Une *Approche généralisée avec EDS*

Rapport de Stage Master ATIAM

Université Paris 6

Frédéric ROSKAM

Sony CSL Paris, 6 rue Amyot 75005 Paris

Septembre 2006

Résumé

L'explosion des contenus numériques audio crée corrélativement une demande accrue pour les systèmes de classification et d'indexation automatiques. L'équipe musique de SONY CSL a développé depuis plusieurs années une approche originale à la classification audio automatique, fondée sur la programmation génétique: le système EDS.

Ce système a déjà trouvé des features (descripteurs bas-niveaux audio) meilleures que celles de la littérature (en particulier MPEG-7), sur des problèmes de classification supervisée audio classiques. Ce stage a consisté à formaliser la comparaison de l'approche EDS avec d'autres formes d'approches. Plus précisément, le stage a comporté les phases suivantes:

- Familiarisation avec le système EDS en mode classification supervisée. Création de base de données de référence (training) sur des problèmes de classification de bruits urbains ou des classifications musicales de type "sens commun", tel que des sons percussifs, et expérimentation avec EDS pour produire des classifieurs.
- Mise en place d'un cadre pour comparer les performances du système à l'approche classique d'extraction d'information dans un sons.
- Enfin nous avons mis en évidence la supériorité de l'approche de manière quasi-systématique, mais avec un écart relativement faible.

Mots clés: Automatic features extraction, MPEG-7, Audio Information Retrieval, bag of frames, EDS

Remerciements

Je tiens à remercier toute l'équipe de SONY CSL pour leurs bons conseils, en particulier Anthony Beurivé, mais aussi Romain Campana pour sa culture musicale, Pierre Roy et bien sûr mon maître de stage François Pachet.

Table des matières

1 Introduction **5**
1.1 Contexte 5
1.2 MPEG-7 5
1.3 Extraction de features 6
1.4 Sélection d'attibuts 6
1.5 Apprentissage 7
1.6 Objectif et Plan 7

2 L'extraction automatique de descripteurs par traitement du signal **8**
2.1 Types de descripteurs 8
2.1.1 Descripteurs acoustiques et culturels 8
2.1.2 Niveaux de description 8
2.2 Vers une automatisation 9
2.3 Extractor Discovery System 10
2.3.1 Construction automatiques de fonctions de traitement du signal 10
2.3.2 Selection des fonctions 11
2.3.3 Création du modèle 11

3 Evaluation **12**
3.1 Justification 12
3.2 Difficultés 13
3.2.1 constitution des bases 13
3.2.2 Reference set 15
3.2.3 Sélection d'attributs 18
3.2.4 Learning 19
3.3 Protocole mis en place 20

4 Expériences **21**
4.1 Cas d'école : sinusoïde dans un bruit rose 21
4.1.1 Contexte et critère d'évaluation 21
4.1.2 Les bases 22
4.1.3 Sélection d'attributs 22
4.1.4 Apprentissage 22
4.1.5 Resultats 22
4.2 Dog Barks 24
4.2.1 Contexte et critère d'évaluation 24

4.2.2 Les bases 25
4.2.3 Resultats 25
4.3 Urban Noises 27
4.3.1 Contexte et critère d'évaluation 28
4.3.2 Les bases 28
4.3.3 Resultats 29

5 Plus loin, la musique populaire du 20ème siècle 34

6 Conclusion 35

Liste des tableaux

1 Configuration des bases d'apprentissage et de test 15
2 Résultats regression sinus et bruit rose 23
3 Matrice de confusion avec InfoGain + SVM pour les aboiements de chiens 27

Table des figures

1 MPEG-7 Audio Low Level Descriptors 16
2 Apperçu du spectre du signal, sinus + bruit rose (fréquences en abcisses) 21
3 Résultats de l'apprentissage (précision) - Aboiements 26
4 Résultats de l'apprentissage (précision) - Bus 30
5 Résultats de l'apprentissage (précision) - Cyclomoteurs 30
6 Résultats de l'apprentissage (précision) - Motos 30
7 Résultats de l'apprentissage (précision) - Oiseaux 31
8 Résultats de l'apprentissage (précision) - Voitures 31
9 Résultats de l'apprentissage (précision) - Voix 31
10 Résultats de l'apprentissage (précision) - Urbain 32

1 Introduction

1.1 Contexte

La récente explosion de la quantité de données multimédia accessible sur Internet a créé de nouveaux intérêts dans le domaine de la distribution de contenu, aussi bien pour les utilisateurs qui recherchent des moyens pratiques de gérer et d'utiliser leurs collections personnelles, que pour les distributeurs, qui cherchent à donner accès au plus grand nombre à ce qui les intéresse. Cette gigantesque quantité d'information ne pouvant être gérée directement, l'utilisation de méta-données, c'est-à-dire de données qui décrivent elles-mêmes les données recherchées, devient incontournable. Des recherches actuelles ont pour but d'extraire des descripteurs pertinents dans le cadre des applications visées. C'est le cas entre autres des descripteurs audio, pour l'indexation des bandes-sons de cinéma par exemple [4], ou la reconnaissance de bruits urbain [3].

1.2 MPEG-7

Une norme fait de plus en plus parler d'elle : la norme MPEG-7[1], formellement nommée Multimedia Content Description Interface. Contrairement à la norme MPEG-4 elle ne décrit pas une forme de compression de l'information audio et/ou vidéo, mais plutôt normalise des descriptions pour faciliter l'indexation et l'exploitation de documents multimédia. Cette norme permet de stocker des métadonnées qui peuvent être liées au temps. Ces données peuvent ainsi être transférées en 'streaming', permettant ainsi à n'importe quel appareil à l'autre bout du réseau d'utiliser en temps réel ce supplément d'information.
Pour ce qui concerne la partie audio, un jeu de 'descripteurs bas-niveaux' (Low Level Descriptors) divers couvrant les domaines temporels et spectraux, et des outils de description plus haut-niveaux sont précisés. Mais sous-entendu dans cette normalisation audio qui servira entres autres de cadre à la reconnaissance, il y a le principe d'une analyse par 'paquet de trame' ou 'bag of frames'. Cette approche, la plus répendue aujourd'hui pour analyser un signal audio, consiste à saucisonner un signal en fenêtre de taille constante, loin du fonctionnement plus discriminant ou contrastant ou encore intégrateur d'information de l'oreille humaine. De plus, la notion de temporalité et de lien entre les éléments sonore est le plus souvent inutilisé ou perdu. On peut également parler du défaut d'avoir une résolution fréquentielle en concurrence avec la résolution temporelle, conduisant quoiqu'il arrive à faire des compromis. Ce-

[1] référence : http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm

pendant dans notre étude nous nous placerons toujours dans ce contexte.

1.3 Extraction de features

Les travaux menés par Aymeric Zils [17] ont porté sur l'extraction automatiques de descripteurs par traitement du signal qui touche à trois domaine : le domaine perceptif dans le choix de descripteurs pertinents pour un problème donné. Le domaine du traitement du signal dans la phase de recherche de d'extraction de modèles descriptifs à partir de ces signaux. Et finalement le domain informatique avec une phase d'intégration de ces descripteurs dans des applications de traitement sonore. Le système créé, EDS pour Extractor Discovery System, est capable de construire automatiquement des fonctions performantes de modèles descriptifs constitués de fonctions générale (mathématiques ou de traitement du signal). Nous parlerons d'EDS un peu plus en détails dans la partie 2.
On peut remarquer qu'aujourd'hui très peu de recherches vont dans cette direction, une étude des derniers papiers publiés dans le domaine de la reconnaissance audio montre une prépondérance de l'utilisation de nouveaux algorithmes d'apprentissage ou de sélection d'attributs (voir 1.4). Ces algorithmes, plus particulièrement depuis les travaux de Vapnik, ont bien sûr fait grandement progressé l'efficacité des outils de reconnaissance, mais peut-on négliger l'extraction de l'information utile dans le signal ? Pour caricaturer : les MFCC (Mel-Frequency Cepstrum Coefficients) ou le ZCR (Zéro Crossing Rate) peuvent-ils tout exprimer ?

1.4 Sélection d'attibuts

La sélection d'attributs est un outil d'optimisation de classification largement étudié dans les années 80 et 90 (e.g. [12]) qui a continué d'être un sujet de recherche dans le domaine de l'apprentissage. Il n'y a toujours pas de consensus sur la meilleure méthode, sachant que de toute façon ces sélections n'ont pas toujours le même objectif. La sélection d'attributs peut avoir principalement deux intérêts : d'une part lorsqu'un problème complexe tel que le classification de sons [9] est rencontré, il est difficile de savoir quels sont les descripteurs pertinents pour classifier. Devant un choix d'un dizaine, centaine ou millier de 'features', il devient indispensable de sélectionner ceux qui sont efficaces mais aussi orthogonaux (non redondants en quelque sorte). D'autre part cela peut devenir une nécessité dans le cas de calculs lourds sur de grandes bases de données, de prendre en compte la notion de temps de calcul. Un point supplémentaire qui va nous intéresser dans la sélection d'attributs est le

fait qu'elle permet de comparer deux jeux d'attributs (descripteurs bas-niveaux dans notre cas) qui n'ont pas initialement la même taille puisque cette taille est réduite de manière équivalente.

1.5 Apprentissage

Domaine sur lequel beaucoup de chercheurs se concentrent, l'apprentissage a pendant longtemps opposé les chercheurs où chacun tentait de prouver la suprématie de sa méthode. Aujourd'hui un consensus s'est établi autour de l'idée qu'il n'y a pas de meilleure méthode. Chacune est plus ou moins adaptée à un problème posé. Par contre sur le plan de la méthodologie, comparer les méthodes est un exercice délicat qui demande à être conduit avec soin. La tendance va à souligner l'importance de construire des modèles parcimonieux quelle que soit la méthode utilisée. Néanmoins certains algorithmes 'hybrides' tels que le 'bagging' ou le 'boosting' (dont l'AdaBoost que nous allons utiliser) ne suivent pas cette règle.

1.6 Objectif et Plan

Dans la suite de ce rapport de stage nous allons présenter quelques concepts d'extraction automatique de descripteurs puis le système EDS. Ensuite nous rentrerons dans le coeur de notre problématique : comment prouver la validité et l'utilité d'une nouvelle approche ? A quoi se comparer ? Nous tenterons d'y répondre dans la partie suivante en mettant progressivement en place un protocole de test. Enfin nous appliquerons ce protocole à deux expériences intéressantes et complémentaires de reconnaissance de sons : les bruits urbains et les aboiements de chien. Le temps ne nous a pas permis d'appliquer ce même protocole non pas cette fois pour comparer EDS, mais pour étudier la contextualisation des attributs dans la musique du 20ième siècle.

2 L'extraction automatique de descripteurs par traitement du signal

La volonté d'extraire automatiquement des descripteurs des signaux audio ne date pas d'hier. De nombreuses tentatives ont été faites par exemple dans le domaine de la musique pour tenter d'identifier et d'extraire des descripteurs utiles. L'un des descripteurs les plus simples à appréhender est le tempo : les méthodes les plus simples utilisent l'autocorrélation du signal audio pour la détection de la mesure, ou l'excitation d'oscillateurs non-linéaires pour le suivi du tempo ('beat tracking'). [Scheirer, 1998] propose une méthode complète de suivi de tempo, qui donne de très bons résultats sur des signaux de musique populaire au rythme prononcé. Le timbre - ensemble des caractéristiques acoustiques qui permettent de différencier deux sons indépendamment de leur hauteur et de leur volume - reste quant à lui plus compliquer à cerner et à extraire [1]. On peut aussi rechercher à détecter la présence de voix (qui trouve de nombreuses applications dans le domaine de l'audio-visuel) ou encore la nuisance qu'un son peut provoquer chez les personnes. Le problème de la mesure automatique de bruits urbains fait justement l'objet d'actives recherches, et l'une des conlusions de B. Defreville est que cette mesure passe par l'identification des sources sonores.

2.1 Types de descripteurs

Les descripteurs audio sont extrêmement variées, et on peut les classer suivant divers critères : leur origine, leur niveau d'abstraction, leur subjectivité, etc...

2.1.1 Descripteurs acoustiques et culturels

Les descripteurs culturels sont des données liées au contexte dans lequel a été créé un son ou un extrait musical. Ils sont en général textuels (nom de l'artiste, titre, etc...), et extractibles par recherche dans des bases de données sur internet. Cependant, ces descripteurs ne donnent aucune information sur les caractéristiques sonores intrinsèques à la musique. Ici nous nous intéressons uniquement aux descripteurs extraits à partir de l'étude de signaux acoustiques.

2.1.2 Niveaux de description

L'ensemble des descripteurs acoustiques peut être divisé en plusieurs niveaux de description, mesurant l'abstraction des propriétés qu'ils évaluent par rapport à celles

du signal acoustique brut :

- les descripteurs **bas-niveaux** expriment des propriétés sonores significatives directement observables ou calculables à partir du signal audio, mais qui n'ont pas forcément de signification évidente, comme par exemple les caractéristiques des enveloppes d'énergie ou du spectre du signal ;

- les descripteurs de **haut niveaux** expriment des propriétés qui ont une signification plus évidente et compréhensible par les auditeurs, mais nécessitent le calcul de modèles acoustiques plus complexes.

2.2 Vers une automatisation

Il n'existe pas de méthode universelle pour extraire des descripteurs audio. Dans sa thèse, Zils présente deux types de techniques d'extraction automatique de descripteurs musicaux : empirique et paramétrique.
Dans le cas d'une méthode empirique , on met en oeuvre une chaîne de traitements du signal aboutissant à un système d'extraction complexe (extracteur de tempo, modèle de timbre global, transcription polyphonique. Leur difficulté réside à la fois dans le choix de traitements efficaces des signaux, et dans le long travail de mise au point et d'ajustage des paramètres.
Dans le cas d'une méthode paramétrique, utilisées pour la plupart des études de timbre, on utilise toutes sortes de caractéristiques temporelles, spectrales ou cepstrales, ainsi que leurs statistiques (moments). Les données à traiter sont ensuite projetées dans l'espace des caractéristiques pour être analysées (extraction de tempo, classification de sons instrumentaux, classification de titres musicaux, détection de voix). La difficulté de ces méthodes est de trouver un espace de caractéristiques pertinent, qui décrit suffisamment bien le problème posé.
Pour cela, on utilise habituellement les fonctions caractéristiques comme celles proposées dans le framework audio de la norme MPEG-7. Ces fonctions sont puissantes et robustes, dans la mesure où elles correspondent à des propriétés générales des signaux audio. Cependant, elles peuvent être mal adaptées pour modéliser des problèmes descriptifs plus spécifiques comme on va le voir plus loin.

2.3 Extractor Discovery System

L'idée clé derrière EDS est de substituer aux descripteurs bas-niveaux basiques des combinaisons d'opérateurs mathématiques et de traitement du signal. L'architecture globale d'EDS est composée de deux parties : modélisation du descripteur et synthèse de l'extracteur.
La modélisation du descripteur est en réalité le vrai coeur d'EDS, elle consiste à chercher automatiquement un jeu de fonctions caractéristiques pertinentes pour obtenir un modèle optimal pour le descripteur qui combine ces fonctions. La recherche dans l'espace des fonctions est basée sur de la programmation génétique. Pour chaque fonction créée par croisement ou mutation est calculée un indice de 'fitness' qui représente 'à quel point cette fonction extrait bien tel descripteur' sur la base de test d'EDS. L'espace des fonctions étant potentiellement infini, une dose d'heuristique a été ajoutée pour par exemple simplifier des expressions.
Nous allons décrire ci-dessous les 3 principaux ingrédients du système EDS : la génération automatique de fonctions de traitement du signal, la sélection des meilleures fonctions pour un descripteur donné, et la combinaison de ces fonctions afin de créer un modèle.

2.3.1 Construction automatiques de fonctions de traitement du signal

Les fonctions sont représentées dans EDS comme une combinaison d'opérateurs simples applicables sur n'importe quel signal audio. Il y a par exemple des filtres passe-bas, un maximum, une corrélation, un centroid, un banc de filtre paramétrable, un fenêtrage réglable...
L'une des propriétés clés introduites dans EDS et bien connue du domaine de la programmation génétique est le **typage**. EDS s'appuie sur 3 types de données : la donnée de type temps, de type fréquence, et de type amplitude. N'importe quelle fonction peut en général être appliquée sur une seule valeur, un vecteur, ou une matrice. Et avec jusqu'à 2 paramètres supplémentaires. Enfin l'heuristique intervient au moment de la génération d'un nouvelle population de fonctions : en favorisant ou défavorisant des configurations précises d'opérateurs, en interdisant des mauvaises combinaisons (comme deux filtres passe-bas l'un à la suite de l'autre), en bornant les valeurs que peuvent prendre les paramètres des fonctions, et enfin en simplifiant certaines écritures inutiles.

2.3.2 Selection des fonctions

Une fois que le système a construit des fonctions avec les types demandés, elles sont appliquées sur les bases d'apprentissage. Tout le travail ensuite est de sélectionner les fonctions qui permettent d'extraire au mieux l'information, ceci étant réalisé par plusieurs critères au choix : critère de Fisher, taux de reconnaissance après apprentissage par la méthode des kNN, gain d'information, critère de corrélation de Pearson pour la régression...etc.
Les n meilleures fonctions d'après de critère choisi sont gardées et les autres sont supprimées. L'algorithme de recherche génétique crée alors une nouvelle génération en appliquant des transformations comme 'mutation', 'variation constante d'un des paramètres' et 'cross-over'.
On peut faire deux remarques : tout d'abord il n'y a pas de critère d'arrêt. Ensuite chaque fonction est évaluée individuellement, conduisant souvent à des minimums locaux (voir conclusion).

2.3.3 Création du modèle

Une fois phase de recherche génétique stoppée, il reste à choisir à la main les k meilleures fonctions et à construire un modèle du descripteur. EDS est capable en interne de réaliser un apprentissage avec les méthodes kNN et GMM. Cependant nous n'utiliserons pas cette fonctionnalité lors de notre étude, nous exporterons à la place les données pour la bibliothèque Weka.

3 Evaluation

Pour évaluer et pouvoir tenir un discours valable sur l'intérêt de l'approche EDS nous nous proposons de mettre en place une approche générale des problèmes de reconnaissance audio. Nous allons définir premièrement un cadre, puis nous expliciterons une démarche de référence applicable mécaniquement à différentes expériences. Cette démarche étant appliquée à la fois aux extracteurs issus de la recherche génétique d'EDS, et aux extracteurs de référence que nous préciserons et justifierons également.

3.1 Justification

Comparer un système comme EDS à ce qui correspond à l'approche standard n'est pas trivial. Les problèmes de reconnaissance sont d'une part extrêmement variés, et d'autre part mesurer et comparer la performance de ces systèmes alimente un débat permanent dans la communauté.

Premièrement nous ne pouvons pas nous contenter d'une seule expérience pour prouver ou infirmer l'utilité de l'approche EDS. D'où le choix d'inclure quelques problèmes classiques de reconnaissance. Les sons percussifs (enregistrement de pandeiro par Sergio Krakowski, http://www.pandeiro.com/sergio.php) et harmoniques avec des accords de guitare (Giordano Cabral [2]). A cela nous ajoutons deux autres problèmes originaux qui font l'objet d'études en ce moment même : la reconnaissance d'aboiements de chiens dans différentes situations citedogbarks et l'identification de bruits urbains [3]. Je ne me suis occupé que de ces deux derniers problèmes, et ce sont donc ceux qui seront présentés. Un premier exemple choisi pour montrer une bonne application d'EDS est de plus introduit.

Mais finalement à quoi ou à qui peut-on se comparer ? Qu'est-ce que l'approche standard, et peut-on parler d'opérateurs du commerce ? En parcourant les publications dans le domaine de l'extraction d'information audio, on retrouve bien je pense une approche classique qui combine deux aspects. L'aspect "bag of frames" consiste à réaliser un fenêtrage du signal en segments fixes de 10 à 100 millisecondes puis à appliquer un seul ou un jeu d'opérateurs sur chacun de ces segments. L'autre aspect est le choix d'opérateurs qui se restreint dans la plupart des cas à un ensemble maintenant établi d'une vingtaine. J.J. Aucouturier a réalisé une critique dans sa thèse [1] de cette approche typiquement utilisée dans la communauté MIR (Music Information Retrieval) pour définir le timbre. Il en ressort entre autres la présence

supposée d'un "plafond de verre" (glass ceilling) à 70% de précison. De nombreuses études ont montré l'intérêt des Mel-Frequency Cepstrum Coefficients (MFCC) [5] mais concédent que les détails temporels tels que l'attaque, ignorés par l'approche "bag of frames", sont la clé pour reconnaître un instrument.
Néanmoins nous nous comparons ici à cette approche que l'on pourrait qualifier de naïve puisqu'elle est choisie par de nombreux chercheurs par défaut.
D'autres chercheurs s'écartent (heureusement) de ce cadre et développent des algorithmes ad-hoc. On peut citer par exemple le cas de la reconnaissance de tempo [10]. De nombreux extracteurs supplémentaires sont aussi présentés comme particulièrement adaptés à certains problèmes spécifique, comme celui de la discrimination voix / musique (Modulation d'énergie à 4 Hz) [10]. Ces travaux ont tout à fait leur place mais nécessitent le travail de chercheurs expérimentés en traitement du signal. Nous y reviendrons plus loin.

3.2 Difficultés

Définir un protocole précis et justifiable n'est non seulement pas trivial, mais je n'ai trouvé que très peu d'exemples dans la littérature de papiers publiés décrivant et justifiant leurs expériences de manière satisfaisante. De même que penser de résultats de taux de reconnaissances donnés souvent sans comparaison et un intervale de confiance ? Nous ne sommes certainement pas arrivés à un résultat parfait, mais ce qui suit tente au moins de poser un cadre à notre approche.

3.2.1 constitution des bases

La taille des bases d'apprentissage et de test a été le premier problème auquel je me suis attaqué. Peut-être le plus compliqué. L'approche facile est de prendre "une base raisonnablement grande et pas mal distributée". Et qu'en est-il de l'impacte sur le taux de reconnaissance et ce que l'on peut en dire ? Intuitivement on peut penser que plus une base d'apprentissage est petite, plus le taux de reconnaissance obtenu sur la base de test risque d'être éloigné de celui obtenu sur la première. Mais dans quelle mesure ? A ce sujet le cours CS229[2] du professeur Andrew Ng de l'université de Stanford m'a été précieux.
Notre besoin ici est de connaître la taille minimum de la base d'entrainement pour espérer obtenir une erreur de généralisation aussi proche que possible de l'erreur d'entrainement. Y a t'il un rapport formel entre ces deux erreurs ? En quoi est-il lié

[2]Machine Learning, Lecture Notes, http://www.stanford.edu/class/cs229/

à la complexité du modèle ?
Dans le choix du modèle de classification, il y a un compromis entre la variance et le biais. La variance augmente lorsque l'erreur de généralisation de l'algorithme d'apprentissage "apprend" trop par coeur les données d'entrainement (overfitting). Elle augmente en général avec la complexité du modèle. Inversement le biais d'un modèle augmente plus le modèle est simple. Si le modèle est trop simple, il ne pourra pas capturer toute la complexité de la répartition des données. On s'interesse d'abord aux algorithmes d'apprentissage de type à minimisation du risque empirique (les algorithmes les plus classiques) et on suppose que les données d'entrainement et les données utilisées pour valider le modèle sont tirées de la même distribution, elle-même indépendantes et identiquement distribuées. Alors dans le cas des modèles d'apprentissage à classe d'hypothèse infinie (qui est le cas par exemple de la classification linéaire, et pour la plupart de ceux qui nous interesse) on a :
Soit **H** les hypothèses du modèle et $d = VC(H)$ la dimension de Vapnik-Chervonenkis du modèle (c'est à dire la taille du plus grand sous ensemble des données d'entrées X pulvérisé par **H**). Pour que l'écart entre l'erreur d'apprentissage et l'erreur de généralisation soit inférieur à γ avec une probabilité d'au moins 1-δ il suffit que la taille des données d'apprentissage soit de
$m = O_{\gamma,\delta}(d)$, avec $O_{\gamma,\delta}$ une constante qui peut dépendre de γ et δ
Il se trouve que dans la plupart des cas la dimension de Vapnik-Chervonenkis des modèles est proportionnelle au nombre de ses paramètres. Par conséquent la conclusion est que pour un algorithme à minimisation du risque empirique la taille de la base d'apprentissage doit être à peu près proportionnelle au nombre de paramètres de H. Cette conclusion ne répond malheureusement pas totalement aux questions de départ, mais donne seulement un guide. Faire le calcul pour un cas réel se révele en fait très complexe (impossible?). Donc dans la pratique on a pris des tailles de base largement supérieures au nombre de paramètres du modèle testé.

La question du nombre de bases d'exemples est également importante. Pour nos expériences nous avons besoin de 3 bases. Le schéma classique consiste à entrainer l'algorithme sur une base d'entrainement, puis de le tester sur la base de test. Mais nous avons besoin d'une base supplémentaire pour notre système EDS. On ne souhaite pas en effet que les algorithmes soit entrainés sur la base servant à la recherche génétique pour ne pas biaiser le résultat, ce qui se comprend puisqu'EDS par essence choisit le meilleur extracteur pour une base donnée via un critère de "fitness" (critère

de Fisher[3] ou kNN[4]).
Au final nous retenons la configuration suivante : une base de recherche génétique est séparée en 2 pour les besoins du système EDS et une grande base qui sera utilisée en Validation croisée à 10 blocs [5]. La taille de la base utilisée pour la recherche génétique est influencée par un autre critère : le temps de calcul. En effet la recherche génétique produit des résultats intéressants et originaux que si de nombreuses générations sont produites, après croisements et autres mutations. Donc nous avons revu à la baisse la taille de certaines bases afin de limiter ce temps de calcul, au détriment de la corrélation entre la qualité d'un extracteur perçu par EDS et sa qualité sur une base plus grande et plus représentative.

Entrainement EDS	Test EDS	Entrainement classifieur et test
taille = S	taille = S/2	validation croisée à 10 blocs

TAB. 1 – Configuration des bases d'apprentissage et de test

Pour garder une recherche génétique suffisament rapide, la taille empirique de la base d'entrainement d'EDS (S) doit être de l'ordre de 100 à 200 secondes maximum à une fréquence d'échantillonnage de 44100 Hz. Dans le cas des samples de l'ordre de 500 millisecondes cela correspond à des tailles de 200 à 400 samples.

3.2.2 Reference set

Le jeu de référence a pour cahier des charges de couvrir tous les opérateurs standards utilisés en reconnaissance des sons. La stratégie adoptée a été de ne pas prendre en compte les opérateurs trop spécifiques à un problème donné. C'est le cas par exemple de la modulation d'énergie à 4 Hz [11], qui a certes prouvé sont utilité pour la discrimination voix/musique. Nous sommes partis des descripteurs audio bas-niveaux de la norme MPEG-7. Ceux-ci sont listés et décrits dans le papier "MPEG-7 Overview (version 10)" [7], en voici le résumé en tableau :

Nous nous sommes heurtés à la difficulté de trouver des encodeurs MPEG-7[6] dis-

[3] puissance de discrimination d'une caractéristique déterminée

[4] k plus proches voisins

[5] La base originale est partitionnée aléatoirement en 10 sous-ensembles. Parmis ces 10 sous-ensembles, un seul servira de base de test et les 9 autres serviront de base d'apprentissage. Le processus est répété 10 fois, chaque sous-ensemble étant utilisé une fois comme base de test. Le résultat est alors la moyenne des résultats obtenus pour chaque estimation. En anglais 10-fold cross-validation.

[6] http://www.mpegif.org

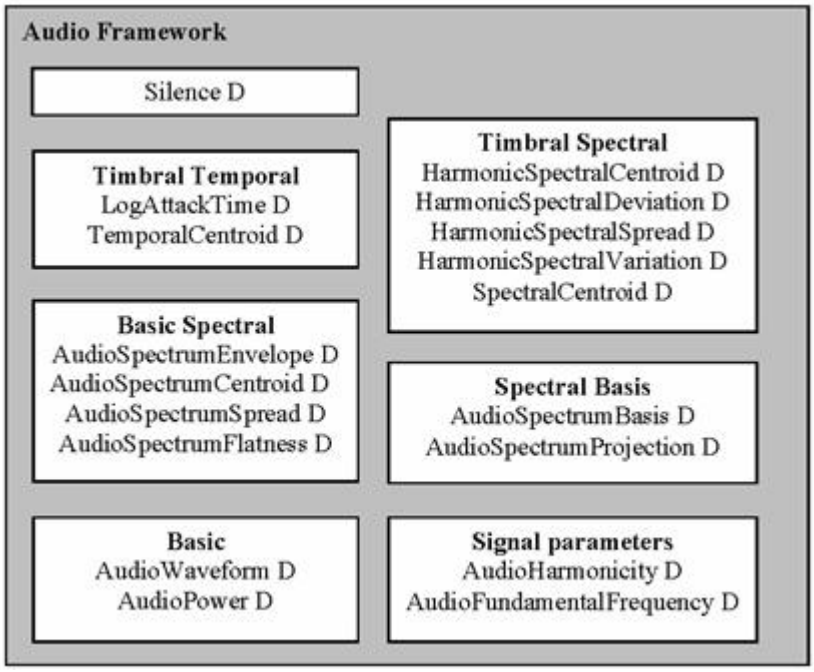

FIG. 1 – MPEG-7 Audio Low Level Descriptors

ponibles et de bonne qualité. Le "MPEG-7 Audio Analyzer" de l'Université Technique de Berlin par exemple n'est disponible qu'en ligne, et le "MPEG-7 Audio Encoder" de Holger Crysandt en Java a des limitations génantes sur la taille des fenêtres et les descripteurs non fonctionnels (Log Attack Time, etc...). Ces limitations nous ont conduit à utiliser les fonctions d'EDS pour extraire ces même informations. Il nous manquait quelques opérateurs (ceux concernant l'harmonicité, le temps de montée et le pitch) que j'ai codé en C puis ajoutés. Ainsi on a :

→ **Basic**

- AudioWaveForm n'apporte pas d'information utiles pour la reconnaissance de sons complexes
- AudioPower est équivalent à Rms dans EDS

→ **Basic Spectral**

- AudioSpectrumEnvelope => PitchBands dans EDS
- AudioSpectrumCentroid => SpectralCentroid même si dans EDS on n'utilise pas des fréquences logarithmiques. La quantité d'information n'est pas radicalement différente

- AudioSpectrumSpread => SpecralSpread dans EDS (même remarque que AudioSpectrumCentroid)
- AudioSpectrumFlatness => SprectralFlatness dans EDS (même remarque que AudioSpectrumCentroid)

→ **Signal Parameters**

- AudioHarmonicity => HarmonicityPraat, provenant de la librairie Praat[7]
- AudioFundamentalFrequency => Pitch, ajouté à EDS. Il manque à cet opérateur un indice de confiance.

→ **Timbral Temporal**

- TemporalCentroid => Centroid
- LogAttackTime => Log10(AttackTime(x)) ajouté à EDS

→ **Timbral Spectral**

- HarmonicSpectralCentroid => HarmonicSpectralCentroid ajouté à EDS
- HarmonicSpectralDeviation => HarmonicSpectralDeviation ajouté à EDS
- HarmonicSpectralSpread => HarmonicSpectralSpread ajouté à EDS
- HarmonicSpectralVariation => HarmonicSpectralVariation ajouté à EDS, nécessite au moins 2 trames
- SpectralCentroid => équivalent en terme d'information à AudioSpectrumCentroid

→ **Spectral Basis**

- AudioSpectrumBasis et AudioSpectrumProjection sont l'équivalent d'une transformée de Fourier à court terme. Ces informations sont bien trop volumineuses pour être utilisées en apprentissage telles quelles. Elles sont raisonnablement remplacées par le MFCC appliqués sur des fenêtres glissantes.

→ **Silence**, dans notre cas (des sons isolés de durée courte) ce descripteur n'est pas utile

A cet ensemble MPEG-7 nous ajoutons les descripteurs bas-niveaux suivant :

- MFCC, Les Coefficients Cepstraux permettent une représentation compacte du spectre du signal en prenant en compte, grâce à l'échelle Mel, des informations perceptives.
- HFC (High-Frequency Content), contenu en hautes-fréquences
- RHF (High Frequency Ratio)
- Spectral Kurtosis, mesure le degré d'écrasement du spectre
- Spectral Skewness, mesure le degré d'asymétrie du spectre

[7] http://www.praat.org/

– Spectral Rolloff Point, [16] and [11]
– Iqr (écart interquartile), utilisé comme indicateur de dispersion, il correspond à 50% des effectifs situés dans la partie centrale de la distribution
– Chroma, spectre ramené sur un octave
– 10 bandes de Mel
– 24 bandes de Barks
– Zcr, taux de passage par zéro

3.2.3 Sélection d'attributs

La grande difficulté de la selection d'attributs est que la notion de meilleur attribut est mal définie. De plus elles sont toutes nécessairement biaisées, c'est-à-dire sensibles à certains types de régularité et pas à d'autres, et limitées face au bruit dans les données ou à un nombre insuffisant de données. Sans prétendre être exhaustifs nous prenons ici trois algorithmes assez différents mais reconnus pour leur efficacité. Leur implémentation est celle de la bibliothèque Weka[8] [14].

– **InfoGain** mesure le gain d'information (ou réduction d'entropie) en découvrant la classe des exemples par rapport à l'entropie de l'ensemble d'exemples. Cette mesure est simple, mais a l'avantage d'être rapide et expliquable. En revanche, InfoGain va seulement classer les attributs, nous fixons donc une limite arbitraire du nombre d'attributs que nous voulons garder.

– **SubSetEval** va chercher les attributs prédisant le mieux la classe. Pour évaluer le jeu de descripteurs, nous utilisons l'algorithme des arbres de décision **J48** (version Weka de C4.5 de Quinlan) qui a l'avantage d'avoir un bon compromis entre précision et rapidité. Cependant ces arbres de décisions sont instables, c'est à dire qu'une petite variation dans la base d'entrainement peut conduire à différents choix d'attributs à chaque noeud de l'arbre. La recherche dans l'espace des features est ensuite conduite via la méthode **Greedy Step Wise** qui effectue des recherches dans l'espace des attributs, et s'arrête dès lors que l'ajout ou la suppression d'un attribut restant fait baisser le taux de reconnaissance.

– **AdaBoost** [6], le plus utilisé pour implémenter le boosting, cherche à produire des classifieurs "forts" (très précis) en combinant de façon itérative des

[8] http://www.cs.waikato.ac.nz/ml/weka/

classifieurs faibles. Attention, AdaBoost est une méthode d'apprentissage d'ensemble, et non pas seulement un algorithme de selection d'attributs. AdaBoost doit aussi être lié à un classieur, rapide de préférence. Nous prenons celui qui est conseillé par Weka, à savoir **Decision Stump**.

3.2.4 Learning

Notre stratégie pour choisir des algorithmes d'apprentissage est de prendre un ensemble assez représentatif, et ayant un bon compromis biais/variance (voir plus haut).

- Les **SVM** (appelés parfois Séparateurs à Vaste Marge) sont parmis les meilleurs (et certains pensent en effet le meilleur) algorithmes d'apprentissage. Les SVM sont liés à l'idée de séparer les données avec une vaste marge. Pour cela les données sont projetées dans un espace de dimension supérieure (en utilisant des fonctions noyaux) et il reste à trouver l'hyperplan dont la distance minimale aux exemples d'apprentissage est maximale. On appelle cette distance marge entre l'hyperplan et les exemples. Pour définir cet hyperplan, le modèle n'a besoin que des 'vecteurs de support' qui sont les exemples les plus proches. Le nombre de ces vecteurs de support nous donne en plus une idée de la complexité du modèle. Nous choisissons la fonction à base radiale[9] (noyau gaussien). De même nous prenons l'implémentation Sequential Minimal Optimization (ou SMO) des SVM.

- La méthode des **K plus proches voisins** [Weiss and Kulikowski 1990], est plus connue sous le nom kNN pour K-nearest neighbor . Elle diffère des traditionnelles méthodes d'apprentissage car aucun modèle n'est induit à partir des exemples. Les données restent telles quelles : elles sont simplement stockées en mémoire. Pour prédire la classe d'un nouveau cas, l'algorithme cherche les K plus proches voisins de ce nouveau cas et prédit la réponse la plus fréquente de ces K plus proches voisins. L'un de ses principaux défauts est de demander en calcul autant de comparaisons que de points existants sauvegardés, et ce pour chaque nouvelle donnée évaluée. Nous prenons les 2, 3 et 5 plus proches voisins dans notre comparaison.

[9] RBF

3.3 Protocole mis en place

En rassemblant ce qui précède nous arrivons au protocole retenu pour nos expériences.

1. Construire 3 bases aussi uniformément réparties que possible. 2 bases de 200 à 400 extraits de 1 secondes maximum servent pour EDS lors de la phase de recherche génétique. Une autre base va servir à réaliser la sélection d'attributs, à l'entrainement de l'algorithme d'apprentissage et à évaluer les performances obtenues par validation croisée. La taille de cette base devrait être au moins de 10 fois le nombre de paramètres du modèle (dans le cas des SVM, une borne pessimiste de ce chiffre est le nombre de vecteurs de support).
2. Lancer la recherche génétique avec EDS ses 2 bases. On cherchera à obtenir à la fois des attributs scalaires (comme SpectralCentroid) et des attributs vectoriels (comme des MFCC à 20 coefficients). On laisse tourner la recherche génétique pendant au moins une nuit pour chaque expérience. On ne conserve que les meilleurs attributs (d'après le critère de "fitness" choisi, kNN 3 voisins ou Fisher) et les plus différents entre eux via un outil simpliste d'EDS qui se base sur la formulation des opérateurs.
3. Appliquer les opérateurs issus de la recherche génétique et exporter le résultat vers Weka pour apprentissage.
4. Appliquer les opérateurs du jeu de référence (MPEG-7 et autres) et exporter le résultat vers Weka pour apprentissage.
5. Lancer le script qui réalise les opérations suivantes :
   * AdaBoostM1 avec 10,50,100 et 200 itérations
   * InfoGain ne gardant que 10,20 et 50 attributs puis SVM(RBF)
   * InfoGain ne gardant que 10,20 et 50 attributs puis kNN 5 voisins
   * SubsetEval et GreedyStepwise puis SVM(RBF)

   Chacun des algorithmes d'apprentissage est évalué en validation croisée à 10 blocs

Notes: Les mauvais résultats et les temps de calcul interminables de SubsetEval et GreedyStepwise nous ont conduit à ne plus l'utiliser. Ses mauvais résultats restent inexpliqués.
La combinaison InfoGain et kNN n'a été utilisée que pour l'expérience des aboiements de chiens, pour compenser le fait que c'est un problème multiclasse qui ne convient pas à l'AdaBoost.

4 Expériences

4.1 Cas d'école : sinusoïde dans un bruit rose

4.1.1 Contexte et critère d'évaluation

Considérons le problème simple de détecter la **fréquence** d'un sinus pur dans la bande de fréquence 600-900 Hz, noyé dans un bruit rose de forte puissance dans une autre bande de fréquence 1000-2000 Hz.

Du fait de sa forte puissance, le bruit rose est la composante principale du signal. En représentant le spectre on se rend compte que le pic est visible, mais non prédominant, et qu'une simple détection de maximum d'amplitude ne suffirait pas.

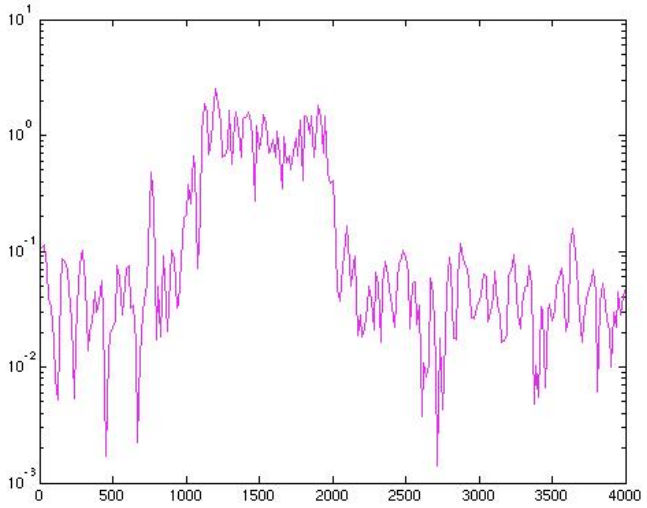

FIG. 2 – Apperçu du spectre du signal, sinus + bruit rose (fréquences en abcisses)

Cette expérience est un excercice type pour EDS. Nous allons comparer ses performances avec ce qu'on pourrait espérer sans connaissance à priori sur la nature du bruit avec les jeu de référence. EDS se base sur coefficient de corrélation de Pearson pour évaluer la précision de ses fonctions caractéristiques trouvées. Plus celui-ci est proche en valeur absolue de 1, plus la relation linéaire est forte. Pour comparer les deux approches, nous passons par Weka.

4.1.2 Les bases

Seules 2 bases sont nécessaires pour cette petite expérience. Une base d'entrainement composée de 151 échantillons de 1 secondes, et une base de test de la 76 échantillons de la même taille. Le bruit rose est générée à partir d'un bruit blanc filtré par un passe-bande 1000-2000 Hz (algorithme de Remez - maintenant nommé firpm - de matlab) sans chercher à obtenir un filtrage parfait. Les paramètres sont :

```
Amplitudes = [0, 0, 1, 1, 0, 0]
Frequences = [0, 950, 1050, 1900, 2100, 8000]
Fréquence d'échantillonnage = 16 kHz
ordre = 150
```

Le sinus a une amplitude de [-0,02;+0,02] et ses fréquences vont de :
600 à 900 Hz par pas de 2 Hz pour la base d'entrainement.
601 à 901 Hz par pas de 4 Hz pour la base dde test.

4.1.3 Sélection d'attributs

Dans ce cas on ne retient que le meilleur attribut côté EDS, que l'on compare à la combinaison des attributs du jeu de référence.

4.1.4 Apprentissage

L'apprentissage est réalisé avec l'algorithme de régression linéaire de Weka[10], et ce pour le meilleur attribut provenant de la recherche génétique d'EDS, et pour le jeu de référence. L'évaluation d'un descripteur se fait à partir d'un modèle de régression.

4.1.5 Resultats

La recherche génétique d'EDS a selectionné la fonction caractéristique suivante comme étant la meilleure : $Abs(Max(FFT(LpFilter(x, 441))))$. En clair un filtrage passe-bas est effectué sur le signal d'entré avec comme fréquence de coupure 441 Hz (filtre butterworth). Puis une FFT[11] est effectuée sur l'ensemble sur signal filtré (pas de zero-padding). Enfin la valeur absolue du maximum du spectre est conservée. La

[10] paramètres: weka.classifiers.functions.LinearRegression -S 0 -R 1.0E-8

[11] la FFT ou Transformée de Fourier Rapide d'EDS ne conserve que l'amplitude et se débarrasse de la phase

valeur maximale du spectre est forcément positive donc il est surprenant de voir la fonction "Abs" ici qui n'apporte rien.

La meilleure fonction caractéristique issu de la recherche génétique ainsi que le jeu de référence sont appliqués sur la base d'entrainement et sur la base de test puis exportés vers Weka (fichiers de type ARFF). Après évaluation nous obtenons les performances suivantes :
Relation linéaire trouvée pour EDS :
$-76.1288 * Abs(Max(Fft(LpFilter(x, 441)))) + 1156.5365$
Relation linéaire trouvée pour le jeu de référence :

```
  55653.9079 * RHF(Hanning(x)) +
  -1082.7348 * HFC(Hanning(x)) +
   -291.3752 * SpectralSkewness(Hanning(x)) +
 -22682.8504 * Iqr(x) +
   4205.2131 * Zcr(x) +
   -350.1112 * HarmonicSpectralDeviation(Hanning(x)) +
     -0.6249 * Pitch(Hanning(x)) +
     -0.7506 * SpectralCentroid(Hanning(x)) +
 -29314.0251 * SpectralFlatness(Hanning(x)) +
  -4263.5806 * SpectralSpread(Hanning(x)) +
      0.1406 * Centroid(x) +
  22877.1759 * Rms(x) +
   6824.5355
```

Attributs	**Coefficient de corrélation**
$Abs(Max(FFT(LpFilter(x, 441))))$	0.9895
Jeu de référence	0.8378

TAB. 2 – Résultats regression sinus et bruit rose

En évaluant individuellement les descripteurs bas-niveaux du jeu de référence le coefficient de corrélation est bien plus bas.

On peut dire que le jeu de référence n'est pas adapté à ce problème, et en effet le choix d'autres descripteurs serait facile pour n'importe quelle personne connaissant un peu le traitement du signal. Le choix de l'opération à ajouter (le filtrage passe-bas) est guidé par un double ensemble de connaissances : celle du probléme à résoudre

(les fréquences des sinus sont limitées à une bande donnée), et celle des effets des opérations (le filtrage permet de faire émerger les caractéristiques dans une bande de fréquences), c'est-à-dire le savoir-faire des experts en traitement du signal.
Néamoins on a mis en évidence la remarquable adaptabilité "à l'aveugle" du système EDS à un problème. La fonction caractéristique trouvée par EDS semble tout à fait logique et triviale. Par ailleurs, la valeur de la fréquence de coupure peut surprendre (on aurait pu penser que elle se situerait plus proche de 1000 Hz) mais cependant il faut prendre en compte le fait que les filtres utilisés aussi bien dans EDS que pour la génération du bruit rose ne sont pas du tout idéaux.

4.2 Dog Barks

Dans son document (en cours de publication) intitulé "A machine learning approach to the classification of dog barks" [8], Frédéric Kaplan présente un étude sur la communication acoustique entre les chiens domestiques. De rares études se sont penchées au préalable sur ce problème, et même de nombreux chercheurs ont supposé que du fait de sa domestication les aboiements du chien ont perdu de leur rôle spécifique à l'espèce [15]. Mais depuis les dernières années, des études ont montré que les aboiements sont caractérisés par certains paramètres acoustiques dépendants du contexte.

4.2.1 Contexte et critère d'évaluation

Des nombreux enregistrements de chiens de la race Mudi - un berger de hongrie - ont été effectués (Enregistreur Sony TCD-100 DAT et microphone Sony ECM-MS907 avec des cassettes Sony PDP-65C DAT) dans 7 situations différentes :

- 'Stranger': un étranger pénètre dans le jardin ou l'appartement du propriétaire en son absence
- 'Fight': le propriétaire tient son chien en laisse, un étranger encourage le chien à mordre son gant
- 'Walk': le propriétaire du chien fait comme s'il allait partir promener son chien
- 'Alone': le propriétaire attache le chien à un arbre dans un parc et s'en va
- 'Ball': Le propriétaire tient une balle (ou le jouet favoris du chien) à une hauteur d'environ 1,5 mètre devant le chien
- 'Food': Le propriétaire tient de la nourriture à une hauteur d'environ 1,5 mètre devant le chien
- 'Play': On demande au propriétaire de jouer avec le chien

Il est à noter que plus d'une vingtaine de chiens différents de cette même race ont été enregistrés.

L'approche de Frédéric Kaplan a été dans sont étude de prendre tout ce qui était possible (approche génétique, opérateurs de traitement du signal de la litérature, opérateurs provenant de la librairie Praat) et d'utiliser parmis les plus puissants algorithmes de sélection d'attribut et d'apprentissage (kNN, J48, PART et MultiLayer Perceptron). Le meilleur résultat obtenu pour le problème à 7 classes en validation croisée à 10 blocs est de 68 % avec des KNN à 10 voisins sur la base complète de 7440 aboiements. Ces 68 % de précision sont bien au dessus du hasard ($100/7 = 17.3\%$), et c'est bien ce que voulait démontrer Frédéric Kaplan.

Nous reprenons dans ce qui suit son problème en appliquant notre protocole dans le but de comparer plus distinctement les 2 approches EDS et approche standard. Nous espérons aussi secrètement trouver de meilleurs extracteurs et dépasser ces fameux 68 %.

4.2.2 Les bases

Nous avons nos 3 bases de rigueur :

- Entrainement EDS => 350 enregistrements de 0,3 seconde en moyenne, 50 enregistrements par classe
- Test EDS => 175 enregistrements de 0,3 seconde en moyenne, 25 enregistrements par classe
- Evaluation avec Weka en validation croisée à 10 blocs => 2100 enregistrements de 0,3 seconde en moyenne, 300 enregistrements par classe

Note : Nous avons apporté un soin particulier à éviter d'avoir les même chiens dans les bases pour EDS et dans la base d'évaluation afin d'éviter de ne pas apprendre par coeur telle ou telle signature acoustique d'un individu. Il aurait été mieux dans cette optique de ne pas faire de validation croisée automatique mais de mettre au point pour la validation une ou des bases d'entrainement et une base de test bien définie.

4.2.3 Resultats

Les résultats le l'évaluation ont été réuni dans le tableau qui suit :

2100 instances			DOGBARKS REFERENCE	DOGBARKS EDS
			96 attributs	63 attributs
Evaluateur	Recherche	Classifieur	10-fold X-Valid	10-fold X-Valid
InfoGain	Ranker, N=10	SMO RBF	47,2%	41,3%
InfoGain	Ranker, N=20	SMO RBF	50,8%	51,7%
InfoGain	Ranker, N=50	SMO RBF	64,0%	**71,0%**
InfoGain	Ranker, N=10	kNN, k=5	54,8%	49,5%
InfoGain	Ranker, N=20	kNN, k=5	58,0%	59,6%
InfoGain	Ranker, N=50	kNN, k=5	**64,4%**	69,6%
		Min	**47,2%**	**41,3%**
		Max	**64,4%**	**71,0%**
		Moyenne	**56,5%**	**57,1%**

FIG. 3 – Résultats de l'apprentissage (précision) - Aboiements

La première observation est que pour les 2 approches, 'Référence' et 'EDS', le taux de reconnaissance global est meilleur lorsqu'on augmente le nombre d'attributs sélectionné (limite associée à l'algorithme InfoGain). C'est un problème de reconnaissance complexe à 7 classes, et il n'est pas surprenant qu'il faille extraire une quantité importante d'info pour espérer séparer ces classes. En théorie on arrive assez rapidement à un palier lorsque l'on augmente le nombre d'attributs, car ceux ci deviennent redondants les uns avec les autres. Malheureusement on n'a pas retenu assez d'attributs d'un côté comme de l'autre pour mettre en évidence ce palier.
La deuxième observation immédiate est que le taux de reconnaissance global moyen est meilleur pour le jeu de référence lorsque seulement 10 attributs sont sélectionnés, équivalent lorsque 20 attributs sont sélectionnés, meilleur pour EDS lorsque 50 attributs sont sélectionnés. On peut interpréter ce phénomène par la présence dans le jeu de référence de descripteurs réputés pour donner des classifieurs 'fort', c'est à dire aptes à extraire beaucoup d'information dans de nombreux contextes. C'est le cas des MFCC, du ZCR, ou encore du Spectral Centroid. A l'inverse les fonctions caractéristiques retenues par EDS sont souvent plus spécialisées, donnant naturellement des classifieurs 'faibles'. Par conséquent si peu d'attributs sont sélectionnés, le jeu de référence est capable de fournir des extracteurs corrects, alors que le jeu d'EDS est bien pauvre en comparaison. En revanche il semble que la diversité des fonctions caractéristiques trouvées par EDS se combine mieux et permet d'atteindre le bon taux de reconnaissance de 71,0 % avec la configuration InfoGain (limite à 50 attributs) et SVM à noyau gaussien. Pour être plus complet voici la matrice de confusion de ce meilleur cas :

a	b	f	fo	p	s	w	<-- classés en tant que
247	10	10	6	8	12	7	a = alone
13	170	13	40	13	28	23	b = ball
1	7	270	3	4	10	5	f = fight
5	18	4	221	23	14	15	fo= food
2	40	20	3	219	4	12	p = play
14	21	24	16	3	205	17	s = stranger
12	38	11	20	40	17	162	w = walk

TAB. 3 – Matrice de confusion avec InfoGain + SVM pour les aboiements de chiens

On peut noter la confusion fréquente entre la situation walk et [ball ou food], et entre ball et [food]. Dans son étude, Frédéric Kaplan a ensuite laissé tomber la catégorie 'food' qui en effet vient fortement perturber la reconnaissance. Nous n'irons pas plus loin dans l'interprétation de ces confusions et je vous invite plutôt à consulter l'article de Frédéric Kaplan lorsqu'il sera publié !

4.3 Urban Noises

Le projet ORUS porte sur l'étude de la perception sonore des résidents dans un site urbain, considéré comme lieu de vie.
Cette étude cherche à mieux modéliser les réactions de gêne sonore en ville, en particulier la perception des sources sonores qui vont générer cette insatisfaction. Une recherche d'indicateurs adaptés au type de source mesurée est développée de manière à ne plus se limiter à la mesure d'indices purement énergétiques, mais à intégrer la perception de «l'objet bruit » identifié par le riverain. Ces nouveaux indicateurs permettent de mieux traduire les données sensibles des résidents et d'être ainsi plus proche de leur vécu. Les indicateurs tiennent compte de la perception des situations multi-sources et sources évènementielles, leur calcul passe par la résolution d'un problème métrologique et donc par l'extraction de descripteurs physiques propres à l'environnement sonore urbain.
Au final, des points de mesures seront installés et une extrapolation des résultats ponctuels permettra de visualiser le calcul des nouveaux indicateurs dans une cartographie sonore dynamique à l'échelle de vie des résidents et directement applicable vis à vis de la directive européenne 2002/49/CE en vue de communiquer avec le public.
(source : http://www.imageauditive.com/wikilistening/index.php?title=ORUS)

4.3.1 Contexte et critère d'évaluation

L'une des conclusions des travaux de Boris Defreville [3] sur les bruits urbains est que la gêne pour les habitants est plus liée à la relation qu'entretiens la personne avec la source, qu'à des caractéristiques intrinsèques acoustiques du bruit. Le niveau sonore joue bien sûr un rôle, mais l'identification doit faire partie du processus pour rendre compte de la gêne.
C'est donc l'un des objectifs du projet FDAI de créer un système général qui calcule la gêne sonore en se basant sur l'identification de 6 catégories de sons. 4 types de véhicules : voitures, bus, motos et cyclomoteurs. 2 non-véhicules : oiseaux et voix. L'une des voies suivies actuellement est de séparer en premier les véhicules des non-véhicules (tâche facile à réaliser) pour obtenir de meilleures performances globales. Cependant pour rester dans une démarche simple et lisible, nous resterons dans la configuration plus simple sans traitement préalable.
Pour évaluer nos 2 approches, nous allons nous baser sur les 6 confrontations en classe contre classe. Par exemple "Voiture / Non-voiture" ou "Oiseau / Non-oiseau". Encore une fois nous voulons mettre en évidence les différences amenées par nos deux approches, donc nos deux jeux de descripteurs.

4.3.2 Les bases

BUS / NON-BUS

- Entrainement EDS ⇒ enregistrements de 500 ms avec et sans présence d'un bruit de bus
- Test EDS ⇒ enregistrements de 500 ms avec et sans présence d'un bruit de bus
- Validation ⇒ 272 enregistrements de 500 ms autres

CYCLOMOTEURS / NONCYCLOMOTEURS

- Entrainement EDS ⇒ enregistrements de 500 ms avec et sans présence d'un bruit de cyclomoteurs
- Test EDS ⇒ enregistrements de 500 ms avec et sans présence d'un bruit de bus
- Validation ⇒ 224 enregistrements de 500 ms autres

MOTOS / NON-MOTOS

- Entrainement EDS ⇒ enregistrements de 500 ms avec et sans présence d'un bruit de motos
- Test EDS ⇒ enregistrements de 500 ms avec et sans présence d'un bruit de bus

– Validation ⇒ 875 enregistrements de 500 ms autres

OISEAUX / NON-OISEAUX

– Entrainement EDS ⇒ enregistrements de 500 ms avec et sans présence d'un bruit de oiseaux
– Test EDS ⇒ enregistrements de 500 ms avec et sans présence d'un bruit de bus
– Validation ⇒ 745 enregistrements de 500 ms autres

VOITURES / NON-VOITURES

– Entrainement EDS ⇒ enregistrements de 500 ms avec et sans présence d'un bruit de voitures
– Test EDS ⇒ enregistrements de 500 ms avec et sans présence d'un bruit de bus
– Validation ⇒ 340 enregistrements de 500 ms autres

VOIX / NON-VOIX

– Entrainement EDS ⇒ enregistrements de 500 ms avec et sans présence d'un bruit de voix
– Test EDS ⇒ enregistrements de 500 ms avec et sans présence d'un bruit de bus
– Validation ⇒ 741 enregistrements de 500 ms autres

Tous les enregistrements ont été réalisés par l'équipe de Boris Defreville dans Paris. Les enregistrements ont été découpés en fenêtre de 500 ms puis étiquettés à la main (ou plutôt à l'oreille experte). Grâce à cet effort indispensable nous avons pu avoir des bases de taille acceptable.

4.3.3 Resultats

Pour aider à la lecture, si la performance moyenne ou maximale est significativement meilleure, elle a un fond vert et son alter ego un fond rouge. Si les performances sont comparables (écart inférieur à 1 %) alors les deux (REFERENCE et EDS) ont un fond blanc. Plus spécifiquement, pour les valeurs spécifiques à un algorithme, les valeurs en rouge sont significativement en dessous de la performance moyenne, et les valeurs en vert sont parmis les meilleurs. Tous les pourcentages correspondent à des taux de reconnaissance (précision).

Avant de faire la synthèse de ces résultats dégageons les points les plus notables : **Premièrement** il n'y a pas UN meilleur algorithme. AdaBoost et SVM produisent

			BUS REFERENCE	BUS EDS
			96 attributes	1651 attributes
Evaluator	Search	Classifier	10-fold X-Valid	10-fold X-Valid
		IBk kNN = 2	89,0%	91,5%
		IBk kNN = 3	87,5%	91,5%
		IBk kNN = 5	86,8%	90,4%
		SMO RBF	90,8%	
		AdaBoost(DecisionStump), iterations : 10	78,3%	81,6%
		AdaBoost(DecisionStump), iterations : 50	84,6%	89,3%
		AdaBoost(DecisionStump), iterations : 100	86,0%	**92,6%**
		AdaBoost(DecisionStump), iterations : 200	86,0%	91,2%
InfoGain	Ranker, N=10	SMO RBF	80,9%	**76,8%**
InfoGain	Ranker, N=20	SMO RBF	86,4%	82,7%
InfoGain	Ranker, N=50	SMO RBF	**91,5%**	83,5%
		Max w/FS	**91,5%**	**92,6%**
		Average w/FS	**84,8%**	**85,4%**

FIG. 4 – Résultats de l'apprentissage (précision) - Bus

			CYCLO REFERENCE	CYCLO EDS
			96 attributes	158 attributes
Evaluator	Search	Classifier	10-fold X-Valid	10-fold X-Valid
		IBk kNN = 2	96,9%	96,0%
		IBk kNN = 3	96,4%	96,0%
		IBk kNN = 5	96,0%	95,5%
		SMO RBF	93,8%	92,0%
		AdaBoost(DecisionStump), iterations : 10	96,0%	**94,6%**
		AdaBoost(DecisionStump), iterations : 50	97,3%	96,4%
		AdaBoost(DecisionStump), iterations : 100	**99,1%**	96,0%
		AdaBoost(DecisionStump), iterations : 200	**99,1%**	**96,9%**
InfoGain	Ranker, N=10	SMO RBF	95,5%	95,5%
InfoGain	Ranker, N=20	SMO RBF	95,5%	96,0%
InfoGain	Ranker, N=50	SMO RBF	95,5%	96,0%
		Max w/FS	**99,1%**	**96,9%**
		Average w/FS	**96,9%**	**95,9%**

FIG. 5 – Résultats de l'apprentissage (précision) - Cyclomoteurs

			MOTOS REFERENCE	MOTOS EDS
			96 attributes	222 attributes
Evaluator	Search	Classifier	10-fold X-Valid	10-fold X-Valid
		IBk kNN = 2	84,7%	89,1%
		IBk kNN = 3	85,7%	86,7%
		IBk kNN = 5	85,5%	88,0%
		SMO RBF	87,7%	88,6%
		AdaBoost(DecisionStump), iterations : 10	**73,6%**	78,2%
		AdaBoost(DecisionStump), iterations : 50	80,1%	80,2%
		AdaBoost(DecisionStump), iterations : 100	82,1%	84,2%
		AdaBoost(DecisionStump), iterations : 200	82,1%	84,8%
InfoGain	Ranker, N=10	SMO RBF	81,4%	82,4%
InfoGain	Ranker, N=20	SMO RBF	85,7%	83,3%
InfoGain	Ranker, N=50	SMO RBF	**87,4%**	**89,3%**
		Max w/FS	**87,4%**	**89,3%**
		Average w/FS	**81,8%**	**83,2%**

FIG. 6 – Résultats de l'apprentissage (précision) - Motos

			OISEAUX REFERENCE	OISEAUX EDS
			96 attributes	143 attributes
Evaluator	Search	Classifier	10-fold X-Valid	10-fold X-Valid
		IBk kNN = 2	90,2%	92,6%
		IBk kNN = 3	91,5%	93,0%
		IBk kNN = 5	91,0%	93,0%
		SMO RBF	92,1%	92,6%
		AdaBoost(DecisionStump), iterations : 10	89,3%	87,9%
		AdaBoost(DecisionStump), iterations : 50	**91,9%**	90,9%
		AdaBoost(DecisionStump), iterations : 100	90,7%	91,0%
		AdaBoost(DecisionStump), iterations : 200	90,7%	**91,7%**
InfoGain	Ranker, N=10	SMO RBF	**87,0%**	86,7%
InfoGain	Ranker, N=20	SMO RBF	87,2%	**86,6%**
InfoGain	Ranker, N=50	SMO RBF	90,6%	90,2%
		Max w/FS	**91,9%**	**91,7%**
		Average w/FS	**89,6%**	**89,3%**

FIG. 7 – Résultats de l'apprentissage (précision) - Oiseaux

			VOITURES REFERENCE	VOITURES EDS
			96 attributes	146 attributes
Evaluator	Search	Classifier	10-fold X-Valid	10-fold X-Valid
		IBk kNN = 2	89,7%	91,2%
		IBk kNN = 3	88,8%	92,6%
		IBk kNN = 5	87,6%	90,9%
		SMO RBF	87,9%	87,6%
		AdaBoost(DecisionStump), iterations : 10	**78,5%**	90,0%
		AdaBoost(DecisionStump), iterations : 50	87,6%	**89,1%**
		AdaBoost(DecisionStump), iterations : 100	87,6%	91,2%
		AdaBoost(DecisionStump), iterations : 200	87,6%	90,3%
InfoGain	Ranker, N=10	SMO RBF	85,6%	91,5%
InfoGain	Ranker, N=20	SMO RBF	88,2%	**92,6%**
InfoGain	Ranker, N=50	SMO RBF	**92,1%**	92,1%
		Max w/FS	**92,1%**	**92,6%**
		Average w/FS	**86,8%**	**91,0%**

FIG. 8 – Résultats de l'apprentissage (précision) - Voitures

			VOIX REFERENCE	VOIX EDS
			96 attributes	143 attributes
Evaluator	Search	Classifier	10-fold X-Valid	10-fold X-Valid
		IBk kNN = 2	87,6%	91,4%
		IBk kNN = 3	89,5%	91,9%
		IBk kNN = 5	88,4%	91,1%
		SMO RBF	90,3%	92,3%
		AdaBoost(DecisionStump), iterations : 10	**81,4%**	88,3%
		AdaBoost(DecisionStump), iterations : 50	87,3%	91,8%
		AdaBoost(DecisionStump), iterations : 100	86,9%	91,6%
		AdaBoost(DecisionStump), iterations : 200	86,9%	**93,1%**
InfoGain	Ranker, N=10	SMO RBF	86,1%	**86,6%**
InfoGain	Ranker, N=20	SMO RBF	88,3%	88,9%
InfoGain	Ranker, N=50	SMO RBF	**91,9%**	92,3%
		Max w/FS	**91,9%**	**93,1%**
		Average w/FS	**87,0%**	**90,4%**

FIG. 9 – Résultats de l'apprentissage (précision) - Voix

des résultats comparables et bons comparés aux autres algorithmes essayés au cours de nos expériences. C'est ce que l'on espérait, à savoir obtenir une bonne cohérence dans les résultats en essayant de se dégager d'une querelle prévisible sur le choix forcément discutable d'un algorithme plutôt qu'un autre. Plus spécifiquement pour l'algorithme AdaBoost, en augmentant le nombre d'itérations, on arrive à un plateau où l'algorithme ne trouve plus d'attributs permettant d'améliorer le taux de reconnaissance. Augmenter le nombre d'itérations ne change donc plus grand chose. Ce plateau se situe entre 30 et 50 itérations. 10 itérations donne en général de mauvais résultats. Concernant les SVM, on observe de manière générale une meilleure constance en fonction des différents cas avec une précision autour de 87-90 %

Deuxièmement la hiérarchie est le plus souvent bien conservée en dépit des paramètres des algorithmes, mais comme on l'a vu pour les aboiements de chiens, le jeu de descripteur issu d'EDS atteint son maximum d'efficacité avec plus d'attributs que le jeu de référence.

Troisièmement on peut noter figure 10 que si on ne regarde que la performance maximal et moyenne par problème, EDS fait presque toujours aussi bien et souvent mieux que le jeu de référence. L'exception notable est le cas CYCLOMOTEURS, déjà relevé par Boris Defreville comme étant très facile. Le jeu de référence MPEG-7 produit de toute évidence de très bons extracteurs ! En revanche pour les cas MOTOS et VOIX, l'avantage est pris par EDS.

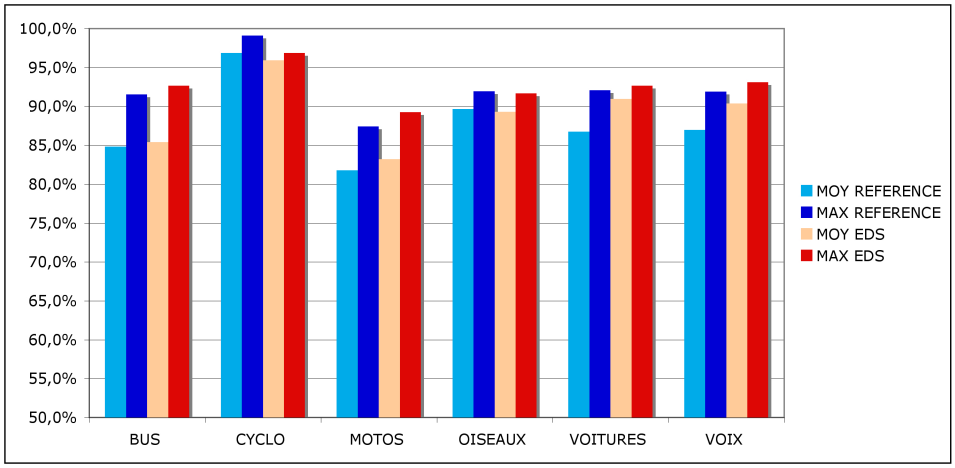

FIG. 10 – Résultats de l'apprentissage (précision) - Urbain

Restons sur la figure 10. Du point de vue des résultats maximums obtenus on peut dire que le cas de la moto est le plus délicat avec 89,3 % de taux de reconnaissance pour EDS. Ce n'est pas une surprise car cette catégorie n'est typiquement pas cohérente ou plutôt homogène au niveau acoustique. Cette classe abstraite 'motos' renferme aussi bien les 4 cylindres d'une moto japonaise que le 2 cyclindres d'un Harley-Davidson ! Boris Defreville utilise cette comparaison : ce serait comme mettre dans la même case guitare électrique et guitare acoustique...
D'une manière plus générale on se heurte aussi ici à la difficulté d'obtenir des bases homogène et suffisament grandes pour des problèmes aussi complexes. De nombreux sons parasites viennent perturber les enregistrements, et il devient difficile en annotant les extraits de dire que le bus repartant au feu rouge qu'on entend est plus important que le klaxon de la voiture de derrière au même moment.

5 Plus loin, la musique populaire du 20ème siècle

Application de ces approches à la musique populaire du 20ième siècle sur une base unique en son genre. Expériences en cours...

6 Conclusion

Avant de conclure on peut revenir sur les défauts d'EDS de part son algorithme génétique. Premièrement les temps de calculs sont parfois très long car on explore un espace potentiellement infini des fonctions. Cela nous a poussé à réduire la taille des bases d'apprentissage. Deuxièmement de nombreux paramètres relèvent un peu de la magie noire appelée heuristique. Par conséquent il est parfois nécessaire de relancer plusieurs fois les recherches afin d'obtenir de bons resultats. Enfin un autre problème important est celui des minimums locaux, en effet l'algorithme peut partir dans une 'direction' qui semble bonne, et finir par ignorer d'autres solitions possibles. On se retrouve alors avec 500 versions d'une même fonction, ce qui n'apporte pas grand chose en général.
La comparaison de l'approche évolutionniste [17] avec l'approche identifiée comme standard - nous laisse sur notre faim. EDS sort globalement certes vainqueur aux points mais la différence entre les meilleurs taux de reconnaissance obtenus avec les deux approches est souvent de l'ordre de seulement 1-2 %. A la décharge d'EDS, nous avons manqué de temps pour faire tourner la recherche génétique qui certainement n'a pas donné le meilleur d'elle-même. Mais de même on nous reprochera sûrement de n'avoir pas inclus dans le jeu de référence tel ou tel descripteur. Peut-on tirer des conclusions ? On a tenté de théoriser l'approche avec la création des bases de données, mais on se heurte dans ces études de données complexes à un ecueil : la plupart des théories de l'appentissage font des hypothèses sur la répartition uniforme des données d'entrées. Or on n'a pas accès à cette information pour des sons réels. On peut aussi douter que des bases de 200 à 400 sons vont pouvoir être réparti de manière uniforme. Mais alors que dire ? L'hypothèse de départ, peut-être était elle mal exprimée, était qu'avec un système comme EDS on va aller chercher PLUS d'information. Et donc que naturellement les résultats des taux de reconnaissance seraient meilleurs. Mon avis est que l'amélioration un peu décevante de l'approche EDS provient du cadre restrictif que l'on s'est donné au début. Cette fameuse approche 'bag of frames'. Pour l'oreille humaine, les évenements sonores sont interprétés dans leur contexte temporel et culturelle. La contextuallisation culturelle est importante pour la musique car par exemple un son de saxophone ne véhiculera pas la même information si celui-ci est entendu dans un titre jazz ou dans un titre de musique classique. C'est cette contextualisation qui permet à l'homme de devenir expert pour reconnaître la marque un amplificateur de guitare électrique. Pour ce qui est du contexte temporel les modèles dynamiques (tels les chaînes de Markov cachées) ne donnent pas pour le moment de meilleurs résultats **??**. Nous avons eu cette année un remarquable exposé

de psycho-acoustique avec Mme Tillman [13] au cours duquel elle nous a présenté ses travaux. Par de multiples expériences, certaines avec des réseaux de neurones, elle a mis en évidence l'importance de la mémoire dans la perception de la tonalité. Notre environement culturel implique des anticipations sur la musique que nous écoutons. Et finalement c'est souvent le delta entre ce que nous attendons et ce que nous recevons qui est l'information. Or ces aspects là sont encore totalement ignorés par la plupart des approches en apprentissage des signaux audio.

Merci d'avoir pris le temps de lire ce rapport.

Références

[1] J.-J. Aucouturier. *Dix Expériences sur la Modélisation du Timbre Polyphonique*. PhD thesis, UNIVERSITE PARIS 6, 2006.

[2] Giordano Cabral, François Pachet, and Jean-Pierre Briot. Automatic x traditional descriptor extraction: The case of chord recognition. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR2005), London, U.K.*, September 2005.

[3] B. Defréville, P. Roy, C. Rosin, and F. Pachet. Automatic recognition of urban sound sources. In *Proceedings of the 120th AES Conference*, 2006.

[4] Bertrand Delezoide. *Modèles d'indexation multimedia pour la description automatique de films de cinéma*. Thèse de doctorat, UNIVERSITÉ PARIS VI – PIERRE ET MARIE CURIE, 2006.

[5] G. DePoli and P. Prandoni. Sonological models for timbre characterization, 1997.

[6] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[7] José M. Martínez. Mpeg-7 overview (version 10). Technical report, 2004.

[8] Cs. Molnar, F. Kaplan, P.Roy, F. Pachet, P. Pongracz, and A. Miklosi. A machine learning approach to the classification of dog barks. Not Yet published.

[9] Geoffroy Peeters, Xavier Rodet, and Ircam Centre Pompidou. Automatically selecting signal descriptors for sound, November 29 2002.

[10] E. Scheirer. Tempo and beat analysis of acoustic musical signals, 1998.

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

[12] W. Siedlecki and J. Sklansky. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(2):197–220, 1988.

[13] B. Tillman and J.J. Bharucha. Implicit learning of tonality: A self-organizing approach. *Psychological Review*, 107(4):885–913, 2000.

[14] Frank E. Witten I.H. *Data Mining*. Morgan Kaufmann, 1999.

[15] S Yin. A new perspective on barking in dogs (canis familiaris). *Journal of Comparative Psychology*, pages 189–193, 2002.

[16] T. Zhang and C. Kuo. Hierarchical system for contentbased audio classification and retrieval.

[17] Aymeric Zils and François Pachet. Automatic extraction of music descriptors from acoustic signals using eds.