Université Pierre & Marie Curie, Paris VI
Équipe Applications Temps-Réel - IRCAM

MÉMOIRE DE STAGE DE MASTER 2 ATIAM

Directeurs de stage : Arshia CONT et Diemo SCHWARZ

SÉLECTION DE DESCRIPTEURS POUR MODÈLES D'OBSERVATION AUDIO TEMPS-RÉEL

Olivier RICORDEAU
<olivier@ricordeau.org>

Paris, Le 22 Juin 2006

Résumé

L'interaction musicale entre un interprète humain et une machine suppose dans un premier temps que la machine ait la capacité d'"écouter" le jeu de l'interprète. Pour atteindre ce but, les modèles d'observation audio temps-réel tels que celui du système de suivi de partition de l'Ircam se basent sur des descripteurs audio et un algorithme de classification pour segmenter le signal pendant le jeu de l'interprète. Les descripteurs audio existants sont nombreux et leur caractère pertinent dépend du problème de classification considéré, ce qui pose le problème de la sélection des descripteurs : lesquels faut-il utiliser pour un problème de segmentation donné ? L'utilisation dans le contexte des applications temps-réel des algorithmes de sélection de descripteurs usuels est problématique car ceux-ci ne permettent pas de maîtriser la quantité de calcul nécessaire à l'obtention des valeurs des descripteurs sélectionnés.

Après avoir présenté les algorithmes de sélection de descripteurs usuels et mis en exergue les problèmes suscités par leur utilisation dans le cadre d'applications où la quantité de calcul est un paramètre critique, nous proposons deux approches permettant d'intégrer le modèle calculatoire des descripteurs au processus de sélection. La première est naïve et repose sur l'utilisation des méthodes *filter*. La seconde se base sur une taxonomie de recherche dérivée du modèle calculatoire des descripteurs et sur un parcours de l'espace de recherche.

Enfin, nous donnons les résultats de l'évaluation des deux approches proposées à laquelle nous nous sommes livrés. Ceux-ci confirment d'une part le bien-fondé de nos constatations initiales concernant les algorithmes usuels de sélection de descripteurs, et d'autre part mettent en évidence l'intérêt de la deuxième approche proposée.

Abstract

Musical interaction between a human and a synthetic performer firstly relies on the machine's ability to "listen" to the sounds made by the human performer. In order to reach this goal, real-time audio observation models like the one in Ircam's score follower use audio descriptors and a classification algorithm to segment the signal as the human performer plays. There are many audio descriptors and there relevance depends on the considered classification problem, which leads to the problem of feature selection : which ones should be used for a given segmentation problem ? Usual feature selection algorithms are problematic regarding real-time applications since they don't let the user master the amount of calculation needed to compute the selected features' values.

In this document, we present classical feature selection algorithms, and stress the issues related to their use in the context of real-time applications. Then, we propose two approaches meant to integrate the features' computational model to the selection process. The first method is naive and relies on the use of *filter* methods. The second is based on a search taxonomy derived from the features' computational model and on an exploration of the search space.

Finally, we give the results of our evaluation of the two proposed approaches. They confirm the validity of our initial observations regarding classical feature selection techniques, and tend to show the usefulness of the second suggested approach.

Remerciements

Je tiens en premier lieu à remercier ceux qui m'ont encadré pendant ce stage, m'ont accueilli dans leur équipe et m'ont aidé dans ma découverte du suivi de partition : Arshia Cont et Diemo Schwarz. J'ai grandement apprécié mon sujet de stage, qui a en plus le mérite d'être au coeur des problématiques actuelles en matière d'apprentissage.

Mes remerciements vont également à Gérard Assayag, Cyrille Defaye, et à l'équipe pédagogique qui permet l'existence de ce Master pas comme les autres qu'est ATIAM. Je les remercie de m'avoir permis de faire partie du petit nombre d'étudiants qui ont la chance d'aborder des problématiques aussi diverses, originales, et . . . artistiques !

À Nicolas Rasamimanana, pour sa sympathie et sa clairevoyance dans les moments critiques.

À Geoffroy Peeters, pour ses conseils précieux et avisés.

À mes camarades d'ATIAM, que je ne suis pas prêt d'oublier.

À mon frère Marc

À mes parents

Table des matières

Introduction

L'interaction musicale entre un interprète humain et une machine suppose dans un premier temps que la machine ait la capacité d'"écouter" le jeu de l'interprète. Pour atteindre ce but, les modèles d'observation audio temps-réel tels que celui du système de suivi de partition de l'Ircam se basent sur des descripteurs audio et un algorithme de classification pour segmenter le signal pendant le jeu de l'interprète. Les descripteurs audio existants sont nombreux et leur caractère pertinent dépend du problème de classification considéré.

Ceci pose le problème de la sélection des descripteurs : lesquels faut-il utiliser pour un problème de segmentation donné ? Au cours du stage, nous avons réalisé les problèmes liés à l'utilisation dans le contexte du temps-réel des algorithmes de sélection de descripteurs usuels. Aussi avons nous pris le parti d'essayer d'apporter des éléments allant dans le sens de l'adaptation des méthodes de sélection existantes aux applications où le temps de calcul est un paramètre critique.

Le premier chapître situe le contexte dans lequel s'est effectué le stage : la problématique de suivi de partition y est présentée, ainsi que le système de suivi de l'Ircam. Quelques notions concernant les descripteurs audio sont par ailleurs données.

Le deuxième chapitre aborde la sélection de descripteurs et ses enjeux, et offre une vue d'ensemble des différentes solutions algorithmiques connues servant à réduire la dimensionnalité des problèmes de classification. Les algorithmes utilisés pendant nos expérimentations y sont décrits, et les principaux critères permettant d'évaluer les algorithmes sont présentés.

Le troisième chapitre aborde la problématique qui constitue le coeur du travail effectué pendant le stage : la sélection de descripteurs pour les applications où la quantité de calcul est un paramètre critique. Dans cette perspective, nous donnons un modèle naïf du calcul des descripteurs audio. De plus, nous suggérons deux approches de sélection de descripteurs utilisant des algorithmes déjà existants et permettant d'intégrer le modèle calculatoire des descripteurs au processus de sélection.

Le quatrième chapitre décrit les expérimentations effectuées durant le

stage. Celles-ci mettent en exergue les problèmes posés par les algorithmes de sélection de descripteurs connus pour les applications ayant de fortes contraintes concernant le temps de calcul. Les deux approches présentées au chapitre précédent y sont évaluées et comparées.

Enfin, le cinquième et dernier chapitre donne quelques pistes de recherche constituant des suites possibles au travail présenté dans ce mémoire.

CHAPITRE 1

Pré-requis

Sommaire

Dans ce chapitre, nous décrivons la problématique du suivi de partition, puis nous donnons les notions de base nécessaires à la compréhension de notre travail dans son contexte : le suiveur de partition de l'Ircam.

1.1 Suivi de partition

Plusieurs définitions du suivi de partition ont été données au cours des années. Nous nous tiendrons à une définition pratique : le suivi de partition consiste à trouver en temps-réel un *alignement* entre une ou plusieurs sources de signal (correspondant à un ou plusieurs interprètes) et une représentation symbolique de la musique : la partition. Le but est de disposer d'un système capable de spécifier à n'importe quel instant la position des interprètes dans la partition pendant l'exécution de la pièce. Les applications ouvertes sont principalement artistiques et concernent l'interaction entre musiciens et machines.

La réalisation d'un tel système pose de nombreux problèmes théoriques et pratiques. Les problèmes théoriques sont relatifs aux stratégies à adopter pour extraire de l'information du signal et pour effectuer l'alignement avec la partition de la pièce. Les problèmes pratiques sont liés à la possibilité d'utiliser le système sur scène dans le cadre de performances artistiques.

Les premiers travaux connus dans le domaine du suivi de partition sont ceux de Roger Dannenberg [6] et Barry Vercoe [31]. Par un hasard de l'histoire, ces travaux ont été présentés la même année, après avoir été menés

indépendamment l'un de l'autre. Dannenberg définit comme but le fait d'obtenir un programme capable détecter le jeu du soliste, d'effectuer l'alignement avec la partition, et de réaliser un accompagnement en temps-réel. Son approche concerne principalement la problématique de l'alignement, qu'il effectue par programmation dynamique. Vercoe, lui, cherche à créer un "interprète synthétique" capable d'écouter le jeu de l'interprète humain, de jouer, et d'apprendre pour s'améliorer. Les deux approches se basent uniquement sur la fréquence fondamentale instantanée extraite du signal en temps-réel pour détecter les notes jouées par l'interprète.

Suite à ces premières approches, les systèmes ont évolués tant dans les techniques utilisées pour détecter les notes dans le jeu de l'artiste que dans la manière de réaliser l'alignement. Les modèles de Markov cachés [24] se sont révélés être un solution performante au problème d'alignement dans les travaux de Christopher Raphael [25].

1.2 Le suiveur de partition de l'Ircam

Nous allons présenter ici la structure globale du suiveur de partition de l'Ircam. Nous ne rentrerons pas dans les détails, étant donné que beaucoup d'éléments dépassent le cadre du stage. Nous donnerons les aspects qui permettent de comprendre le rôle du modèle d'observation dans le système de suivi automatique. Une présentation plus détaillée est donnée dans [2] et [4].

Le suiveur de partitions de l'Ircam permet à la fois d'effectuer un suivi à partir du signal (ce qui constitue le problème "classique" de suivi tel qu'expliqué précédemment), mais aussi à partir d'évènements reçus au travers du protocole MIDI [27]. Il se base sur l'approche décrite par Raphael, reposant sur l'utilisation d'un modèle de Markov caché. Le modèle est généré à partir de la partition. C'est lui qui incorpore la structure de la partition et qui permet d'effectuer le lien entre le signal et la structure symbolique. Pour chaque note, quatre états sont générés, tel que représenté dans la figure 1.1 (source : [2]). L'état *Attack* correspond au début d'une note, l'état *Sustain* à la partie soutenue, et l'état *Rest* représente un silence suivant la fin de la note. Le lien entre l'état *Sustain* et l'état *Attack* de la note suivante permet de passer directement d'une note à l'autre si il n'y a pas de silence à la fin de la première note.

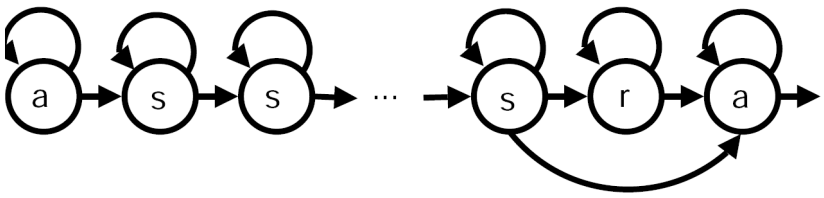

FIG. 1.1 – Modèle de Markov correspondant à une note.

Le modèle de Markov correspondant à la partition est formé en joignant les groupes d'états correspondant à chaque note, tel que représenté figure 1.2.

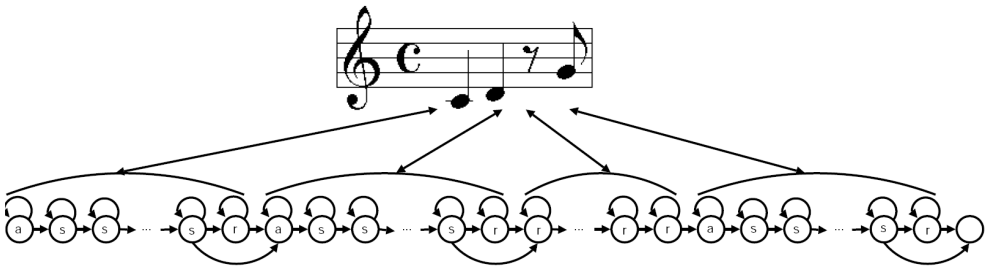

FIG. 1.2 – Génération du HMM à partir de la partition.

Pour effectuer l'alignement en temps-réel, le suiveur de partition utilise une variante causale de l'algorithme Viterbi permettant de trouver en temps-réel l'état du modèle de Markov auquel correspond le signal sonore reçu en entrée. Il est à noter que le système décrit ici est quelque peu différent du système tel qu'il est au moment de l'écriture de ce mémoire, suite à l'ajout récent d'un modèle de Markov hiérarchique [3].

1.3 Modèle d'observation

Comme l'a noté dès le début Vercoe, l'une des problématiques liées à la création d'un interprète synthétique est de concevoir un programme capable d'*écouter* le jeu de l'interprète en temps-réel. Dans le système de suivi de partition de l'Ircam, nous nommons *modèle d'observation* la partie du système qui reçoit en entrée des fenêtres de valeurs du signal et qui en déduit la distribution de probabilité d'appartenance de chaque fenêtre à chacun des trois états utilisés dans la modélisation des notes : *Attack*, *Sustain* et *Rest*. Nous distinguons la problématique de l'observation et celle de l'alignement, et le travail réalisé pendant ce stage porte sur le modèle d'observation.

Nous avons vu précédemment que les premières approches historiques du suivi de partition étaient uniquement basées sur la détection de la fréquence fondamentale instantanée dans le signal. Les avancées des dernières années en matière d'indexation audio ont montré qu'il est possible d'utiliser le paradigme descripteurs/classifieurs habituel en apprentissage statistique dans le but de catégoriser automatiquement des échantillons sonores. Ce paradigme trouve une application naturelle dans un modèle d'observation audio. L'idée est d'utiliser des descripteurs audio compatibles avec l'utilisation en temps-réel et des classifieurs rapides dans le but de *segmenter automatiquement* les notes de l'interprète dans le signal.

La figure 1.3 page suivante montre le rôle du modèle d'observation dans le système. Comme son nom l'indique, il *observe* les fenêtre de signal. Pour chaque fenêtre les valeurs des descripteurs audio utilisés sont calculées, puis sont données en entrée à un classifieur qui calcule une distribution de probabilité par rapport aux trois classes *Attack*, *Sustain* et *Rest*. Enfin, c'est cette distribution de probabilités qui permet d'effectuer l'alignement en utilisant le modèle de Markov.

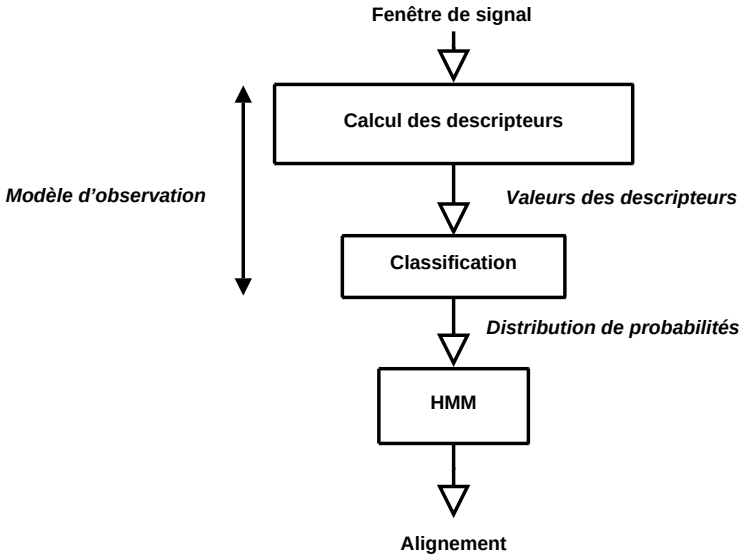

FIG. 1.3 – Rôle du modèle d'observation

L'approche par apprentissage statistique possède un avantage non négligeable pour le domaine du traitement du signal musical : elle permet de construire une connaissance à partir d'une expertise humaine. L'expertise réside dans la base d'échantillons sonores préalablement segmentés manuellement, sur laquelle un modèle algorithmique (le classifieur) est entraîné. Les concepts musicaux sont difficiles à formaliser sous la forme d'équations portant sur le signal. Aussi, dans le cadre d'un modèle d'observation audio, il est intéressant de disposer d'algorithmes capables d'apprendre à partir d'exemples.

1.4 Descripteurs audio

Différents types de descripteurs audio

Le nombre croissant de descripteurs audio a fait naître un besoin de pouvoir catégoriser ceux-ci. Nous retiendrons la taxonomie de G. Peeters [23], qui se base sur les représentations du signal à partir desquelles sont calculés les descripteurs.

Tout d'abord, nous pouvons catégoriser les descripteurs suivant la durée de l'échantillon sonore pour laquelle ils sont valides :

- **Descripteurs globaux -** Ce sont des descripteurs calculés pour l'ensemble du signal. Par exemple, le temps d'attaque (*attack time*), qui est le temps nécessaire pour que le signal atteigne son intensité maximale.
- **Descripteurs instantanés -** Ce sont des descripteurs calculés sur des fenêtres temporelles (généralement recouvrantes).

À l'intérieur de chaque classe de descripteurs, nous pouvons catégoriser les descripteurs selon le type de représentation(s) du signal utilisée(s) pour extraire ceux-ci à partir du signal :

- **Descripteurs temporels (globaux ou instantanés)** Ils sont calculés à partir de la forme de l'onde ou de l'énergie du signal : temps d'attaque, Temporal Decrease, Temporal Centroid, Effective Duration, Zero-crossing rate, Cross-correlation.
- **Descripteurs spectraux (instantanés)** Ils sont calculés à partir de la STFT (Short Time Fourier Transform) du signal : Spectral Centroid, Spread, Skewness, Kurtosis, Slope, Decrease, Roll-off point, variation.
- **Descripteurs harmoniques (instantanés)** Ils sont calculés à partir d'un modèle harmonique du signal : Fréquence fondamentale, Noisiness, Odd-to-Even Harmonic Ratio, Tristimulus, Deviation, Centroid, Spread, Skewness, Kurtosis, Slope, Decrease, Roll-off point, Variation
- **Descripteurs perceptuels (instantanés)** Ils sont calculés en utilisant un modèle perceptif de l'audition humaine : MFCC, DMFCC, DDMFCC, Loudness, Specific Loudness, Sharpness, Spread, Roughness.

Quelques descripteurs classiques pour l'analyse du signal musical

Zero Crossing Rate (ZCR) Ce descripteur est la fréquence de passage par zéro du signal sur la fenêtre étudiée. Il est simple à calculer et constitue un bon discriminant pour beaucoup de problèmes de classification d'échantillons sonores.

Mel Frequency Cepstral Coefficients (MFCC) Le cepstre d'un signal réel $x(t)$ est une transformation de ce signal du domaine temporel vers un autre domaine temporel, celui des quéfrences. Le cepstre est défini comme étant le résultat de la transformée de Fourier inverse appliquée au logarithme de la transformée de Fourier du signal :

$$C(\tau) = C(x(t)) = FT^{-1}(log_{10}(FT(x(t))))$$

Par application du logarithme, la source (*ex :* la corde d'une guitare) et la fonction de transfert associée au conduit (*ex :* l'ensemble de la caisse de la guitare) peuvent être séparés. Ce principe a trouvé sa première application dans le traitement de la parole, et il est très utilisé pour le traitement du signal musical. Les *MFCC*'s sont les coefficients cepstraux, exprimés dans l'échelle (perceptive) de Mel.

Fréquence fondamentale Étant donné l'importance des hauteurs dans les compositions, la fréquence fondamentale est un descripteur important pour l'utilisation d'algorithmes d'apprentissage dans la segmentation de flux musical. Elle est calculée sur des fenêtre se recouvrant, pour lesquelles une fréquence fondamentale instantanée est extraite.

Descripteurs audio et temps-réel

Les applications temps-réel imposent deux *contraintes fortes* sur les systèmes de segmentation basés sur des algorithmes de classification. D'une part, les descripteurs globaux ne sont pas utilisables car par définition, ils

sont calculés à partir de valeurs futures du signal. Des réflexions visant à créer des descripteurs simulant les descripteurs globaux pour les applications temps-réel sont présentées au 5.3 page 42.

D'autre part, le calcul des descripteurs et la classification doivent pouvoir être réalisés en un temps inférieur à la durée séparant le début de deux fenêtres successives. Cette contrainte rend impossible l'utilisation de bon nombre des classifieurs les plus performants de la littérature. Par exemple, les *Support Vector Machines* avec un noyau de dimension plus grande que l'unité sont proscrits.

Le *délai lié à la segmentation* doit être acceptable en fonction de l'application pour laquelle est utilisé le système. Dans le cadre du suivi de partition, ceci dépend de la pièce et donc des choix du compositeur. En pratique, les descripteurs utilisés pour le temps-réel sont non-causaux dans le sens où ils dépendent de valeurs du signal futures par rapport à la date correspondant au milieu de chaque fenêtre. De faibles longueurs de fenêtres permettent d'utiliser ces descripteurs dans les applications temps-réel, mais ceci se fait au prix de l'ajout d'un délai lié au système de segmentation. Il est souhaitable de réduire autant que possible de le temps nécessaire au calcul des descripteurs ainsi que celui qui est consacré à la classification du vecteur de descripteurs obtenu.

Un système de suivi de partition (comme cas particulier d'application temps-réel utilisant la segmentation) induit ses propres contraintes. Celles-ci sont d'une part liées à l'interaction avec l'interprète, et d'autre part aux ingénieurs et techniciens qui manipulent le système dans le cadre d'une performance mêlant interprête humain et interprête synthétique. Ainsi, on peut citer les contraintes suivantes :

- Le *nombre de paramètres* à manipuler doit être aussi réduit que possible. Dans le cadre de la sélection de descripteurs pour le modèle d'observation, ceci est une contrainte importante.
- Le *délai maximum* acceptable est fonction de la pièce pour laquelle le système est utilisé. Ce délai est une borne supérieure sur la durée nécessaire au système pour réagir au son produit par l'interprète. Il peut toutefois être contourné par anticipation (ceci ne relève pas du modèle d'observation mais de la partie qui effectue l'alignement avec la partition en utilisant le modèle de Markov caché).

Enfin, pour de meilleures performances d'alignement pendant l'utilisation sur scène, il est souhaitable d'entraîner le modèle sur des *exemples issus de l'interprétation d'une pièce précise*. Dans ce cas, la sélection des descripteurs doit pouvoir être faite en un temps raisonnable étant donné les contraintes liées à l'utilisation du système de suivi dans un contexte de production.

CHAPITRE 2

Sélection de descripteurs

Sommaire

2.1 Définition

Les données sur lesquelles travaillent les classifieurs sont des coordonnées de points dans un espace à n dimensions. Dans le cadre de la fouille de données audio, chaque point de l'espace correspond à une fenêtre du signal, et chaque coordonnée correspond à la valeur d'un descripteur pour la fenêtre en question. Un algorithme sélection de descripteurs a pour but de *réduire le nombre de dimensions* de l'espace dans lequel sont projetées les données. Ceci revient à projeter les points dans un sous-espace.

Le problème scientifique posé par ce principe est de trouver l'espace d'arrivée le mieux adapté à la résolution du problème. Comme nous le verrons, les différents algorithmes existants se basent sur des critères différents pour sélectionner les descripteurs.

2.2 Intérêts et buts

Les motivations justifiant la diminution du nombre de descripteurs servant à décrire les données sont multiples. Elles peuvent d'une part toucher

à la performance de classification (c'est-à-dire la capacité à classifier correctement des vecteurs de descripteurs), mais aussi à la charge de calcul lors de la phase d'utilisation du classifieur. Enfin, des algorithmes de sélection performants et peu coûteux en temps de calcul sont la clé de la création de descripteurs à partir d'exemples.

Améliorer la performance de classification

L'intuition nous pousse à dire que plus le nombre de dimensions servant à décrire les données est élevé, plus il est facile pour un classifieur d'apprendre à distinguer les classes. Les algorithmes de classification sont conçus pour identifier les descripteurs appropriés à la prise de décision. L'expérience montre malheureusement qu'en pratique, une réduction du nombre de descripteurs améliore souvent les performances.

D'une part, les attributs bruités ont une souvent une influence néfaste sur les performances. Considérons une expérience simple : rajouter à un ensemble d'apprentissage un nouveau descripteur dont la valeur est tirée au hasard pour chaque exemple de la base d'entraînement. Avec des arbres de décision, cet ajout a tendance à faire baisser les taux de classifications correctes de 5 à 10% [32]. Ce problème est d'autant plus présent que l'algorithme de classification utilisé est sensible aux attributs bruités.

D'autre part, des attributs "trop pertinents" peuvent également détériorer les performances des systèmes de classification (ce qui est contre-intuitif). Pourtant, des résultats empiriques mettent ce phénomène en évidence. Pour un problème à deux classes, si l'on ajoute un descripteur qui prédit la classe dans 65% des cas et la classe opposée dans les autres cas, la performance de classification est diminuée de 1 à 5% [32].

Ces constatations mettent clairement en évidence l'intérêt de la sélection de descripteurs dans le but d'améliorer la résolution de la tâche de classification. Les systèmes de classification travaillent maintenant sur des exemples projetés selon de nombreux descripteurs (actuellement jusqu'à quelques centaines dans le cas de le domaine de l'audio, des millions dans celui de la bio-informatique), d'où la nécessité de connaître des combinaisons algorithmiques à l'entrée desquelles nous pouvons greffer de nouveaux descripteurs en pouvant espérer au mieux une amélioration, au pire des résultats identiques. Ce but est atteint au prix de l'ajout d'une phase de sélection des descripteurs précédant l'entraînement du classifieur.

Par ailleurs, le problème du *manque d'exemples* est présent dans beaucoup d'applications pour lesquelles des algorithmes de classification sont utilisés. Il est particulièrement présent dans le problème de la détection de transitoires tant l'annotation manuelle des fichiers audio est laborieuse. Or, l'espace de recherche lié au problème de la sélection de descripteurs possède une cardinalité exponentielle par rapport au nombre de descripteurs (voir 2.3 page suivante). Le manque d'exemples est d'autant plus problématique que le nombre ne dimensions décrivant les exemples est élevé. Ceci expose au *sur-apprentissage* (*overfitting*), c'est-à-dire à un apprentissage à la suite duquel les limites de classes apprises sont tellement calquées sur les exemples de la

base d'entraînement que le système a perdu toute généralisation.

Diminuer l'utilisation des ressources pendant la classification

Pour beaucoup de classifieurs, diminuer le nombre de dimensions selon lesquelles les vecteurs sont décrit permet de diminuer la quantité de calcul nécessaire à la classification des vecteurs.

En plus de la diminution du nombre d'opérations à effectuer à chaque fenêtre de descripteurs, la réduction de dimensionnalité a pour conséquence de diminuer la quantité de mémoire nécessaire au classifieur. Pour la plupart des classifieurs, plus le nombre de dimensions selon lesquelles les exemples sont décrits est faible, moins la quantité de mémoire nécessaire pour classifier un vecteur est élevée.

Dans le domaine de l'interaction musicale en temps-réel, les environnements tels que Max/MSP ou PureData imposent des contraintes fortes sur le temps de calcul des descripteurs à partir du signal et le temps de classification d'un vecteur de descripteurs. Or, ces deux grandeurs dépendent précisément du nombre de descripteurs utilisés. Réduire la dimensionnalité du problème de classification permet donc de rendre possible la prise en compte d'un nombre de descripteurs de la base d'entraînement supérieur à ce que le système est capable de calculer en temps-réel. Ceci pose le problème de la prise en compte du coût de calcul des descripteurs, que nous avons abordé durant ces recherches.

But caché : créer de descripteurs

Les descripteurs audio courament utilisés sont des descripteurs bas-niveaux, dans le sens où ils représentent des concepts dérivés (de façon plus ou moins complexe) du signal. Nous savons déjà qu'ils peuvent avoir un caractère discriminant quand ils sont utilisés pour entraîner des algorithmes de classification. Or, certains évènements que nous cherchons à détecter dans le signal correspondent à des concepts musicaux de haut niveau (*ex :* métrique, rythme, *etc.*). Dans cette perspective, certaines recherches ([29], [30]) ont pris la direction de la *création automatique de descripteurs*. Ce principe a été appliqué récemment au domaine des applications audio [33].

2.3 Formalisation du problème

D'une part, nous considérons une fonction de détection $f : x \mapsto label(x)$, où $x \in \mathbf{R}^n$ est un vecteur de valeurs de descripteurs et $label(x) \in \{0, 1\}$ est le label associé à ce vecteur (0 si il s'agit d'une attaque, 1 sinon). D'autre part, nous disposons d'un ensemble E d'exemples d'apprentissage. Chaque exemple est un couple (d, l), où $d \in \mathbf{R}^n$ est un vecteur de valeurs de descripteurs et $l \in \{0, 1\}$ est le label associé à ce vecteur. Un sous-ensemble de l'espace dans lequel sont décrit les exemples est donné par un vecteur $s \in \{0, 1\}^n$. Notre but est de trouver un vecteur s maximisant la performance de détection de f. Nous aborderons le problème de l'évaluation de cette performance dans la suite de ce texte.

Si les exemples sont décris par n descripteurs, l'*espace de recherche* (c'est-à-dire l'ensemble des sous-ensembles de descripteurs constituables) comporte 2^n sous-ensembles de descripteurs distincts. Ceci rend impossible l'exploration de toutes les combinaisons avec les ordinateurs actuels. Dans la théorie de la complexité, il s'agit d'un *problème d'optimisation NP-difficile* [12]. La figure 2.1 (source : [1]) donne une représentation visuelle de l'espace de recherche.

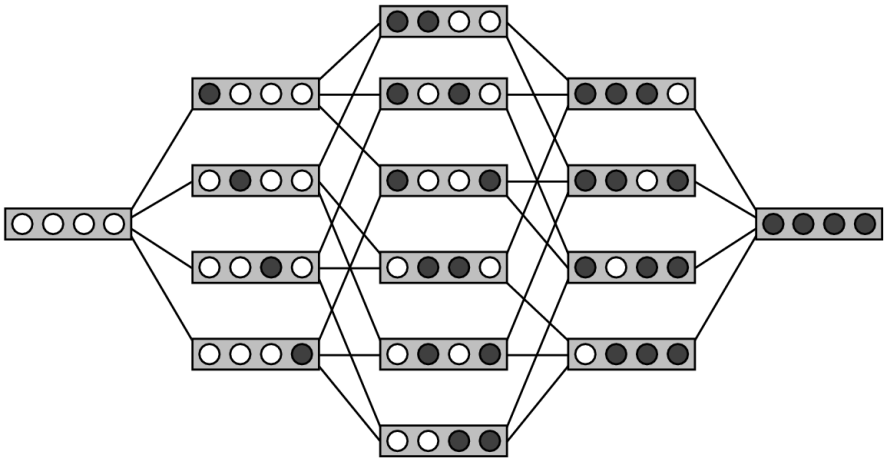

FIG. 2.1 – Espace de recherche : représentation schématique (pour n= 4).

2.4 Vue d'ensemble des différentes approches

La littérature de référence ([12], [1]) en matière de sélection de descripteurs distingue habituellement deux types d'approches algorithmiques :

- Les *"filters"* peuvent être supervisés ou non, et ont pour principe d'*ordonner* les descripteurs en leur attribuant un mérite (ce qui revient à évaluer les descripteurs). Ils posent le problème du nombre de descripteurs à retenir dans leur résultat.
- Les autres méthodes combinent un *algorithme permettant de parcourir l'espace de recherche* et un *algorithme évaluant des sous-ensembles de descripteurs*. Elles offrent un parcours de l'espace de recherche moins naïf que les *filters*.

Les deux approches diffèrent par leur façon d'aborder l'espace de recherche. Pour chacune, il existe différentes stratégie permettant d'attribuer un mérite respectivement aux descripteurs et aux sous-ensembles de descripteurs. Notamment (et c'est un point important), certaines prennent en compte le classifieur utilisé, et d'autres pas. Nous laissons volontairement de côté les algorithmes non supervisés qui constituent des projections.

2.5 Approches Filter

Les approches de types *filter* sont les premières réponses qui ont été données au problème de sélection des descripteurs. Les scores attribués aux descripteurs sont calculés à partir de la base d'entraînement, et les manières de calculer ce score diffèrent selon les algorithmes. Les critères usuels se basent sur des critères issus de la théorie de l'information ou sur la corrélation. D'autres algorithmes calculent les scores en se basant sur des notions de voisinage. Les *filters* ont l'avantage de nécessiter peu de calculs. Cependant ils souffrent de limites théoriques connues, que nous donneront après avoir présenté les algorithmes que nous avons utilisé pour nos expérimentations.

Gain d'information L'une des stratégies permettant d'attribuer des scores aux descripteurs consiste à mesurer le gain d'information de chacun des descripteurs par rapport à la classe. La notion de gain d'information est issue de la *théorie de l'information*, initiée par Claude Shannon à la fin des années quarante. Pour une classe c et un descripteur i, le gain d'information de i par rapport à c est :

$$G(i, c) = H(c) - H(c|i)$$

$H(c)$ est l'entropie de c et $H(c|i)$ est l'entropie conditionnelle de c sachant i. Cette méthode destinée à donner un ordre entre les descripteurs est basée sur un critère naïf, mais elle demande peu de calculs.

Ratio du gain d'information Son calcul consiste à diviser le gain d'information par l'entropie associée au descripteur. Comme le gain d'information, c'est un critère dont le calcul est rapide. Pour une classe c et un descripteur i, le ratio du gain d'information de i par rapport à c est :

$$R(i, c) = \frac{H(c) - H(c|i)}{H(i)}$$

ReliefF Il s'agit ([19], [26]) d'une amélioration de l'algorithme Relief (proposé dans [17], puis analysé et discuté dans [12]). Relief traite uniquement les problèmes de classification binaires. Son principe est d'estimer le mérite d'un descripteur selon sa *capacité à permettre de séparer les instances éloignées les unes des autres*. À chaque tour dans la boucle principale de l'algorithme un vecteur de la base d'entraînement est tiré au hasard et ses deux plus proches voisins dans chaque classe sont identifiés. Le mérite de chaque descripteur est alors mis à jour. Si le vecteur tiré au hasard et son plus proche voisin dans la même classe ont des valeurs différentes pour un descripteur, ce descripteur ne permet pas de distinguer deux vecteurs de la même classe et son score est diminué. Si le vecteur et son plus proche voisin dans l'autre classe ont des valeurs différentes pour un descripteur, ce descripteur permet de distinguer deux vecteurs de classes différentes et son score est augmenté.

Avec ReliefF, Kononenko généralise Relief pour son utilisation dans des problèmes multi-classes. Il modifie également l'algorithme de manière à ne plus considérer seulement les voisins les plus proches appartenant à chaque

classe, mais à constituer autant de voisinages de k points (k étant un paramètre de l'algorithme) qu'il y a de classes, et à moyenner sur ces voisinage. Cet apport rend l'algorithme moins sensible au bruit. La méthode possède un biais [8] qui est d'autant plus fort que le nombre de descripteurs est grand devant le nombre d'exemples d'apprentissage. Cependant, elle est peu coûteuse en calculs et constitue, parmi les *filter*, une alternative aux critères issus de la théorie de l'information.

Limite des approches Filter

En 1997, Kohavi & John [18] donnent différentes définitions de la notion de pertinence d'un descripteur et mettent en évidence les limites des approches de type *filter* en montrant que *la pertinence n'implique pas l'optimalité* et *l'optimalité n'implique pas la pertinence*. L'optimalité dépend du classifieur employé, et les méthodes *filter* connues jusqu'alors ne le prennent pas en compte.

Le problème du choix du nombre de descripteurs à retenir à partir de l'ordre donné par un algorithme *filter* est lié au fait que le classifieur n'est pas pris en compte lors de la sélection. Le nombre de descripteurs retenus a un impact imprévisible sur les différents critères servant à évaluer un classifieur.

2.6 Recherches et évaluateurs de sous-ensembles

Les approches combinant un algorithme de parcours de l'espace de recherche et un évaluateur de sous-ensembles de descripteurs mesurent la qualité d'un descripteur en prenant en compte un contexte : les autres descripteurs. Plus raffinée que les *filters*, cette approche aborde l'espace de recherche dans sa totalité avec une méthode "générer et tester". La génération des solutions candidates est faite par l'algorithme de parcours de l'espace de recherche. Le test est effectué par l'évaluateur qui attribue un mérite aux sous-ensembles de descripteurs, permettant à l'algorithme de recherche de les comparer entre eux pour diriger la recherche.

Heuristiques de recherche

Nous présentons ici les deux algorithmes de parcours de l'espace de recherche les plus couramment utilisés : la recherche naïve pas à pas et la recherche génétique. La première offre généralement un optimum local et est facilement paramétrable, alors que la seconde offre de plus grandes preuves théoriques de possibilité de convergence vers un optimum global mais ne converge pas systématiquement en un temps raisonnable.

Recherche pas à pas Avec cet algorithme, un descripteur est sélectionné à chaque itération. Dans sa variante "sélection en avant" (*forward selection*), pour chaque descripteur candidat à la sélection, le mérite du sous-ensemble de descripteurs incluant le candidat et tous les descripteurs sélectionnés aux

itérations précédentes est évalué. Le descripteur candidat maximisant le mérite est alors sélectionné. La recherche s'arrête quand aucun des descripteurs candidats n'améliore le mérite. À l'inverse, la variante "élimination en arrière" (*backward elimination*) consiste à débuter la recherche en sélectionnant tous les descripteurs, puis à retirer itérativement les descripteurs permettant d'augmenter le mérite des sous-ensembles de descripteurs. La recherche pas à pas est purement locale : elle suppose qu'un optimum global est constitué avec une succession d'optima locaux. Des algorithmes moins locaux existent, mais ils impliquent des temps de calculs encore plus élevés. Par ailleurs, dans [12], Isabelle Guyon donne des exemples de données montrant que la sélection en avant "manque" des descripteurs pertinents (au sens où ils permettent d'augmenter la valeur du mérite). *Le pouvoir discriminant d'un descripteur peut ne s'exprimer qu'en la présence d'un autre descripteur*. Dit différemment, deux descripteurs peuvent ne pas améliorer le mérite si ils sont pris séparément, et l'améliorer lorsqu'ils sont combinés.

Recherche génétique La recherche génétique [11] est un algorithme basé sur une analogie avec la sélection qui s'opère dans le génome des espèces au cours de leurs évolutions. Un individu est une solution du problème (ici, un sous-ensemble de descripteurs), et son génome le caractérise (ici, les descripteurs sélectionnés). L'algorithme fait évoluer une population initialisée aléatoirement par croisements et mutations, éliminant les individus minimisant le mérite à chaque itération. Cette recherche nécessite une grande quantité de calculs et son paramétrage est délicat, mais elle est connue pour sa capacité à ne pas se perdre dans des optima locaux comme la recherche pas à pas. Nous l'avons utilisée pendant nos expérimentations, mais pas pour les résultats présenté dans ce mémoire.

Évaluateurs de sous-ensembles de descripteurs

CFS Cet évaluateur [14], basé sur la *corrélation*, repose sur l'idée que pour un bon sous-ensemble de descripteurs, les valeurs des descripteurs sont très corrélées avec la classe et peu corrélées entre elles. Pour calculer le mérite d'un sous-ensemble de descripteurs, les matrices de corrélation entre descripteurs et classe, et entre descripteurs et descripteurs sont calculées. Le score attribué permet de favoriser les sous-ensembles de descripteurs corrélés avec la classe et de défavoriser ceux où les descripteurs sont très corrélés entre eux.

Wrapper Le *wrapper* [18] est la solution qui a été trouvée au problème majeur des *filters* : ces derniers ne prennent pas en compte le classifieur lors de la phase de sélection. Or, les descripteurs ont une influence peu maîtrisable d'un classifieur à l'autre. Ceci a fait naître la nécessité d'intégrer le classifieur au processus de sélection. Ainsi, pour évaluer un sous-ensembles de descripteurs, un classifieur est entraîné puis ses performances sont évaluées. Le mérite associé à un sous-ensemble est l'opposé de l'erreur de classification estimée pendant l'évaluation des performances du classifieur. L'algorithme a l'inconvénient de nécessiter une grande quantité de calculs (l'entraînement

et l'évaluation du classifieur sont longs, et très sujets à la dimensionnalité). En revanche, il a le mérite de permettre d'*optimiser en fonction d'un critère qui prend en compte le classifieur*. C'est ce qui le distingue autres algorithmes présentés.

2.7 Classifieurs utilisés

Les stratégies de sélection de descripteurs abordées précédemment sont applicables à tout type de classifieur. Nous allons ici présenter les deux classifieurs que nous avons utilisé lors de nos expériences. Leur entraînement et leur évaluation demande peu de calculs, ce qui les rend cohérents avec l'utilisation dans le cadre du suivi de partition.

Classifieur bayésien naïf Pour deux évènements A et B, la formule de Bayes permet d'exprimer la probabilité conjointe de A et B en fonction de la probabilité *a posteriori* de l'évènement B sachant la réalisation de l'évènement A, et de la probabilité de l'évènement A :

$$P(A \cap B) = P(B|A) \times P(A)$$

La formule de Bayes donne la probabilité *a posteriori* de l'évènement A sachant la réalisation de l'évènement B :

$$P(A|B) = \frac{P(A) \times P(B|A)}{P(B)}$$

Le classifieur bayésien naïf [16] s'appuie sur cette formule pour estimer les probabilités conditionnelles $P(C_i|D)$, où les C_i sont les différentes classes possibles, et D un vecteur de valeurs de descripteurs à classifier. La classe choisie est celle qui maximise la probabilité conditionnelle. Il convient de noter que le modèle de Bayes suppose une indépendance statistique entre les paramètres (ici, les descripteurs), ce qui constitue bien sûr une approximation dans notre application.

Gaussiennes multi-dimensionnelles modélisant chaque classe L'autre approche utilisée (décrite dans [4]) consiste à modéliser chaque classe à l'aide d'une gaussienne multi-dimensionnelle, où chaque dimension correspond à un descripteur. Les paramètres μ et σ de chaque gaussienne sont évalués grâce à l'algorithme *Expectation Maximization* [7]. Les deux gaussiennes multi-dimensionnelles ainsi formées permettent de déduire une probabilité d'appartenance pour chaque classe. Pour étiqueter un vecteur de descripteurs, la classe pour laquelle la probabilité d'appartenance est maximale est retenue.

Le fonctionnement de ce classifieur est naïf dans la mesure où il suppose que les classes sont représentables par des groupes de points. En revanche, l'étiquetage d'un vecteur demande une quantité de calculs extrêmement faible, ce qui fait de cet algorithme un bon candidat pour les applications temps-réel. Il est très sensible au bruit, ce qui en fait un classifieur pertinent dans le cadre d'un travail sur la sélection de descripteurs.

2.8 Évaluation de la performance

Concernant l'évaluation des algorithmes de sélection de descripteurs, le problème posé par les approches *filter* (c'est-à-dire la non connaissance de l'influence du nombre de descripteurs sur les performances du classifieur) met en avant la nécessité d'utiliser des critères d'évaluation relatifs à la performance de l'algorithme de classification qui effectue la segmentation. D'autres critères [10] indépendants du classifieur existent : une mesure de la séparabilité des classes (pour lequel un bon sous-ensemble de descripteurs assure une bonne séparabilité) ou encore une mesure d'entropie (pour laquelle un bon sous-ensemble de descripteurs est non-redondant). Nous ne les avons pas retenus dans la mesure où ils ne permettent pas de juger de la performance du système.

L'évaluation des performances des algorithmes de sélection de descripteurs passe donc par l'évaluation de la performance de détection du classifieur. Celle-ci peut être estimée en calculant la proportion de vecteurs mal classifiés (erreur de classification) sur une base de test distincte de la base d'apprentissage. Une variante très courante de cette méthode, la *k-fold cross-validation*, est plus adaptée aux situations où les utilisateurs des algorithmes de sélection ne disposent pas d'une base de test séparée de la base d'entraînement. Elle consiste à partitionner la base d'entraînement en k parties, à entraîner le modèle de classification utilisé sur $k-1$ parties, puis à estimer la proportion de classifications correctes en utilisant la partie restante comme base de test. Le procédé est répété k fois de façons à ce que chacune des k parties joue une fois le rôle de base de tests. Les performances obtenues dans les k configurations différentes sont finalement moyennées.

Enfin, il existe un dernier critère d'évaluation : l'aire sous la courbe ROC. L'utilisation d'un classifieur passe par le choix d'un seuil de sensibilité permettant de définir à partir de quelle probabilité d'appartenance un vecteur est déclaré "positif", soit appartenant à la classe. Issue de la théorie de la détection du signal, la courbe ROC (*Reciever Operating Characteristic*) est calculée en mesurant le compromis entre faux positifs et faux négatifs pour les différentes valeurs du seuil de sensibilité. La figure 2.2 page suivante en donne un exemple. Un classifieur qui étiquette les vecteurs au hasard correspond à la diagonale entre (0,0) et (1,1), soit une aire sour la courbe de valeur 0,5. À l'opposé, un classifieur "parfait" correspond à une droite entre (0,1) et (1,1), soit une aire sous la courbe de valeur 1.

Comparé à l'erreur de classification, l'aire sous la courbe ROC a le mérite de prendre en compte les différentes valeurs possibles du seuil de détection. De plus, certaines publications récentes ([15], [5]) affirment que l'aire sous la courbe ROC est un meilleur critère que l'erreur de classification pour caractériser la "qualité" de détection d'un modèle d'apprentissage.

Pour savoir comment sont évalués les algorithmes de sélection de descripteurs, nous avons considéré comme référence le concours de sélection de descripteurs NIPS 2003, dont les résultats sont analysés dans [13]. Les algorithmes de sélection sont évalués selon les critères suivants :

– L'erreur de classification équilibrée (*balanced error rate*), qui est la pro-

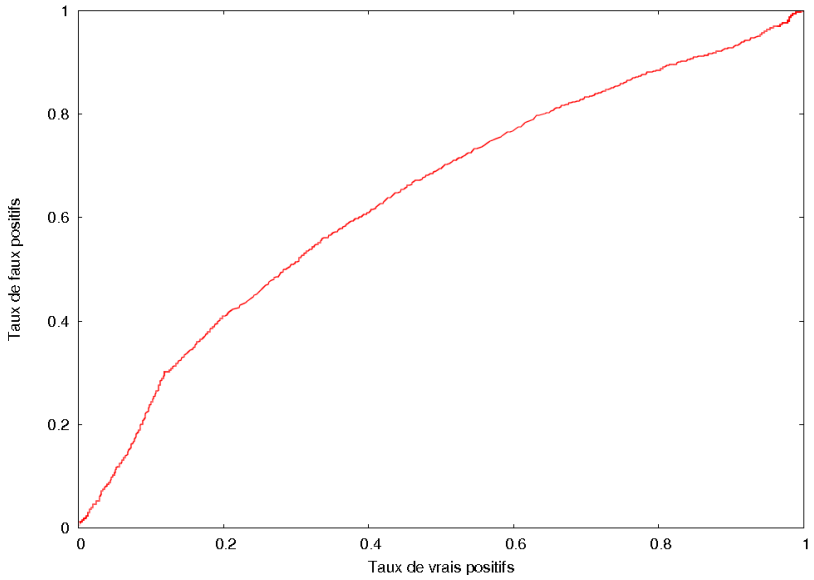

FIG. 2.2 – Exemple de courbe ROC.

portion de classifications fausses, moyennée dans chaque classe.
- L'aire sous la courbe ROC.
- Le pourcentage de descripteurs utilisés par rapport au nombre de descripteurs dans la base d'entraînement.
- Le pourcentage de descripteurs aléatoires sélectionnés par rapport au nombre de descripteurs bruités préalablement ajoutés.

Le critère retenu pour le classement des différents systèmes est l'erreur de classification pondérée. Mais il convient de remarquer que dans le cas d'un problème à deux classes, l'erreur de classification pondérée est égale à l'unité moins l'aire sous la courbe ROC.

CHAPITRE 3

Sélection de descripteurs et calcul des descripteurs

Sommaire

3.1 Problématique

Les algorithmes de sélection de descripteurs ont habituellement pour seul but de trouver un sous-ensemble des descripteurs aboutissant à la meilleure performance de classification possible, parmi n descripteurs. Cependant, pour les applications temps-réel le temps de calcul des descripteurs est un élément critique car il est impératif que le calcul des descripteurs et la classification puissent se faire en une durée inférieure à la longueur d'une fenêtre de signal.

Nous nous sommes donc intéressés aux algorithmes pouvant trouver non pas un mais *plusieurs* sous-ensembles de descripteurs, assurant chacun une performance de classification optimale par rapport au coût de calcul des descripteurs. On notera que les coûts de calcul maximaux sont donnés par l'implémentation des descripteurs utilisée.

La seule publication trouvée abordant le sujet du coût de calcul des descripteurs lors de la sélection est [22], où les auteurs proposent une stratégie de sélection de descripteurs intéressante basée sur des groupes de descripteurs : *group-wise forward selection*. Ils définissent un groupe comme un ensemble de descripteurs obtenus par un même calcul. Si un descripteur d'un groupe a été calculé, tous les autres descripteurs du groupe sont disponibles pour un coût de calcul nul. L'idée principale de leur algorithme est d'effectuer une

sélection dans chaque groupe, puis de retenir les meilleurs descripteurs de chaque groupe. La sélection utilise comme critère le ratio $C = \frac{\Delta P}{\Delta T}$, où ΔP représente la variation de performance apportée par le descripteur candidat à la sélection, et ΔT l'augmentation du coût de calcul liée à l'ajout du descripteur. Les descripteurs disponibles pour un coût de calcul nul (dans le cas où au moins un descripteur du groupe auquel ils appartiennent a déjà été sélectionné) et améliorant la performance de classification sont sélectionnés en priorité. Si aucun descripteur ne remplit cette condition, le descripteur maximisant le ratio C est sélectionné. La méthode a été appliquée à un problème de traitement d'images et offre une performance de classification plus élevée que la *forward selection* pour un coût de calcul des descripteurs donné. L'intérêt de cette approche réside dans la prise en compte du coût de calcul dans le critère servant à évaluer les descripteurs pendant la sélection, ainsi que dans le fait d'effectuer la sélection dans des groupes de descripteurs.

Cependant, cette technique n'est pas directement applicable à notre problème. Tout d'abord, elle cherche un seul sous-ensemble de descripteurs pseudo-optimal, et pas plusieurs. De plus, la modélisation des groupes de descripteurs ne correspond pas à la manière dont les descripteurs audio sont calculés. Les auteurs supposent que les groupes de descripteurs sont totalement indépendants les uns des autres. Or, les descripteurs audio sont basés sur des représentations du signal desquelles sont dérivés les descripteurs (par exemple, les descripteurs spectraux sont dérivés du résultat de la transformée de Fourier). Certaines représentations du signal sont dérivées d'autres représentations du signal (par exemple, les descripteurs harmoniques sont dérivés du résultat de l'analyse harmonique, qui est elle-même dérivée du spectre). Il nous a donc été nécessaire de modéliser la manière dont les descripteurs sont calculés.

3.2 Modélisation du calcul des descripteurs audio

L'estimation du coût de calcul des descripteurs pose de nombreuses questions pratiques. L'obtention de la valeur d'un descripteur nécessite le calcul de représentations du signal, ainsi qu'une quantité de calcul spécifique à chaque descripteur. Nous nous sommes basés sur la taxonomie des descripteurs audio donnée par G. Peeters [23], en négligeant la partie du calcul propre à chaque descripteur. Ceci constitue une approximation, mais le fait que les différents descripteurs soient groupés en fonction des données à partir desquelles ils sont calculés en fait une estimation du coût de calcul proche de la réalité (dans le sens où la mesure est proche du temps réellement consacré au calcul de chaque descripteur). D'autre part, cette modélisation a l'avantage d'être générique et applicable à d'autres types de descripteurs que ceux utilisés lors de nos expérimentations, y compris hors du domaine de l'audio.

Nous avons considéré les représentations du signal suivantes :
- Le *signal* lui-même, de coût de calcul nul.
- La *Transformée de Fourier à Court Terme*, dont le coût de calcul est noté c_{fft}. Elle se fait à partir du signal.
- Le *modèle sinusoïdal-harmonique*, dont le coût de calcul est noté $c_{harmonique}$.

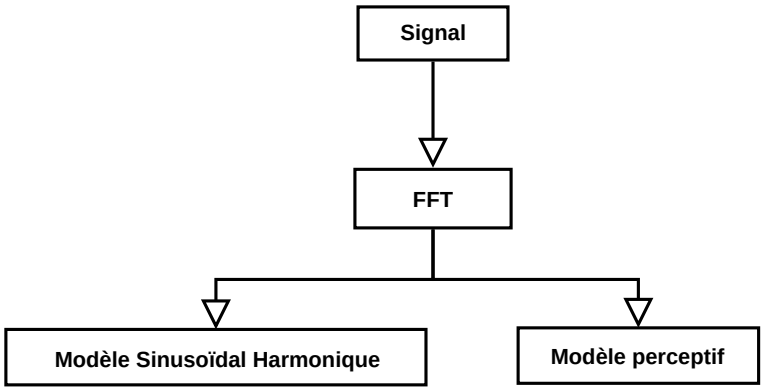

FIG. 3.1 – Représentations du signal servant au calcul des descripteurs

Il est calculé à partir du résultat de la Transformée de Fourier à Court Terme.

– Le *modèle perceptif*, dont le coût de calcul est noté $c_{perceptif}$, utilise également le résultat de la Transformée de Fourier à Court Terme.

Le nombre de cycles du microprocesseur utilisés par les fonctions calculant les différentes représentations du signal sont mesurables au moyen d'appels système qui sont disponibles sur les systèmes d'exploitation courants. Il est possible d'estimer ces nombres de cycles et de les moyenner sur plusieurs exécutions pendant la phase d'extraction des descripteurs servant à générer la base d'entraînement à partir d'échantillons sonores segmentés, ce qui a été fait durant nos expérimentations.

Coût de calcul d'un sous-ensemble de descripteurs La figure 3.1 illustre le processus de calcul des différentes représentations du signal. Cette structure d'arbre doit être prise en compte lors de l'évaluation du coût de calcul d'un sous-ensemble de descripteurs. Nous avons retenu la somme de tous les coûts de calcul correspondant aux représentations du signal nécessaires à l'obtention de tous les descripteurs du sous-ensemble.

En nous basant sur les représentations du signal citées précédemment, nous avons distingué les ensembles de descripteurs suivants (illustrés par la figure 3.2 page suivante) :

1. L'ensemble $E_{temporel}$ regroupant les *descripteurs temporels*. Ils sont calculés directement à partir des valeurs du signal (*ex :* Zero-Crossing Rate, auto-corrélation). Le coût de calcul de cet ensemble est considéré comme nul.
2. L'ensemble $E_{spectral}$ regroupant les *descripteurs spectraux*. Ils sont calculés à partir de la Transformée de Fourier à Court Terme. *Ex : Audio Spectrum Centroid, Audio Spectrum Spread*. Le coût de calcul associé à cet ensemble est c_{fft}.
3. L'ensemble $E_{harmonique}$ regroupant les *descripteurs harmoniques*. Ils sont calculés à partir du modèle sinusoïdal harmonique, dont l'obtention nécessite le calcul de la Transformée de Fourier à Court Terme. *Ex :*

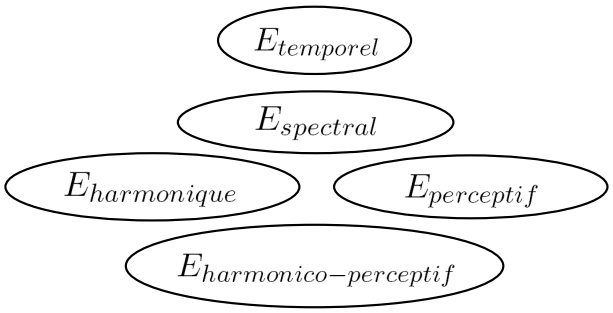

FIG. 3.2 – Les différents ensembles de descripteurs, basés sur leur mode de calcul

Audio Fundamental Frequency, Harmonic Spectral Tristimulus. Le coût de calcul associé à cet ensemble est $c_{fft} + c_{harmonique}$.

4. L'ensemble $E_{perceptif}$ regroupant les *descripteurs perceptifs*. Ils sont calculés à partir d'un modèle perceptuel, qui est lui-même calculé à partir de la Transformée de Fourier à Court Terme. *Ex :* coefficients *MFCC, Audio Filterbank Centroid, AudioLoudness*. Le coût de calcul associé à cet ensemble est $c_{fft} + c_{perceptif}$.
5. L'ensemble $E_{harmonico-perceptif}$ regroupant les *descripteurs harmonico-perceptifs*. Quelques descripteurs (*Audio Filterbank Deviation, Audio Filterbank Odd/Even Ratio, Audio Filterbank Tristimulus*) sont calculés à la fois à partir d'un modèle sinusoïdal harmonique et à partir d'un modèle perceptif. Le coût de calcul associé à cet ensemble est $c_{fft} + c_{harmonique} + c_{perceptif}$.

3.3 Recherche et évaluateurs de sous-ensembles

À l'exception de certains algorithmes de recherche permettant d'ordonner les descripteurs (principalement, la recherche pas à pas), les algorithmes combinaisons algorithmiques se basant sur un algorithme de parcours de l'espace de recherche et un algorithme d'évaluation de sous-ensembles de descripteurs ne sont pas adaptés au problème posé. En effet, ils n'apportent *aucune garantie sur le nombre de descripteurs* qui sera retenu dans le résultat. Si l'on fait travailler ces algorithmes sur la totalité des n dimensions dont on dispose en entrée, il est *impossible* de prévoir le coût de calcul du sous-ensemble de descripteurs sélectionné.

Toutefois une modification de l'évaluateur de sous-ensemble de descripteurs *wrapper* permettant d'intégrer le coût de calcul des descripteurs au calcul du "mérite" des sous-ensembles est possible. Nous l'avons d'ailleurs implémentée (combinée avec une recherche génétique) et nous avons constaté les problèmes qu'elle pose. Il faut choisir la fonction de coût qui combine les deux critères (le premier est l'erreur de classification, le second est le coût de calcul). Diverses fonctions candidates ont été testées, mais nous avons conclu que les approches qui suivent méritaient plus d'être étudiées car elles

se basent sur des façons d'aborder l'espace de recherche qui sont plus abordables du point de vue du temps de calcul.

3.4 Première approche : ordonner les descripteurs

Notre première approche est naïve et se base sur l'utilisation d'un algorithme de sélection de descripteurs qui ordonne les descripteurs selon un mérite. Ceux-ci sont de deux types :

- *Filters* purs (algorithmes attribuant un mérite à chaque descripteur, indépendamment du classifieur utilisé).
- Algorithmes basés sur une recherche permettant d'ordonner les descripteurs sélectionnés (comme la recherche pas à pas) et sur un évaluateur de sous-ensembles de descripteurs.

Dans les deux cas, l'ordre obtenu permet de former facilement plusieurs sous-ensembles de descripteurs de coût de calcul croissant et de mérite croissant (la notion de mérite variant selon les algorithmes). Nous formons n sous-ensembles de descripteurs, où le n-ième sous-ensemble est constitué des n premiers descripteurs selon l'ordre donné par l'algorithme de sélection.

Les algorithmes en question ayant pour principe d'ordonner les descripteurs suivant un mérite, notre première approche consiste à évaluer (pour un classifieur donné) la performance de classification obtenue avec les n sous-ensembles de descripteurs, puis à retenir le sous-ensemble aboutissant à la meilleure performance pour chaque coût de calcul maximum possible. Cette approche a le défaut de ne pas garantir l'obtention d'un sous-ensemble de descripteurs pour tous les coûts de calcul maximums possibles. En effet, nous ne pouvons pas prédire le coût de calcul du premier descripteur (de mérite maximal) trouvé par l'algorithme.

3.5 Deuxième approche : taxonomie de recherche

Une autre approche consiste à former plusieurs ensembles de descripteurs (un pour chaque coût de calcul maximum) pour lesquels tout sous-ensemble de descripteurs a un coût de calcul inférieur au coût de calcul maximum, et à effectuer une sélection dans chaque ensemble. Pour ces sélections, n'importe quel algorithme de sélection de descripteurs peut être utilisé.

Taxonomie de recherche

Notre taxonomie de recherche s'appuie sur les représentations du signal qui sont nécessaires à l'obtention des descripteurs. Les sous-ensembles de descripteurs formés en combinant des descripteurs des ensembles $E_{temporel}$, $E_{spectral}$, $E_{harmonique}$, $E_{perceptif}$ et $E_{harmonico-perceptif}$ ont pour coût de calcul une combinaison linéaire des coûts de calcul des différentes représentations du signal. En ré-utilisant le formalisme donné à la partie 2.3 page 11, si n est le nombre de descripteurs selon lesquels les exemples sont décrits, p le nombre de représentations du signal prises en compte dans le modèle, $c_{representation}(i)$

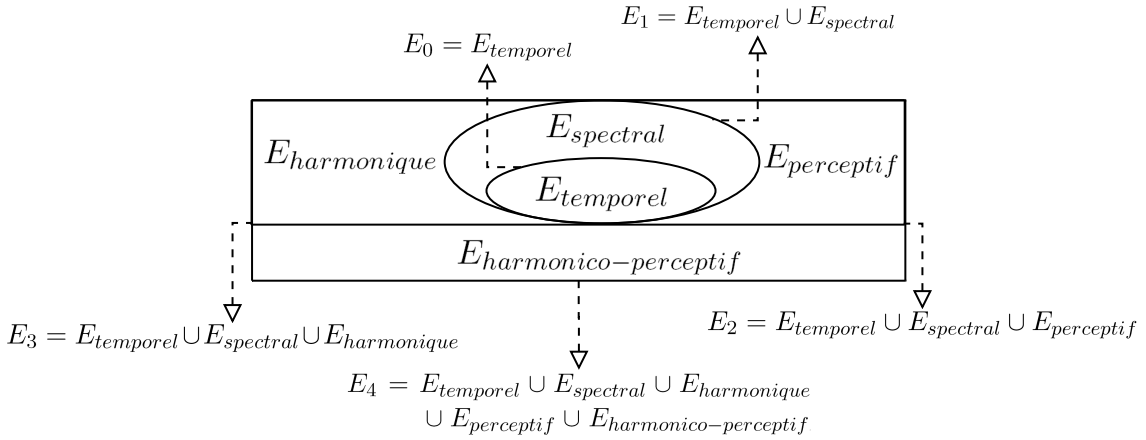

FIG. 3.3 – Taxonomie de recherche

le coût de calcul associé à la i-ème représentation du signal, alors pour tout sous-ensemble de descripteurs $s \in \{0,1\}^n, \exists(\lambda_i) \in \{0,1\}^p$ tel que

$$coût(s) = \sum_{i=0}^{p-1} \lambda_i \times c_{representation}(i)$$

.

Avec la modélisation du coût de calcul des descripteurs audio explicitée précédemment, nous pouvons former les ensembles de descripteurs suivants. Pour chacun d'eux, nous disposons d'une borne supérieure sur le coût de calcul d'un sous-ensemble quelconque, que nous notons c_i :

- Pour tout sous-ensemble de $E_0 = E_{temporel}$, le coût de calcul est inférieur ou égal à $c_0 = 0$, ce qui revient à dire qu'il est nul.
- Pour tout sous-ensemble de $E_1 = E_{temporel} \cup E_{spectral}$, le coût de calcul est inférieur à $c_1 = c_{fft}$.
- Pour tout sous-ensemble de $E_2 = E_{temporel} \cup E_{spectral} \cup E_{perceptif}$, le coût de calcul est inférieur à $c_2 = c_{fft} + c_{perceptif}$.
- Pour tout sous-ensemble de $E_3 = E_{temporel} \cup E_{spectral} \cup E_{harmonique}$, le coût de calcul est inférieur à $c_3 = c_{fft} + c_{harmonique}$.
- Pour tout sous-ensemble de $E_4 = E_{temporel} \cup E_{spectral} \cup E_{harmonique} \cup E_{perceptif} \cup E_{harmonico-perceptif}$, le coût de calcul est inférieur à $c_4 = c_{fft} + c_{harmonique} + c_{perceptif}$.

Ces différents ensembles constituent une taxonomie de recherche dont la figure 3.3 donne une représentation visuelle. La caractéristique principale de notre taxonomie est d'être basée sur un modèle de calcul des descripteurs. D'une part, l'utilisation de la taxonomie permet de *combattre la dimensionnalité en réduisant l'espace de recherche*. Plus le nombre de descripteurs en entrée est élevé, plus il est difficile pour les algorithmes de sélection de trouver un résultat s'approchant d'un optimum global. Ainsi, la sélection dans les ensembles de descripteurs de faible cardinalité consititue un problème d'optimisation plus simple à résoudre pour les algorithmes (voir 4.5 page 37).

D'autre part, pour chaque coût de calcul maximum, la sélection de des-

cripteurs est effectuée sur le plus grand ensemble de descripteurs (en terme de cardinalité) *garantissant* que le résultat sera inférieur au coût de calcul maximum. Ceci permet d'éviter l'utilisation d'un critère prenant en compte le coût de calcul lors de la sélection tel que dans [22].

Enfin, la démarche présentée est parallélisable, dans la mesure où les sélections ayant lieu dans les différents ensembles de descripteurs sont indépendantes. Si l'algorithme de recherche utilisé le permet, la recherche peut par ailleurs être arrêtée pendant son exécution et donner la meilleure solution trouvée jusqu'alors. La recherche pas à pas utilisée pendant nos expérimentations satisfait d'ailleurs ce critère.

CHAPITRE 4

Expérimentations

Sommaire

4.1 Données d'entraînement

Les échantillons sonores déjà segmentés utilisés proviennent de la base de P. Leveau [20] (14 échantillons représentants divers styles musicaux), ainsi que d'une base d'extraits de la partie vocale de *"En Echo"* de Phillipe Manoury pour soprano et électronique (11 échantillons). La base d'apprentissage (telle que formalisée au 2.3 page 11) est constituée de 15009 vecteurs et des étiquettes de classe associées. Parmi vecteurs, 1945 ont le label de classe "Attaque", et 13064 le label "Autre".

Observations sur la segmentation manuelle La segmentation utilisant un classifieur implique une expertise humaine : nous supposons que l'on dispose d'exemples correctement annotés, puis nous entraînons un modèle de classification sur ces exemples. Or, il convient de remarquer que la notion d'attaque est un concept flou ([9], [23]). Ceci se ressent particulièrement au moment de l'annotation des exemples d'apprentissage : dans de nombreux cas la segmentation des attaques relève d'un choix de l'expert.

D'autre part, l'annotation se fait le plus souvent à l'aide de logiciels de traitement du son dans lesquels il est possible de constituer une suite de marqueurs temporels. Mais ces logiciels offrent une vue graphique du signal qui se base elle-même sur une représentation du signal (la forme d'onde, ou bien

une représentation spectrale). Il est très probable que cette représentation graphique influence largement l'expert dans son annotation. Cette influence peut avoir pour conséquence de mettre en valeur certains descripteurs par rapport à d'autres lors du processus de sélection. Par exemple, si le logiciel utilise une représentation du spectre, l'expert pourra avoir tendance à se référer à ce qu'il voit pour effectuer son annotation, et donc à segmenter en fonction de propriétés spectrales du signal.

Extraction des descripteurs et pré-traitements

Nous avons utilisé l'extracteur de descripteurs du projet CUIDADO, décrit dans [23]. Seuls les descripteurs basés sur le fenêtrage (ou "descripteurs instantanés") ont été retenus. Au total, 49 descripteurs ont été utilisés. La majorité d'entre eux sont multi-dimensionnels, et nous avons considéré chaque dimension de ces descripteurs comme un descripteur indépendant. Ainsi, la base d'entraînement était constituée de 223 descripteurs mono-dimensionnels. Les différents descripteurs figurent en annexe p.53.

Pré-traitements Les valeurs des descripteurs extraits ont été normalisées, car ce procédé a pour effet d'améliorer les performances des classifieurs [10] et n'est pas incompatible avec notre application. Il est tout à fait envisageable d'appliquer en temps-réel les mêmes facteurs de normalisation des descripteurs que ceux qui ont été obtenus pendant la normalisation de la base d'apprentissage.

Estimation du coût de calcul des descripteurs

Lors de la phase d'extraction des descripteurs, les temps d'exécution des phases de pré-traitement (*"pre-computing"* dans [23]) suivantes ont été mesurés :

- Transformée de Fourier à Court Terme (t_{fft})
- Modèle sinusoïdal harmonique ($t_{harmonique}$)
- Modèle perceptif ($t_{perceptif}$)

Le temps de calcul de l'enveloppe d'énergie a été ignoré car ce calcul sert à obtenir des descripteurs temporels globaux que nous n'utilisons pas. Les estimations ont été divisées par $t_{fft} + t_{harmonique} + t_{perceptif}$, qui est le coût de calcul maximum que peut avoir un sous-ensemble de descripteurs en partant de nos données. Ceci revient donc à normaliser le coût de calcul des sous-ensembles de descripteurs.

4.2 Critère d'évaluation

Parmi les critères d'évaluations (présentés au 2.8 page 17), nous avons retenu l'erreur d'apprentissage (estimée par cross-validation sur 5 folds) pour estimer la "qualité" d'un sous-ensemble de descripteurs. D'une part, ce critère prend en compte le classifieur utilisé (contrairement à la corrélation, ou au gain d'information, par exemple). D'autre part, notre but étant d'obtenir

un système performant (c'est-à-dire un système qui segmente le plus correctement possible), la mesure de la proportion de classifications correctes effectuée par le classifieur semble mesurer ce que l'on attend du système. Enfin, c'est un critère généralement accepté dans la littérature ([12], [18], [28]).

4.3 Algorithmes retenus pour les expériences

Classifieurs

La classification en temps-réel impose des contraintes fortes concernant le choix du classifieur. Pour nos expériences, nous avons retenu deux classifieurs pour lesquels la quantité de calcul nécessaire à l'étiquetage d'un vecteur de descripteurs est faible : un classifieur bayésien naïf, et une gaussienne multidimensionnelle modélisant chaque classe. Ces deux algorithmes ont été décrits précédemment. La comparaison des deux approches que nous proposons sur deux classifieurs différents permet de donner une idée de ce qui est propre à un classifieur et ce qui est généralisable à plusieurs classifieurs.

Algorithmes de sélection de descripteurs

Pour évaluer les deux approches de résolution du problème présenté au chapître 3, nous avons retenu cinq algorithmes de sélection de descripteurs. Trois d'entre eux ordonnent les descripteurs, alors que les deux autres sont basés sur un évaluateur de sous-ensembles de descripteurs et un algorithme de recherche.

Algorithmes ordonnant les descripteurs selon un mérite - Les trois premiers algorithmes sont des approches *filter* : ReliefF (en utilisant toutes les instances et avec des voisinages de taille $k = 10$), gain d'information et ratio du gain d'information. Leur principe de fonctionnement a été expliqué dans la partie 2.5.

Évaluateurs de sous-ensembles et recherche Les deux autres algorithmes sont des évaluateurs de sous-ensembles de descripteurs combinés à des recherches pas à pas. D'une part, nous avons retenu *CFS* car c'est un évaluateur de sous-ensembles de descripteurs demandant peu de calcul. D'autre part, nous avons retenu l'évaluateur de sous-ensemble de descripteurs *wrapper* pour ses performances déjà établies dans la littérature. C'est un algorithme d'évaluation demandant beaucoup de calculs, mais contrairement aux autres méthodes il a pour lui le mérite d'optimiser un critère dépendant du classifieur. La recherche pas à pas est également coûteuse, mais elle a le mérite de pouvoir être à la fois utilisée pour parcourir l'espace de recherche avec un évaluateur de sous-ensemble de descripteurs, et pour ordonner les descripteurs (et donc de l'utiliser comme un *filter*). Ceci nous permet de comparer le *wrapper* dans les deux approches.

Nous employons l'expression "*filters* purs" pour désigner les méthode attribuant un score aux descripteurs, à l'exception du cas particulier du *wrapper* utilisé comme un *filter*.

4.4 Première approche : ordonner les descripteurs

Nous traitons ici l'approche présenté dans la partie 3.4 page 23, qui consiste à former des ensembles de descripteurs de plus en plus grands à partir de l'ordre donné par des algorithmes permettant d'ordonner les descripteurs selon un mérite.

Influence du nombre de descripteurs sur l'erreur d'apprentissage

Dans un premier temps, nous avons étudié l'influence du nombre de descripteurs retenus à partir de l'ordre donné par les algorithmes sur la performance du classifieur. Les figures 4.1 et 4.2 page suivante donnent l'erreur de classification en fonction du nombre de descripteurs inclus.

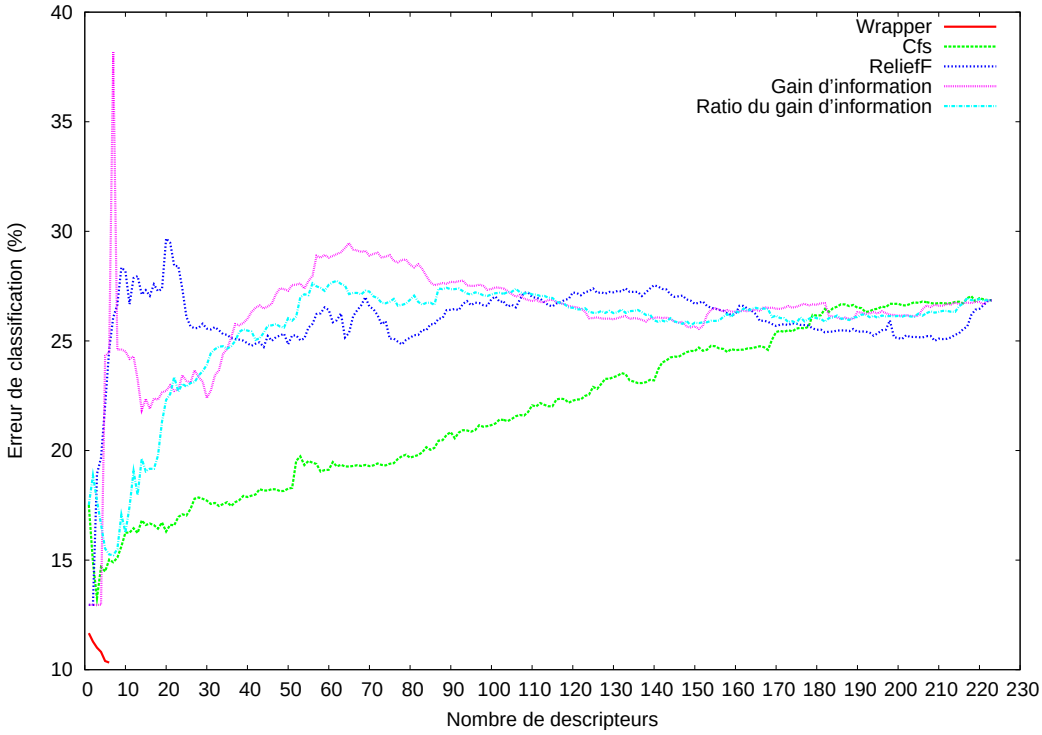

FIG. 4.1 – Influence du nombre de descripteurs retenus sur l'erreur de classification pour un *classifieur bayésien naïf*.

***Filters* purs** Tout d'abord, il convient de noter que les courbes illustrent bien les limites déjà connues des méthodes *filter*. Contrairement à ce que peut laisser penser le principe consistant à ordonner les descripteurs suivant un mérite, *l'erreur d'apprentissage n'est pas du tout une fonction strictement décroissante du nombre de descripteurs sélectionnés*. Dans certains cas l'ajout d'un descripteur dégrade considérablement la performance de classification, et l'ajout du descripteur suivant l'améliore (le cas du critère du gain d'information et du classifieur bayésien naïf en est un bon exemple). Pour un algorithme donné,

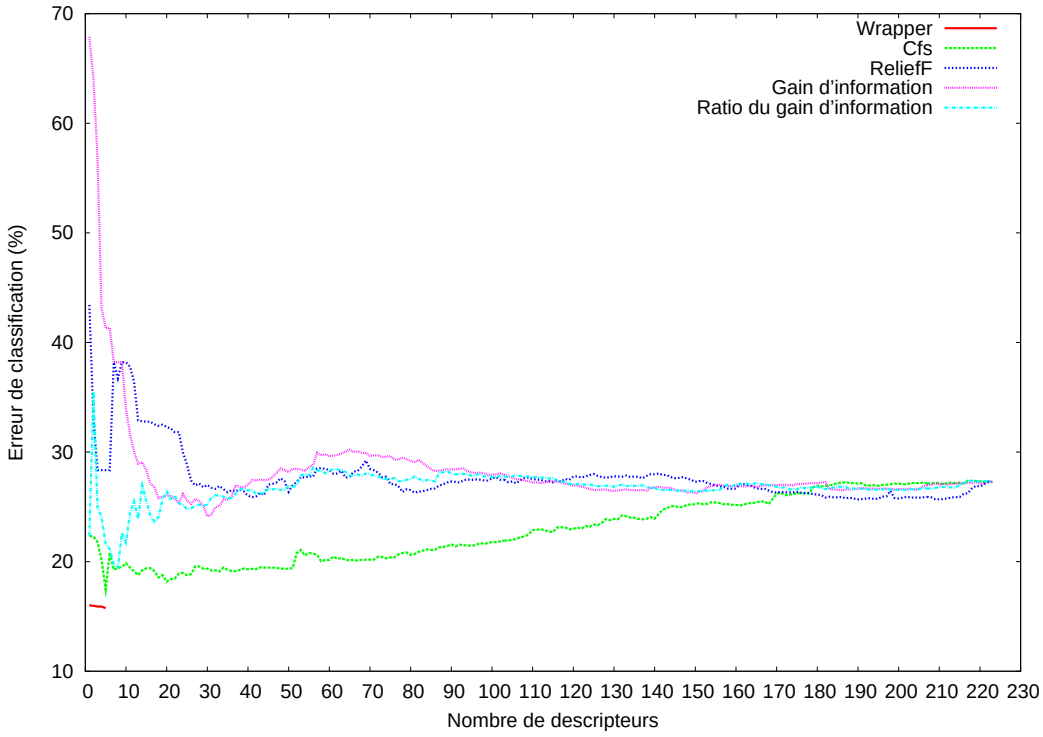

FIG. 4.2 – Influence du nombre de descripteurs retenus sur l'erreur de classification avec une *gaussienne multi-dimensionnelle modélisant chaque classe*.

le nombre de descripteurs aboutissant à un optimum est différent selon le classifieur. Ceci implique la nécessité d'évaluer les sous-ensembles avec un classifieur (à la manière du *wrapper*) pour choisir le nombre de descripteurs à retenir de l'ordre donné par l'algorithme de sélection.

De plus, nous constatons que les meilleurs résultats (c'est-à-dire les sous-ensembles de descripteurs menant à la meilleure performance de classification) sont trouvés avec un nombre faible de descripteurs. Pour certains algorithmes (gain d'information, ReliefF), le meilleur résultat est même obtenu uniquement avec le premier descripteur. Ceci laisse penser que le problème du choix du nombre de descripteurs peut ne pas ce poser avec ces algorithmes (une confirmation requiert un test sur d'autres données de les notres). Toutefois, la solution trouvée constitue un optimum très local (des sous-ensembles de descripteurs plus performants ont été découverts dans la suite nos expérimentations).

Wrapper* utilisé comme un *filter Pour les deux classifieurs, c'est l'évaluateur de sous-ensemble de descripteur *wrapper* qui permet d'obtenir l'erreur de classification la plus basse. Ceci s'explique par le fait que cette approche consiste à optimiser l'erreur de classification elle-même.

Enfin, le dernier point de chaque courbe correspond à l'évaluation de la performance de classification lorsque tous les descripteurs sont utilisés. Les

valeurs mesurées sont données par le tableau 4.1.

Classifieur	Bayésien naïf	Gaussienne multi-dimensionnelle modélisant chaque classe
Erreur de classification	26,9%	27,3 %

TAB. 4.1 – Erreur d'apprentissage obtenue *sans sélection de descripteurs*.

Influence du nombre de descripteurs sur le coût de calcul

La courbe 4.3 montre l'évolution du coût de calcul des sous-ensembles de descripteurs en fonction du nombre de descripteurs inclus suivant l'ordre donné par l'algorithme de sélection pour un classifieur bayésien naïf. Comme pouvons le constater, *le coût de calcul des sous-ensemble croît extrêmement vite quel que soit l'algorithme de sélection* utilisé. Le coût de calcul maximal est atteint avec un ou deux descripteurs dans le pire des cas, avec dix descripteurs dans le meilleur des cas. Des résultats similaires ont été obtenus avec l'autre classifieur retenu pour notre étude.

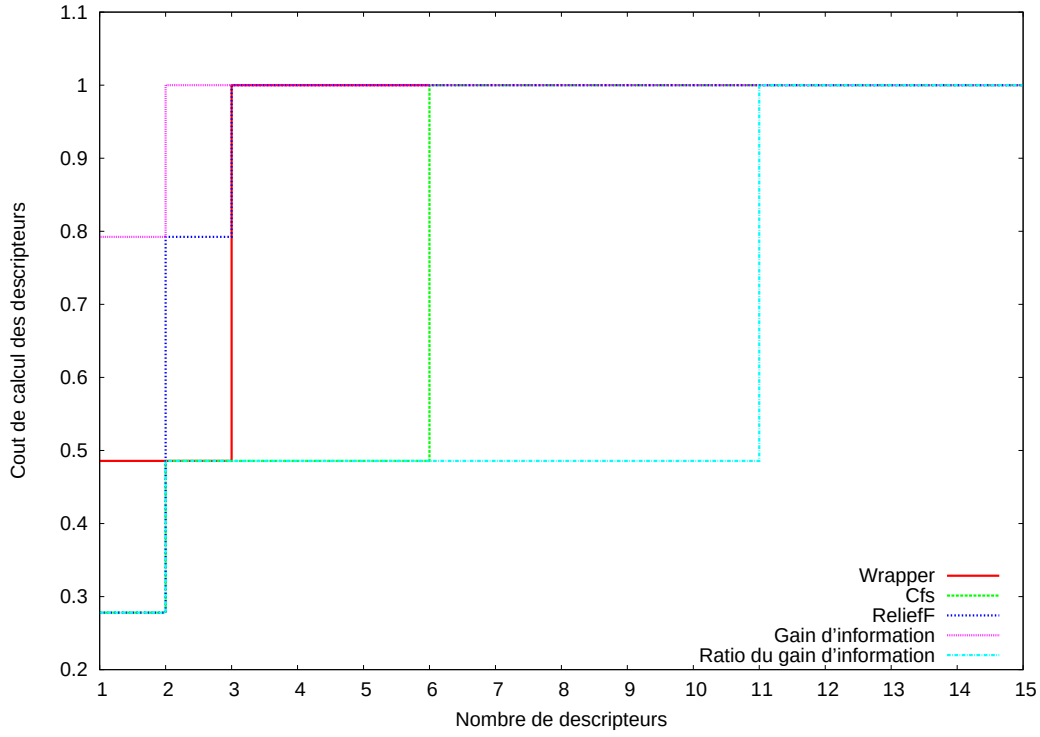

FIG. 4.3 – Influence du nombre de descripteurs retenus sur le coût de calcul du sous-ensemble pour un classifieur bayésien naïf

Notre approche consiste à évaluer la performance de classification des n sous-ensemble de descripteurs formés en incluant les n premiers descripteurs

suivant l'ordre donné par l'algorithme de sélection, à évaluer la performance de classification obtenue avec chaque sous-ensemble de descripteurs (pour un classifieur donné), puis à retenir la meilleure performance pour chaque coût de calcul maximum. Les résultats donnés par la courbe 4.3 montrent que *concernant les coût de calculs inférieurs au coût de calcul maximum, notre première méthode prend ses décisions à partir d'un nombre très faible d'évaluations* (dix dans le meilleur des cas). Nous avons donc de bonnes raisons de penser que pour ces coûts de calcul, la méthode proposée donne des résultats éloignés de l'optimum global.

4.5 Deuxième approche : taxonomie de recherche

Le principe de la deuxième approche (présentée dans la partie 3.5 page 23) est d'effectuer autant de sélections de descripteurs qu'il y a de coûts de calculs maximums, dans des ensembles de descripteurs pour lesquels tout sous-ensemble a un coût de calcul borné par le coût de calcul maximum. Elle a été implémentée et comparée à la première approche.

Ensembles de descripteurs et cardinalité de l'espace de recherche

Le tableau 4.2 donne le nombre de descripteurs des différents ensembles de decripteurs utilisés dans la taxonomie de recherche, ainsi qu'un ordre de grandeur de la cardinalité des espaces de recherche associés (c'est-à-dire $2^{Card(E_i)}, i \in \{0,1,2,3,4\}$). Il est important d'avoir une idée du nombre de solutions possibles pour chaque groupe pour pouvoir analyser les résultats.

Comme on peut le constater, les ensembles E_i ne sont pas croissants en nombre de descripteurs car ils sont formés à partir des coûts de calcul associés aux représentation du signal nécessaires à l'obtention des descripteurs qu'ils contiennent, et non selon le nombre de descripteurs dérivés de chaque représentation.

Ensemble de descripteurs	E_0	E_1	E_2	E_3	E_4
Valeur normalisée du coût de calcul maximum associé (c_i)	0	0,28	0,49	0,79	1
Nombre de descripteurs	14	49	155	102	223
Cardinalité de l'espace de recherche	16384	6.10^{14}	5.10^{46}	5.10^{30}	1.10^{67}

TAB. 4.2 – Ensembles de descripteurs associés à la taxonomie de recherche, tels que décrit au 3.5 page 23

Résultats

Les courbes 4.4 page suivante et 4.5 page 35 donnent l'erreur de classification évaluée en fonction du cout de calcul maximum des descripteurs pour

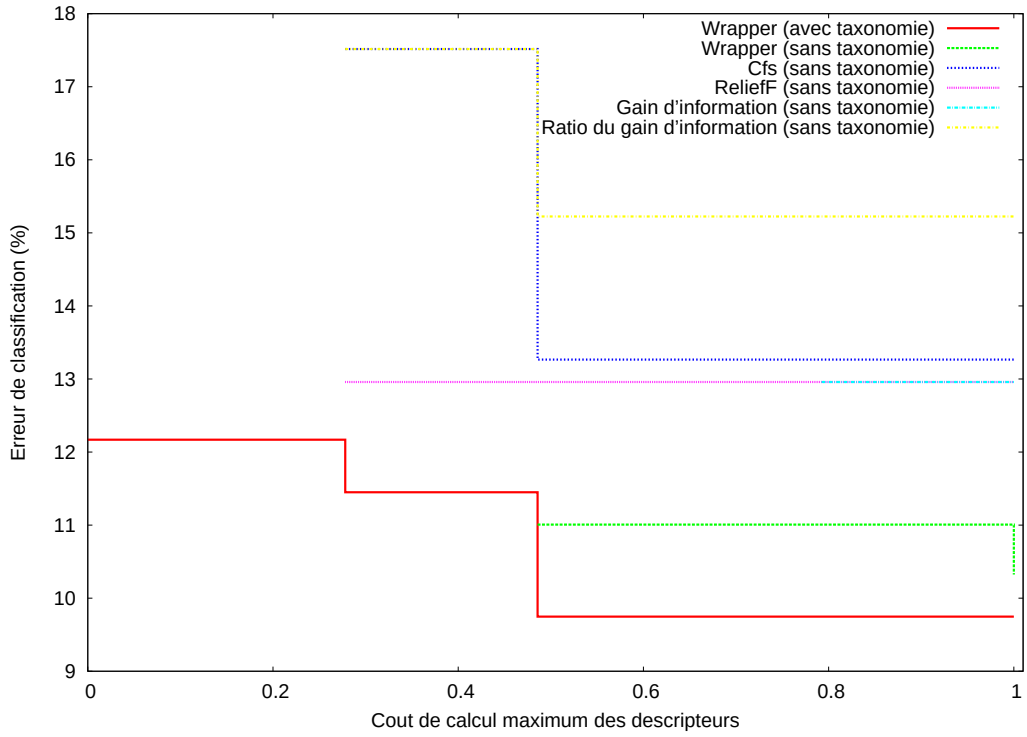

FIG. 4.4 – Erreur de classification associée au meilleur sous-ensemble de descripteurs trouvé en fonction du coût de calcul maximum, pour un *classifieur bayésien naïf*

les deux approches proposées, respectivement pour un classifieur bayésien et pour une gaussienne multi-dimensionnelle modélisant chaque classe. Ces courbes donnent la performance du meilleur sous-ensemble de descripteurs correspondant à chaque coût maximum.

Pour les deux classifieurs choisis, les deux cas où il y a combinaison de l'évaluateur de sous-ensembles de descripteurs *wrapper* et de la recherche pas à pas aboutissent aux meilleurs performances. Par ailleurs, *l'utilisation de la taxonomie de recherche permet d'obtenir des solutions pour les coûts de calcul faibles, et donne une solution plus performante quel que soit le coût de calcul maximum.*

Première approche Tout d'abord, nous constatons que les *filters* (nous incluons ici le cas du *wrapper* utilisé comme un *filter*) ne répondent que partiellement au problème de la sélection de descripteurs par rapport à un coût de calcul maximum des descripteurs. En effet, la première approche ne donne pas de sous-ensembles de descripteurs pour les coûts de calcul les plus faibles quel que soit l'algorithme de sélection utilisé.

Par ailleurs, nous remarquons que dans le cas de la gaussienne multi-dimensionnelle modélisant chaque classe, certains *filters* (ReliefF et le gain d'information) donnent des résultats médiocres pour les coûts de calcul des descripteurs inférieurs au maximum. Ceci est dû au fait que les sous-ensembles

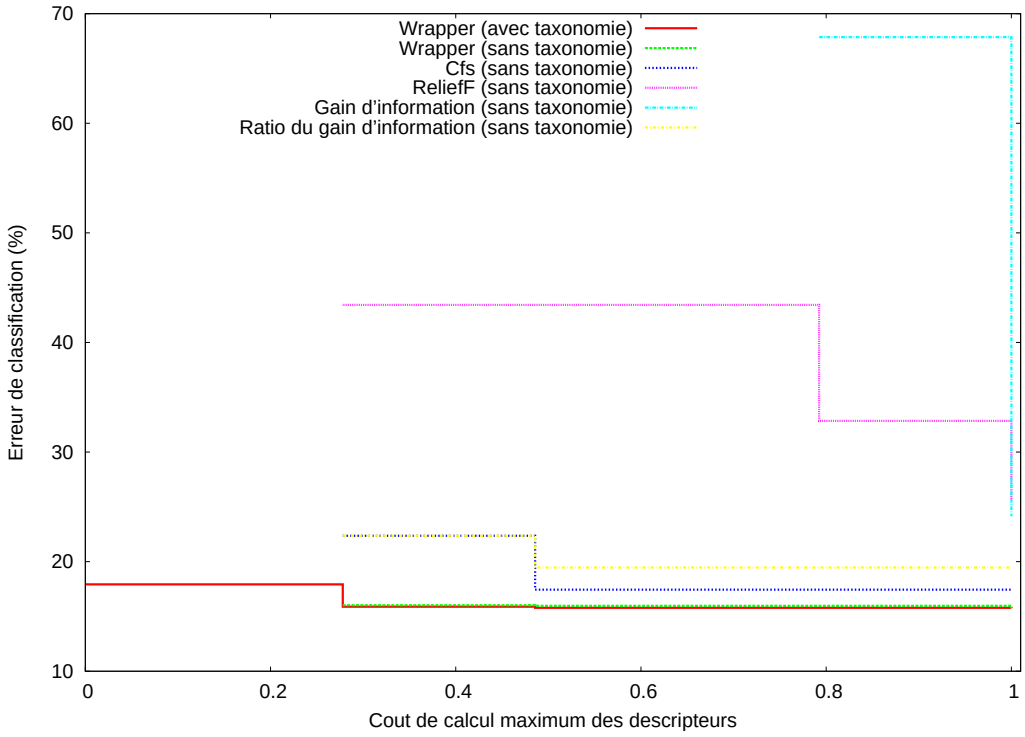

FIG. 4.5 – Erreur de classification associée au meilleur sous-ensemble de descripteurs trouvé en fonction du coût de calcul maximum avec une *gaussienne multi-dimensionnelle modélisant chaque classe*.

de descripteurs de faible coût de calcul formés à partir de l'ordre donné par ces algorithmes sont d'une part peu nombreux, et d'autre part peu performants.

Deuxième approche Avec les deux classifieurs, *l'ajout des descripteurs spectraux permet dans tous les cas d'améliorer la performance de classification*. Par ailleurs, les descripteurs trouvés dans l'ensemble E_2 (incluant les descripteurs temporels, spectraux et perceptifs) sont dans les deux cas plus performants que ceux obtenu en sélectionnant dans l'ensemble E_3 (incluant les descripteurs temporels, spectraux et harmoniques). Ceci se voit au fait qu'il n'y a pas de palier au coût de calcul correspondant à E_3, dont la valeur est 0,79. La signification de ce résultat est que *l'ajout des descripteurs perceptifs est plus bénéfique à la performance du système que l'ajout des descripteurs harmoniques*.

Par ailleurs toujours avec les deux classifieurs, *aucune meilleure performance n'est trouvée dans l'ensemble E_4 qui contient tous les descripteurs*. Nous expliquons ceci par le fait que la dimensionnalité du problème devient trop élevée pour la recherche pas à pas, ce qui se traduit par un résultat plus éloigné d'un optimum (en tout cas, un meilleur optima est trouvé dans des ensembles de plus petite taille).

Gain de performance lié à la sélection

Pour dresser un bilan de l'intérêt de la sélection de descripteurs comme moyen de réduire le coût de calcul tout en améliorant les performances, il convient de comparer les performances de classification obtenues après sélection avec celles qui sont obtenues en utilisant la totalité des descripteurs.

Les gains de performance indiqués dans les tableaux suivants sont la différence entre l'erreur de classification obtenue avec tous les descripteurs et celle correspondant au meilleur sous-ensemble de descripteurs trouvé. Ils font clairement ressortir l'intérêt de la réduction de dimensionnalité. En fonction des différents coûts maximums de calcul, la sélection des descripteurs permet de *réduire l'erreur de classification de 17,2% dans le meilleurs des cas*, et de *14,7% dans le pire des cas* pour le classifieur bayésien naïf, et de *11,5% dans le meilleur des cas* et *9,4% dans le pire des cas* pour les gaussiennes multi-dimensionnelles modélisant les classes.

Nous pensons que ces performances sont encore améliorables. Des pistes de recherche allant dans cette direction sont données dans la partie 5.4.

Coût de calcul maximum c_{max}	$0 \leq c_{max} < c_1$	$c_1 \leq c_{max} < c_2$	$c_2 \leq c_{max} \leq c_4$
Erreur de classification associée au meilleur sous-ensemble de descripteurs trouvé	12,2%	11,4%	9,7%
Gain de performance lié à la sélection de descripteurs	**14,7%**	**15,5%**	**17,2%**

TAB. 4.3 – Résultats obtenus avec l'évaluateur de sous-ensembles de descripteurs *wrapper* et la recherche pas à pas utilisés avec la taxonomie de recherche, pour un *classifieur bayésien naïf*. Erreur de classification obtenue avec tous les descripteurs : 26,9%.

Coût de calcul maximum c_{max}	$0 \leq c_{max} < c_1$	$c_1 \leq c_{max} < c_2$	$c_2 \leq c_{max} \leq c_4$
Erreur de classification associée au meilleur sous-ensemble de descripteurs trouvé	17,9%	15,9%	15.8%
Gain de performance lié à la sélection de descripteurs	**9,4%**	**11,4%**	**11,5%**

TAB. 4.4 – Résultats obtenus avec l'évaluateur de sous-ensembles de descripteurs *wrapper* et la recherche pas à pas utilisés avec la taxonomie de recherche, avec une *gaussienne multi-dimensionnelle modélisant chaque classe*. Erreur de classification obtenue avec tous les descripteurs : 27,3%.

Descripteurs sélectionnés

Les descripteurs sélectionnés dans la configuration qui aboutit à la meilleure performance de classification lors de nos expérimentations (c'est-à-dire l'évaluateur de sous-ensemble de descripteurs *wrapper* combiné à une recherche pas à pas, appliqué à la deuxième approche proposée) sont donnés tableau 4.5 (pour le classifieur bayésien naïf) et tableau 4.6 page suivante (pour une modélisation de chaque classe par une gaussienne multi-dimensionnelle).

Coût de calcul maximum c_{max}	**Meilleur sous-ensemble de descripteurs**
$0 \leq c_{max} < c_1$	{ AudioXcorr 6}
$c_1 \leq c_{max} < c_2$	{ AudioXcorr 1, AudioXcorr 12, AudioSpectrumSkewness 6, AudioSpectrumSlope 2, AudioSpectrumVariation 2 }
$c_2 \leq c_{max} \leq c_4$	{ AudioXcorr 12, AudioSpectrumSlope 3, AudioSpectrumRolloff, AudioSpectrumVariation 2, AudioFilterbankKurtosis 4, AudioFilterbankSlope 2, AudioDeltaMFCC 1, AudioDeltaDeltaMFCC 2, AudioSpectrumCrest 3, AudioRelativeSpecificLoudness 17 }

TAB. 4.5 – Meilleurs sous-ensembles de descripteurs (en terme d'erreur de classification) trouvés avec l'évaluateur de sous-ensembles de descripteurs *wrapper* et la recherche pas à pas utilisés avec la taxonomie de recherche, pour un *classifieur bayésien naïf*. Pour les descripteurs multi-dimensionnels, la dimension est donnée en comptant à partir de 1.

Taille des ensembles *vs.* performance du résultat

Dans notre deuxième approche, nous retrouvons plusieurs fois la situation où nous effectuons une sélection dans un ensemble de descripteurs E, puis dans un ensemble $E \cup E'$, avec $E' \neq \emptyset$. Nous rappelons que les sélections à l'intérieur des ensembles sont faites avec le même algorithme. En première approche, nous pourrions nous attendre à ce que le sous-ensemble de descripteurs $s_{E \cup E'}$ trouvé à partir de l'ensemble $E \cup E'$ soit au moins aussi performant (au sens de l'erreur d'apprentissage) que le sous-ensemble de descripteurs s_E trouvé à partir de l'ensemble E.

Or, il n'en est rien. Parfois, la performance associée à s_E est meilleure que celle associée à $s_{E \cup E'}$. Dans ce cas, notre implémentation associe s_E au coût maximum de calcul correspondant à $E \cup E'$. Ceci a du sens car $E \subset E \cup E'$, et car le coût de calcul de s_E est bien inférieur au coût de calcul maximum

Coût de calcul maximum c_{max}	**Meilleur sous-ensemble de descripteurs**
$0 \leq c_{max} < c_1$	{ AudioXcorr 2, AudioPower }
$c_1 \leq c_{max} < c_2$	{ AudioPower , AudioSpectrumCentroid 2, AudioSpectrumSpread 4, AudioSpectrumSlope 2, AudioSpectrumVariation 1 }
$c_2 \leq c_{max} \leq c_4$	{ AudioSpectrumSpread 4, AudioFilterBankCentroid 2, AudioMFCC 6 }

TAB. 4.6 – Meilleurs sous-ensembles de descripteurs (en terme d'erreur de classification) trouvés avec l'évaluateur de sous-ensembles de descripteurs *wrapper* et la recherche pas à pas utilisés avec la taxonomie de recherche, avec une *gaussienne multi-dimensionnelle modélisant chaque classe*. Pour les descripteurs multi-dimensionnels, la dimension est donnée en comptant à partir de 1.

associé à $E \cup E'$.

Nous attribuons ce phénomène à la difficulté qu'ont les algorithmes de sélection à se confronter à des espaces de recherche immenses. Par définition de la relation d'inclusion, l'optimum global sur l'ensemble $E \cup E'$ est au mieux plus performant, au pire aussi performant que l'optimum global sur l'ensemble E. Mais *la recherche d'un optimum parmi $2^{Card(E \cup E')}$ possibilités constitue un problème d'optimisation autrement plus difficile que celle d'un optimum parmi $2^{Card(E)}$ possibilités*.

Par ailleurs, dans le cas où le sous-ensemble de descripteurs trouvé par sélection sur un ensemble E_i correspond à une performance de classification moins élevée que le résultat de la sélection dans un autre ensemble E_j (pas forcément inclus dans E_i) de coût maximum de calcul associé plus faible que celui de E_i, notre algorithme retient le résultat de la sélection dans E_j pour le coût maximum de calcul associé à E_j.

4.6 Aire sous la courbe ROC

Comme nous l'avons vu dans la partie 2.8, l'aire sous la courbe ROC est un critère également utilisé pour estimer les performances d'un classifieur. Nous avons voulu étudier si les performances mesurées suivant ce critère étaient similaires à celles que nous avons obtenu par *cross-validation*. Les courbes 4.6 page ci-contre et 4.7 page 40 montrent l'évolution de l'aire sous la courbe ROC en fonction du nombre de descripteurs (pris dans l'ordre donné par l'algorithme de sélection) pour les méthodes *filter* que nous avons utilisées.

Tout d'abord, il convient de noter que l'allure générale de la courbe est en opposition avec celle que nous avons obtenu en prenant en compte l'erreur de

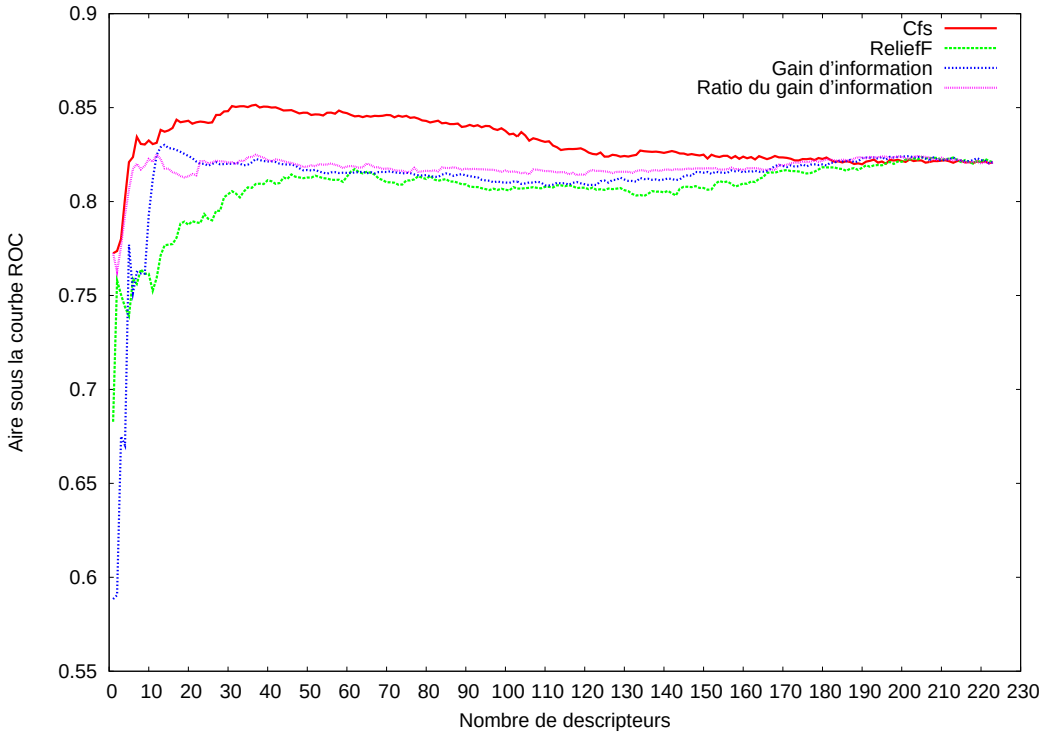

FIG. 4.6 – Influence du nombre de descripteurs retenus sur l'aire sous la courbe ROC pour un *classifieur bayésien naïf*.

classification (figures 4.1 et 4.2). Pour tous les algorithmes évalués, le meilleur résultat (c'est-à-dire le sous-ensemble de descripteurs maximisant l'aire sous la courbe ROC) est obtenu avec un nombre de descripteurs plus grand que lorsque l'erreur de classification est choisie comme critère d'évaluation. Par ailleurs, les ensembles formés à partir des dix premiers descripteurs sont les moins performants au sens de l'aire sous la courbe ROC, alors que ce sont parmi les plus performants pour le critère de l'erreur de classification.

Ces résultats, ainsi que certaines publications ([15], [5]) découvertes à la fin de nos travaux et affirmant la supériorité de l'aire sous la courbe ROC sur l'erreur de classification comme critère pour évaluer la performance de détection d'un classifieur, suggèrent la nécessité d'une étude plus poussée sur la question du critère d'évaluation.

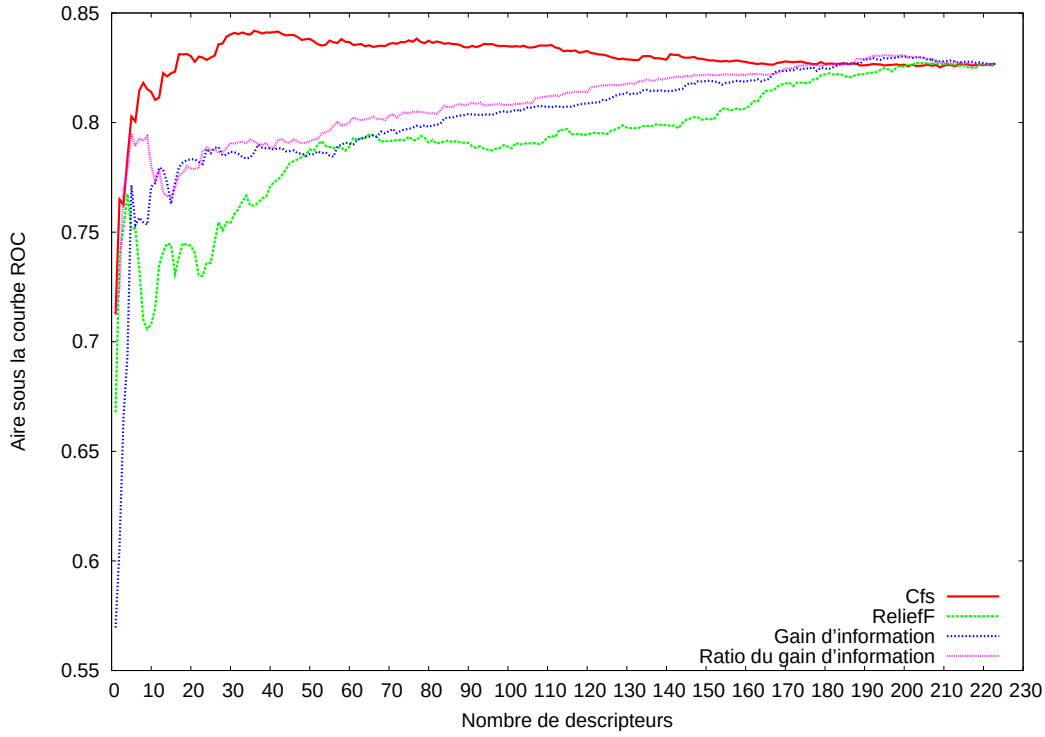

FIG. 4.7 – Influence du nombre de descripteurs retenus sur l'aire sous la courbe ROC avec une *gaussienne multi-dimensionnelle modélisant chaque classe*.

CHAPITRE 5

Discussion et directions futures

Sommaire

5.1 Intégration du modèle dans un environnement temps-réel

Les systèmes audio temps-réel tels que Max/MSP ou PureData reposent sur un paradigme visuel basé sur des boîtes représentant un calcul, et sur des flèches symbolisant le passage des données d'une boîte à une autre. Le calcul des descripteurs audio à partir du signal est un élément des *patchs* utilisés dans l'audio temps-réel.

Notre modèle de sélection de descripteurs pseudo-optimale par rapport au coût de calcul des descripteurs trouve son application dans la structure d'un patch. En effet, celle-ci isole les différentes opérations nécessaires au calcul des descripteurs audio.

Une modélisation fine de la structure du calcul des descripteurs

La structure d'un *patch* permet d'isoler les différentes parties du calcul des descripteurs pour estimer leurs temps d'exécution, et contient dans sa structure des informations pouvant faire espérer les bénéfices suivants :

– Une *estimation très fine du temps de calcul des descripteurs*, comparé à la modélisation naïve que nous avons utilisé.

– Une modélisation plus proche de la réalité de la structure du calcul des descripteurs, qui a pour conséquence d'augmenter le nombre de groupes, ce qui rend les paliers de la fonction $erreur = f(\text{cout de calcul maximum})$ moins larges, soit *des optima de performance de classification plus fins*.
– Une *automatisation* du processus de sélection des descripteurs en fonction des exemples d'entraînement, du classifieur, et de l'implémentation du calcul des descripteurs.

Prise en compte du temps de classification

Le calcul des descripteurs est un élément contribuant au délai lié au système de suivi de partition, et la classification d'un vecteur de descripteurs en est un autre, que nous avons volontairement mis de côté lors de notre étude. L'intégration d'un algorithme de sélection de descripteurs à un *patch* temps-réel rend envisageable de *mesurer le temps que le classifieur met à classifier un vecteur de descripteurs*. Ce temps dépend du sous-ensemble de descripteurs dans lequel les exemples sont décrits, et peut donc être *incorporé comme critère pendant le processus de sélection* dans un algorithme évaluant des sous-ensembles de descripteurs. La manière de combiner le temps de classification avec les autres "mérites" possibles est un problème ouvert, bien que nous possédions quelques idées sur le sujet.

5.2 Création de descripteurs et coût de calcul

Une des conséquences logiques de ce travail est d'aborder la question du coût de calcul dans les problèmes de création de descripteurs (dont un aperçu a été donné dans la partie 2.2 page 11). De même qu'il est intéressant de pouvoir sélectionner des descripteurs offrant un bon compromis entre performance et temps de calcul, il semble intéressant de créer des descripteurs à la fois performants et peu nécessiteux en calcul. L'utilisation d'une fonction de mérite combinant un coût lié au coût de calcul des descripteurs et la performance estimée pour un classifieur donné (tel que nous l'avons brièvement expérimenté, voir 3.3 page 22) semble indiquée.

5.3 Descripteurs globaux et temps-réel

Comme nous l'avons vu dans la partie 1.4, les descripteurs audio globaux (tels que définis dans [23]) ne sont pas utilisables pour l'entraînement d'un modèle d'observation. Cependant, il semble intuitivement intéressant de décrire les exemples d'entraînement à la fois selon la valeur d'un descripteur instantané et selon un *descripteur contextuel* dérivé, combinant les valeurs instantanées du descripteur sur des fenêtres passées. La différence entre la valeur du descripteur instantanée et celle du descripteur contextuel (qui est également instantané) peut constituer un descripteur rendant les classifieurs plus performant pour la segmentation en temps-réel, dans la mesure où nous

maîtrisons mal l'influence des conditions de scène (placement des micros, niveaux, *etc.*) sur le système de suivi. Il est probable que les différences entre les échantillons servant à construire la base d'entraînement et le signal que reçoit le système dans le cadre d'une utilisation scénique dégradent la qualité de la segmentation pendant l'utilisation sur scène.

La combinaison des valeurs passées peut par exemple se faire par une moyenne sur les n dernières fenêtres, ce qui pose le problème du choix de n. Un oubli exponentiel est également envisageable, et là aussi il faut choisir un paramétrage.

5.4 Techniques de sélection et coût de calcul

Taxonomie de recherche et algorithmes qui ordonnent les descripteurs

La première approche que nous avons abordé (partie 3.4) pour tenter d'apporter des éléments de réponse au problème de l'optimalité par rapport au coût de calcul est naïve dans la mesure où les algorithmes ordonnant les descripteurs ne donnent pas de garantie sur la façon dont croît le coût de calcul lorsque nous formons des sous-ensembles de descripteurs en incluant les n premiers descripteurs selon l'ordre donné par la sélection. Ce problème est mis en avant par la courbe 4.3 page 32.

Cependant, il existe une variante envisageable de la première approche permettant d'utiliser une taxonomie de recherche. Elle consiste, pour chaque coût maximum de calcul, à ne retenir que les n premiers descripteurs selon l'ordre donné par l'algorithme de sélection parmi les descripteurs de l'ensemble de recherche correspondant. Cette technique devrait aboutir à des performances meilleures que la première approche expérimentée (car le choix des sous-ensembles de descripteurs dans les petites dimensions se fait parmi un plus grand nombre de candidats). Par contre, les performances devraient rester moins bonnes que la combinaison de l'évaluateur de sous-ensembles de descripteurs *wrapper* et de la recherche pas à pas que nous avons expérimenté (les *wrappers* aboutissent en général à de meilleurs résultats car ils optimisent un bon critère par rapport au problème de la sélection de descripteurs : l'erreur de classification).

Tirer partie de l'inclusion des groupes

D'une part, l'utilisation d'une taxonomie de recherche basée sur le processus de calcul des descripteurs audio telle que nous l'avons présenté (partie 3.5) et la réalisation de plusieurs sélections de descripteurs dans des groupes de decripteurs induit des *relations d'inclusion entre les ensembles de descripteurs dans lesquels les sélections sont effectuées*. Cette inclusion vient du fait que plusieurs représentations du signal peuvent être calculées à partir d'une même représentation de départ. Par exemple, ceci se produit dans le cas de l'ensemble E_0 qui est inclu dans E_1, E_2, E_3 et E_4, et dans celui de l'ensemble E_1, qui est inclut dans E_2, E_3 et E_4.

D'autre part, la réponse majeure au problème de la sélection de descripteurs réside dans la manière dont l'espace de recherche est parcouru. L'initialisation des algorithmes de recherche peut accélérer la sélection (dans le cas des recherches pas à pas), ou bien guider dès le début une recherche génétique dans des optima locaux lors de l'initialisation de la population. Quand $E_i \subset E_j$, le fait d'*utiliser les données issus d'une sélection effectuée dans E_i pour initialiser l'algorithme de recherche avant le parcours de l'espace de recherche lié à E_j* est probablement intéressant du point de vue de la performance de sélection (*i.e.* de sa capacité à trouver un sous-ensemble de descripteurs optimal selon un critère, l'erreur de classification dans notre cas).

Améliorer les performances avec d'autres algorithmes de recherche

Dans le but d'obtenir plusieurs sous-ensembles de descripteurs associés à une performance pseudo-optimale pour un coût de calcul maximum donné, la deuxième approche que nous avons expérimenté est générique du point de vue de l'algorithme de sélection à utiliser. Nous avons obtenus les meilleurs résultats avec la combinaison de l'évaluateur de sous-ensembles de descripteurs *wrapper* et d'une recherche naïve de type *hill climbing*, mais il est probable que des sous-ensembles de descripteurs aboutissant à une meilleure performance du système puissent être trouvés au sein des groupes associés à la taxonomie de recherche.

La combinaison de l'évaluateur *wrapper* avec une recherche génétique, par exemple, semble tout-à-fait indiquée mais est d'une part difficile à paramétrer, et d'autre part problématique étant donné que notre modèle effectue plusieurs sélections. Contrairement à la recherche génétique, l'algorithme *race search* [21] a l'avantage (non négligeable pour le problème de l'optimalité par rapport à un coût de calcul) de fournir un ordre sur les descripteurs. De plus, il semble ne pas avoir certains des défauts de la recherche naïve de type *hill climbing*, notamment une certaine tendance intrinsèque à se perdre dans des minima locaux.

5.5 Sélection de descripteurs et sur-apprentissage

Les algorithmes de sélection de descripteurs supervisés sont exposés au sur-apprentissage. Cette exposition est d'autant plus forte que le nombre d'exemples correctement étiquetés est faible, ce qui est fréquemment le cas dans le cadre d'utilisation du système de suivi de partition. Ceci s'explique par le temps que requiert l'annotation manuelle ainsi que par la nécessité de disposer d'enregistrements des pistes correspondant aux voix non synthétiques, effectués lors de répétitions ou de concerts précédents.

Nous envisageons les directions de recherche suivantes (qui sont complémentaires et pas concurrentes) dans le but de maîtriser le sur-apprentissage dans le modèle d'observation :

Nettoyage de données (*data cleaning*)

Il s'agit d'un domaine de recherche très actif ces dernières années. Son principe est de *retirer de la base d'entraînement des vecteurs étiquetés jugés comme étant de mauvais exemples*. Ceci pose le problème de la détection automatique de tels vecteurs dans la base, et donc des critères selon lesquels les trier. Étant donné les remarques déjà faites sur la précision de l'annotation manuelle des échantillons musicaux (voir 4.1 page 27), l'utilisation de ce genre de techniques paraît avoir beaucoup de sens dans le cadre du suivi automatique de partition.

Entraîner le classifieur sur un sous-ensemble d'exemples

La nature des problèmes de segmentation que l'on rencontre dans les applications musicales donne lieu à une sur-représentation de certaines classes dans la base d'entraînement. Pour la détection d'*onsets*, nous disposons de beaucoup plus d'exemples de la classe "Attaque" que d'exemples de la classe "Autres" (voir 4.1 page 27). Aussi, il semble intéressant d'étudier l'impact des techniques permettant de filtrer les exemples de manière à équilibrer la représentation des différentes classes dans la base à partir de laquelle s'effectue la sélection. Dans cette perspective, le *Balanced Error Rate* ou l'aire sous la courbe ROC (voir 2.8 page 17) peuvent être des critères à considérer, dans la mesure où ils prennent en compte l'erreur sur les différentes classes. Il convient également de mener une réflexion sur une possible différence de criticité entre les faux négatifs et les faux positifs dans le cadre du suivi de partition.

Conclusion

Durant ce stage, nous avons abordé la problématique de la sélection de descripteurs dans le cadre des modèles d'observation audio temps-réel tels que celui du suiveur de partition de l'Ircam. Après avoir étudié les différentes approches algorithmiques classiques en matière de sélection de descripteurs, nous avons réalisé les difficultés que pose leur utilisation pour des applications où le temps de calcul est un paramètre critique.

Nous avons mis en évidence l'intérêt d'intégrer le modèle calculatoire des descripteurs audio au processus de sélection des descripteurs, et proposé deux approches permettant d'utiliser les algorithmes usuels de sélection de descripteurs pour les applications temps-réel. Nous avons ensuite évalué ces approches. Nos résultats montrent l'intérêt de la deuxième méthode proposée, qui se base sur une taxonomie de recherche dérivée du modèle calculatoire des descripteurs et sur un parcours de l'espace de recherche.

Enfin, nous avons proposé diverses pistes et idées accumulées au fil de notre travail pouvant constituer des points de départ pour de futures recherches dans le domaine de la classification audio.

Ce travail ouvre la perspective d'une sélection de descripteurs adaptée aux contraintes des applications utilisant des algorithmes de classification. Cette idée dépasse le cadre des modèles d'observation temps-réel, et nous pouvons espérer qu'elle fera son chemin dans d'autres domaines, permettant d'améliorer les performances des systèmes tout en réduisant la quantité de resources nécessaire à leur fonctionnement.

Bibliographie

[1] Avrim Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2) :245–271, 1997.

[2] Arshia Cont. Improvement of observation modeling for score following. Dea atiam, University of Paris 6, IRCAM, Paris, 2004.

[3] Arshia Cont. Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical hmms. In *IEEE International Conference in Acoustics and Speech Signal Processing (ICASSP)*. Toulouse, May 2006.

[4] Arshia Cont, Diemo Schwarz, and Norbert Schnell. Training ircam's score follower. In *IEEE International Conference on Acoustics and Speech Signal Processing (ICASSP)*. Philadelphia, March 2005.

[5] Corinna Cortes and Mehryar Mohri. Auc optimization vs. error rate minimization, 2003.

[6] R. B. Dannenberg. An on-line algorithm for real-time accompaniment. In *Proc. of the 1984 Int. Computer Music Conf.*, pages 193–198. Computer Music Association, June 1984.

[7] Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the em algorithm, 1977.

[8] Bruce Draper, Carol Kaito, and José Bins. Iterative relief, 2003.

[9] S. Essid, P. Leveau, G. Richard, L. Daudet, and B. David. On the usefulness of differentiated transient/steady-state processing in machine recognition of musical instruments. In *Proc. AES 118th Convention, Barcelona, Spain*, May 2005.

[10] Slim Essid. *Classification automatique de signaux audio-fréquences : reconnaissance des instruments de musique*. PhD thesis, ENST, 2006.

[11] David E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.

[12] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3 :1157–1182, 2003.

[13] Isabelle Guyon, Steve Gunn, Asa Ben Hur, and Gideon Dror. Result analysis of the nips 2003 feature selection challenge. 2004.

[14] M. A. Hall. *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, University of Waikato, Hamilton, New Zealand, 1998.

[15] Jin Huang and Charles X. Ling. Using auc and accuracy in evaluating learning algorithms, 2003.

[16] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, San Mateo, 1995. Morgan Kaufmann.

[17] Kenji Kira and Larry A. Rendell. A practical approach to feature selection. In *ML92 : Proceedings of the ninth international workshop on Machine learning*, pages 249–256, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[18] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1-2) :273–324, 1997.

[19] Igor Kononenko. Estimating attributes : Analysis and extensions of RELIEF. In *European Conference on Machine Learning*, pages 171–182, 1994.

[20] P. Leveau, L. Daudet, and G. Richard. Methodology and tools for the evaluation of automatic onset detection algorithms in music. In *In Proc. ISMIR 2004, Barcelona*, 2004.

[21] Andrew W. Moore and Mary S. Lee. Efficient algorithms for minimizing cross validation error. In *International Conference on Machine Learning*, pages 190–198, 1994.

[22] Pavel Paclik, Robert P.W. Duin, Geert M.P. van Kempen, and Reinhard Kohlus. On feature selection with measurement cost and grouped features, 2002.

[23] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the cuidado project. Technical report, 2004. CUIDADO I.S.T. Project Report.

[24] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. pages 267–296, 1990.

[25] C. Raphael. Automatic segmentation of acoustic musical signals using hidden markov models. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, page 21(4), 1999.

[26] Marko Robnik-Sikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Mach. Learn.*, 53(1-2) :23–69, 2003.

[27] Diemo Schwarz, Nicola Orio, and Norbert Schnell. Robust polyphonic midi score following with hidden markov models. In *Proceedings of the International Computer Music Conference (ICMC)*, November 1-6 2004. Miami, Florida.

[28] Michèle Sebag. Sélection d'attributs, 2005. Transparents de cours.

[29] Matthew G. Smith and Larry Bull. Feature construction and selection using genetic programming and a genetic algorithm, 2003. EuroGP 2003, LNCS 2610.

[30] Haleh Vafaie and Kenneth De Jong. Genetic algorithms as a tool for restructuring feature space representations, 1995.

[31] Barry Vercoe. The synthetic performer in the context of live performance. In *ICMC proceedings*.

[32] Ian H. Witten and Eibe Frank. *Data Mining : Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2 edition, 2005.

[33] Aymeric ZILS. *Extraction de descripteurs musicaux : une approche évolutionniste*. PhD thesis, Université Paris 6, Septembre 2004.

Annexes

Nom du descripteur	Nombre de dimensions
cuidado :AudioXcorr	12
cuidado :AudioZcr	1
mpeg7 :AudioPower	1
	Total : 14

TAB. 5.1 – Les 3 descripteurs temporels ($E_{temporel}$).

Nom du descripteur	Nombre de dimensions
mpeg7 :AudioSpectrumCentroid	6
mpeg7 :AudioSpectrumSpread	6
cuidado :AudioSpectrumSkewness	6
cuidado :AudioSpectrumKurtosis	6
cuidado :AudioSpectrumSlope	6
cuidado :AudioSpectrumDecrease	1
cuidado :AudioSpectrumRolloff	1
cuidado :AudioSpectrumVariation	3
	Total : 35

TAB. 5.2 – Les 8 descripteurs spectraux ($E_{spectral}$).

Nom du descripteur	Nombre de dimensions
cuidado :AudioFilterbankCentroid	6
cuidado :AudioFilterbankSpread	6
cuidado :AudioFilterbankSkewness	6
cuidado :AudioFilterbankKurtosis	6
cuidado :AudioFilterbankSlope	6
cuidado :AudioFilterbankDecrease	1
cuidado :AudioFilterbankRolloff	1
cuidado :AudioFilterbankVariation	3
cuidado :AudioMFCC	12
cuidado :AudioDeltaMFCC	12
cuidado :AudioDeltaDeltaMFCC	12
mpeg7 :AudioSpectrumFlatness	4
cuidado :AudioSpectrumCrest	4
cuidado :AudioLoudness	1
cuidado :AudioSharpness	1
cuidado :AudioSpread	1
cuidado :AudioRelativeSpecificLoudness	24
	Total : 106

TAB. 5.3 – Les 17 descripteurs perceptifs ($E_{perceptif}$).

Nom du descripteur	Nombre de dimensions
cuidado :AudioHarmonicPower	1
cuidado :AudioNoisePower	1
mpeg7 :AudioFundamentalFrequency	1
mpeg7 :AudioHarmonicity	1
cuidado :AudioInharmonicity	1
mpeg7 :HarmonicSpectralCentroid	6
mpeg7 :HarmonicSpectralSpread	6
HarmonicSpectralSkewness	6
cuidado :HarmonicSpectralKurtosis	6
cuidado :HarmonicSpectralSlope	6
mpeg7 :HarmonicSpectralVariation	3
mpeg7 :HarmonicSpectralDeviation	3
cuidado :HarmonicSpectralOERatio	3
cuidado :HarmonicSpectralTristimulus1	3
cuidado :HarmonicSpectralTristimulus2	3
cuidado :HarmonicSpectralTristimulus3	3
	Total : 53

TAB. 5.4 – Les 16 descripteurs harmoniques ($E_{harmonique}$).

Nom du descripteur	Nombre de dimensions
AudioFilterbankDeviation	3
AudioFilterbankOERatio	3
cuidado :AudioFilterbankTristimulus1	3
cuidado :AudioFilterbankTristimulus2	3
cuidado :AudioFilterbankTristimulus3	3
	Total : 15

TAB. 5.5 – Les 5 descripteurs harmonico-perceptifs ($E_{harmonico-perceptif}$).

Table des figures

Liste des tableaux