurs d'attributs de manière implicite vers l'espace de caractéristiques et d'estimer une fonction de décision linéaire dans cet espace.

Diverses fonctions satisfont le théorème de Mercer :

- Polynômes

$$K(u,v)=(\langle u\cdot v\rangle+1)^d$$

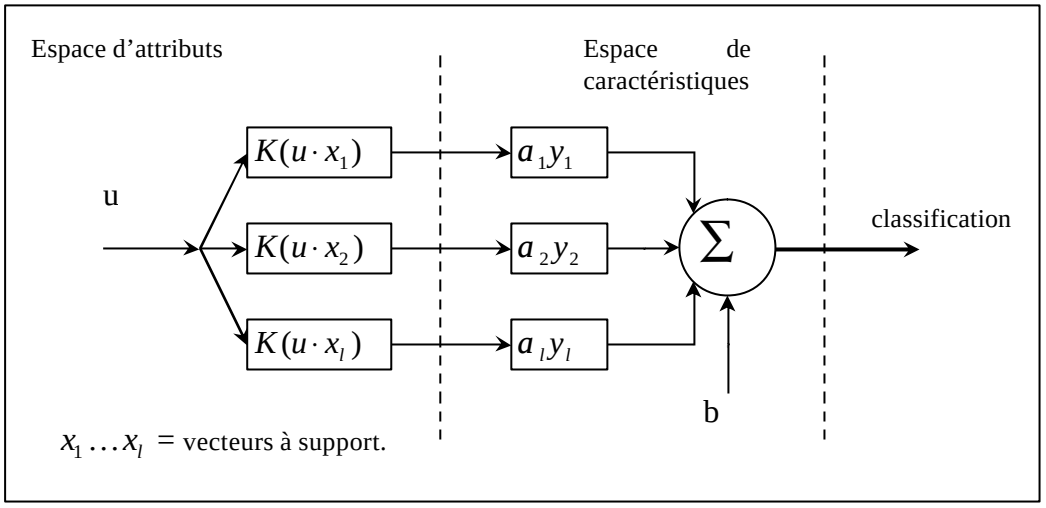


Figure 17: Classification du vecteur **u** par une SVM. Le vecteur est transformé de manière non linéaire dans l'espace d'attributs. Une fonction linéaire détermine la sortie du classificateur. La dimension de l'espace de caractéristiques n'affecte pas les calculs.

- Gaussiennes

$$K(u,v) = e^{-\|u-v\|^2/2s^2}$$

- Sigmoïdes

$$K(u,v) = \tanh(k\langle u \cdot v\rangle + r)$$

- Combinaisons des noyaux précédents

$$K(u,v) = \sum_i K_i(u,v), \prod_i K_i(u,v)$$

Comme exemple de transformation « explicite », un noyau polynomial de deuxième ordre change un espace d'attributs à deux dimensions en un espace de caractéristiques à six dimensions, comme le montrent les lignes suivantes :

$$u = [u_1\ u_2],\ v = [v_1\ v_2]$$
$$(\langle u \cdot v\rangle + 1)^2 = f(u) \cdot f(v)$$
$$(u_1 v_1 + u_2 v_2 + 1)^2 = f(u) \cdot f(v)$$
$$u_1^2 v_1^2 + u_2^2 v_2^2 + \sqrt{2}u_1 u_2 \sqrt{2}v_1 v_2 + \sqrt{2}u_1 \sqrt{2}v_1 + \sqrt{2}u_2 \sqrt{2}v_2 + 1 \cdot 1 = f(u) \cdot f(v)$$
$$f(u) = \left[u_1^2 \quad u_2^2 \quad \sqrt{2}u_1 u_2 \quad \sqrt{2}u_1 \quad \sqrt{2}u_2 \quad 1\right]$$
$$f(v) = \left[v_1^2 \quad v_2^2 \quad \sqrt{2}v_1 v_2 \quad \sqrt{2}v_1 \quad \sqrt{2}v_2 \quad 1\right]$$

En plus du paramètre C de pénalisation des erreurs, chaque noyau doit déterminer un certain nombre de paramètres pour ajuster sa forme à la distribution des données d'apprentissage. Le choix des paramètres adaptés est une étape cruciale, un ensemble trop encadré peut ne pas parvenir à séparer les données initiales, et au

contraire un ensemble trop libre peut aboutir à l'incapacité de généraliser. La méthode de validation croisée est utilisée pour trouver les valeurs les plus adaptées.

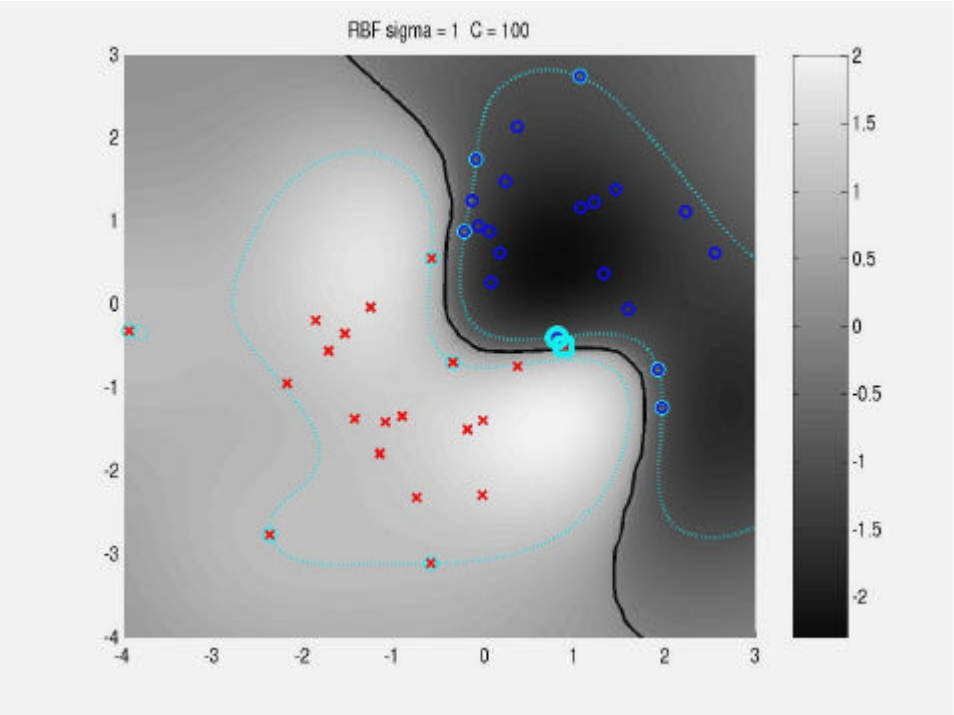


Figure 18: Classification de données avec un SVM non linéaire de type gaussien avec C = 100 et s = 1. La frontière est obtenue en coupant l'hypersurface $\sum_{i=1}^{l} a_i y_i K(u, x_i)$ à l'altitude –b.

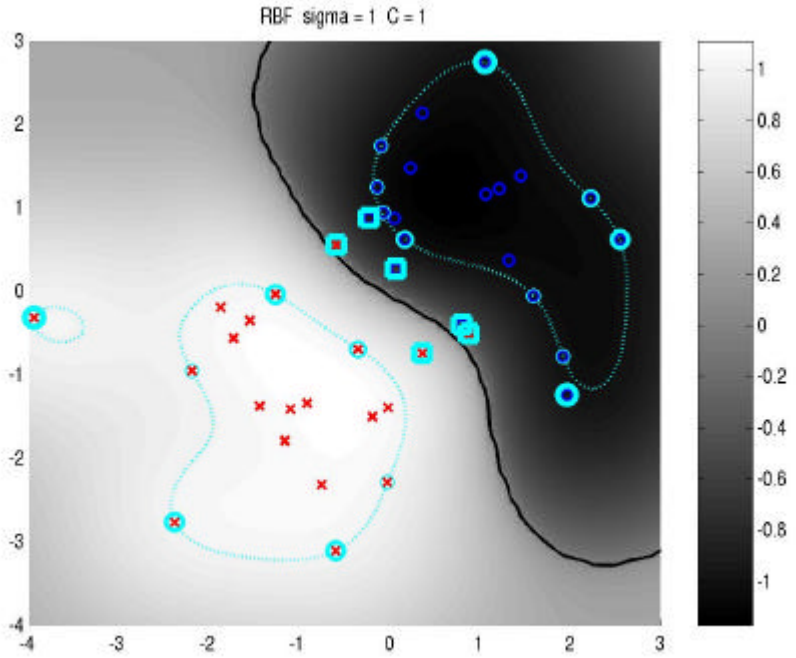


Figure 19: Classification de données avec un SVM non linéaire de type gaussien avec C= 1 et s = 1. La pénalisation des erreurs diminue donc les hyperplans canoniques s'écartent.

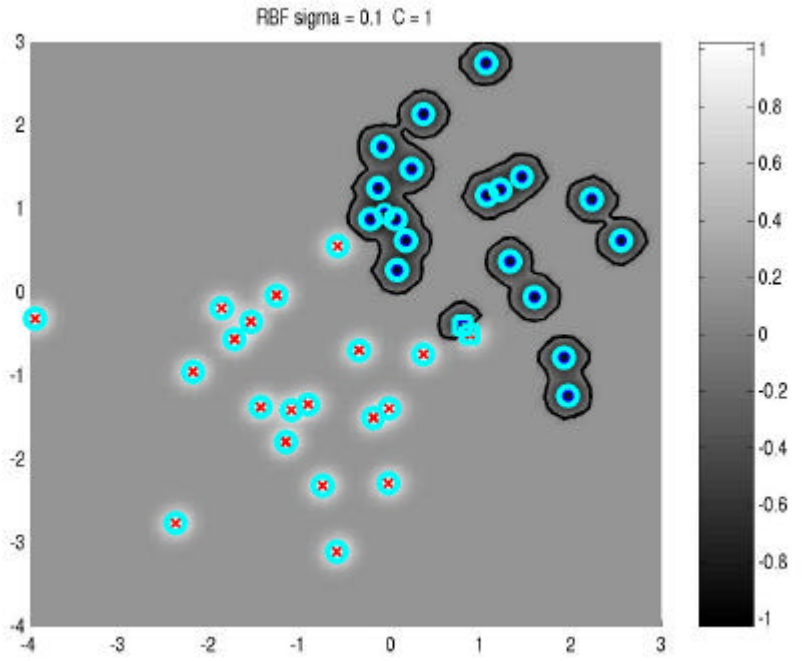


Figure 20: Classification de données avec un SVM non linéaire de type gaussien avec C = 1 et s = 0,1. Les risques de définition trop étroite (généralisation pauvre) peuvent se présenter.

Dans ce chapitre on a présenté des possibles représentations paramétriques des signaux et la description de certaines méthodes de classification. Une analyse de la théorie des machines à vecteurs de support est faite. Dans cette dernière apparaît l'idée d'utilisation d'une fonction noyau qui permet de faire de la classification dans les espaces de Hilbert à partir des coordonnées d'un espace « d'attributs ». Cette affirmation est particulièrement importante.

La modélisation de deux systèmes de classification basés sur l'approche bayesienne et les SVM est mise en œuvre dans le chapitre suivant. Une étape de validation de ces systèmes est présentée par la suite.

Chapitre 2 Modélisation

Nous allons présenter dans ce chapitre les modélisations effectuées pour classer deux types de sons : les rires et les applaudissements. Nous proposons pour chacun de ces sons un système de détection qui consiste à identifier sur chaque trame de signal, la présence ou l'absence du son en question. Chacun des systèmes se ramène donc à un système de classification en deux classes : présence du son / absence du son. Il se décompose en deux modules principaux qui sont le pré-traitement et le module de décision. Nous avons exploré deux approches :

1. La plus classique consiste à représenter chacune des deux classes par une distribution de type mélange de gaussiennes (MMG).
2. L'alternative innovante est basée sur l'utilisation des machines à vecteurs de support (SVM) pour séparer ces deux classes.

Dans la suite de ce chapitre, après avoir donné quelques indications sur la nature des sons étudiés, nous préciserons chacune des deux approches, les paramétrisations et les modèles utilisés ainsi que les problèmes de mise en œuvre.

2.1 Sons d'intérêt : rires et applaudissements

Les applaudissements (figure 21) sont des signaux d'un contenu spectral et d'une durée très uniforme. Par contre, les rires (figure 22) présentent une grande variabilité car les personnes rient de plusieurs manières différentes. Le signal généré peut ressembler à une voyelle plus une fricative velaire ou glottale, entre autres possibilités.

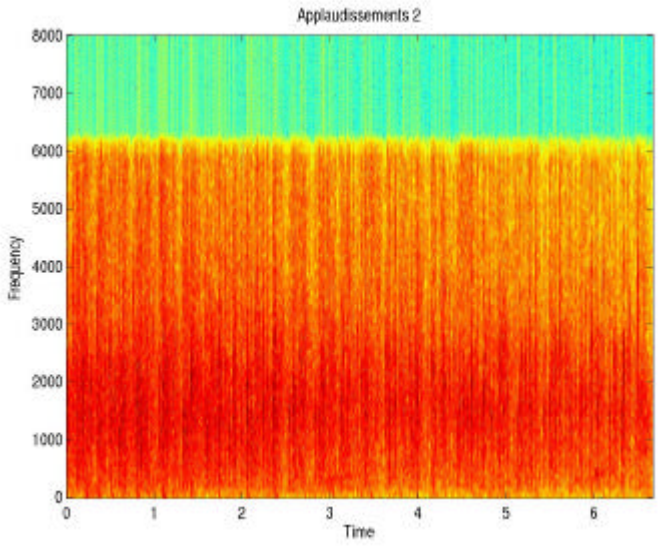
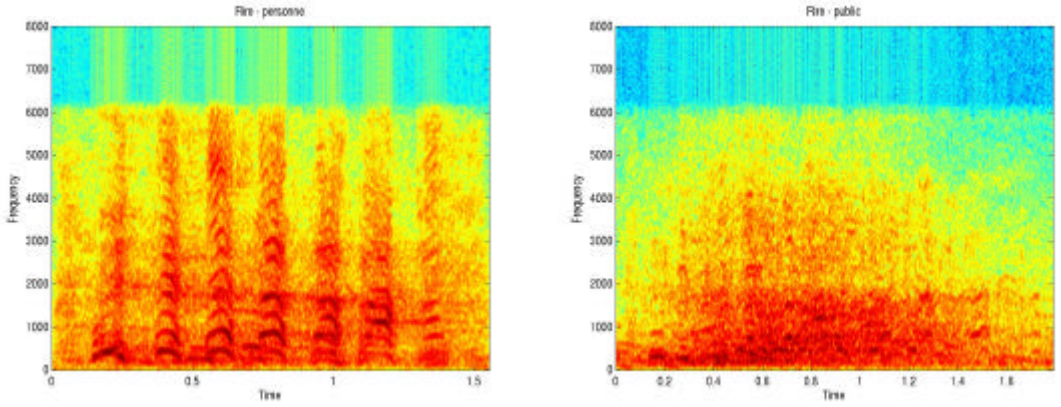


Figure 21: Spectrogramme d'une séquence d'applaudissements. Il s'agit d'un signal très régulier dont l'énergie est concentrée autour des 2 kHz.



(a) (b)

Figure 22: Rires d'une personne (a) et du public (b) au cours d'une émission télévisée. Avec une forte structure formantique, le premier exemple est plus facile à caractériser que le deuxième, qui est difficile à distinguer d'un bruit.

2.2 Pré-traitement

Le signal acoustique est numérisé. Il est échantillonné à 16 kHz avec 16 bits de résolution en amplitude.

Une accentuation des aigus est réalisée car les composantes de fréquence élevée sont plus atténuées pendant la transmission du son dans l'air. On fait un filtrage du type passe-haut avec la fonction de transfert :

$$H(z) = 1 - 0.98z^{-1}$$

La représentation paramétrique du signal est faite sur des trames de 10 à 50 ms, durées où ce signal est considéré stationnaire. Le fait de limiter le nombre d'échantillons peut être vu comme la multiplication terme à terme de la totalité du signal par une fenêtre rectangulaire. Cette multiplication équivaut, du point de vue spectrale, à convoluer la transformée de Fourier du signal par la transformée de Fourier de la fenêtre. Cette opération de convolution a pour effet d'introduire des ondulations parasites dans le spectre.

La pondération par une fenêtre de Hamming est utilisée pour atténuer cet effet. Elle est un bon compromis entre la limitation du signal et les effets de distorsion dus à la convolution des transformées.

Entre les diverses possibilités de représentation des trames de signal acoustique de manière paramétrique, nous utilisons les analyses cepstrale et spectrale.

Analyse cepstrale (MFCC, Mel Frequency Cepstral Coefficients)

Après l'accentuation des aigus et le fenêtrage de Hamming, une transformée de Fourier (FFT) est calculée sur la trame d'analyse.

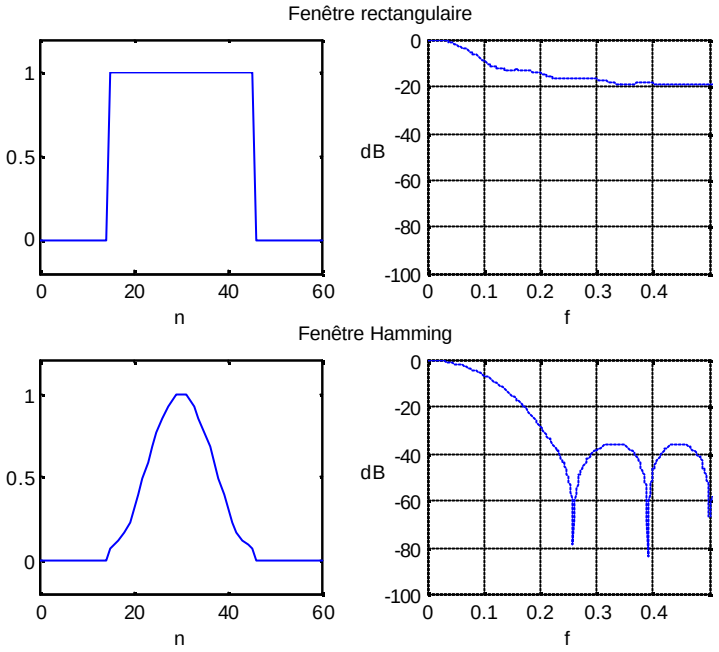


Figure 23: Transformée de Fourier des fenêtres rectangulaire et Hamming. La réduction en hauteur des lobes secondaires s'accompagne de l'élargissement du lobe principal.

Un filtrage de type Mel[6] s'effectue sur la transformée : des filtres triangulaires sont centrés sur les fréquences 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1150, 1300, 1500, 1700, 2000, 2350, 2700, 3100, 3550, 4000, 4500, 5050, 5600, 6200 et 6850 Hz. Ils sont appliqués après chaque FFT.

Ensuite, on calcule le logarithme du module de ce spectre puis la FFT inverse pour extraire l'énergie et les 8 premiers coefficients cepstraux. A l'aide des quatre trames adjacentes, on calcule aussi les dérivées de chaque coefficient et de l'énergie.

Pour normaliser l'effet du canal de transmission, chaque coefficient cepstral est diminué de la valeur moyenne des coefficients, prise sur plusieurs secondes.

A la sortie de l'analyse cepstrale, un vecteur de 18 paramètres représente la trame. Les paramètres sont :

- L'énergie
- 8 coefficients cepstraux
- La dérivée de l'énergie
- 8 dérivées des coefficients cepstraux

Analyse spectrale

Après le fenêtrage de Hamming et le calcul de la FFT sur la trame, on effectue un filtrage triangulaire centré sur les fréquences 100, 200, 300, 400, 500, 600, 700, 800,

[6] L'échelle de Mel est le résultat d'études qui montrent que la perception humaine de la fréquence de sons ne suit pas une échelle linéaire.

900, 1000, 1200, 1400, 1600, 1800, 2000, 2300, 2600, 2900, 3200, 3600, 4000, 4400, 4900, 5400, 5900, 6400, 6900 et 7400 Hz.

On calcule l'énergie totale et l'énergie pour chaque filtre. A la sortie de l'analyse spectrale, 29 coefficients sont extraits par trame :

- L'énergie
- 28 coefficients spectraux

2.3 La classification

Les deux stratégies de décision étudiées résolvent un problème de classification à deux classes que nous noterons C_1 et C_2, et nécessitent un apprentissage supervisé.

Dans l'approche MMG, les vecteurs d'observation de chacune des deux classes sont modélisés par une distribution probabiliste de type mélanges de lois gaussiennes. Dans l'approche SVM, la frontière de décision est obtenue après définition d'un noyau et des vecteurs support.

Ces deux approches sont dites de type paramétrique puisqu'elles nécessitent l'estimation de paramètres pour définir soit les pdfs soit les vecteurs support. Il s'en suit que la mise en œuvre d'un tel système se décompose en deux phases (figure 24) :

- La phase d'apprentissage au cours de laquelle les estimations de paramètres sont faites.
- La phase de classification proprement dite au cours de laquelle tout nouveau signal est prétraité pour donner une succession de vecteurs d'observations, qui sont classés en fonction de la stratégie. Un éventuel lissage permet d'éliminer les décisions aberrantes (figure 25).

Nous appliquons ces deux types de systèmes à la classification {Rire, Non rire} et la classification {Applaudissement, Non applaudissement}. Nous allons maintenant détailler ces deux systèmes et les algorithmes nécessaires à leur mise en œuvre.

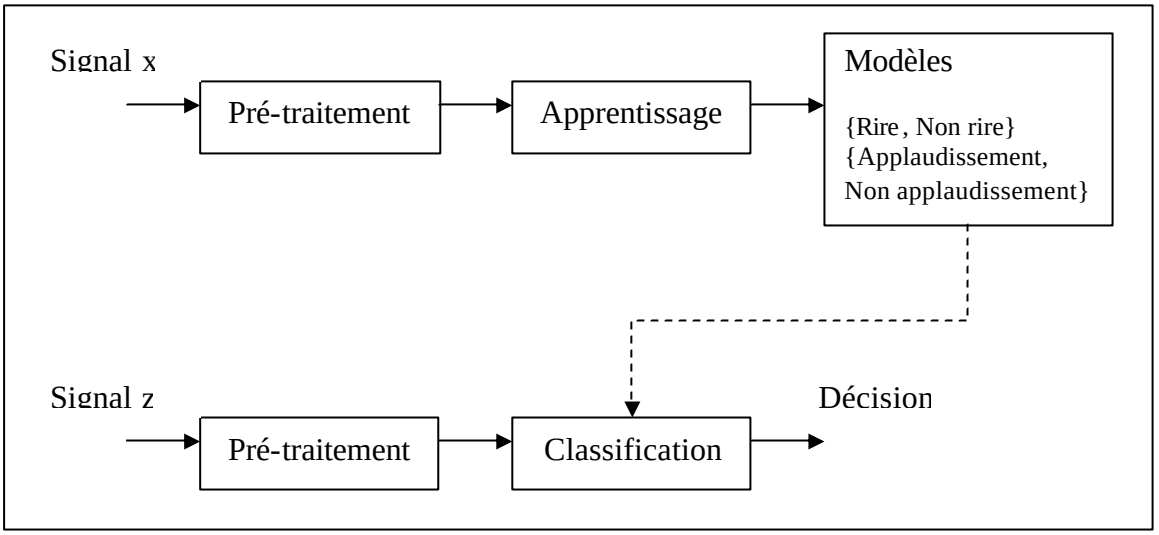


Figure 24: Système de classification. Une étape d'apprentissage est nécessaire pour construire les modèles de référence. La reconnaissance d'information inconnue est faite grâce à une mesure de similitude du nouveau signal par rapport aux modèles issus de l'apprentissage.

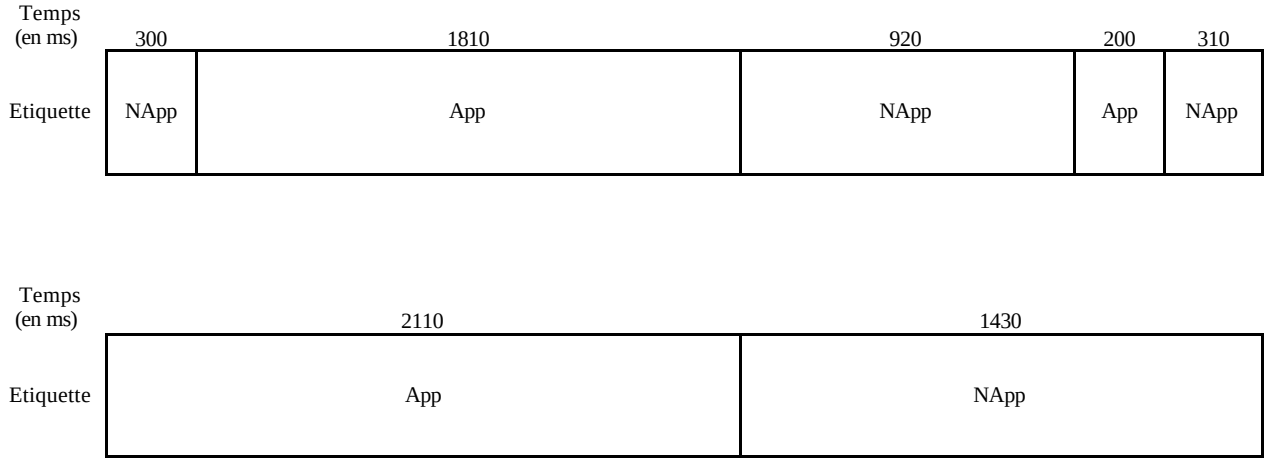


Figure 25: Lissage de 500 ms sur une indexation Applaudissement/Non applaudissement. Les segments de taille inférieure à l'intervalle défini sont fusionnés avec leurs voisins.

2.3.1 Système de classification de référence : Mélange de lois gaussiennes (MMG).

Les vecteurs d'observations relatifs à chacune des classes C_i sont les réalisations issues d'une densité de probabilité de la forme :

$$f(x) = \sum_{k=1}^{K} p_k N(x, m_k, \Sigma_k) = \sum_{k=1}^{K} p_k N_k(x, q)$$

$$p_k \geq 0 \quad et \quad \sum_{k=1}^{K} p_k = 1$$

Les paramètres q de la loi sont p_k, m_k, Σ_k.

Phase d'apprentissage

Après l'étiquetage manuel et le pré-traitement des vecteurs d'apprentissage, on peut effectuer l'estimation des paramètres pour chacune des deux lois. L'apprentissage des paramètres du mélange se réalise par maximisation de la fonction de vraisemblance :

$$\Lambda(q) = \prod_{n=1}^{N} f(x_n)$$

où $\{x_n, n = 1,\ldots,N\}$ est l'ensemble d'apprentissage correspondant.

Cette maximisation est réalisée à l'aide de l'algorithme EM.

Algorithme EM

La probabilité conditionnelle que le vecteur **x** soit généré par la loi gaussienne N_k est :

$$p(N_k / x) = \frac{p_k N_k(x,q)}{\sum_{k=1}^{K} p_k N_k(x,q)}$$

Pour trouver les expressions des optima locaux, on calcule les dérivées du logarithme de la fonction de vraisemblance par rapport à p_k, m_k, Σ_k. Ces dérivées donnent comme résultat des expressions itératives, où les variables $p_k^{t+1}, m_k^{t+1}, \Sigma_k^{t+1}$ sont exprimées en termes de $p^t(N_k / x)$ à la (t+1) itération.

1. Utilisation de l'algorithme de quantification vectorielle (VQ) pour initialisation des lois N_k : m_k^0, Σ_k^0 et des poids p_k^0.
2. Calcul de $p^t(N_k / x_n)$ pour chaque loi N_k et chaque vecteur d'apprentissage $\mathbf{x}_n$ (« Expectation »).
3. Calcul de $p_k^{t+1}, m_k^{t+1}, \Sigma_k^{t+1}$. (« Maximisation »).

$$p_k^{t+1} = \frac{1}{N}\sum_{n=1}^{N} p_k^t(N_k / x_n)$$

$$m_k^{t+1} = \frac{\sum_{n=1}^{N} p_k^t(N_k / x_n)\cdot x_n}{\sum_{n=1}^{N} p_k^t(N_k / x_n)}$$

$$\Sigma_k^{t+1} = \frac{\sum_{n=1}^{N} p_k^t(N_k / x_n)\cdot \left\| x_n - m_k^{t+1} \right\|^2}{\sum_{n=1}^{N} p_k^t(N_k / x_n)}$$

4. Evaluation de la fonction de vraisemblance. Si sa variation relative est suffisamment faible alors arrêt sinon aller à l'étape 2.

A la sortie de cette phase, on obtient un modèle de densité de probabilité adapté à nos données d'apprentissage ; il correspond à un maximum local de la fonction de vraisemblance.

Cet algorithme suppose l'existence d'un mélange initial obtenu à l'aide d'un algorithme de quantification vectorielle (VQ).

Ici, la VQ est exécutée au moyen de l'algorithme LBG (Linde, Buzo et Gray) : on détermine les *K* meilleurs représentants d'une distribution inconnue à partir d'une séquence de vecteurs d'apprentissage. *K* est un paramètre fourni par l'utilisateur et représente le nombre de lois gaussiennes avec lesquelles on veut modéliser une classe particulière.

L'ensemble des étapes de cet apprentissage est résumé par la figure 26. Deux modèles correspondant à chacune des deux classes sont ainsi estimés :

Pour le système du « rire », un modèle représente les trames de rire et un modèle représente les autres trames.

Pour le système des « applaudissements », un modèle représente les trames d'applaudissements et un modèle représente les autres trames.

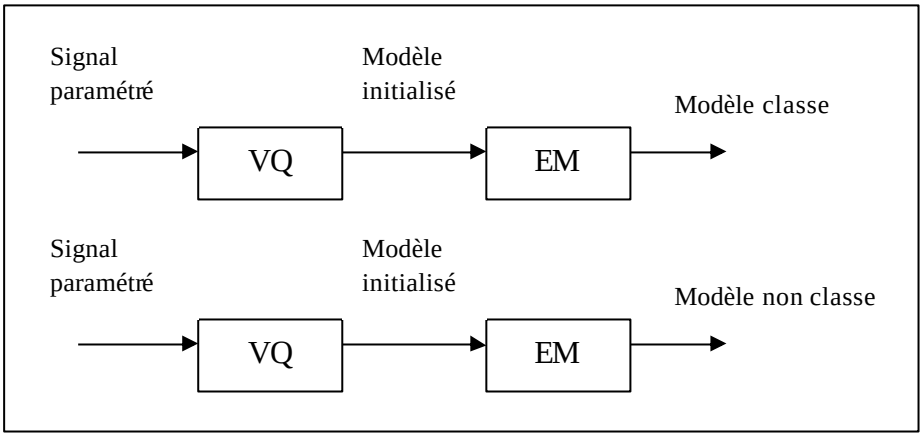


Figure 26: Apprentissage de modèles MMG. L'algorithme d'initialisation VQ est suivi par l'optimisation EM.

Reconnaissance

Un vecteur à classer est évalué par chaque modèle de référence (classe, non classe). Le maximum de vraisemblance définit son appartenance à une classe.

Le lissage est une fonction créée pour trouver les segments représentatifs des sons indexés. L'intervalle utilisé varie de 300 à 2500 ms. L'étiquette d'un segment lissé est celle qui est la plus représentée dans l'intervalle choisi.

2.3.2 Système de classification basé sur les Machines à Vecteurs de Support (SVM).

Ce système met en œuvre la nouvelle technique de classification, SVM, que sera comparé à celle de référence, MMG.

Apprentissage

L'apprentissage d'une SVM est en réalité un problème d'optimisation. On doit trouver les solutions a_i de la formulation du Lagrangien :

$$L_D = \sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i,j=1}^{l} a_i a_j y_i y_j K(x_i \cdot x_j) \quad (7)$$

sujet aux contraintes :

$$\sum_{i=1}^{l} a_i y_i = 0$$
$$0 \le a_i \le C \quad i = 1,\ldots,l$$

l est le nombre de données d'apprentissage. y_i sont les étiquettes de la classe de chaque donnée $\mathbf{x}_i$. Le paramètre C et la fonction noyau sont choisis par l'utilisateur.

L_D est une fonction convexe alors on peut toujours trouver une solution qui satisfait les conditions KKT. Elles sont nécessaires et suffisantes pour convergence.

$$y_i f(x_i) \begin{cases} \geq 1 & \textit{pour vecteurs avec} \quad a_i = 0 \\ = 1 & \textit{pour vecteurs avec} \quad 0 < a_i < C \\ \leq 1 & \textit{pour vecteurs avec} \quad a_i = C \end{cases}$$

où $f(x_i)$ est la fonction de classification évalué en (x_i).

Dans la pratique, toutes ces conditions sont approximées numériquement avec un certain niveau de tolérance.

Les valeurs a_i, x_i, b constituent notre modèle d'apprentissage. Dès que l'on connaît une solution pour a et b**,** la fonction de classification s'écrit :

$$f(u) = sign\left(\sum_{i=1}^{m} a_i y_i K(u, x_i) + b\right)$$

u est la donnée à reconnaître
m est le nombre de vecteurs support, pour lesquels $0 < a_i < C$.

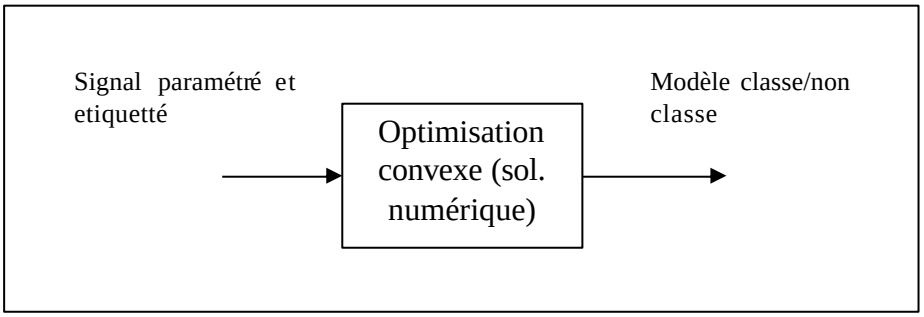


Figure 27: Apprentissage du modèle SVM. Divers algorithmes d'optimisation peuvent résoudre le problème de programmation quadratique posé par le Lagrangien L_D.

Comme nous l'avons présenté au chapitre 1, le problème réside dans l'estimation des paramètres a_i .

Nous présentons deux algorithmes de solution au problème d'optimisation. Le premier (gradient ascendant) a été utilisé au début du stage pour résoudre les problèmes simples. Une méthode plus performante (optimisation séquentielle minimale) a été utilisée pour l'indexation des émissions télévisuelles.

Première solution : gradient ascendant

La solution numérique la plus simple est celle de « gradient ascendant » : l'algorithme commence par l'estimation d'une solution a^0 et il actualise itérativement

le vecteur en suivant la direction d'incrément de L_D. La version séquentielle de cet algorithme (gradient stochastique) modifie seulement une variable à la fois avec l'incrément d :

$$da_i = n\frac{\partial L_D}{a_i} = n\left(1 - y_i \sum_{j=1}^{l} a_j y_j k(x_i \cdot x_j)\right)$$
$$a_i^{t+1} = a_i^t + da_i$$
$$si\ a_i < 0, \quad a_i = 0$$
$$si\ a_i > C, \quad a_i = C$$

n est le taux d'adaptation. Cette stratégie n'est pas optimale en temps d'exécution mais elle est simple. En plus, elle évite le stockage en mémoire RAM de la matrice carrée $K(x_i, x_j)$. Les critères d'arrêt des itérations sont : la surveillance des conditions KKT et le taux d'incrément de la fonction L_D.

Seconde solution : optimisation séquentielle minimale (SMO)

L'idée principale des algorithmes de décomposition est de travailler avec un sous-ensemble réduit de données du problème, garder les solutions et continuer avec le reste des données, où les solutions antérieures doivent être encore testées.

La SMO prend cette idée à l'extrême : elle optimise seulement deux vecteurs par itération. Cette optimisation admet une solution analytique[7]. A chaque itération, la SMO choisit deux coefficients a_i et a_j pour les optimiser ensemble, trouver ses valeurs optimales étant donné que toutes les autres sont fixes, et actualiser le vecteur solution a.

La solution pour a_1 et a_2 est trouvée grâce à la décomposition de L_D :

$$L_D = a_1 + a_2 - \frac{1}{2}K(x_1 \cdot x_1)a_1^2 - \frac{1}{2}K(x_2 \cdot x_2)a_2^2 - y_1 y_2 K(x_1 \cdot x_2)a_1 a_2 - y_1 a_1 v_1 - y_2 a_2 v_2 + cte$$
$$v_i = \sum_{j=3}^{l} y_j a_j K(x_i \cdot x_j)$$

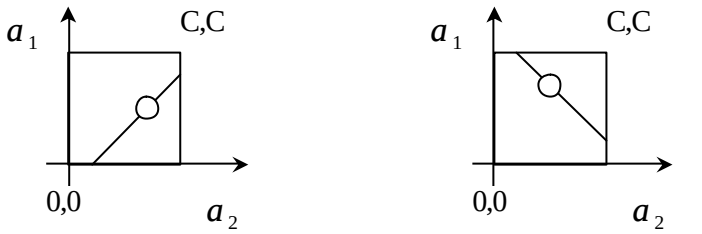
En respectant les contraintes $0 \le a_1, a_2 \le C$ et $\sum_{i=1}^{l} a_i y_i = 0$ (figure 28), on dérive L_D par rapport à a_2 et on obtient des expressions de cette variable en fonction de l'erreur de classification.

[7] J. Platt, "Fast training of Support Vector Machines using Sequential Minimal Optimization" [Schölkopf99].

$$a_2^{new} = a_2^{old} - \frac{y_2(E_1 - E_2)}{k}$$
$$a_1^{new} = a_1^{new} + y_1 y_2 (a_2^{old} - a_2^{new})$$

$$où \quad k = K(x_1 \cdot x_1) + K(x_2 \cdot x_2) - 2K(x_1 \cdot x_2)$$
$$et \quad E_1 \; et \; E_2 :$$
$$E_i = f(x_i) - y_i = \left(\sum_{j=1}^{l} a_j y_j K(x_j \cdot x_i) + b \right) - y_i$$

La SMO optimise deux coefficients à chaque itération. Un des deux doit violer les conditions KKT pour être choisi dans l'itération courante.



$y_1 \neq y_2 \Rightarrow a_1 - a_2 = g$ $\qquad$ $y_1 = y_2 \Rightarrow a_1 + a_2 = g$

Figure 28: Les deux multiplicateurs de Lagrange choisis doivent satisfaire les contraintes du problème.

Le critère d'arrêt de la SMO est aussi la surveillance des conditions KKT. La comparaison entre les deux formulations du Lagrangien (L et L_D) peut être utilisée pour assurer la convergence : la différence entre les deux est de (presque) zéro dans la solution.

Reconnaissance

Après apprentissage, la fonction de décision s'écrit :

$$f(u) = sign\left(\sum_{i=1}^{m} a_i y_i K(u, x_i) + b \right)$$

où m est le nombre de vecteurs de support. Les données sont présentées au modèle *f(u)* et par conséquence, chaque vecteur reçoit une étiquette de classification {(sign=+1 -> classe), (sign=-1 -> non classe)}.

Chapitre 3 Réalisations et expériences

Dans le chapitre précédent, nous avons détaillé deux modélisations pour réaliser la classification de deux types de sons : rires et applaudissements. Nous proposons maintenant une validation de ces modélisations à travers un ensemble d'expériences réalisées dans le cadre du projet RIAM FERIA (Framework pour l'Expérimentation et la Réalisation Industrielle d'Applications Multimédias). Le corpus a été défini par l'Institut National de l'Audiovisuel, il est constitué d'émissions de type plateau où ces deux événements sonores apparaissent en grand nombre. Après avoir détaillé le corpus, des séries d'expériences sur chacune des modélisations sont présentées.

3.1 Corpus

Le corpus est composé de deux documents télévisuels qui correspondent à deux émissions "Le grand échiquier" dont le contenu est de la musique (classique, jazz, variété française), des interviews et des sketches. Chaque émission a une durée de 180 minutes environ.

Le canal audio est séparé du canal vidéo et mis au format suivant : 16 bits, échantillonnage à 16 kHz, canal monophonique, fichier type WAV et RAW (brut).

Les applaudissements et les rires sont présents tout au long des émissions. Pour chaque type nous distinguons les sons purs des « pseudo-sons » (figure 29). Ces derniers sont des applaudissements ou rires, bruités ou mélangés avec d'autres sons de l'environnement, de la parole ou de la musique.

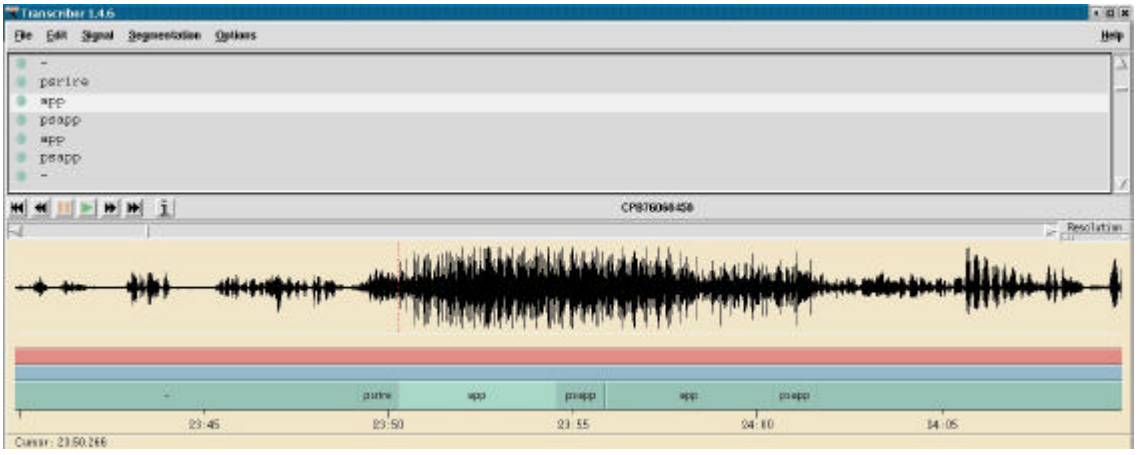


Figure 29: Relation entre les étiquettes pseudo-applaudissement (psapp) et applaudissement (app). Seuls, les signaux «propres » sont étiquettés app. Une relation similaire est établie entre les rires et les pseudo-rires.

Les séquences d'applaudissements sont en général claires et bien définies, avec des durées comprises entre 5 et 8 s. Par contre, les segments de rire sont plus variables,

ils peuvent durer de 0,5 s à 4 s. Les signaux de rires les plus réguliers sont les rires du public, mais ce type de signal est difficile à distinguer d'un bruit.

Le tableau 1 résume l'étiquetage manuel des émissions. Les segments « pseudo » ne sont pas pris en compte pendant la phase d'apprentissage. Néanmoins, pour la reconnaissance, les parties étiquetées « pseudo » sont supposées être reconnues.

Fichier	Applaudissements		Pseudo applaudissements		Total	
	Segments	Durée	Segments	Durée	Segments	Durée
CPB81052332	47	243s	74	228s	121	471s
CPB76068458	72	638s	72	268s	144	906s

Fichier	Rires		Pseudo rires		Total	
	Segments	Durée	Segments	Durée	Segments	Durée
CPB81052332	33	57s	15	23s	48	80s
CPB76068458	175	354s	184	298s	359	652s

Tableau 1 : Analyse de l'indexation manuelle des émissions d'apprentissage et de test.

3.2 Système MMG

Ce système est considéré comme « référence » car plus classique. Ses résultats seront comparés à ceux du système SVM. Il comporte divers modules :

Apprentissage

- Etiquetage manuel des données d'apprentissage
- Création des paramètres
- Création des modèles de référence MMG pour chaque classe

Reconnaissance

- Création des paramètres
- Comparaison entre les vecteurs de test et les modèles par maximum de vraisemblance
- Assemblage
- Lissage

3.2.1 Les paramètres

En cherchant la paramétrisation la plus appropriée pour la création des modèles d'apprentissage et pour la reconnaissance des applaudissements et des rires dans les émissions, le signal audio a été prétraité et soumis à diverses expérimentations. Les

premiers tests ont été faits pour choisir entre les analyses cepstrale (MFCC) et spectrale (SPL), et trouver la taille de la trame la plus adaptée pour calculer ces paramètres.

- Analyse MFCC et SPL. L'analyse spectrale consiste en 29 paramètres par trame (l'énergie + 28 coefficients spectraux), l'analyse cepstrale en considère 18 (l'énergie + 8 coefficients cepstraux + ses dérivées) [Pinquier02]. Pendant certains tests, on a augmenté le nombre de coefficients spectraux en ajoutant les dérivées.
- Variations de la taille de la trame. On a utilisé des trames de 128, 256, 512 et 1024 échantillons. Le recouvrement utilisé a toujours été la moitié de la taille de la trame.

3.2.2 L'apprentissage

L'information obtenue par l'étiquetage manuel est utilisée pour découper l'émission d'apprentissage. Les trames sont regroupées en deux ensembles : Classe et Non classe. Par exemple, tous les vecteurs qui sont étiquetés comme « rire » constituent l'ensemble d'apprentissage de la classe rire. Le reste de l'émission donne l'ensemble d'apprentissage de la classe « non rire » (les données «pseudo » sont ignorées à cette étape).

A partir de chaque ensemble est calculé un mélange de lois gaussiennes. Sont ainsi obtenus les modèles de référence de chaque classe (rire, non rire, app, non app) pour la phase de reconnaissance. La variable à contrôler dans cette phase est le nombre de lois gaussiennes dans le mélange, qui dépend directement de la quantité et de la distribution statistique des données. On a utilisé les valeurs K = 32, 64 et 128.

Dans la plupart des systèmes, les lois normales sont définies avec une matrice de covariance diagonale (supposant l'indépendance statistique entre les coordonnées des vecteurs). Néanmoins, dans un certain nombre de tests nous avons considéré l'utilisation de matrices de covariances pleines.

3.2.3 La reconnaissance

Les données de test après paramétrisation sont confrontées aux modèles de référence. Un calcul de maximum de vraisemblance entre les vecteurs de test par rapport aux modèles permet l'affectation à chaque trame d'une étiquette de la classe à laquelle elle appartient. Les résultats de cette indexation subissent ensuite un lissage, c'est à dire, une fusion des étiquettes dans un intervalle pour trouver des segments significatifs. Les intervalles utilisés varient de 0,5 s à 1 s.

3.2.4 L'évaluation

Le critère d'évaluation des résultats se base sur le rapport (exprimé en %) entre le temps correctement segmenté d'une émission et sa durée totale. Un outil fournit par le NIST (National Institute of Standards and Technology) utilisé pour l'évaluation de campagne sur les systèmes de segmentation a été employé.

On a considéré l'émission CPB81052332 comme celle d'apprentissage et l'émission CPB76068458 comme celle de test. Les deux ont été étiquetées manuellement pour extraire les données d'apprentissage et pour faire les évaluations des résultats.

Indexation Applaudissements / Non applaudissements

Test	256 MFCC 32G	256 MFCC 64G	128 SPL 64G	256 SPL 32G	256 SPL 64G	512 SPL 64G	1024 SPL 64G	1024 SPLD 64G	1024 SPL 1GP	1024 SPL 4GP
Score (%)	91.7	85.3	93.7	93.9	94.2	97	98.58	98.49	98.47	98.6

Tableau 2 : Résultats de l'indexation MMG pour les applaudissements.

Par rapport à la paramétrisation, les résultats montrent une meilleure performance de l'analyse SPL sur les MFCC. Une autre variante est explorée : celle d'incrémenter la taille du vecteur SPL avec les valeurs des dérivées de ses coefficients (tests SPLD).

Pour la taille de la trame utilisée, les meilleures évaluations sont obtenues avec les fenêtres de 512 (32 ms) et 1024 (64 ms) échantillons (tableau 2).

Les matrices de covariances utilisées pour les MMG sont diagonales, sauf dans les tests marqués comme GP.

Avec la configuration, trame 1024 échantillons, analyse SPL, 64 lois gaussiennes, lissage 1 s, on arrive à d'excellentes performances pour la reconnaissance des applaudissements. Tous les événements importants ont été repérés (figure 31).

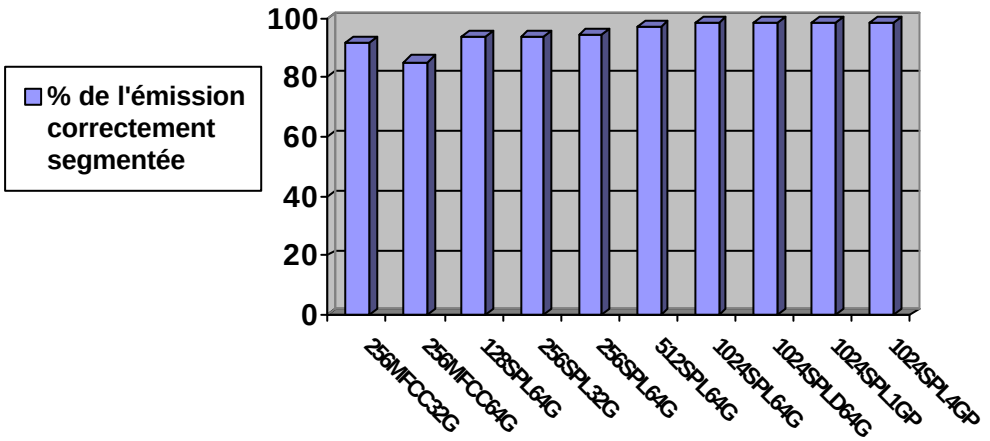


Figure 30: Synthèse des expériences significatives d'indexation d'applaudissements avec le système MMG. Le code de test est : FPG, F = taille de la fenêtre d'analyse, P = type de paramétrisation et G = Nombre de lois gaussiennes.

Temps total étiqueté : 906 s – 144 segments (dont 72 significatifs).
Evaluation NIST : 98.58 %.
Temps total récupéré : 771 s – 97 segments.

On dit qu'un segment est significatif en référence à son étiquetage manuel, quand il est considéré comme un son « pur ».

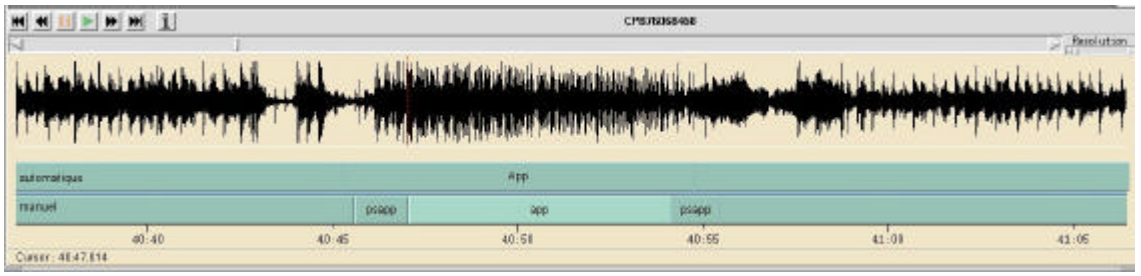


Figure 31: Résultats de l'indexation app / non app. Un événement « applaudissement » a été détecté. Les étiquettes de haut sont celles données par la sortie du système, celles en bas sont le résultat de l'étiquetage manuel. Les frontières du segment reconnu montrent que les signaux « pseudo » ont été repérés.

Toutes les séquences longues (5 - 6 s) d'applaudissements ont été identifiées. Seuls certains signaux pseudo et segments courts (< 1 s) n'ont pas été reconnus.

Indexation Rire / Non rire

Le signal rire est difficile à modéliser dû à sa variabilité importante. A cause de cela, plusieurs erreurs sont arrivées pendant l'indexation automatique : les signaux de rires d'amplitude faible sont supprimés, des séquences de parole et musique sont considérées comme des rires, les événements « pseudo » sont partiellement récupérés. Il y a aussi des problèmes importants de décalage des frontières (figure 33).

Test	256 MFCC 32G5	256 MFCC 64G5	128 SPL 64G5	256 SPL 32G5	256 SPL 64G	512 SPL 64G	1024 SPL 64G	1024 SPLD 64G	1024 SPL 1GP	1024 SPL 4GP	1024 SPL 128G	FUS MUS APP
NIST (%)	74.3	74.5	82.7	74.4	89	91	95.98	95.75	56.47	14.4	97.26	97.45

Tableau 3 : Résultats de l'indexation MMG pour les rires.

Dans le but d'améliorer l'évaluation de l'indexation rire/non rire, une fusion des résultats avec un système d'indexation musique/non musique [Pinquier02] est réalisée (test FUS MUS APP). La fusion analyse les intervalles déclarés musique pour enlever les « fausses » étiquettes de rire insérées dans ces intervalles. De cette manière, l'évaluation de l'indexation rire/non rire passe de 97.26 à 97.45%.

Avec la configuration, trame 1024 échantillons, analyse SPL, 128 lois gaussiennes, lissage 1 s (figure 33), le nombre de suppressions d'événements importants n'est pas trop élevé, néanmoins, une certaine quantité d'insertions pendant les segments de musique et parole dégrade les résultats de l'évaluation (figure 34).

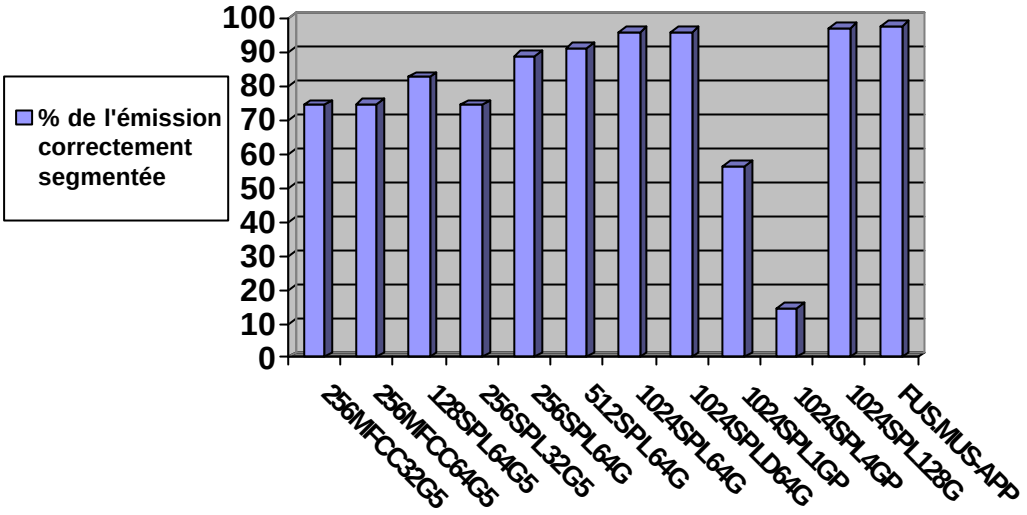


Figure 32: Synthèse des expériences significatives d'indexation de rires avec le système MMG. Divers variables ont été modifiées.

Temps total étiqueté : 652 s – 359 segments (dont 175 significatifs).
Evaluation NIST : 97. 26 %.
Temps total récupéré : 212 s – 102 segments.

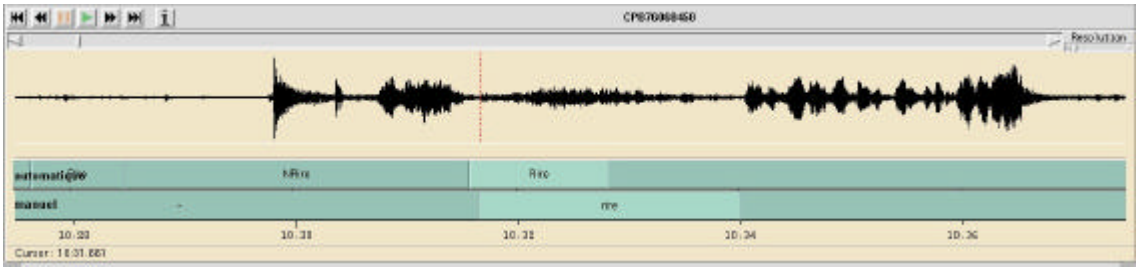


Figure 33: Résultats de l'indexation rire / non rire. Un événement rire est identifié dans l'émission. On remarque un décalage important vers la fin du segment.

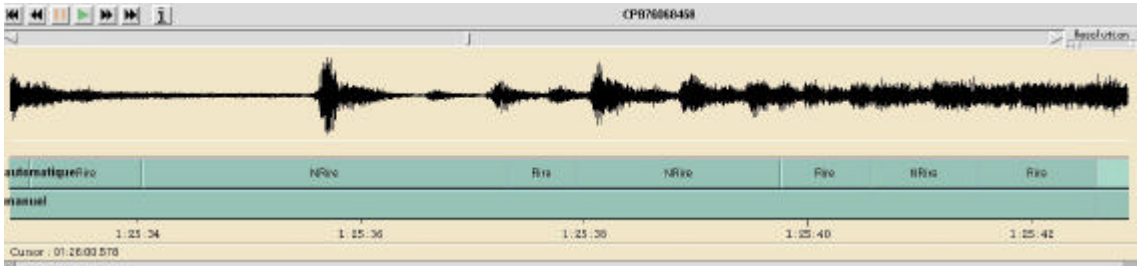


Figure 34: Résultats de l'indexation rire / non rire. Plusieurs insertions se sont produites pendant les segments de musique.

3.3 Système SVM

Ce système consiste des modules :

Apprentissage

- Etiquetage manuel des données d'apprentissage
- Création des paramètres
- Calcul des vecteurs support qui définissent l'hyperplan séparateur de classes

Reconnaissance

- Création des paramètres
- Calcul de la fonction de décision pour chaque vecteur de test
- Assemblage
- Lissage

3.3.1 Les paramètres

Pour être capable de comparer les systèmes de classification SVM et MMG, on a utilisé la paramétrisation SPL avec les trames de 512 et 1024 échantillons. L'émission CPB81052332 est utilisée aussi comme information d'apprentissage et la CPB76068458 comme celle de test.

3.3.2 L'apprentissage

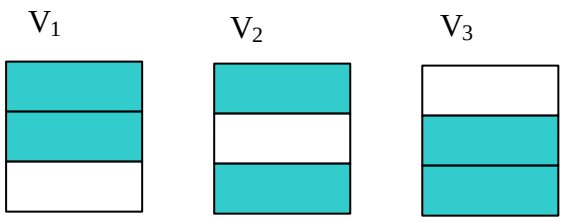
A la différence du cas MMG, le système SVM ne sépare pas les données Classe et Non classe dans l'étape d'apprentissage. Les deux informations sont nécessaires en même temps pour la création des modèles (dans ce cas, les poids de l'hyperplan séparateur).

La variable à contrôler dans cette phase est la fonction noyau utilisée pour modéliser la frontière qui sépare les deux classes de données. Trois fonctions ont été essayées : polynômes, gaussiennes et sigmoïdes. Chaque fonction utilise un nombre de

variables choisi par l'utilisateur. Ces variables définissent la capacité de généralisation de la SVM.

Etant donné le degré de liberté dans le choix des paramètres, la méthode de validation croisée est utilisée pour trouver les valeurs les plus adaptées pour le modèle (figure 35). Les données d'apprentissage sont alors divisées en *n* pièces. On teste le morceau n_i par rapport à l'apprentissage issu des segments n_j i<>j. Un premier passage permet de déterminer un ensemble de valeurs V qui sont affinées dans un deuxième tour.

Les expériences pratiques avec les SVM montrent qu'il y a deux procédures qui peuvent améliorer la création de modèles d'apprentissage : (1) la mise en échelle et (2) la balance de données. Dans (1) on normalise les vecteurs avec une échelle calculé par rapport aux valeurs maximales de chaque coordonnée, et (2) est importante quand la quantité de données classe/non classe est très asymétrique.



$$V = \{d, C, g\}$$

$$V_k = \frac{1}{n}\sum_{i=1}^{n} v_i$$

Figure 35: Procédure de validation croisée.

3.3.3 La reconnaissance

Chaque vecteur de test est soumis à la fonction de décision du modèle d'apprentissage. Cette fonction donne comme résultat l'étiquette de la classe à laquelle le vecteur appartient. Les résultats passent par le même lissage que dans le cas MMG pour trouver des segments significatifs. Les intervalles utilisés varient de 0,5 s à 1 s.

3.3.4 L'évaluation

Indexation Applaudissements / Non applaudissements

La fonction noyau qui donne les meilleurs résultats est la gaussienne. Elle a besoin de deux paramètres : C (pénalisation aux erreurs) et *s* (amplitude de la fonction). Le noyau polynomial pose de sévères problèmes de temps d'exécution quand sa dimension est d'ordre 4 ou 5 (test KPOL).

Test	512 SPL C2 G.8B	512 SPL C1000 G.5B	1024 SPL C1000 G.5B	1024 SPL C1000 G.5	1024 SPL C1000 G.51C	1024 SPL KPOL	1024 SPL KSIG
Score (%)	97.84	98.35	98.3	93.47	93.38	87.41	52.7

Tableau 4 : Résultats de l'indexation SVM pour les applaudissements.

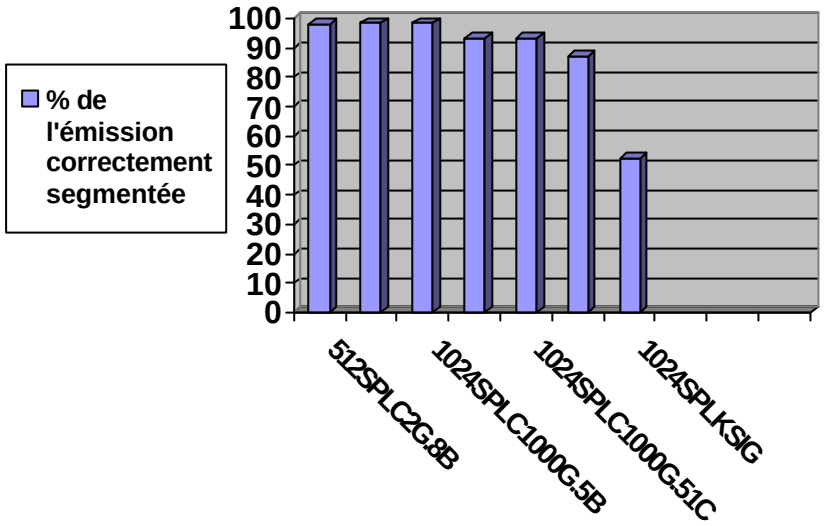


Figure 36: Synthèse des expériences significatives d'indexation d'applaudissements avec le système SVM. Diverses variables ont été modifiées et la procédure de validation croisée a été employée. Le code de test est : FPC, F = taille de la fenêtre d'analyse, P = type de paramétrisation et C = Variables du noyau.

Avec la configuration, trame 512 échantillons, analyse SPL, kernel rbf (C=1000 g=0,5), lissage 1s (figure 36), on a une très bonne performance de reconnaissance des applaudissements. Tous les événements importants ont été repérés.

Temps total étiqueté : 906 s – 144 segments (dont 72 significatifs).
Evaluation NIST : 98.35 %.
Temps total récupéré : 760 s – 96 segments.

Indexation Rire / Non rire

Le déroulement de l'indexation rire a demandé, d'abord, d'équilibrer le nombre des données d'apprentissage des classes rire et non rire car une grande asymétrie dans la quantité de données affecte les résultats du processus d'indexation. Ensuite, grâce à la procédure de validation croisée, on remarque qu'une diminution de la quantité de données améliore les scores NIST (tests M3).

Test	512 SPL C2 G.5M3	512 SPL C2 G.5B	1024 SPL C2 G.5B	1024 SPL C2 G.5M3L.5	1024 SPL C2 G.51C
Score (%)	94	27	86.63	97.12	96.98

Tableau 5 : Résultats de l'indexation SVM pour les rires.

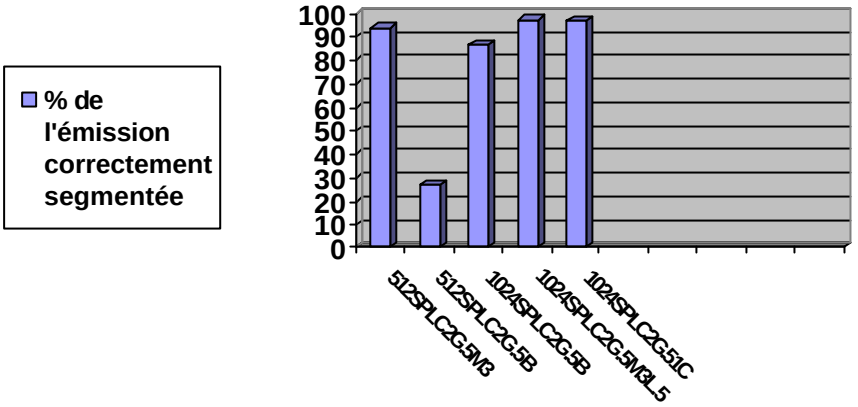


Figure 37: Synthèse des expériences significatives d'indexation de rires avec le système SVM. Diverses variables ont été modifiées.

Avec la configuration, trame 1024 échantillons, analyse SPL, kernel rbf (C=2 g=0.5), lissage 0.5 s (figure 37), malgré un nombre important de suppressions, on retrouve les segments de rire les plus intéressants dans l'émission. Pour les segments retrouvés, leurs frontières automatiques sont un peu courtes par rapport à celles déterminées manuellement (figure 38). Il n'y a presque aucune insertion dans les segments de musique.

Temps total étiqueté: 652 s – 359 segments (dont 175 significatifs).
Evaluation NIST: 97.12 %.
Temps total récupéré: 112.6 s – 106 segments.

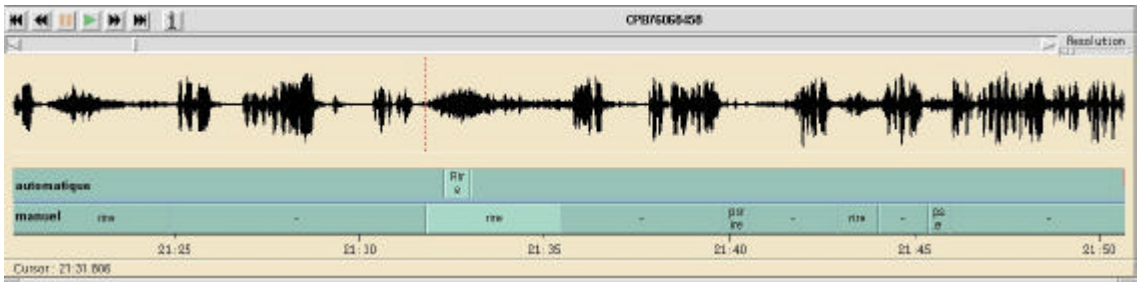


Figure 38: Résultats de l'indexation rire / non rire avec le système SVM. Sans un nombre important d'insertions, les problèmes de la indexation automatique sont les suppressions et les décalages des segments aux frontières.

3.4 Conclusion

Le tableau 6 montre un résumé des meilleurs résultats d'indexation. Les deux systèmes donnent des résultats pratiquement équivalents, mais il y a certaines différences à souligner :

1. L'évaluation de l'indexation de rires avec le système MMG est fortement affectée par les insertions de rires dans les segments de musique. Si apparemment on récupère plus du temps de signal «rire » que dans le cas SVM (tableau 7), la précision de l'indexation des deux systèmes est pourtant équivalente.
2. Le nombre de vecteurs nécessaires pour modéliser les classes est très différent : le système SVM a besoin de beaucoup moins de données d'apprentissage que le système MMG (tableau 8).

Système	Applaudissements	Rires
MMG	98.58%	97.26%
SVM	98.35%	97.12%

Tableau 6 : Evaluations NIST des meilleurs cas.

Système	Applaudissements		Rires	
Indexation	Manuel	Automatique	Manuel	Automatique
MMG	906s	771s	652s	212s
SVM	906s	760s	652s	112s

Tableau 7 : Temps récupéré par les indexations manuel et automatique de l'émission de test (CPB76068458).

Système	Applaudissements	Rires
MMG	App 7512 Non-App 337 160	Rire 1714 Non-Rire 342 874
SVM	App 7512 Non-App 20 000	Rire 283 Non-Rire 1000

Tableau 8 : Nombre des vecteurs nécessaires pour faire l'apprentissage des modèles.

Chapitre 4 Conclusions et perspectives

Ce stage m'a permis d'étudier deux méthodologies de classification : l'approche de décision bayesienne et la toute nouvelle théorie des machines à vecteurs de support.

Au même temps, une initiation aux problèmes pratiques d'indexation audio a été accomplie : comment définir un corpus d'apprentissage et de test, les procédures d'étiquetage manuel, quelle choix faire parmi les différentes alternatives de paramétrisation, la théorie des systèmes de classification, la gestion de grandes quantités d'information, les outils d'analyse de résultats…

Grâce à sa mise en œuvre robuste, sa vulgarisation et sa performance, les systèmes issus de mélanges de lois gaussiennes sont une très bonne alternative comme plateforme pour le développent de systèmes d'indexation audio. Les expériences d'identification de sons caractéristiques dans des émissions télévisuelles ont donné de bons résultats. Le seul inconvénient de cette approche est la quantité de données nécessaires pour l'apprentissage des modèles.

Issus d'un fort et élégant bagage théorique, les méthodes à vecteurs de support commencent à être plus utilisées. Pendant la réalisation des expériences on a pu constater son efficacité : on arrive presque aux même résultats que dans la classification bayesienne avec beaucoup moins de données. Diverses subtilités de sa mis en œuvre ont été découvertes : d'abord sa complexité, la validation croisée pour la définition de paramètres, le comportement de différentes fonctions noyau, entre autres.

Néanmoins, l'expérience que j'ai appréciée le plus lors de mon travail avec l'équipe SAMOVA, a été le développement de l'intuition pour chercher des alternatives d'expérimentation quand les résultats ne sont pas ceux que l'on attend.

La segmentation est la base de systèmes plus complexes d'analyse de contenu sémantique. L'intégration avec d'autres sources d'information pour construire des descriptions formelles plus précises de documents multimédia est envisagée. Modélisation de documents multimédia, recherche d'information, reconnaissance du locuteur, sont de futures possibilités de poursuite de ce travail.

Références

[Ash93] C. Ash « The Probability Tutoring Book », Ed. IEEE Press, 1993.

[Bennet98] K. P. Bennet, E. Bredensteiner « Geometry in learning », Geometry at work, 1998.

[Burges98] C. Burges « A Tutorial on Support Vector Machines for Pattern Recognition », Data Mining and Knowledge Discovery 2, pp. 121-167, 1998.

[Chang04] C-C Chang, C-J Lin « LIBSVM: a Library for Support Vector Machines », http://www.csie.ntu.edu.tw/~cjlin, National Taiwan University, 2004.

[Cortes95] C. Cortes, V. Vapnik « Support Vector Networks », Machine Learning, vol. 20, pp. 273-297, 1995.

[Cristianini00] N. Cristianini, J. Shawe-Taylor « An Introduction to Support Vector Machines », Ed. Cambridge University Press, 2000.

[Davy02] M. Davy, S. J. Godsill « Audio information retrieval: A bibliographical study », CUED/F-INFENG/TR429, Cambridge University Engineering Department, Février 2002.

[Duda01] R. O. Duda, P. E. Hart, D. G. Stork « Pattern Classification », Ed. John Wiley & Sons, 2001.

[Guo03] G. Guo, S. Z. Li « Content-Based Audio Classification and Retrieval by Support Vector Machines », IEEE Trans. on Neural Networks, vol. 14, n°1, Janvier 2003.

[Gunn98] S. Gunn « Support Vector Machines for Classification and Regression », ISIS Technical Report, Mai 1998.

[Kao03] W-C Kao, K-M Chung, C-L Sun, C-J Lin « Decomposition Methods for Linear Support Vector Machines », Manuscript Number: 2752, National Taiwan University, 2003.

[Lu02] L. Lu, H-J Zhang, H. Jiang « Content Analysis for Audio Classification and Segmentation », IEEE Trans. on Speech and Audio Processing, vol. 10, n°7, Octobre 2002.

[Mariani02] J. Mariani « Analyse, synthèse et codage de la parole », Ed. Hermes, 2002.

[Pinquier02] J. Pinquier, C. Sénac, R. André-Obrecht « Indexation de la bande sonore: recherche des composantes Parole et Musique », RFIA'2002, Angers, pp. 163-170, Janvier 2002.

[Saunders96] J. Saunders, « Real-time discrimination of broadcast speech/Music », ICASSP'96, pp. 993-996.

[Schölkopf02] B. Schölkopf, A. Smola « Learning with Kernels », Ed. The MIT Press, 2002.

[Schölkopf99] B. Schölkopf, C. Burges, A. Smola « Advances in Kernel Methods: Support Vector Learning », Ed. The MIT Press, 1999.

[Scheirer97] E. Scheirer, M. Slaney, (1997), « Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator », ICASSP'97, Munich, Vol. II, pp. 1331-1334.

[Rabiner93] L. Rabiner, B-H Juang « Fundamentals of Speech Recognition », Ed. Prentice Hall, 1993.

[Tomasi04] C. Tomasi « Estimating Gaussian Mixture Densities with EM – A Tutorial », http://citeseer.nj.nec.com.

[Vapnik99] V. Vapnik « The Nature of Statistical Learning Theory », Ed. Springer, 1999.

[Wang00] Y. Wang, Z. Liu, J-C Huang « Multimedia Content Analysis », IEEE Signal Processing Magazine, Vol. 17, pp. 12-36, Novembre 2000.