université de BORDEAUX

Utilisation de méthodes statistiques de classification supervisée pour le son

Rapport de stage de fin d'études en Traitement des Signaux et des Images numériques

Stage effectué par :

MANENTI *Céline*

Maître de stage :

ARNAUDON *Marc, IMB (Institut de Mathématiques de Bordeaux)*

Tuteur de stage :

BERTHOUMIEU *Yannick, IMS (Laboratoire de l'Intégration du Matériau au Système)*

Table des matières

Notations et abréviations

I) Notations

Analyse sonore		Classification	
i_t	intensité du signal à l'instant t,	m	nombre de descripteurs,
i_f	intensité de la fréquence f,	n	nombre d'individus,
N	nombre d'échantillons du signal.	n_k	nombre d'individus de la classe c_k,
		C	ensemble des classes,
		c_k	$k^{ième}$ classe,
		c_i	classe du $i^{ième}$ individu,
		μ_k	(vecteur 1xm) moyenne de la classe c_k,
		σ_k^2	(vecteur 1xm) variance de la classe c_k,
		w	axe de projection,
		W	ensemble d'axes de projection.

II) Abréviations

Analyse sonore		Classification	
LPC	Linear Predictive Coefficients – Coefficients Linéaires Prédictifs	EM	Expectation Minimization – Estimation Minimisation
LPCC	Linear Predictive Cepstral Coefficients	MMD	Maximale Marginale diversity - Mean Measure of Divergence
MFCC	Mel Frequency Cepstral Coefficients	CFS	Correlation-based Feature Selection
FFT	Fast Fourier Transform – Transformée de Fourrier rapide	ACP	Analyse en Composantes Principales – Principal Component Analysis
DCT	Discret Cosinus Transform – Transformée en Cosinus Discrète	GAs	Genetics Algorithms – Algorithmes Génétiques
IRR	Degré d'IRRégularité du spectre	ALD	Analyse Linéaire Discriminante - Linear DA, Canonical DA, Quadratic DA, ...
AM	Modulation d'Amplitude	SVM	Support Vector Machines – Machines à Vecteurs de Support, ou encore : Séparateurs à Vastes Marges
EDS	Extractor System Discovery	K-NN	K-Nearest Neighbourgs – k plus proches voisins
ASF	Amplitude Spectral Flatness – Platitude Spectrale	NBC	Naive Bayesian Classifier – Classifier Bayésien Naïf
SCF	Spectral Crest Factor – Facteur de Crêtes Spectral	GMM	Gaussian Mixture Model – Modèle de Mélange de Gaussiennes
ZCR	Zéro Crossing Rate – Taux de passage par zéro		

Introduction

I) Introduction

Le stage s'est déroulé à l'IMB (l'Institut Mathématiques de Bordeaux), avec monsieur ARNAUDON Marc comme encadrant de stage et monsieur BERTHOUMIEU Yannick comme tuteur de stage.

Le but de ce stage était l'analyse, la synthèse et la classification de sons d'instruments de musique. Le cœur du stage était plus précisément l'extraction de descripteurs et l'utilisation d'outils statistiques pour la classification sonore supervisée.

Le stage s'est déroulé en deux parties : tout d'abord l'analyse et la classification des sons, puis l'analyse-synthèse sonore. Les 1500 sons instrumentaux étudiés sont regroupés en 22 instruments différents, de qualité professionnelle, avec une note jouée par fichier sonore (il n'y a donc pas eu besoin de faire de segmentation).

Ce stage m'a prouvé que faire compliqué est rarement mieux que faire simple, surtout en terme de rapport temps/résultats. J'ai mis quelques jours à obtenir 85% de réussite en classification, à l'aide d'une centaine de descripteurs, et après être restée plusieurs mois plafonnée à 93% avec 1500 descripteurs, j'ai fini par frôler les 98% grâce à près de 20.000 descripteurs audio. Concernant les algorithmes, la méthode de sélection de descripteurs utilisant les seules moyennes des classes obtient des scores rivalisant avec des algorithmes de complexité quadratique et la classification la plus simple (k-NN) obtient les meilleurs taux de classification.

II) Résumé du stage

La classification se déroule en 3 étapes principales : l'extraction de descripteurs, la réduction de la taille des données puis la classification en elle-même, faite d'une phase d'apprentissage des données puis d'attribution des classes.

Les descripteurs détaillés sont temporels (enveloppe temporelle, descripteurs statistiques, ...) ou fréquentiels (enveloppe spectrale, répartition des fréquences, …). Ils peuvent être analysés sur tout le signal, ou sur des fenêtres courtes du signal et de nouveaux descripteurs en découlent, décrivant leur évolution temporelle. Nous nous retrouvons alors avec un nombre important de descripteurs, jusqu'à 20000 selon les choix faits. Certains travaux de recherche relatent même l'obtention de plusieurs millions de descripteurs, or il a été montré qu'un trop grand nombre d'informations nuisait à la classification (phénomène appelé la malédiction de la dimension). La réduction de la taille des données extraites est donc indispensable.

Pour réduire la taille des données, il existe plusieurs méthodes qu'il est intéressant de combiner, surtout en cas de très forte réduction nécessaire : la suppression des descripteurs inutiles (ou la sélection des meilleurs), la suppression de la redondance et la projection dans d'autres espaces.

La classification peut se faire directement à partir des individus appris (k-NN, …), en approximant les classes à l'aide de modèles statistiques (GMM, …) ou encore à l'aide d'arbres hiérarchiques (CART, …).

Mots-clefs : Traitement du signal – extraction de descripteurs audio – sélection de descripteurs – classification supervisée – analyse de sons d'instruments de musique.

Extraction de descripteurs audio

I) Introduction

Des échantillons sonores de quelques secondes sont composés d'une centaine de milliers de valeurs. Il est donc impossible de directement les classifier, il faut nécessairement diminuer de manière intelligente la taille des informations. Pour ce faire, nous utilisons des connaissances à priori sur les signaux sonores pour trouver des descripteurs adaptés au problème.

Il existe de nos jours de très nombreux descripteurs, assez souvent cités et explicités dans la littérature, comme par exemple dans : [HER03], [LEM06] ou encore [ESS05].

Il n'existe pas de descripteur universel miracle, tout dépend des caractéristiques des classes à séparer même si certains descripteurs se montrent plus utiles que d'autres. Notons le plus simple : le taux de passage par zéro du signal, qui a lui seul nous a permit d'obtenir un taux de 16%[1] de réussite pour la classification de 22 classes.

Une fois tous les descripteurs d'origines diverses calculés, il est nécessaire de les centrer et de les normer pour les uniformiser. De plus, éliminer les descripteurs constants peut éliminer des sources d'erreurs potentielles.

Certains descripteurs étant parasités par des valeurs aberrantes, l'application de bornes et/ou d'une échelle logarithmique à certains descripteurs s'est souvent avéré utile. Le traitement des rangs des valeurs[2] et non pas directement des valeurs est une autre façon de voir les choses et d'éliminer le problème des valeurs aberrantes.

Synonymes : descripteurs - variables explicatives - variables prédictives

II) Protocole d'extraction des paramètres

Certains paramètres ne s'extraient que sur le signal sonore complet (descripteurs de l'enveloppe temporelle, …), d'autres peuvent s'extraire sur des fenêtres[3] du signal.

Lorsqu'un descripteur est extrait sur les fenêtres du signal, nous obtenons une valeur par fenêtre. Nous pouvons alors conserver toutes ces valeurs, les fenêtres sont donc considérées comme étant chacune un individu, ou

1 Moyenne ZCR + NBC : 16% ; [Moyenne + variance] ZCR + K-NN : 21% ; 82 descripteurs dérivés du ZCR + K-NN : 63%
2 Voir annexe II) Distances et similarités
3 Voir annexe I) Fenêtrage

bien déduire des paramètres statistiques décrivant l'évolution temporelle des valeurs extraites. Les paramètres généralement analysés sont la moyenne, la variance et la moyenne des deux premières dérivées, mais l'utilisation d'autres paramètres plus élaborés a amélioré nos résultats de manière notable, comme nous le verrons plus loin.

Certains préconisent d'analyser séparément les descripteurs pour l'attaque du son et sa décroissance. Cela peut donner des résultats intéressants si les sons conviennent, mais ce n'est pas toujours le cas : les sons percussifs ont une attaque beaucoup trop nette qui peut ne s'étendre que sur quelques valeurs.

III) Les descripteurs

A) Descripteurs statistiques

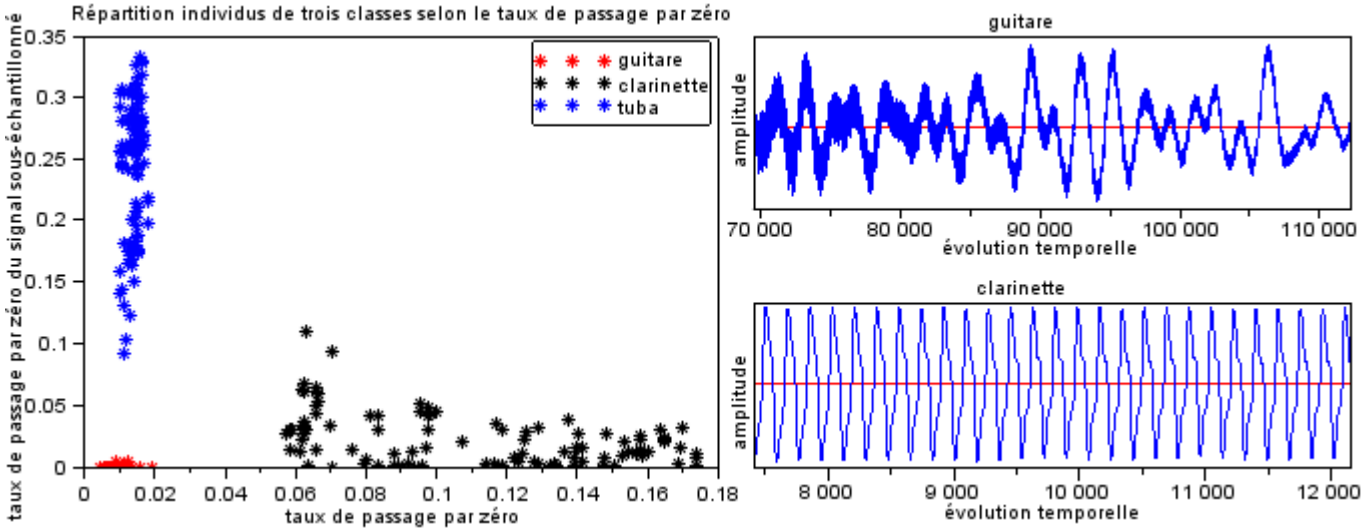
1) Le taux de passage par zéro (zero crossing rate)

Le taux de passage par zéro ZCR est un descripteur intéressant (uniquement pour les signaux centrés). Il est évidement corrélé à la fréquence fondamentale, mais donne aussi notamment une information sur le bruit présent dans le signal :

$$ZCR = \sum_{2 \le t \le N} \frac{|sign(i_t) - sign(i_{t-1})|}{2(N-1)}$$

Bien que extrêmement simple, il est très efficace combiné à d'autres descripteurs. Le taux de passage par zéro peut être calculé directement sur le signal mais apporte aussi des informations supplémentaires s'il est calculé sur le signal modifié, quelle que soit la modification (décomposition en ondelettes, signal lissé, sous-échantillonné, ...).

Exemple de projection des signaux sur les axes du taux de passage par zéro du signal et du signal lissé puis sous-échantillonné :



Ces trois classes sont clairement séparées grâce à ces deux descripteurs. Les taux de passage par zéro des signaux de guitare sont très faibles à cause de modulations très basses fréquences qui décentrent localement le signal. De nombreuses fenêtres ont ainsi un taux de passage par zéro nul, et d'autres un taux plus élevé. La variance de ce paramètre doit donc aussi pouvoir discriminer les sons de guitare. La clarinette, au contraire, est très stable tout au long de l'échantillon sonore.

Ce type de répartition des individus (classes presque rectangulaires) donne de la cohérence aux algorithmes de classification par partitionnement de l'espace, que nous verrons plus loin (ils découpent l'espace en cases puis leur attribue à chacune une classe).

2) L'intensité (& le RMS)

L'amplitude des signaux audio ayant toutes le même ordre de grandeur (ils sont compris entre -1 et 1), leurs différentes normes peuvent nous donner des informations supplémentaires, sur leur intensité mais pas seulement. Par exemple, un son dont l'enveloppe temporelle est exponentielle décroissante aura des normes (1, 2, ...) inférieures à un son de même intensité mais d'enveloppe temporelle de type plateau.

Définition des normes (pour p entier strictement positif) :

$$\|i\|_p = \left(\sum_{1 \le t \le N} |i_t|^p \right)^{\frac{1}{p}}$$

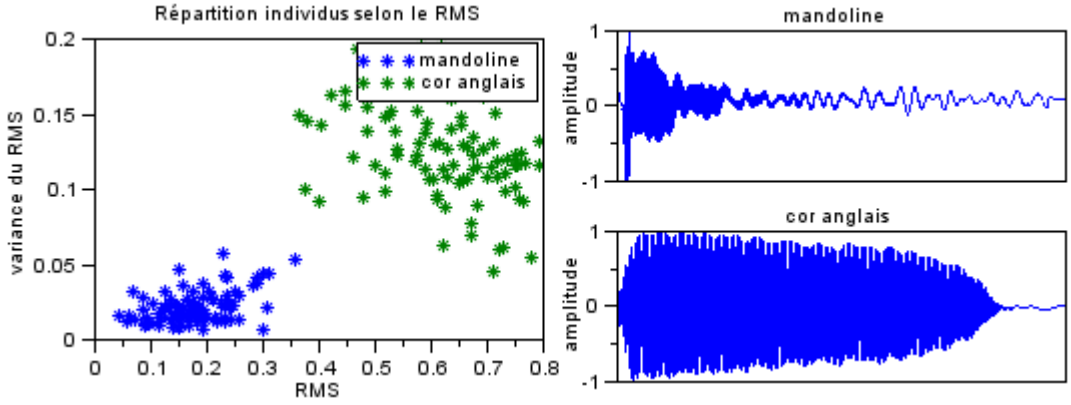
La norme infinie :

$$\|i\|_\infty = \max_{1 \le t \le N} (|i_t|)$$

Le RMS (Root Mean Square), très utilisé pour les signaux audio, est proche de la norme 2 :

$$RMS = \left(\frac{\sum_{1 \le t \le N} |i_t|^2}{N} \right)^{\frac{1}{2}}$$

Exemple de résultat :



Comme supposé, les sons percussifs ont une intensité moyenne plus faible que les sons tenus.

3) Centroïd (moment d'ordre1) : temporel et spectral

Le centroïd, temporel ou spectral, est un des descripteurs statistiques les plus souvent utilisés. Le centroïd temporel permet de décrire l'enveloppe temporelle et le centroïd spectral, aussi appelé brillance, est reconnu comme l'une des plus importantes caractéristiques du timbre.

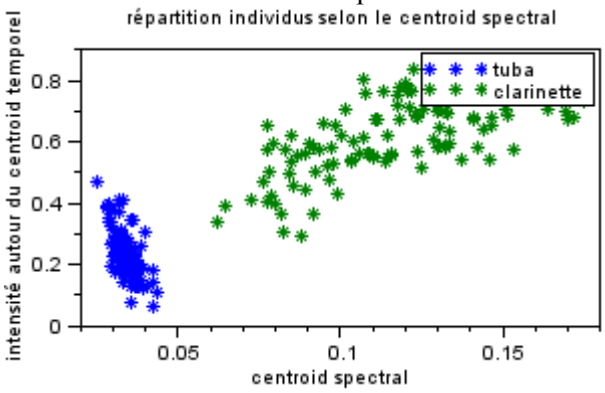
Le centroïd est l'indice (temps ou fréquence) moyen :

$$C = \frac{\sum_{1 \le t \le N} t|i_t|}{\sum_{1 \le t \le N} |i_t|}$$

On peut retrouver dans la littérature l'utilisation d'autres normes que la norme 1. Les centroïds peuvent aussi n'être calculés que sur certaines valeurs, comme le centroïd temporel de l'attaque et/ou de la décroissance, de même que le centroïd fréquentiel des harmoniques (qui peut être approximé par le centroïd du spectre seuillé).

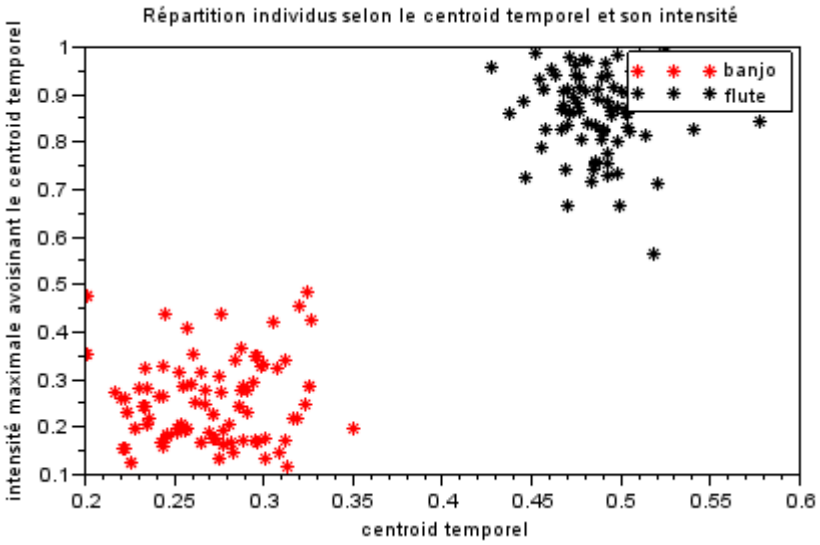
Un calcul de l'intensité des valeurs avoisinant le centroïd est un descripteur intéressant pour situer le centroïd par rapport à l'intensité maximale du signal et ainsi décrire plus précisément l'enveloppe.

Exemple de disposition des individus selon le centroïd spectral et son inensité :



Le centroïd spectral, l'intensité aux alentours et des descripteurs dérivés (135 au total) permettent d'obtenir 65% de classification réussie pour nos 22 classes à l'aide de l'algorithme k-NN.

Autre exemple, pour le centroïd temporel :



Ces résultats ne sont pas surprenants : le banjo est un son percussif, avec une attaque forte et une décroissance rapide, la majorité de l'intensité se situe donc au début du son. Au contraire, la flute est un son tenu, avec une enveloppe temporelle de type plateau et un centroïd temporel plutôt au centre de l'échantillon sonore.

Le centroïd temporel et son intensité permettent à eux seuls d'obtenir 21% de classification réussie avec l'algorithme k-NN.

4) Les moments (spectraux, temporels)

Les descripteurs statistiques comme la moyenne, la variance, de même que les moments d'ordre plus élevé, apportent une information supplémentaire à faible coût.

Les moments spectraux :

$$\mu_k = \frac{\sum_{1 \le f \le N} f^{k.} |i_f|}{\sum_{1 \le f \le N} |i_f|}$$

Les moments temporels :

$$\mu_k = \frac{\sum_{1 \le t \le N} t^{k.} |i_t|}{\sum_{1 \le t \le N} |i_t|}$$

A partir des moments spectraux peuvent être calculés le centroïd spectral : μ_1, la largeur spectrale (ou «bandwith » : la variance du spectre autour de son centroïd) : $\sqrt{\mu_2 - \mu_1^2} = \frac{\sum_{1 \le f \le N} |f - \mu_1| . |i_f|}{\sum_{1 \le f \le N} |i_f|}$, l'asymétrie spectrale $\frac{2\mu_1^3 - 3\mu_1\mu_2 + \mu_3}{(\mu_2 - \mu_1^2)^{\frac{3}{2}}}$ et la platitude spectrale : $\frac{-3\mu_1^4 + 6\mu_1\mu_2 - 4\mu_1\mu_3 + \mu_4}{(\mu_2 - \mu_1^2)^2} - 3$.

5) Auto-corrélation

Les premiers coefficients d'auto-corrélation du signal donnent notamment des informations sur l'importance du bruit dans le signal. Le nombre de coefficients conservés peut varier, parfois uniquement le premier.

Calcul de l'auto-corrélation :

$$\tau(k) = \sum_t i_t i_{t+k}$$

6) Centroid temporel / taille de l'attaque

Ce descripteur fait le lien entre le temps mis par le son pour atteindre son maximum et le temps mis pour émettre la moitié de l'intensité sonore. Dans le cas d'une enveloppe symétrique, ce descripteur vaut 1, et est supérieur à 1 dans le cas d'un son percussif.

De nombreuses combinaisons entre les différents descripteurs peuvent être faites. Il existe même des algorithmes de recherche automatique de nouveaux descripteurs, comme nous le verrons un peu plus tard.

B) Descripteurs de l'enveloppe temporelle

1) La rapidité d'attaque

La rapidité d'attaque est la durée mise par le signal pour atteindre son intensité maximale : $\underset{t}{\text{Arg max}}(i_t)$.
Pour les sons percussifs, cette durée est extrêmement faible. Au contraire, les sons dont l'enveloppe temporelle est parabolique atteindront leur intensité maximale vers la moitié de leur durée.

Ce descripteur devient néanmoins indésirable lorsqu'il s'agit d'une enveloppe temporelle de type plateau : le maximum est alors aléatoire. De fortes variations très basses fréquences peuvent aussi influencer ce descripteur dans le cas de sons non percussifs.

Pour des raisons d'échelles, le logarithme de la durée d'attaque est souvent utilisé (Log Attack Time).

2) Fréquence de l'enveloppe temporelle

La fréquence principale de l'enveloppe temporelle donne une idée des variations basses fréquences de l'amplitude du signal. L'enveloppe temporelle peut s'obtenir de plusieurs façons, en lissant l'absolu du signal ou à l'aide d'un sous-échantillonnage. On peut par exemple prendre pour chaque valeur de l'enveloppe temporelle le RMS de 10 ms du signal.

3) Modulation d'Amplitude (AM)

La Modulation d'Amplitude s'intéresse à certaines bandes de fréquences, dans le but de décrire l'enveloppe temporelle. L'intervalle 4-8Hz correspond au débit syllabique de la voix humaine et l'intervalle 10-40Hz à la granularité, ou rugosité, des sons. Seul le deuxième nous intéresse pour les sons d'instruments de musique.

De cette bande de fréquences, nous calculons : sa fréquence maximale, la différence entre son amplitude maximale et moyenne, de même que la différence entre son amplitude maximale et l'amplitude moyenne de tout le spectre.

4) Paramètres de modélisation de l'enveloppe temporelle

Dans le cadre de la synthèse sonore abordée en dernière partie, la modélisation de l'enveloppe temporelle a été nécessaire. Les paramètres extraits ont permis une amélioration du taux de classification, nous les avons donc conservés comme descripteurs.

Dans le cas où seuls des sons percussifs seraient considérés, une enveloppe exponentielle décroissante conviendrait. Dans notre cas, nous optons pour une exponentielle décroissante multipliée à une fonction puissance :

$$x^{a_1}.e^{-a_2.x}$$

Calcul des paramètres :

Nous pouvons penser à une première façon de les calculer (avec ici i_1 et i_2 les deux première valeurs de l'enveloppe temporelle du signal) :

$$\begin{cases} i_1 = 1^{a_1}.e^{-a_2.1} = e^{-a_2} \\ i_2 = 2^{a_1}.e^{-a_2.2} = 2^{a_1}.i_1^2 \end{cases} \Leftrightarrow \begin{cases} a_2 = -\ln(i_1) \\ a_1 = -\log 2\left(\frac{i_2}{i_1^2}\right) \end{cases}$$

Néanmoins cette résolution a deux défauts : elle ne se base que sur les deux premières valeurs de l'enveloppe (qui sont bruitées) et dépend de l'échelle du signal.

Nous allons donc nous intéresser à deux valeurs plus fiables : l'indice maximal (x_1) et le temps mit ensuite pour que l'amplitude du signal soit divisée par deux (x_2-x_1).

Or :

$$\begin{aligned} (x^{a_1}.e^{-a_2.x})' &= a_1.x^{a_1-1}.e^{-a_2.x} - a_2.x^{a_1}.e^{-a_2.x} \\ &= a_1.x^{a_1-1}.e^{-a_2.x} - a_2.x.x^{a_1-1}.e^{-a_2.x} \\ &= x^{a_1-1}.e^{-a_2.x}.(a_1 - a_2.x) \end{aligned}$$

Comme $a_1>0$ et $a_2>0$:

$$x_1 = \underset{x}{\text{indmax}}\left(x^{a_1}.e^{-a_2.x}\right) \Leftrightarrow \left(x_1^{a_1}.e^{-a_2.x}\right)' = 0 \Leftrightarrow \left(a_1 - a_2.x_1\right) = 0 \Leftrightarrow \left(a_2.x_1\right) = a_1 \Leftrightarrow x_1 = \frac{a_1}{a_2}$$

De plus :

$$x_1^{a_1}.e^{-a_2.x_1} = 2.x_2^{a_1}.e^{-a_2.x_2} \Leftrightarrow \left(\frac{x_1}{x_2}\right)^{a_1}.e^{-a_2.(x_1-x_2)} = 2 \Leftrightarrow a_1.\ln\left(\frac{x_1}{x_2}\right) - a_2.(x_1 - x_2) = \ln(2)$$

d'où pour $x_1 = \frac{a_1}{a_2}$:

$$\begin{cases} x_1 = \dfrac{a_1}{a_2} \\ a_1.\ln\left(\dfrac{x_1}{x_2}\right) - \dfrac{a_1}{x_1}(x_1 - x_2) = \ln(2) \end{cases} \Leftrightarrow \begin{cases} a_2 = \dfrac{a_1}{x_1} \\ a_1 = \dfrac{\ln(2)}{\ln\left(\dfrac{x_1}{x_2}\right) - 1 + \dfrac{x_2}{x_1}} \end{cases}.$$

Ces paramètres nous serons également utiles pour synthétiser l'enveloppe temporelle des sons.

C) Descripteurs fréquentiels

1) Avant-propos : la Transformée de Fourrier et ses dérivées

La transformée en fréquences la plus connue est la Transformée de Fourrier : elle projette les signaux dans la base des exponentielles complexes. La transformée de Fourier d'un signal quelconque réel est complexe et symétrique, et peut être décomposée en deux signaux réels : le rayon et l'angle. Ces informations peuvent s'avérer utiles dans certains cas, mais nous préfèrerons une transformée réelle, comme la Transformée en Cosinus.

La Transformée en Cosinus Discrète (DCT) se base sur la Transformée de Fourier : c'est la transformée de Fourier du signal modifié de telle sorte que sa transformée soit réelle et non symétrique.

Pour cela, le signal est tout d'abord rendu symétrique, puis les valeurs sont placées sur les indices pairs, des « 0 » sont mis aux indices impairs. La transformée en cosinus est alors la transformée de Fourier de ce signal 4 fois plus grand.

En pratique, le son sorti est de la même taille que le son d'entrée, la formule exacte de la transformée en cosinus est :

$$i_f = w(f)\sum_{t=1}^{N} i_t \cos\left(\frac{\pi(2t-1)(f-1)}{2N}\right), \text{ pour : } f = 1,2,\ldots,N$$

où :

$$w(f) = \begin{cases} \dfrac{1}{\sqrt{N}} & f = 1 \\ \sqrt{\dfrac{2}{N}} & 2 \le f \le N \end{cases}$$

Il existe plusieurs sortes de Transformées en Cosinus. La première est la plus proche de la Transformée de Fourier, les autres versions de la DCT ont étés modifiées dans l'optique de la compression de données.

2) Fréquence fondamentale (période)

La fréquence fondamentale exacte est difficile à obtenir, elle peut se calculer dans le domaine temporel (recherche de la période) ou dans le domaine fréquentiel. Ce n'est pas forcément ni la fréquence ni le coefficient d'auto-corrélation les plus élevé : l'erreur la plus courante est de trouver la moitié ou le double de la fréquence fondamentale.

A première vue, la fréquence d'un son est un descripteur plutôt parasite dans le cadre de la séparation d'instruments de musique. Néanmoins, ayant une influence sur de nombreuses caractéristiques du son (comme le taux de passage par zéro), elle peut s'avérer utile pour les en décorréler. Pour plus de renseignements sur les différents algorithmes d'estimation de la fréquence fondamentale, on pourra regarder [OBI05].

3) Les harmoniques

Les harmoniques sont des fréquences d'intensité élevée situées à des multiples de la fréquence fondamentale. Elles ne sont pas entendues comme des hauteurs, mais modifient notre perception du son.

Le nombre d'harmoniques est une information importante sur les sons, de même que l'intensité des harmoniques paires par rapport aux impaires (ce détail peut différencier certains instruments de musique, comme par exemple la clarinette qui n'a pas d'harmonique paire). Une approximation du nombre d'harmoniques peut être obtenu en divisant la fréquence d'intensité importante la plus élevée par la fréquence fondamentale.

Pour un modèle de synthèse plus poussé, d'autre paramètres peuvent êtres utilisés.

Le coefficient d'inharmonicité :

Le coefficient d'inharmonicité représente le déplacement des harmoniques dans le spectre par rapport à l'indice multiple de la fréquence fondamentale où elles auraient été si l'harmonicité était parfaite.

Les sons inharmoniques laissent entendre deux hauteurs distinctes, principalement au niveau de l'attaque pour des sons percussifs.

Calcul d'un coefficient d'inharmonicité :

$$\sum_{1 \le k \le 4} \frac{|p_i - k.f_0|}{k.f_0}$$

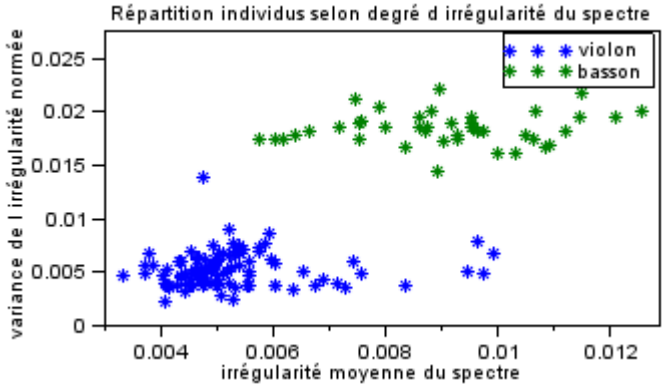
L'énergie de l'inharmonicité (Harmonic energy skewness) :

Mélange énergie et inharmonicité de chaque partiel :

$$\sum_{1 \le k \le 4} \frac{|p_i - k.f_0|}{k.f_0} . i_{p_i}$$

4) Degré d'Irrégularité du spectre (IRR)

Le degré d'irrégularité du spectre est souvent cité, avec plusieurs définitions différentes. Il peut être la norme de la dérivée du spectre ou de son enveloppe par exemple, ou encore la distance à l'enveloppe spectrale. Nos tests ont montrés que les résultats des différentes méthodes de calculs étaient fortement corrélés entre eux.



Une étude plus approfondie permet de constater que ce descripteur apporte des informations importantes pour une classification par modèle gaussien.

5) Fréquence de coupure (RollOff)

La fréquence de coupure Fc_p est la fréquence en-dessous de laquelle se trouve p% (généralement 95% ou 99%) du contenu spectral. C'est une approximation de la fréquence de coupure entre les parties majoritairement sinusoïdales et celles majoritairement bruitées du signal.

La fréquence de coupure du signal peut se calculer à l'aide de la somme cumulée de l'absolu de l'amplitude du signal :

$$Sc(k) = \sum_{1 \le t \le N} |i_t|$$

$$Fc_p = \min\left\{ k \mid Sc(k) \ge \frac{p}{100} Sc(N) \right\}$$

Plusieurs seuils de coupure peuvent être analysés et comparés entre eux.

6) Flux spectral

Le flux spectral représente la variation temporelle du spectre et se calcul à partir de la corrélation entre deux trames d'analyse consécutives :

$$1-\frac{\sum_{f=1}^{N} i_f(t-1)i_f(t)}{\sqrt{\sum_{f=1}^{N} i_f(t-1)^2}\sqrt{\sum_{f=1}^{N} i_f(t)^2}}$$

7) Pourcentage de fréquences significatives

Généralement, un spectre est majoritairement composé de valeurs faibles et il ne contient que quelques fréquences d'intensité élevée. Le pourcentage de fréquences significatives peut être évalué en cherchant un coude dans la courbe des intensités triées des fréquences. Pour obtenir le coude, nous pouvons tracer une droite entre le premier et le dernier point (fonction *linspace* avec MATLAB ou SCILAB) et lui soustraire la courbe. L'indice du maximum de l'absolu est alors l'indice du coude.

D) Enveloppe spectrale

L'enveloppe spectrale a un très grand impact sur la sonorité des sons : c'est par exemple elle qui permet de différencier les différentes voyelles.

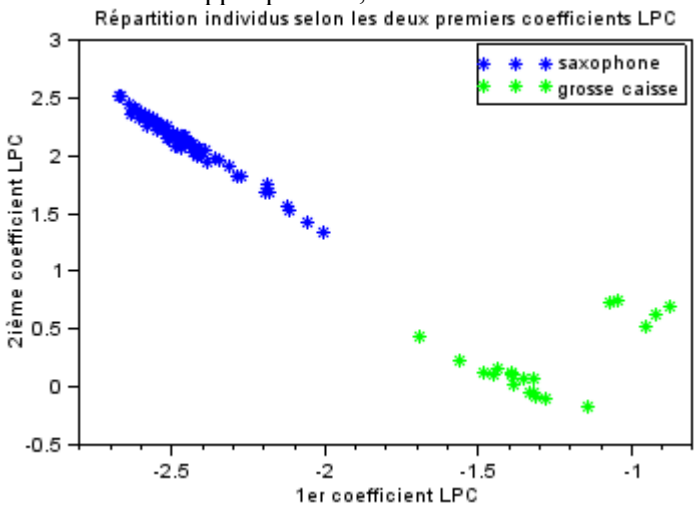
Néanmoins, il ne faut pas perdre de vue que les paramètres des différentes méthodes de description et de modélisation de l'enveloppe spectrale décrivent tous la même chose (l'enveloppe spectrale) et sont donc extrêmement redondants entre eux. Il n'est donc pas nécessaire d'utiliser plusieurs méthodes différentes.

1) Coefficients LPC

Le codage LPC[DOU02] a été pendant longtemps une référence en analyse/synthèse vocale.

Les coefficients LPC sont les coefficients du filtre linéaire auto-régressif minimisant l'erreur quadratique entre le signal original et un bruit blanc gaussien filtré. Ils sont obtenus à partir des coefficients d'auto-corrélation : N coefficients LPC sont obtenus à partir des N premiers coefficients d'auto-corrélation, calculés de manière itérative (nous pouvons citer par exemple l'algorithme de Levinson-Durbin) à partir de l'équation $a = R^{-1} r$, avec a le vecteur des coefficients LPC, R la matrice d'auto-corrélation et r le vecteur d'auto-corrélation. Notons que la première valeur vaudra toujours 1 (initialisation de la récurrence) et sera ignorée dans le cadre de la classification.

Les coefficients LPC décrivent l'enveloppe spectrale, mais se calculent dans le domaine temporel.



Les premiers coefficients LPC d'un son de saxophone (harmonique) sont plus éloignés de l'origine que ceux de la grosse caisse (percussif). Or, plus les sons sont bruités, plus les corrélations sont faibles et plus les coefficients LPC sont proches de zéro. Pour un bruit blanc, les coefficients du filtre auto-régressif sont nuls et la sortie vaut l'entrée, soit un bruit blanc.

Notons l'existence, entre autres, de deux descripteurs proches des coefficients LPC, mais utilisant une échelle plus proche de la perception humaine : les coefficients WLPC (Warped LPC) et PLP (Perceptual LP).

2) Coefficients Cepstraux

Les coefficients Cepstraux sont décrit comme plus robustes au bruit que les coefficients LPC.

Le calcul à effectuer pour obtenir le cepstre d'un signal Y est le suivant :

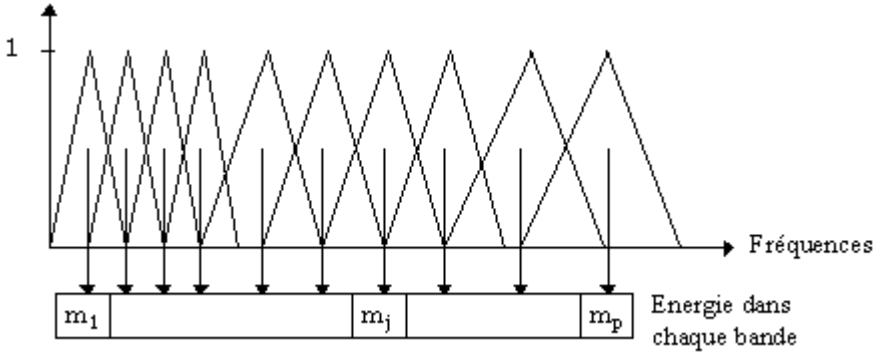
$$C(S) = FFT(\log(FFT(Y)^2))$$

A l'origine, le Cepstre est complexe, mais c'est la version réelle qui est la plus utilisée. La FFT peut être remplacée par une autre transformée, telle que la DCT qui a l'avantage d'être réelle.

Seuls les premiers coefficients cepstraux sont conservés.

3) Coefficients MFCC

Les coefficients MFCC sont plus proches des caractéristiques de l'oreille humaine : une échelle de fréquence logarithmique mel est appliquée au logarithme de la norme du spectre avant la deuxième FFT, réduisant l'étendue des plus hautes fréquences :



La conversion Mef/Hz peut être calculée de la manière suivante :

$$M = \frac{1000}{\log(2)} \log\left(1 + \frac{F}{1000}\right)$$

Algorithme :

Entrées : *y*, nx1, signal sonore, *nb*, entier strictement positif, le nombre de filtres,

Sortie : *mfcc*, nbx1, les coefficients MFCC.

- $l_f = \log(DCT(y)^2)$
- *k* allant de 1 à *nb* :
  - Application du $k^{ième}$ filtre : $m_f(k) = l_f' . f_k$ avec f_k la $k^{ième}$ fenêtre localement triangulaire de l'échelle mel,
- $mfcc = DCT(m_f)$

Nous pouvons ensuite conserver le nombre souhaité de coefficients parmi les *nb* coefficients MFCC obtenus.

4) Coefficients LPCC

Les coefficients LPCC (Linear Predictive Cepstral Coefficients) sont l'équivalent des coefficients cepstraux, mais calculés à partir des coefficients LPC.

Soient $(a_i)_{0 \le i \le p}$ les coefficients LPC. Les *p* premiers coefficients cepstraux sont obtenus par :

$$c_i = a_i + \sum_{k=1}^{i-1} \frac{k}{i} . c_k . a_{i-k}$$

Et les coefficients suivants par :

$$c_i = \sum_{k=i-p}^{i-1} \frac{k}{i} . c_k . a_{i-k}$$

5) Énergie relative

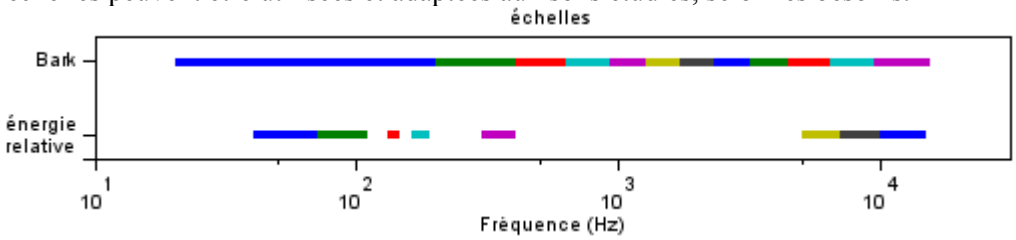
Les spectres peuvent être découpés en bandes de fréquences et différentes échelles peuvent leur être appliquées.

L'énergie relative [HER02] a pour but de comparer l'énergie de différentes bandes de fréquences pour cerner les parties les plus importantes du spectre. L'énergie est calculée sur chacune des bandes de fréquences suivantes :

40-70Hz ; 70-110Hz ; 130-145Hz ; 160-190Hz ; 300-400Hz ; 5-7KHz ; 7-10KHz ; 10-15KHz.

L'échelle de Bark peut aussi être utilisée, le son est alors découpé en 12 bandes : 20-200Hz ; 200-400Hz ; 400-630Hz ; 630-920Hz ; 920-1270Hz ; 1270-1720Hz ; 1720-2320Hz ; 2320-3150Hz ; 3150-4400Hz ; 4400-6400Hz ; 6400-9500Hz ; 9500-15500Hz. L'échelle de Bark place le signal sur une échelle quasiment logarithmique, plus proche de l'audition humaine.

D'autres échelles peuvent être utilisées et adaptées aux sons étudiés, selon les besoins.



6) Pente et décroissance spectrales (Slope & Decrease)

La pente et la décroissance spectrales sont deux autres descripteurs de l'enveloppe spectrale.

La pente spectrale :

$$\frac{N \sum_{1 \le f \le N} f\, i_f - \sum_{1 \le f \le N} f \sum_{1 \le f \le N} i_f}{N \sum_{1 \le f \le N} f^2 - \left(\sum_{1 \le f \le N} i_f \right)^2}$$

La décroissance spectrale :

$$\frac{1}{\sum_{2 \le f \le N} i_f} \sum_{2 \le f \le N} \frac{i_f - i_1}{f - 1}$$

7) Platitude et crête spectrales

ASF (Amplitude Spectral Flatness) :

$$\frac{\prod_{1 \le k \le K} a_k^{\frac{1}{K}}}{\frac{1}{K} \sum_{1 \le k \le K} a_k}$$

avec a_k l'amplitude de l'enveloppe spectrale (découpée en K sous-bandes)

SCF (Spectral Crest Factor - facteur de crête spectrale) : rapport entre le maximum de l'enveloppe du spectre et sa moyenne. (autre descripteur de la platitude spectrale)

$$\frac{\max_{1 \le k \le K} a_k}{\frac{1}{K} \sum_{1 \le k \le K} a_k}$$

Plus le facteur de crête est élevé, plus l'enveloppe est « piquée », plus il est faible plus elle est plate.

E) Transformations du signal

Les descripteurs précédents ont été calculés à partir du signal ou de son spectre, mais ils peuvent l'être aussi à partir d'une autre transformation du signal, par exemple telle que la Transformée en Ondelettes.

Cela créé de très nombreux descripteurs inutiles (doublons des précédents) mais certains sont utiles pour certaines classes.

1) Tenveloppe temporelle

L'enveloppe temporelle est obtenue en lissant le signal et parfois en le sous-échantillonnant. Elle peut être plus ou moins grossière ou fine (selon l'échelle choisie) et peut donc être plus ou moins proche du signal original.

Nous avons vu plus haut que la fréquence principale de l'enveloppe temporelle était un descripteur parfois utilisé. Mais son centroïd temporel, spectral, de même que son énergie, les premiers coefficients LPC et bien d'autres encore, ont montrés de l'intérêt dans le cadre de notre stage.

2) Une mesure de la variation temporelle des fréquences

Les fréquences du signal varient dans le temps différemment selon les signaux. Plusieurs échelles de temps peuvent être considérées, nous nous sommes limités à la variation des fréquences entre deux fenêtres consécutives. Cette variation peut être mesurée par produit scalaire, différence, …, selon les envies.

L'extraction de quelques uns des descripteurs conventionnels listés précédemment sur un signal représentant cette différence nous a donné de bons résultats : 5 descripteurs (moyenne et variance du centroïd spectral, moyenne de la fréquence de coupure, intensité) ont finis bien placés parmi les 50 meilleurs descripteurs.

Deux représentations ont été utilisées avec succès :

$$p_f = i_{1_f} . i_{2_f}$$

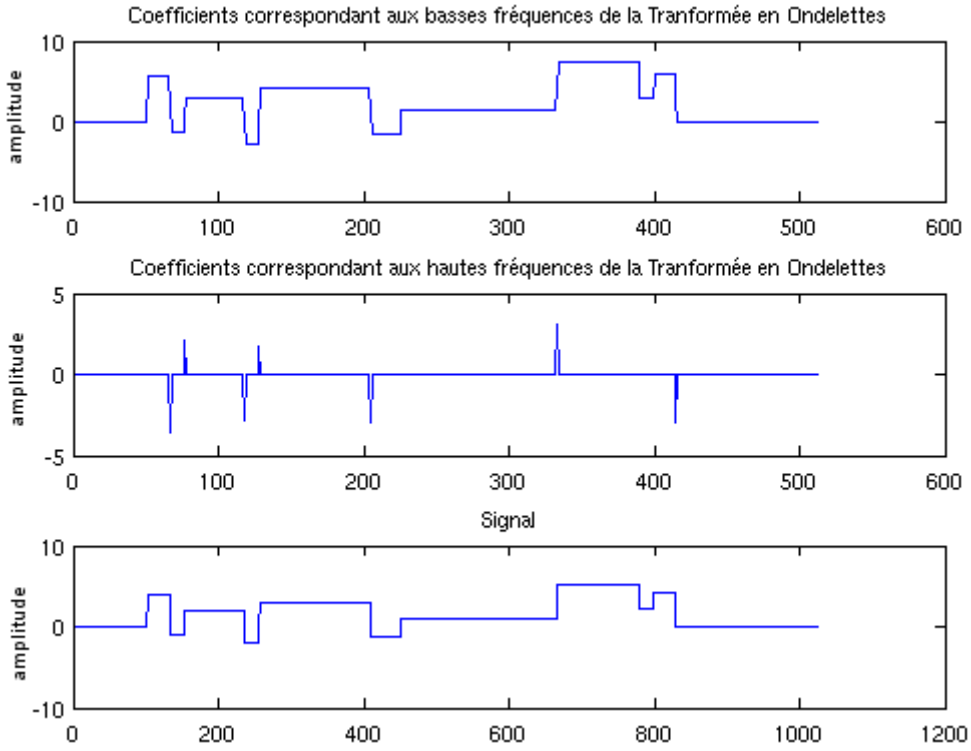
et :

$$s_f = \frac{i_{1_f}}{\max_f |i_{1_f}|} - \frac{i_{2_f}}{\max_f |i_{2_f}|}$$

3) La Transformée en Ondelettes (et par Paquets d'Ondelettes)

La Transformée en Ondelettes (TO) est une transformée conservant de l'information temporelle tout en ajoutant une information fréquentielle. A chaque étape, la TO sépare le signal en deux signaux de taille deux fois inférieure correspondant approximativement aux basses fréquences et aux hautes fréquences du signal original.

Pour mieux visualiser, voici un exemple simple :



La TO peut être répétée plusieurs fois sur les basses fréquences, celles qui habituellement contiennent le plus d'informations. Mais dans certains cas, décomposer de nouveau les hautes fréquences peut aussi être intéressant, cela s'appelle alors la Transformée par Paquets d'Ondelettes. Il reste à choisir l'Ondelette et le niveau de décomposition.

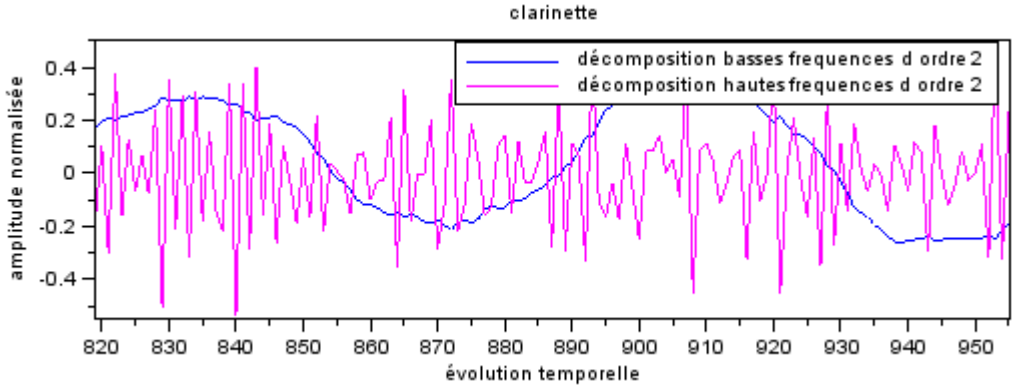
Dans notre cas, décomposer une ou deux est le plus adapté : chaque décomposition doit être suffisamment grande pour qu'il soit intéressant de lui appliquer les descripteurs vus précédemment.

Exemples de résultats :

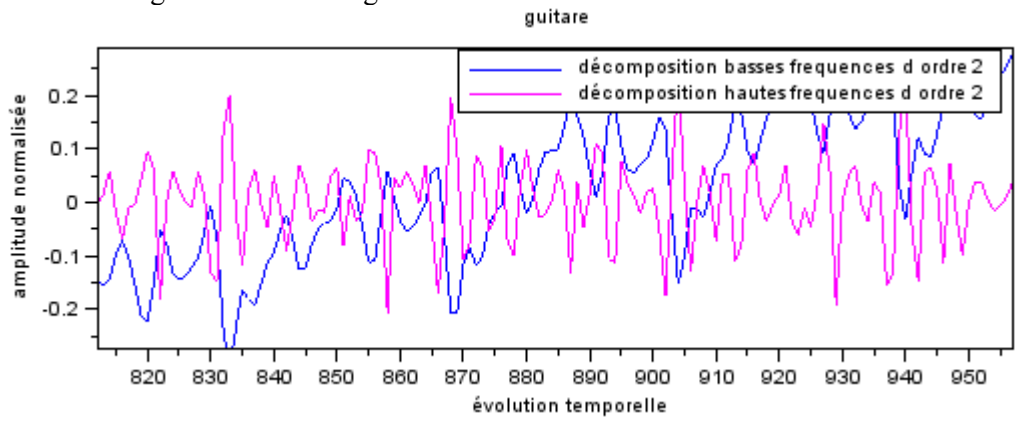
Clarinette *versus* banjo : la clarinette est plus harmonique, en conséquence la décomposition hautes fréquences a une amplitude deux fois moindre pour la clarinette que pour le banjo, et plus de 10 fois moindre dans le cas d'une seconde décomposition des hautes fréquences.

Clarinette *versus* guitare : les composantes hautes fréquences varient beaucoup plus rapidement pour la clarinette, puisque le signal restant est plus proche du bruit, comme nous pouvons le constater :

Décomposition du signal sonore d'une clarinette :



Décomposition du signal sonore d'une guitare :



Malgré l'intérêt visible sur cet exemple, d'autres descripteurs vus précédemment ont déjà apporté des informations similaires. Les descripteurs qui ont étés calculés avec la TO dans le cadre de ce stage n'étaient pas inintéressants mais n'ont pas réellement apporté d'informations supplémentaires. Les descripteurs les plus intéressants ont peut-être été ceux décrivant la décroissance de l'intensité selon le niveau de décomposition des hautes fréquences (les 10 meilleurs permettent d'obtenir près de 64% de classification réussie pour les 22 classes).

IV) Descripteurs représentatifs de l'évolution temporelle du son (intégration temporelle)

A) Introduction

Nous avons maintenant obtenu une réalisation pour chaque descripteur et pour chaque fenêtre du signal. A partir de là, il existe deux tactiques différentes quant à l'utilisation de ces informations. Nous pouvons soit considérer chaque fenêtre du signal comme un individu à part entière, soit prendre en compte l'évolution temporelle des descripteurs. Un compromis entre ces deux possibilités peut aussi être appliqué : considérer chaque fenêtre comme un individu et prendre en compte son évolution par rapport aux k fenêtres précédentes ou voisines. Les k premières et dernières fenêtres doivent donc être négligées, ce compromis est donc plutôt adapté aux sons longs (plusieurs notes jouées, …).

Dans le cas où nous considérons chaque fenêtre du signal comme un individu à part entière, nous conservons le nombre de descripteurs initial (saut les descripteurs globaux) mais augmentons sensiblement le nombre d'individus. Lors de l'attribution des classes, chaque fenêtre de l'individu à classer se voit attribuer une classe et l'attribution par vote majoritaire permet idéalement la diminution du nombre d'erreurs. Cette approche est la plus appropriée pour la segmentation audio.

Dans le deuxième cas, le nombre d'individus reste celui initial, mais la prise en compte de l'évolution temporelle du son implique la création de nouveaux descripteurs.

Graphiquement : (avec D pour Descripteur et F pour Fenêtre) :

		D1	D2			D1	D2
	F1	x11	x12	→	Individu 1	x11	x12
Individu	F2	x21	x22	**si 1F = 1 Individu**	Individu 2	x21	x22
	F3	x31	x32	→	Individu 3	x31	x32
	F4	x41	x42		Individu 4	x41	x42

↓ si analyse de l'évolution temporelle ↓

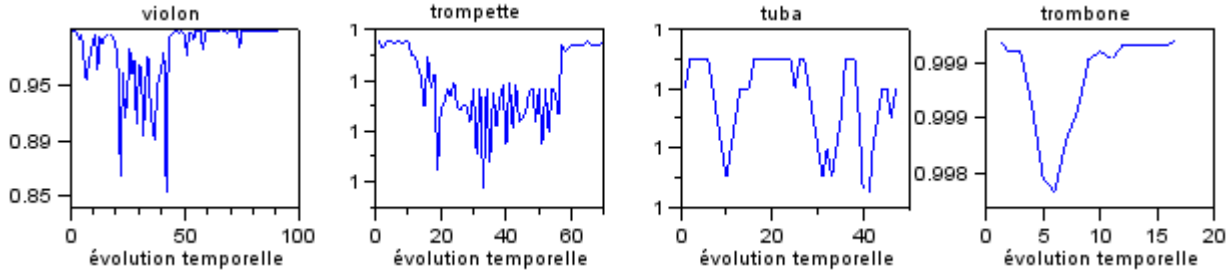
	D1	D2	moyenne (D1)	moyenne (D2)	variance (D1)	variance (D2)
Individu	x11	x12	x11	x12	x11	x12

B) Les caractéristiques temporelles

Pour un découpage du signal en fenêtres d'environ 25 ms (fenêtre de taille 1024 pour un son de 44100 Hz), nous obtenons 43 fenêtres par secondes (sans recouvrement). Nous avons donc généralement entre 30 et 300 fenêtres pour chaque son, ce qui permet d'extraire de nombreux paramètres représentant l'évolution temporelle des descripteurs analysés sur les fenêtres.

Les plus connus et indispensables sont : la moyenne et la variance du descripteur et de ses deux premières dérivées. Néanmoins le rajout d'une vingtaine d'autres nous a permit de gagner quelques pourcentages de réussite supplémentaires.

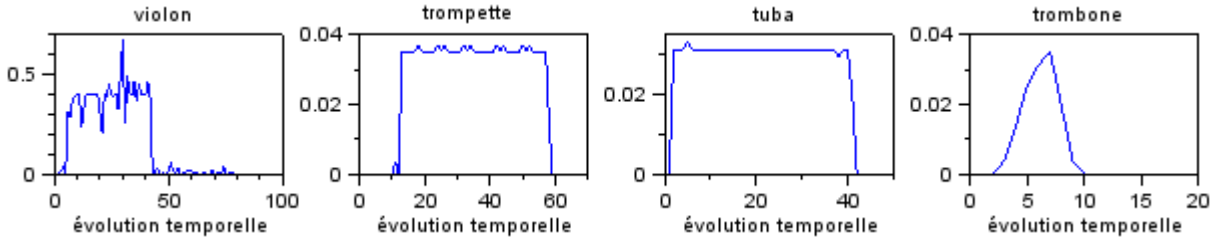
Pour mieux comprendre, étudions différentes courbes d'évolution des descripteurs. Voici 3 exemples de comparaison de 4 courbes de descripteur pour un son des classes violon, trompette, tuba et trombone :



Nous voyons sur ce premier exemple que la moyenne permettra de séparer le premier son des trois autres, et que la moyenne et/ou la variance des deux premières dérivées devraient permettre de séparer les quatre sons.

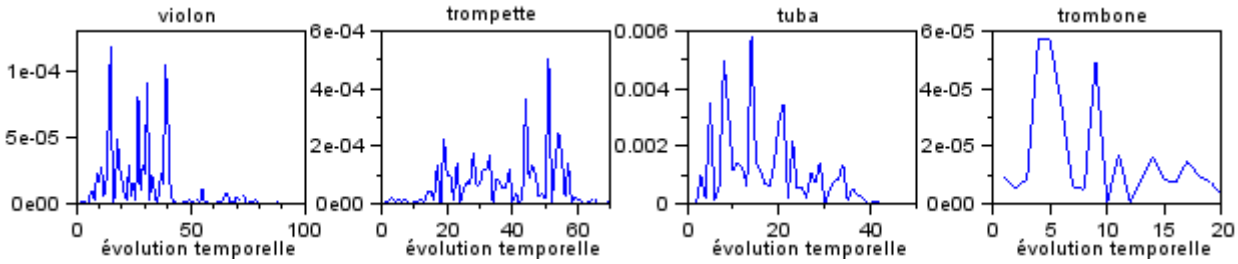
Notons qu'à cause des différences d'échelles, la dérivée du son de trompette a des caractéristiques proches de celle du trombone. Nous avons donc aussi analysés les descripteurs de l'évolution temporelle centrée et normée.

Sur d'autres courbes, d'autres paramètres semblent évidents :



Nous voyons que l'évolution de certains descripteurs peut être de type « plateau », avec un ensemble de valeurs élevées et un autre de valeurs faibles. Nous avons donc séparées ces valeurs en deux lots dont nous avons extrait pour chacun la médiane, la moyenne et la variance. La séparation en 2 du vecteur initial est faite en localisant un coude dans le vecteur trié. Le pourcentage de valeur au-dessus du seuil du coude nous a également servi de descripteur : nous voyons sur cet exemple que le plateau du son de tuba est proportionnellement plus grand que de celui du son de trompette.

Regardons encore un dernier exemple :



Si nous ne regardons pas les différentes échelles, le descripteur qui nous semble ici le plus adapté pour séparer le son de trompette des 3 autres est l'indice de la valeur maximale. Le centroïd et les différents moments peuvent donc eux aussi être envisagés.

Nous avons utilisés d'autres paramètres qui se sont montrés intéressants sur d'autres exemples : le taux de passage par zéro du vecteur centré, le premier coefficient d'auto-corrélation, de même que la fréquence principale du vecteur (descripteur trouvé dans la littérature). Certains calculent même les coefficients LPC.

C) Conclusion

En résumé, la majorité des descripteurs utilisés pour décrire les signaux peuvent l'être avec raison pour décrire l'évolution temporelle des descripteurs.

Certaines conditions restreignent l'étude de l'évolution temporelle des descripteurs, comme des applications temps réel. Ainsi, dans ce type de cas, l'évolution temporelle n'est étudiée que sur quelques fenêtres consécutives, par exemple la fenêtre active et les deux ou trois fenêtres précédentes.

V) Obtenir de nouveaux descripteurs de manière automatique : EDS

Tous les descripteurs vus plus haut sont l'œuvre d'humains, optimisés pour la description de signaux audio en général mais ne prenant pas en compte les spécificités propres aux sons étudiés. De plus, de nombreuses fonctions nécessitent des paramètres (nombre de coefficients LPC, fréquences de coupure, ordre des décompositions en ondelettes, …), qui sont fixés manuellement, parfois après plusieurs essais.

L'idée d'EDS (Extractor Discovery System [ZIL04]) est de découvrir de manière automatique les meilleures combinaisons de fonctions et les meilleurs paramètres, trouvant par itération successives des minimums locaux.

La recherche des meilleurs descripteurs n'a pas de limite théorique, il faut la fixer selon les possibilités matérielles et temporelles. Ce système a l'inconvénient d'être très coûteux en temps, ce qui nous force à limiter ses possibilités de manière drastique.

Besoins d'un algorithme EDS :

- Une liste de fonctions de transformation du signal : l'algorithme testera toutes les combinaisons entre elles. Dans notre cas, nous avons choisi : l'exponentielle, le logarithme, la valeur absolue, les n premiers coefficients LPC, une convolution avec une fenêtre de Hamming de taille n, la dérivée, un filtrage avec un coefficient k, la somme cumulée, le signe, la DCT4, le tri décroissant de même que ses indices, …, avec les paramètres à estimer.
- Une liste de fonctions d'extraction de descripteurs. Nous avons choisi une liste très courte : la première et la dernière valeur, de même que la moyenne. Prendre plus de fonctions n'est pas nécessaire : le maximum est la première valeur du signal trié, par exemple. Par ce choix, le temps d'exécution est nettement plus long pour obtenir les fonctions habituelles, mais laisse plus de liberté au programme pour les combinaisons.

Pour réduire le temps d'exécution, il convient de faire une matrice interdisant certaines combinaisons, comme 2 DCT successives par exemple.

Il serait nécessaire de fonctionner avec encore davantage de logique, comme arrêter lorsque le signal ressemble à un bruit blanc, ou bien réduire le nombre de fonctions de transformations pour réduire le temps d'exécution. Notre programme met 1h pour au plus 4 combinaisons de fonctions sur un signal de taille 10000. Cela rend impossible l'utilisation sur une grande base de données.

Étapes d'un algorithme EDS :

- Test de toutes les combinaisons de fonctions permises
- Sélection des meilleurs combinaisons
- Recherche des minimums locaux pour les paramètres des combinaisons sélectionnées.

Nos tests du système EDS nous ont montré qu'il permettait d'avoir des résultats satisfaisants mais pas

exceptionnels, à cause de la limite due aux coûts calculatoires. Il permet d'obtenir des résultats à moindres coûts humain (il y a seulement les fonctions de base à lister) en l'échange d'un fort coût en temps d'exécution.

Pour réduire le temps de calcul, nous avons réduit l'extraction des descripteurs à une seule fenêtre par signal. Nous avons pu ainsi obtenir jusqu'à 9 combinaisons différentes de fonctions dans un temps raisonnable, mais l'information négligée (l'évolution temporelle des signaux) étant très importante, nous n'avons pas non plus obtenus des résultats exceptionnels.

VI) Conclusion

L'extraction de descripteurs est la base de la classification. C'est une partie essentielle qu'il ne faut pas négliger, pour faciliter le travail des méthodes de classification et améliorer les résultats.

Ce sont fréquemment les mêmes descripteurs (moyenne et variance des MFCC, du taux de passage par zéro, des centroïds temporels et fréquentiels, de leurs premières et deuxièmes dérivées, ...) qui apparaissent dans les différentes publications et ils permettent effectivement d'obtenir des résultats satisfaisants avec un faible nombre de descripteurs. Néanmoins, il n'existe aucun descripteur idéal répondant à tous les problèmes de classification et il est souvent nécessaire d'en extraire d'autres pour améliorer les résultats.

Parmi tous les descripteurs, c'est peut-être l'étude approfondie de leur évolution temporelle qui a eu le plus d'impact sur les résultats. C'est en tout cas ce qui nous a permis de gagner les derniers pourcentages de classification.

Sélection de descripteurs

I) Introduction

Nous nous retrouvons maintenant avec un grand nombre de descripteurs : une centaine au début du stage, de l'ordre de 20000 à la fin. Il est possible d'en extraire beaucoup plus, parfois plusieurs millions.

On peut penser que tous les conserver donne le taux maximum de classification réussie, mais l'inverse se produit, appelé la « malédiction de la taille ». Trop de descripteurs à l'entrée des fonctions de classification font baisser les taux de réussite.

Les tests menés durant ce stage ont plutôt fait penser que ce n'était pas le nombre de descripteurs lui-même le problème, mais le nombre de descripteurs parasites n'ayant aucun lien avec le problème de classification posé. Même si les algorithmes de création de sous-espaces propres et de classification cherchent à ignorer au maximum les descripteurs inutiles, ceux-ci ont toujours une influence, même minime. Cela devient donc problématique lorsqu'ils sont en grande majorité.

Durant les tests, il est apparu que supprimer environ 9/10$^{\text{ièmes}}$ des descripteurs permettait de supprimer les descripteurs le plus inutiles et donnait les meilleurs résultats. Bien sur, cette proportion change suivant le nombre de descripteurs initiaux.

Il est donc nécessaire de trouver des méthodes pour sélectionner les descripteurs (Feature Selection) à conserver. De même que précédemment, il n'existe pas de sélectionneur idéal : tout dépend des cas.

II) Sélection des variables explicatives les plus significatives

A) Introduction

Il existe trois types de sélectionneurs :

- **Embarqués :** à l'intérieur même des algorithmes de classification (comme les arbres binaires)
- **Enveloppeurs :** modifient l'ensemble des descripteurs choisis en fonction des résultats de classification (coûteux)
- **Filtres :** juste après l'extraction, avant toute classification. C'est à eux que nous nous sommes intéressés.

Les sélectionneurs embarqués peuvent être les mêmes que les filtres, ils sont simplement utilisés dans un contexte différent.

L'idée de base est d'attribuer un score à chaque descripteur et de ne conserver que les mieux notés. Nous

allons voir différentes méthodes d'attribution de score.

B) Méthodes de sélection de descripteurs

1) Valeur test

La valeur test [ESC08] est une valeur testant l'intérêt de chaque descripteur. L'intérêt du descripteur est calculé à l'aide des valeurs moyennes de chaque classe et du nombre d'individus par classe. Plus la valeur moyenne des individus d'une classe s'éloigne de la moyenne de tous les individus, plus le descripteur a d'intérêt pour cette classe. Pour des données centrées-réduites, le calcul est :

$$vt_k = \mu_k \sqrt{n_k} \sqrt{\frac{n - n_k}{n-1}}$$

Avec : μ_k le vecteur moyen des individus de la classe c_k, n_k le nombre d'individus appartenant à la classe c_k et n le nombre d'individus total.

C'est le score d'intérêt plus simple et rapide à calculer qui soit et il donne des résultats satisfaisants.

2) Algorithme de Fischer

Dans le cadre de l'algorithme de Fisher, la valeur est le discriminant de Fischer :

Pour deux classes :

$$\frac{(\mu_1 - \mu_2)^2}{|\sigma_1^2 - \sigma_2^2|}$$

Pour plusieurs classes :

$$\frac{\sum_{c_k \in C} \frac{n_k}{n} |\mu_k|}{\sum_{c_k \in C} \frac{n_k}{n} \sigma_k^2}$$

Avec μ_k et σ_i^2 la moyenne et la variance des individus de la classe c_k.

Cette méthode est en lien avec l'ALD : toutes deux sont basées sur le discriminant de Fischer.

3) IRMFSP

L'algorithme IRMFSP est lui aussi inspiré de l'ALD, mais il est plus couteux que l'algorithme de Fischer : il est itératif. A chaque itération, il sélectionne le descripteur maximisant la dispersion inter-classes (l'écart des moyennes) sur la dispersion intra-classe (les variances)[4] dans l'espace orthogonal à celui des descripteurs déjà sélectionnés.

4) Gain d'information

Le gain d'information apporté par un descripteur i à une classe c se calcule comme suit :

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

Avec $H(c)$ l'entropie de c et $H(c|i)$ l'entropie conditionnelle de c sachant i. Les entropies sont calculées à partir d'histogrammes des valeurs rencontrées.

5) Ratio du Gain d'information

Le calcul du ratio du gain d'information est le gain d'information divisé par l'entropie associée au descripteur :

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

6) MMD (Mesure Moyenne de Divergence ou Diversité Marginale Maximale)

Le critère de diversité marginale se base sur les divergence de Kullback-Leibler entre les probabilités conditionnelles des classes $p(i|c_k)$ et leur moyenne $p(i)$:

4 Voir partie « ALD » pour plus de détails.

$$J_{MD}(i)=\sum_k p(i|c_k)\log\left(\frac{p(i|c_k)}{p(i)}\right)$$

Les probabilités se calculent à l'aide d'histogrammes, dont le choix du nombre de classes[5] est libre.

Algorithme de calcul de la diversité maximale [TOL06] :

Entrées : M, la matrice des données, et C, les classes des individus,

Sortie : J_{MD}, la diversité maximale

- Pour chaque descripteur i :
  - Pour chaque classe c_k :
    - Calcul de l'histogramme $h_{i,k}$ estimant $p(i|c_k)$,
  - Calcul de l'histogramme h_i estimant $p(i)$,
  - Calcul de la diversité marginale : $J_{MD}(i)=\sum_k \frac{n_k}{n} h_{i,k}^T \log\left(h_{i,k}./h_i\right)$
- Fin

7) Marge Maximale

Le critère optimal à optimiser pour ensuite faciliter l'utilisation des SVM est la maximisation de la marge entre les classes. Il s'agit de noter les descripteurs en fonction de l'écart entre les extremums des classes. Cette méthode de sélection de descripteurs nous a donné de bons résultats en général, pas spécifiquement avec l'utilisation des SVM.

8) Relief

Relief est un algorithme itératif dédié à la classification binaire (c'est-à-dire avec uniquement deux classes présentes). Il peut donc être utile dans le cadre de classification par arbres binaires.

Cet algorithme attribut des points de mérite à chaque descripteur en fonction de sa capacité à permettre de séparer les individus entre eux.

A chaque tour de boucle, un individu est tiré au hasard. Le score de chaque descripteur est alors modifié en fonction de l'éloignement des valeurs pour le plus proche voisin de chaque classe : le plus proche voisin de la même classe doit être proche et celui de l'autre classe le plus éloigné possible.

9) ReliefF

ReliefF[RIC06] est une généralisation de l'algorithme Relief à plusieurs classes et à un voisinage de taille k. Le score de mérite est attribué en moyennant sur ces voisinages.

10) Descripteurs corrélés avec les axes principaux de l'ACP

Les descripteurs peuvent être ordonnés en fonction des axes principaux de l'ACP (ou de l'ALD, …). Le $n^{ième}$ descripteur est celui, n'ayant pas encore été choisi, le plus corrélé avec le $n^{ième}$ axe.

11) Sélection binaire

La sélection des meilleurs descripteurs pour chaque paire de classes se justifie dans le cadre d'utilisation de SVMs ou d'arbres binaires. Il s'agit simplement de sélectionner des descripteurs pour chaque couple de classes à l'aide d'une des méthodes précédentes. Chaque classe a ainsi des descripteurs adapté à elle, ce qui n'est pas garanti lorsqu'un descripteur n'est utile que pour une seule classe : son score sera plus faible que les descripteurs un peu utiles à toutes les classes.

Cette méthode devient coûteuse et peut sélectionner trop de descripteurs lorsque le nombre de classes devient trop important.

5 On appel classe un partitionnement de l'espace par l'histogramme.

C) Mérite des ensembles de descripteurs

Une fois les méthodes de sélection de descripteurs codées, nous nous retrouvons avec plusieurs choix possibles. Le choix peut-être être fait après des tests de classification sur les ensembles sélectionnés, mais cela peut être trop coûteux en temps selon le ou les algorithmes de classification choisis.

Attribuer une note à chacun des ensemble pour les départager est une autre solution, exploitée par la méthode CFS (Correlation-based Feature Selection) qui attribue une note de mérite aux ensembles selon la corrélation des descripteurs entre eux.

Calcul du mérite de chaque descripteur selon un critère de corrélation :

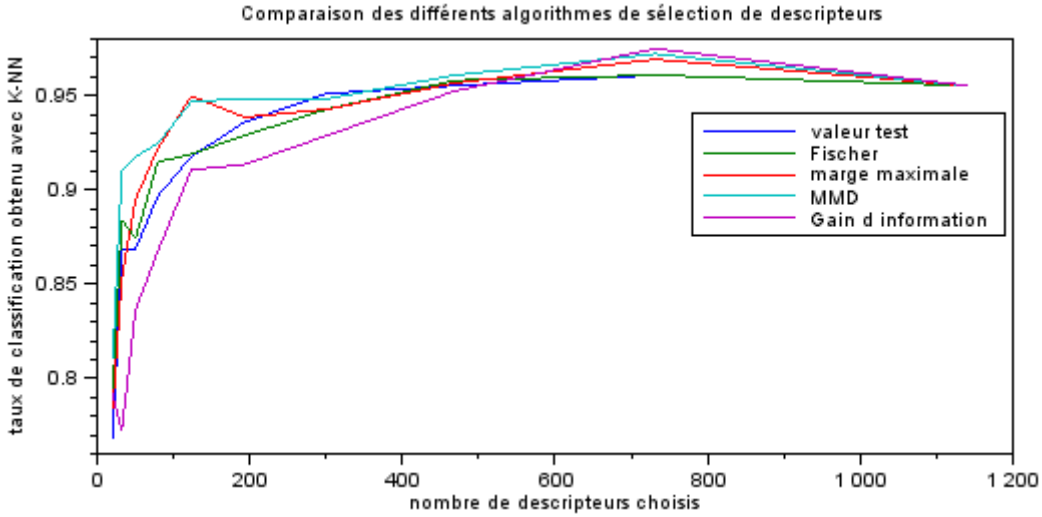
$$merit = \frac{m \,.\, \overline{\rho_{y,x}}}{\sqrt{m + m\,.\,(m-1)\,.\,\overline{\rho_{x,x}}}}$$

avec m le nombre de descripteurs, $\overline{\rho_{y,x}}$ la moyenne des corrélations entre les descripteurs et le descripteur cible, et $\overline{\rho_{x,x}}$ la moyenne des corrélations croisées entre descripteurs.

III) Résultats et conclusion

De nombreuses autres méthodes existent, mais il apparaît clairement, dans la littérature comme dans nos tests, qu'il n'est pas nécessaire de coder des dizaines et des dizaines d'algorithmes de sélection de descripteurs différents. Néanmoins, s'il y a nécessité, on pourra par exemple regarder[CHO11] pour en découvrir d'autres.

Comparaison des résultats obtenus avec les différents descripteurs et la méthode des plus proches voisins (K-NN, voir partie suivante) :



Ces courbes ont été plusieurs fois générées tout au long du stage, après chaque changement important des descripteurs initiaux : ce sont eux qui influencent le plus les résultats et non pas la méthode de sélection. Notons tout de même des différences entre les méthodes :

Le nombre de descripteurs : la méthode MMD dépasse les 90% avec moins de 50 descripteurs quand le Gain d'Information en nécessite plus d'une centaine, avant d'obtenir le meilleur score pour plus de 600.

Variation des courbes : certaines courbes varient plus que d'autres, la valeur test semble être une des méthodes les plus constantes dans ses résultats.

Même s'ils ont bien sûr tous leurs spécificités, les algorithmes de sélection de paramètres donnent globalement des résultats similaires. Il est plutôt conseillé d'en coder deux ou trois et de prendre celui qui semble le plus adapté au cas considéré. Les paramètres qui peuvent influencer le choix du sélecteur de descripteurs sont :

- **les résultats**, qui dépendent des descripteurs et des classes considérées mais aussi et surtout des algorithmes de réduction et de classification ensuite choisis,
- **le temps d'exécution** : trouver le sous ensemble de descripteurs idéal nécessite de tester tous les sous-ensembles possibles. Un compromis doit donc être fait entre le coût calculatoire et la qualité des résultats,
- **le nombre de descripteurs** à sélectionner à partir duquel l'algorithme permet d'obtenir des résultats satisfaisants : comme nous le voyons sur le graphique d'exemple ci-dessous, c'est pour un faible nombre de descripteurs choisis que les différentes méthodes sont les plus inégales.

Projections dans des sous-espaces propres

I) Introduction

La projection des données dans des espaces plus petits permet de les visualiser (2D, 3D) mais aussi de faciliter leur traitement. Projeter de manière intelligente les individus sur des axes où les classes seront plus facilement séparables permet de simplifier les problèmes de classification. Pour ce faire, les méthodes peuvent utiliser les propriétés statistiques des classes comme leur moyenne et leur variance (ACP, ALD) ou se baser sur les individus eux-mêmes (SVMs).

II) Algorithmes de définition de sous-espaces

A) ACP

1) Introduction

L'Analyse en Composantes Principales (ACP)[ESC08] est l'algorithme le plus connu. Il est plus spécifique à des problèmes de classification non supervisée, mais peut aussi être utilisé en classification supervisée.

L'ACP cherche les axes principaux des données à analyser pour les projeter dessus. Ces axes forment le sous espace maximisant la variance des données projetées. Plus précisément, considérons les données disposées en matrice M, avec chaque ligne un individu et chaque colonne un descripteur. Alors les axes principaux sont les vecteurs propres de la matrice de covariance S de M associés aux valeurs propres les plus élevées.

2) Algorithme

Algorithme de l'ACP :

Entrée : M, nxm, matrice des données, avec les individus en ligne,

Sorties : P, nxm, les projections, W, nxm, les vecteurs propres, et D, 1xm, les valeurs propres.

- Centrage des descripteurs : soustraction de sa moyenne à chaque colonne de M,
- Pour économiser en temps de calcul, si $n>m$, $M=M^T$
- Calcul de la matrice de covariance : $S=M.M^T/m$,
- Calcul des valeurs propres D et des vecteurs propres W de S (fonction *spec* sous SCILAB),
- Rangement de D et des lignes de W selon le tri par ordre décroissant de D.
- S'il n'y a pas eu rotation de M au début :
  - $W=M^T.W$
  - Normalisation des vecteurs propres : division de chaque colonne k de W par $\sqrt{m.D(k)}$
  - $P=W^T.M^T$
- Sinon :
  - $P=W^T.M$
- Fin

Le nombre d'axes principaux à conserver dépend du contexte. On pourra fixer un nombre à l'avance (une dizaine, un nombre proportionnel au nombre de classes ou d'individus, …), ou le choisir en fonction des valeurs propres. Il s'agit alors de prendre tous les axes dont la valeur propre est supérieure à un seuil, qu'il soit défini à l'avance ou déterminé selon un « coude » formé par la courbe des valeurs propres.

3) Conclusion

En résumé, les axes représentant le mieux les données (maximisant la variance des projections) sont les vecteurs propres associés aux valeurs propres les plus élevées de leur matrice de covariance.

Diverses méthodes dérivées de l'ACP sont apparues, selon les besoins et donc les critères à minimiser. Il y a par exemple l'ALD, qui est une méthode supervisée cherchant à maximiser la variance entre les classes et à minimiser celle des individus à l'intérieur d'une même classe.

B) ALD

1) Introduction

Le but de l'Analyse Linéaire Discriminante (ALD)[TOL06] est de réduire le nombre de dimensions tout en préservant au maximum les classes. Pour cela, l'ALD cherche à maximiser le critère de Fischer, c'est-à-dire maximiser la variance entre les différentes classes (donc entre leurs moyennes : hypothèse gaussienne) tout en minimisant la variance à l'intérieur des classes. Dans le cas idéal, tous les individus d'une même classe se retrouvent projetés sur un même point (variance intraclasse nulle), à distance des autres classes.

La maximisation du critère de Fischer pour un problème binaire (uniquement deux classes) est plus simple et moins coûteux que pour le cas général. La séparation de deux classes est un problème courant en classification, par exemple lors de l'utilisation d'arbres hiérarchiques binaires. La résolution simplifiée du problème permettant de diminuer sensiblement les coûts calculatoires, nous nous y sommes intéressés.

2) Cas à deux classes

Données :

- $k \in \{1,2\}$
- c_k : $k^{ième}$ classe,
- μ_k : vecteur moyen de la classe c_k ,

- $S_B=(\mu_1-\mu_2)(\mu_1-\mu_2)^T$: matrice de covariance interclasse,
- $S_k=\sum_{x_i\in c_k}(x_i-\mu_k)(x_i-\mu_k)^T$: matrice de covariance des individus de la classe c_k,
- $S_w=S_1+S_2$: matrice de covariance intraclasse.

Posons :

- w : axe discriminant de projection recherché (un vecteur), $S_B.w$ de même direction que $(\mu_1-\mu_2)$,
- $\mu_k^{'}$: vecteur moyen de la projection sur w de la classe c_k
- $S_k^{'}$: matrice de covariance de la projection sur w de la classe c_k.

Le critère de Fischer :

$$J_{ALD}(w)=\frac{(\mu_1^{'}-\mu_2^{'})^2}{S_1^{'2}+S_2^{'2}}=\frac{w^T S_B w}{w^T S_w w}$$

Solution :

$$w=S_w^{-1}(\mu_1-\mu_2)$$

Par rapport au cas général :

- Point fort : beaucoup moins coûteux,
- Point faible : réservé à 2 classes,
- Lien : w est similaire à l'axe principal obtenu avec la méthode générale (à la norme près).

3) Généralisation

Données :

- c_k : $k^{ième}$ classe,
- μ_k : vecteur moyen de la classe c_k,
- μ : vecteur moyen de tous les individus,
- n_{c_k} : nombre d'individus de la classe c_k,
- $S_B=\sum_k n_{c_k}(\mu_k-\mu)(\mu_k-\mu)^T$: matrice de covariance interclasse,
- $S_k=\sum_{x_i\in c_k}(x_i-\mu_k)(x_i-\mu_k)^T$: matrice de covariance de la classe c_k,
- $S_w=\sum_k S_k$: matrice de covariance intraclasse.

De même, posons :

- W : matrice des axes discriminants solutions,
- $S_k^{'}$: matrice de covariance de la projection sur W de la classe c_k.

Le critère de Fischer :

$$J_{ALD}(W)=\frac{|S_B^{'}|}{|S_w^{'}|}=\frac{W^T S_B W}{W^T S_w W}$$

Solution :

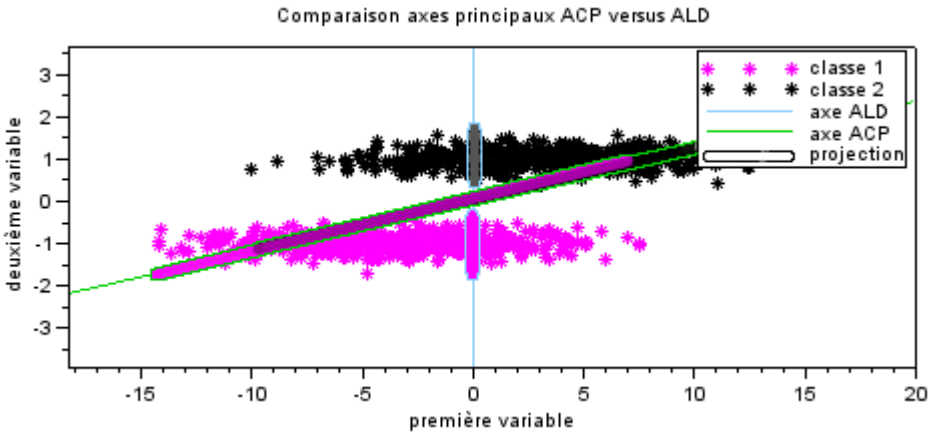
Les colonnes de W sont les vecteurs propres de $S_w^{-1}S_B$.

Inversion de la matrice :

Il arrive que S_w soit singulière. Pour éviter les problèmes de division par zéro lors de l'inversion, une inversion par décomposition en valeurs singulières peut être effectuée, en forçant les valeurs singulières à être supérieures à un seuil.

4) Résultats

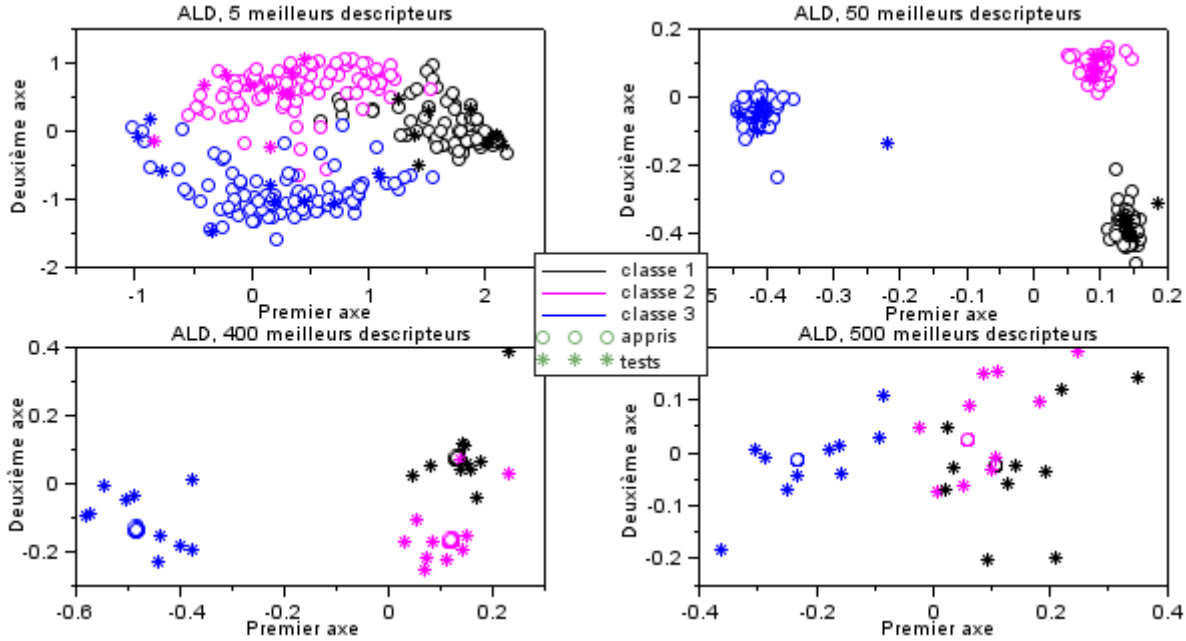
Comme attendu, l'ALD conserve mieux les classes que l'ACP, qui elle conserve mieux l'inertie totale :



Nous voyons sur cet exemple l'axe principal de l'ACP maximise bien l'inertie totale des individus, au détriment des classes, alors que l'ALD conserve au maximum les classes.

Lorsque le nombre de descripteurs est trop élevé, l'ALD montre son plus gros point faible: elle s'attache beaucoup trop aux données d'apprentissage, comme nous allons le voir sur le graphique suivant.

Graphique montrant l'évolution des projections des classes sur les deux axes discriminants de l'ALD en fonction du nombre de descripteurs considérés :



Avec trop peu de descripteurs, l'ALD n'arrive pas à séparer les classes (la variance intraclasse est du même ordre de grandeur que la variance globale). Mais nous voyons aussi qu'avec trop de descripteurs, l'ALD s'attache trop aux individus d'apprentissage (la variance intraclasse des individus tests devient nettement supérieure à celle des individus d'apprentissage). Nous pouvons le voir en regardant les différentes variances de l'exemple ci-dessus dans le tableau suivant.

Évolution des variances pour les projections (réduites) sur le premier axe :

	ALD 5 des.	ALD 50 des.	ALD 400 des.	ALD 500 des.	Idéal
σ^2_{intra} appris	10^{-1}	10^{-1}	10^{-1}	10^{-22}	égales
σ^2_{intra} tests	10^{-1}	10^{-1}	10^{2}	10^{3}	
σ^2_{inter} globale	10^{-1}	10^{1}	10^{3}	10^{2}	élevée

Parmi les 4 propositions ci-dessus, c'est la deuxième (avec 50 descripteurs) qui rempli le mieux les deux conditions souhaitées.

Si nous n'avions pas regardé les projections des individus tests, nous aurions choisie la dernière proposition : son résultat semble idéal, avec une variance intraclasse extrêmement faible (proche de l'erreur machine). Par contre, les individus n'ayant pas servi à l'apprentissage (les individus tests) sont projetés loin des autres individus de leur classe.

5) Conclusion

A la différence de l'ACP, l'ALD est une méthode supervisée, qui permet de conserver au maximum les classes durant la réduction de dimension. L'ALD est donc généralement plus efficace que l'ACP dans le cas de la classification supervisée, mais elle souffre davantage que l'ACP de la malédiction de la dimension. En effet, elle a alors davantage tendance à apprendre les données d'apprentissage par cœur que de créer un modèle général. C'est pourquoi, dans le cas d'un nombre de descripteurs trop élevé, il est parfois conseillé de combiner les deux méthodes, par exemple en utilisant d'abord l'ACP pour réduire le nombre de dimensions.

Comme l'ACP, l'ALD se base sur une hypothèse gaussienne. Dans le cas de données non gaussiennes, les SVM sont à privilégier.

C) SVMs

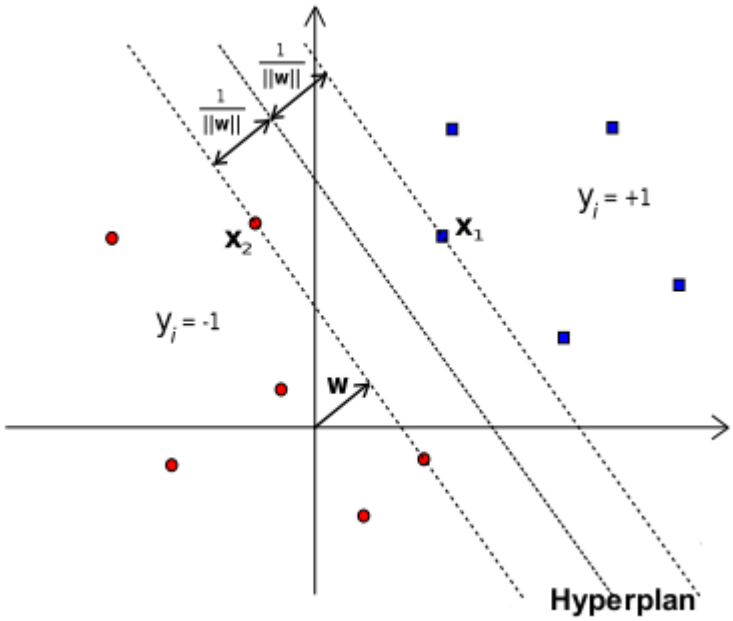
1) Introduction

Les SVM (Supports Vecteurs Machines) sont des algorithmes cherchant à séparer au mieux deux classes par un hyperplan. Contrairement aux méthodes précédentes, les SVM ne font pas d'hypothèse gaussienne : ils se basent uniquement sur les individus et leur classe.

Les SVM permettent d'attribuer une classe à l'aide de la projection sur un axe et sont en principe rangés parmi les méthodes de classification. Mais étant aussi des algorithmes de réduction de données par projection, nous les avons détaillés dans cette partie et uniquement cités dans la partie classification.

Dans le cas de classes linéairement séparables, les SVM ont pour but de trouver l'hyperplan passant entre les deux classes qui maximise la marge entre elles. Pour ce faire, il cherche tout d'abord les individus proches de l'hyperplan recherché (appelés vecteurs supports) puis leur attribut des poids (poids négatifs pour les vecteurs support d'une classe et positifs pour ceux de l'autre classe). L'hyperplan, enfin, est la combinaison linéaire des vecteurs supports pondérés du poids attribué. L'hyperplan ainsi créé passera donc entre les deux classes.

Cas idéal de séparation par SVM :



Il existe plusieurs façons de résoudre le problème des SVM. Nous allons voir tout d'abord deux algorithmes naïfs que nous avons implémentés puis la version avec résolution du problème d'attribution des poids à l'aide du Lagrangien.

2) Perceptron

Le perceptron ressemble à une introduction naïve aux SVM. Sa particularité est de créer l'hyperplan *w* de manière itérative, en le corrigeant à chaque itération selon les individus mal classés obtenus.

Algorithme :

Entrées : M, la matrice des données, et C, le vecteur des classes des individus,

Sorties : p, les projections des individus, et w l'axe de projection.

- w=0
- Tant que condition (n tours, variance de w, …) :
  - Pour chaque individu x_i :
    - Attribution d'une classe c_i à l'individu :
      - $p = w'.M'$
      - Choix d'un seuil b de séparation des deux classes
      - $c_i = c_1$ si $p_i >= b$ et $c_i = c_2$ si $p_i < b$
    - Si c_i différent de C_i
      - $w = w + y_i.x_i$, avec y_i =1 si $C_i = c_1$ et y_i = -1 sinon,
- $p = w'.M'$

Le choix de la classe est faite en fonction de la projection $w'.x_i$ ' et d'un seuil calculé en fonction de l'ensemble des projections.

Durant les tests, cet algorithme a montré des limites en présence de trop de descripteurs. Nous avons codé une variante de l'algorithme qui corrige toutes les erreurs de classification à la fois et qui ne conserve à chaque itération que les valeurs les plus élevées. Il est plus rapide et moins sensible aux descripteurs inutiles.

Algorithme :

Entrées : M, la matrice des données, C, le vecteur des classes des individus, et un seuil s $0 \leq s \leq 1$,

Sorties : p, les projections des individus, et w l'axe de projection.

- w=0
- Tant que condition :
  - Attribution des classes c :
    - $p = w'.M'$
    - Choix d'un seuil b de séparation des deux classes
    - Pour chaque individu i : $c_i = c_1$ si $p_i >= b$ et $c_i = c_2$ sinon
  - Soit I les indices des indices des individus i pour lesquels c_i différent de C_i
  - $m=y_I.x_I$, avec y_i =1 si $C_i = c_1$ et y_i = -1 sinon,
  - Suppression des valeurs de m plus faibles en absolu que max(|m|).s
  - $w = w + m$
- $p = w'.M'$

3) Sans itérations

Nous avons codé un autre algorithme, celui-là non rencontré dans la littérature, mais toujours basé sur l'idée générale des SVMs. Il détermine tout d'abord un ensemble d'individus proches de la marge, puis créé l'hyperplan à l'aide de la combinaison linéaire des individus sélectionnés, en leur attribuant des poids uniformes (dont la somme est nulle) mais de signe différent selon leur classe.

Algorithme :

Entrées : *M*, la matrice des données, et *C*, le vecteur des classes des individus,

Sorties : *p*, les projections des individus, et *w* l'axe de projection.

- Recherche de l'ensemble *I* des individus proches de la marge :
  - Première possibilité : par distance (euclidienne dans nos tests)
    - Pour chaque individu, recherche de l'individu le plus proche parmi les individus de l'autre classe
    - Ajout de l'individu à l'ensemble *I*.
  - Deuxième possibilité : par produit scalaire (plus rapide grâce aux produits matriciels)
    - Soit M_1 l'ensemble des individus de la première classe et M_2 ceux de la deuxième classe,
    - Calcul des produits scalaires de chaque individu de la première classe avec chaque individu de la deuxième classe : $\frac{x_i . x'_j}{\|x_i\| . \|x_j\|}$
    - Pour chaque classe, on a joute à *I* le ou les individus ayant obtenu le produit scalaire le plus élevé.
- Attribution des poids (choisis uniformes) aux individus sélectionnés :
  - Soit n_1 le nombre d'individus sélectionnés dans la première classe et n_2 celui dans la deuxième.
  - Le poids sont y_i=-1/n_1 si l'individu appartient à la première classe et y_i=1/n_2 sinon,
- $w = \sum_{i \in I} y_i . x_i$
- p = w'.M'

4) SVM

Nous arrivons enfin à une version plus officielle des SVM. En tant que classifieur, le but des SVM linéaires est de trouver $f(x) = w' . x + b$ tel que la classe de *x* dépende du signe de *f*(*x*).

La maximisation de la marge amène à devoir résoudre ce système :

$$\begin{cases} \min\limits_{w,b} \|w\| \\ y_i(w' . x_i + b) - 1 \geq 0 \quad \forall i \end{cases}$$

qui peut être résolu grâce au Lagrangien :

Pseudo code SVM (compatible MATLAB, SCILAB)[WIK] :

Entrées : *M*, la matrice des données, et *C*, le vecteur des classes des individus,

Sorties : *p*, les projections des individus, *w,* l'axe de projection, et *b*, scalaire, avec -b le seuil entre les deux classes.

- Recherche de l'ensemble *I* des individus proches de la marge : voir précédemment,
- Soit *y* tel que y_i = -1 si x_i appartient à la première classe et y_i=1 sinon, pour $i \in I$,
- Soit *e* le vecteur unité de la taille de *I*,
- Soit $X_I = diag(y(I)) . M(I, :)$,
- Soit $U = chol(X_I . X_I')$,
- Soit $a = U \backslash (U' \backslash e)$,

- Soit $c = U \backslash (U' \backslash y)$,
- Alors $b = (y'.a) \backslash (y'.c)$,
- Et $\alpha = U \backslash (U' \backslash (e - b.y))$,
- Annuler les coefficients négatifs de α, qui correspondent à des individus trop loin de la marge : $\alpha = \max(\alpha, 0)$,
- Enfin : $w = X_I'.\alpha$,
- $p = w'.M'$

5) SVM multi classes

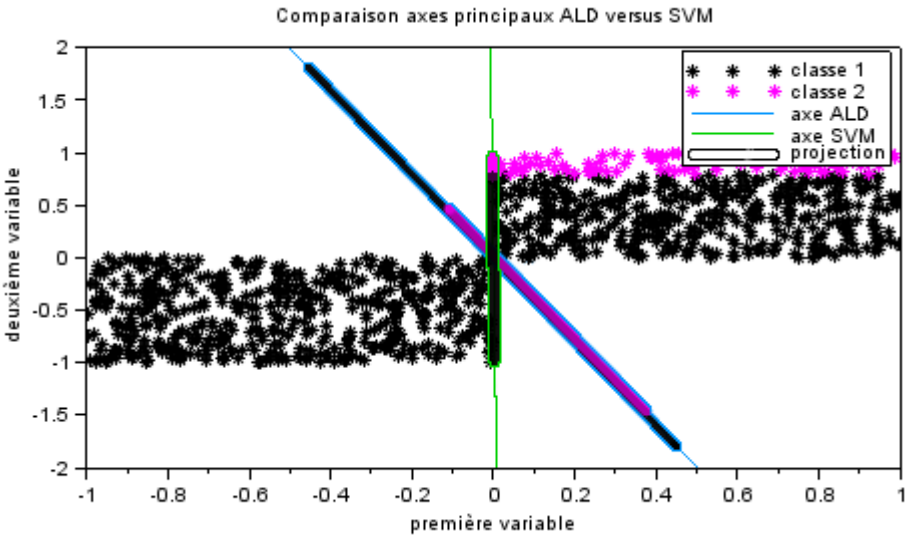
Le principe de base des SVM est de séparer deux classes. Dans le cas de plusieurs classes, il existe par exemple la méthode « un contre tous » : il s'agit de prendre les classes unes à unes et de trouver à l'aide des SVM l'hyperplan la séparant le mieux de toutes les autres classes. Nous obtenons ainsi au final autant d'hyperplans que de classes (ou le nombre de classe -1, la dernière séparation étant redondante aux autres).

6) Résultats et conclusion

Nos trois implémentations des SVM ont donné des résultats similaires en petite dimension, mais la dernière implémentation des SVM a donné des signes de faiblesse en présence de trop de descripteurs ou de descripteurs pas suffisamment adaptés au problème posé. Cela vient probablement du choix approximatif des vecteurs supports, qui a convenu pour la méthode simple que nous avons codé mais qui n'a visiblement pas été suffisamment précis pour la dernière implémentation, nécessitant normalement l'ensemble exact des éléments de la marge. Les méthodes itératives ont l'avantage non négligeable d'améliorer l'ensemble des vecteurs supports à chaque itération.

En terme de temps d'exécution, c'est notre version des SVM sans itération la plus rapide, puis les SVM avec la factorisation de Cholesky (10 fois plus lente) et enfin les algorithmes itératifs, dont le temps d'exécution varie selon le nombre d'itérations.

Les résultats offrent tout de même une réponse satisfaisante aux problèmes non gaussiens. C'est en effet dans les cas non gaussiens que les SVM montrent tout leur intérêt. Le graphique suivant permet de mieux visualiser :



L'axe de projection (orthogonal à l'hyperplan séparateur) le plus adapté est celui des SVM, qui n'a été créé qu'à partir des individus à la frontière entre les deux classes. L'ALD, elle, a pris en considération l'ensemble des individus. Néanmoins, dans le cadre de données gaussiennes, utiliser la moyenne et la variance de tous les individus de chaque classe est plus adapté que de ne s'intéresser qu'à quelques individus, qui ne sont que des réalisations possibles et bruitées.

III) GAs : les Algorithmes Génétiques

Les algorithmes génétiques se basent sur l'idée de mélange et de descendance. Il peuvent être appliqués dans beaucoup de domaines, notamment en classification dans la recherche du meilleur hyperplan séparateur.

Structure d'un algorithme génétique appliqué à la recherche de l'hyperplan idéal :

Entrées : *M*, la matrice des données, *W*, un ensemble d'hyperplans parents,

Sorties : *p*, les individus projetés, *w*, l'hyperplan généré.

- Tant que condition (nombre d'itérations, critère de convergence, …) :
  - Suppression des hyperplans égaux puis choix des 2 meilleurs hyperplans parmi l'ensemble *W* (selon erreurs d'attribution des classes par exemple ou des scores des projections (MMD, gain d'information, Fischer, …)),
  - Création des hyperplans enfants *W* à partir des deux parents choisis :
    - Croisement : mélange d'éléments des hyperplans parents,
    - Mutation : modification, suppression (rendre nul) ou permutation de quelques éléments d'un des deux hyperplans parents (selon un taux de descripteurs mutants).
  - Choix du meilleur hyperplan *w* parmi *W*,
  - $p=w'.M'$

Les GAs permettent de trouver le meilleur hyperplan à proximité d'un lot d'hyperplans donnés (ALD, ACP, SVM, perceptron, …) et permettent donc ainsi de s'assurer d'une certaine stabilité des résultats, que les classes aient une répartition gaussienne ou non, qu'il existe une marge entre les classes ou non. Il n'est pas nécessaire de trop pousser l'optimisation : un faible nombre de tours (une dizaine) permet d'obtenir des résultats concluants.

IV) Conclusion

La projection dans des sous-espaces propres des descripteurs n'est pas nécessairement obligatoire dans tous les contextes. Néanmoins, cela reste une étape importante de la classification, qui permet d'améliorer les résultats et de réduire les temps de calculs par la suite.

Comme pour tout, il n'existe pas de méthode meilleure que les autres, elles ont toutes leurs caractéristiques. C'est l'ALD qui nous a donné les meilleurs résultats dans le cadre de ce stage, suivie du perceptron.

Nous n'avons vu ici que les méthodes linéaires, qui ont semblé bien adaptées à notre problème de classification. Néanmoins, certains cas nécessitent l'utilisation de méthodes non linéaires (quadratiques, …). Il existe par exemple l'Analyse Quadratique Discriminante, mais la séparation non linéaire évoque principalement les SVM non linéaires, bien adaptées au problème.

Classification

I) Introduction

La classification se déroule en deux étapes : l'analyse des données d'apprentissage puis l'attribution de classes aux individus à classer.

Tout d'abord, nous regardons trois algorithmes de classification non supervisée, aussi utilisés en classification supervisée pour mieux approcher les formes hétérogènes des classes.

Puis nous nous intéresserons aux méthodes supervisées que nous pouvons regrouper en trois catégories : les méthodes basées sur les individus eux-mêmes (k-NN, …), les méthodes d'approximation des classes par des lois gaussiennes (NBC, GMM, …) et les arbres hiérarchiques (CART, …).

II) Classification non supervisée

Dans le cadre de la classification supervisée, les algorithmes de classification non supervisée peuvent servir à découper les classes connues en sous-ensembles d'individus naturellement regroupés. Cela permet par exemple de séparer les sons de violon joués avec l'archet de ceux joués en pinçant les cordes, classés tous les deux dans la même classe « violon ».

Pour séparer un ensemble d'individus en *n* classes, nous avons regardé deux types de méthodes différentes : les méthodes par centres mobiles (cartes de Kohonen, EM, k-means, k-medians, ...) et celles par agglomération (arbres ascendants).

Les algorithmes de type centres mobiles cherchent les paramètres (centres, variances, …) supposés des classes. Pour cela, ils dispersent *n* points qu'ils déplacent itérativement jusqu'aux centres supposés des classes, à l'aide de distances (cartes auto-organisatrices, k-means, ...) ou encore de probabilités (EM, ...).

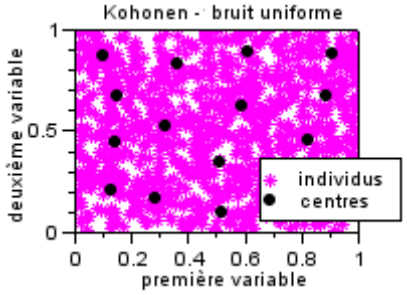
Les algorithmes de type arbre ascendant procèdent par agglomération des individus entre eux, jusqu'à obtenir le nombre de regroupements souhaités.

A) Cartes de Kohonen (Kohonen maps)

Les cartes auto-organisatrices de Kohonen créaient de manière itérative une cartographie d'un ensemble d'individus. Elles sont très utilisées pour la segmentation d'images, par exemple. Dans notre cas, ce sont les classes que nous souhaitons séparer.

Le principe des cartes des Kohonen est simple : à chaque itération, l'ensemble des individus est parcouru et chacun attire vers lui les centres les plus proches de lui. Les centres sont ainsi attirés vers les plus forts regroupements d'individus.

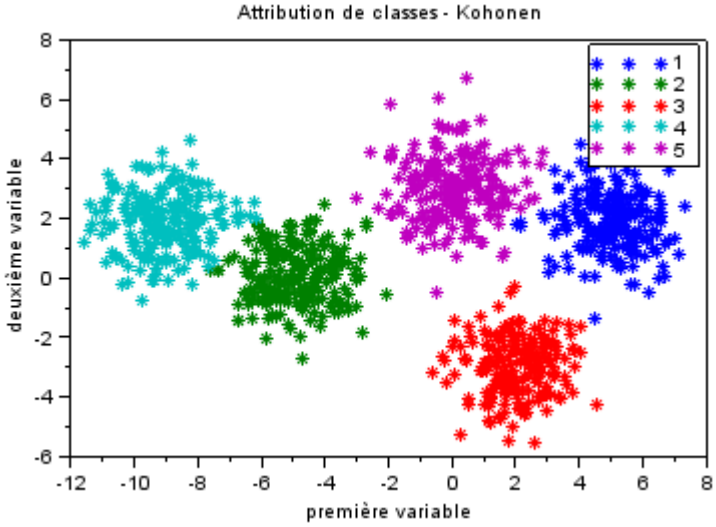
Exemple de répartition des centres des classes sur un bruit uniforme :



Nous voyons que les centres s'alignent pour former une grille (un peu tordue dans notre cas).

En appliquant l'algorithme sur un ensemble de classes bien délimitées, les centres vont se placer au centre de chaque classe. La difficulté réside dans le choix du nombre de classes recherchées. Différents nombres de classes peuvent être testés et le nombre maximisant un critère est alors sélectionné.

Voici un exemple de classification (centres des classes trouvés par les cartes de Kohonen, puis l'attribution des classes aux individus a été faite par maximum de densité de probabilités gaussienne) :



L'exemple est simple et le résultat est celui attendu. Des problèmes peuvent être rencontrés en grande dimension si de nombreux descripteurs inutiles parasitent les distances entre les individus et les centres des classes. D'autres distances que la distance euclidienne peuvent être utilisées pour parer à ce problème.

Dans le cadre d'estimation des classes, les cartes de Kohonen sont appropriées si toutes les classes recherchées suivent à peu près la même loi de probabilité et ont le même nombre d'individus. Sinon, il est plus adapté de se tourner vers des algorithmes tels que l'algorithme EM.

B) EM, k-means, ...

L'algorithme EM (Expectation – Minimization / Espérance - Minimisation) estime à chaque récurrence les paramètres (moyenne, variance) des lois de probabilité gaussiennes des différentes classes estimées. Sa récurrence comporte deux étapes : l'estimation (de la classe supposée pour chaque individu) puis la minimisation (des distances entre les centres des classes et les individus associés aux classes).

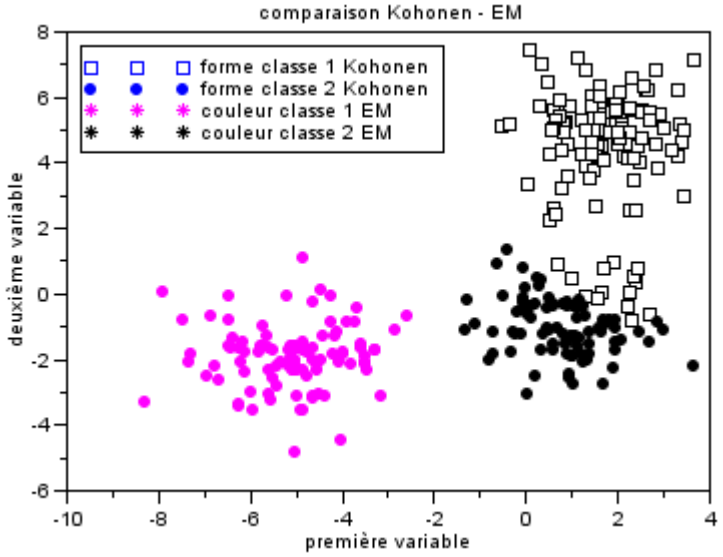
- Estimation :
  - $\mathcal{N}(x_i|\mu_k^{t-1},\Sigma_k^{t-1}) = \frac{1}{\sqrt{\det(2\pi\Sigma_i)}}\exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right)$
  - $\alpha_{ki} = \frac{\omega_k^{t-1}\mathcal{N}(x_i|\mu_k^{t-1},\Sigma_k^{t-1})}{\sum_{k=1,\dots,K}\alpha_{ki}}$
- Minimisation :

- $\omega_k^t = \frac{\sum_{i=1,\dots,n} \alpha_{ki}}{\sum_{i=1,\dots,n} \sum_{k=1,\dots,K} \alpha_{ki}}$
- $\mu_k^t = \frac{\sum_{i=1,\dots,n} \alpha_{ki} x_i}{\sum_{i=1,\dots,n} \alpha_{ki}}$. Ici, d'autres calculs que la moyenne peuvent être utilisés pour estimer les centres (voir algorithme suivant),
- $\Sigma_k^t = \frac{\sum_{i=1,\dots,n} \alpha_{ki} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1,\dots,n} \alpha_{ki}}$

L'algorithme k-means et ses variantes (k-medians, …) sont des versions allégées de l'algorithme EM : seule la moyenne (ou la médiane) est prise en compte, pas la variance. Il n'y a pas de calcul de probabilité à faire, uniquement les centres des classes à estimer :

- Estimation des classes des individus :
  - Calcul des distances entre les individus et les centres,
  - Pour chaque individu, affectation de la classe dont le centre est le plus proche.
- Minimisation des distances entre les centres des classes et les individus qui leurs sont attribués :
  - Les centres peuvent être le vecteur **moyen** (k-means), **médian** (k-medians), ou encore le **centre du cube** ((max-min)/2) ou **de la boule** (centre trouvé de manière itérative, en déplaçant à chaque itération le centre vers l'individu le plus éloigné de lui) des individus de la classe.

Comparaison avec les cartes de Kohonen, en cherchant à séparer en deux 3 classes différentes :



La séparation en deux obtenue par l'algorithme EM semble plus adaptée (il ne sépare pas en deux le groupe du milieu).

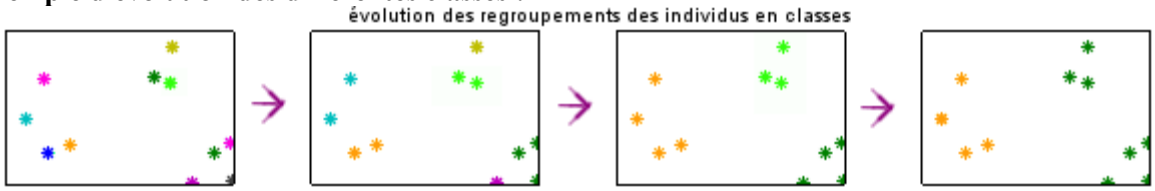
C) Arbres ascendants

Les arbres ascendants ont un principe différent : ils ne partent pas d'un ensemble qu'ils divisent mais d'individus qu'ils agglomèrent en minimisant divers critères. Les arbres ascendants partent donc des feuilles de l'arbre (chaque individu considéré comme une classe), puis regroupent les classes entre elles jusqu'à arriver à la racine de l'arbre (une seule classe). Pour choisir les groupes à agglomérer à chaque nœud de l'arbre, ils peuvent minimiser divers critères, chacun avec son avantage:

- Méthode de Ward [ESC08] : minimisation de l'augmentation de l'inertie intra-classe :

- Augmentation de l'inertie par fusion des classes c_k et c_l = $\frac{n_k . n_l}{n_k + n_l} d^2(g_k, g_l)$, avec g_k et g_l les centres de gravité respectifs des classes c_k et c_l.
- Déplacement minimum des centres,
- Saut minimum (distance minimale entre deux individus des classes) : $\min_{k,l}(\min_{x_k, x_l}(d(x_k, x_l)))$,
- Minimisation du diamètre (distance maximale entre deux individus des classes) : $\min_{k,l}(\max_{x_k, x_l}(d(x_k, x_l)))$,
- Minimisation de la distance moyenne $\min_{k,l}\left(\sum_{x_k, x_l} d \frac{(x_k, x_l)}{n_k . n_l}\right)$,
- Minimisation de la distance moyenne après fusion.

Exemple d'évolution des différentes classes :



Les arbres les plus rapides sont ceux ne s'intéressant qu'à un nombre fixé de valeurs par classe (moyenne, nombre d'individus, …). Les arbres tels que l'arbre du saut minimal oblige à comparer pour chaque classe toutes les distances entre les individus de la classe et ceux des autres classes, pour trouver la distance minimale. Pour de grandes bases de données, c'est beaucoup trop coûteux en temps.

Aucune hypothèse (gaussienne, …) n'est faite sur les classes recherchées, les arbres ascendants ont l'avantage de s'intéresser très localement à la répartition des individus. Ils peuvent obtenir des formes de classes beaucoup plus hétérogènes qu'avec les algorithmes de types centres mobiles, mais aussi plus proches de la réalité.

D) Utilité en classification supervisée

Ces algorithmes sont utilisées à l'intérieur des méthodes de classification, pour séparer les individus d'une même classe en plusieurs groupes qui dessinent mieux la forme de la classe (GMM,...) ou encore pour regrouper les classes en plusieurs groupes de classes (arbres de décision), comme nous le verrons plus tard.

Ces algorithmes peuvent aussi être utilisés pour jauger de la difficulté d'un problème de classification, en cherchant à retrouver avec leur aide les classes connues.

III) Classification supervisée : méthodes « instance-based » : K-NN et ses dérivées

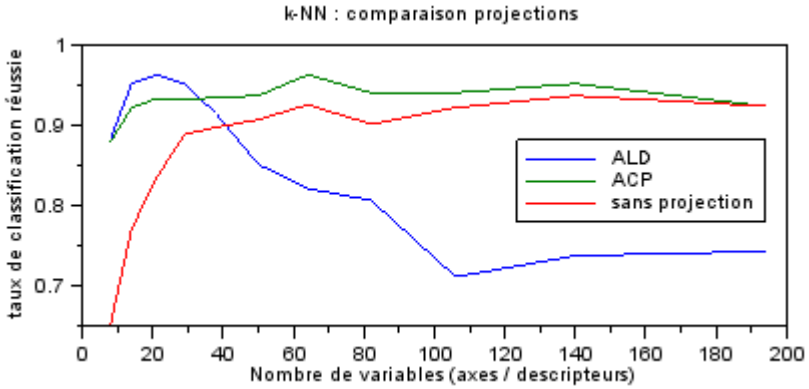
L'algorithme k-NN (k-Nerest Neigbourds - k-PPV : k Plus Proches Voisins) classe les individus selon les classes des *k* individus d'apprentissage les plus proches d'eux.

Ainsi, pour chaque individu à classer, les distances avec les individus de classe connue sont calculées et les *k* plus proches voisins sont conservés, généralement avec k=1, 3, 5 ou 7. La classe qui apparaît le plus chez ses *k* voisins est attribuée à l'individu.

La distance utilisée pour le k-NN standard est la distance euclidienne, mais d'autres distances peuvent être utilisées : la distance de Mahalanobis ou encore l'entropie (K*). Pour plus de renseignements sur les distances, nous pourrons regarder l'annexe II) Distances et similarités.

Nous avons fait plusieurs tests, avec ou sans l'utilisation de méthodes de projection, et les différentes distances obtiennent en moyenne des résultats similaires. Les deux distances qui semblent parfois se démarquer sont la distance de Manhattan (somme des valeurs absolues) et la distance de Canberra (basée sur la différence divisée par la somme).

L'utilisation d'une méthode de projection a davantage d'influence sur les résultats. Nous avons comparé l'algorithme k-NN sur des données projetées (les premiers axes de l'ACP et de l'ALD) et sur les descripteurs (sans projection) :



Nous remarquons que seuls les premiers axes de l'ALD sont utiles à la classification, les autres la perturbent et font baisser son taux de classification. Le nombre d'axes principaux de l'ACP conservés à moins d'influence sur les résultats : au-dessus de 15 axes conservés, la courbe varie lentement entre 90% et 96% de classification. Sans projection, il y a en toute logique besoin de davantage de descripteurs que d'axes de projection conservés : le maximum (98%) est atteint vers 600 descripteurs. Ces trois remarques ne sont pas spécifiques à la classification par les plus proches voisins, cela se remarque aussi avec les mélanges de gaussiennes.

L'algorithme k-NN est le plus basique qui soit en classification et ne demande pas de phase d'apprentissage (il conserve les individus d'apprentissage tels quels). De plus, il donne généralement de très bon résultats et reste une référence connue. Mais il demande en contrepartie beaucoup de ressources pour la phase d'attribution des classes : il nécessite la conservation de toute la base de données d'apprentissage en mémoire et beaucoup de calculs à chaque nouvelle attribution de classe, pour comparer chaque nouvel individu avec toute la base d'apprentissage. Ce problème le rend incompatible avec de nombreux problèmes industriels et laisse le champs libre aux autres algorithmes de classification, plus subtils.

IV) Classification supervisée : méthodes statistiques

A) NBC

Le NBC (Naïve Bayesian Classifier / Classifieur Bayésien Naïf) est la dernière étape de l'algorithme EM : nous avons les classes (donc leurs moyennes et variances) et nous estimons les densités de probabilités des individus à classer pour chaque classe. Nous attribuons ensuite aux individus la classe la plus probable.

La phase d'apprentissage est rapide et la quantité de données nécessaire à la phase d'attribution des classes est faible (la moyenne et la variance de chaque classe).

Malheureusement, les résultats varient considérablement selon les problèmes de classifications étudiés, les classes ne suivant pas toutes une densité de probabilité gaussienne. C'est pourquoi des mélanges de plusieurs gaussiennes par classe (GMM) permettent de mieux représenter les différentes formes des classes.

B) GMM

Les algorithmes de mélange de gaussiennes [PIE09] (Gaussian Mixture Model - GMM) attribuent plusieurs gaussiennes à chaque classe de manière à représenter au mieux la répartition des individus, elles peuvent ainsi dessiner avec plus de précision la marge entre les classes.

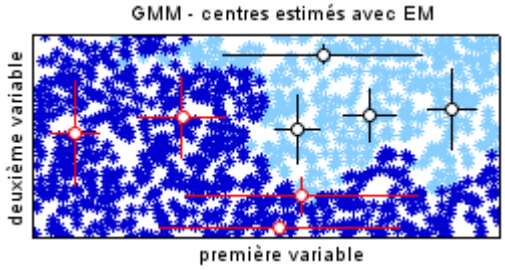
L'estimation des paramètres (moyennes, variances) se fait généralement à l'aide de l'algorithme EM, mais d'autres estimation des centres peuvent être utilisés, comme k-means, k-medians, cartes de Kohonen, … Il existe d'autres modèles, plus élaborés et constitués de davantage de paramètres. Nous pourrons par exemple nous référer à [BOU09] pour un premier aperçu.

Le choix du nombre de classes est difficile et peut être résolu à l'aide de critères de pénalisation tels que le critère BIC. La pénalisation du nombre de classes est primordial dans le cas de la classification non supervisée mais est moins indispensable à l'algorithme GMM : chaque gaussienne ne représente pas une classe estimée mais une particularité d'une classe déjà connue. Avoir trop de gaussiennes n'est pas un problème en soi, en regardant le cas extrême nous voyons qu'une gaussienne par individu nous ramènerait à l'algorithme k-NN.

L'attribution d'une classe à un individu à classer se fait toujours en estimant la probabilité de chaque classe pour l'individu, mais ce calcul peut être effectuée de plusieurs façons différentes. Il faut tout d'abord estimer la

probabilité due à chaque gaussienne puis combiner les probabilités des mêmes classes. Différentes stratégies sont alors possibles : conserver la probabilité la plus élevée, calculer la probabilité moyenne, …

Exemple de répartition de gaussiennes sur deux classes, avec 4 gaussiennes par classe :

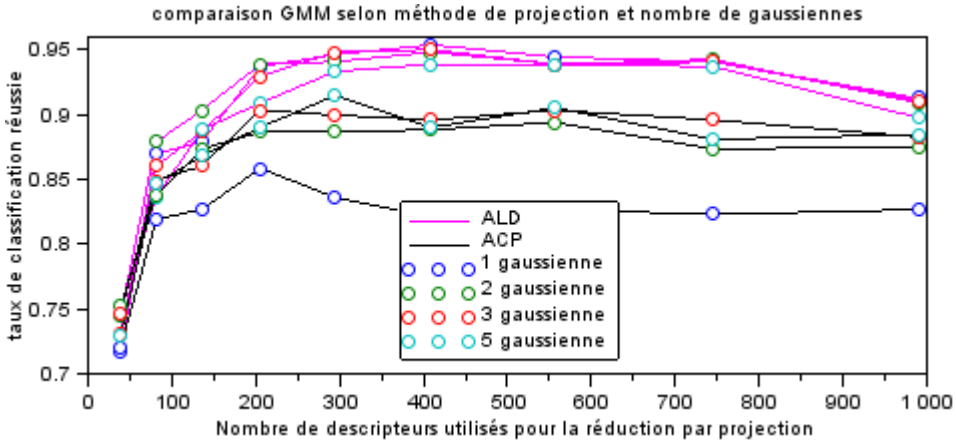


Les formes étranges des deux classes ont été choisies exprès pour l'exemple mais ne coïncident avec aucune répartition d'individus constatée durant ce stage. Pour rendre les algorithmes NBC et surtout GMM efficaces, il est préférable de projeter auparavant les données à l'aide de méthodes faisant des hypothèses gaussiennes (ACP ou ALD).

C) Résultats

Nous avons comparé la classification GMM des 21 (nombre de classes -1) premiers axes des données projetées avec l'ACP ou l'ALD à partir d'un nombre croissant de descripteurs. L'ALD et l'ACP obtiennent des résultats similaires pour un nombre faible ou élevé de descripteurs sélectionnés mais l'ALD a l'avantage quand le nombre de descripteurs est optimal pour elle (voir partie ALD pour le choix du nombre de descripteurs).

Nous avons testé plusieurs nombres de gaussiennes pour l'algorithme GMM : l'ACP préfère au moins 2 gaussiennes par classe alors que l'ALD y est très faiblement sensible. Prendre plus de 2 gaussiennes par classe ne semble pas nécessaire dans notre cas de classification :



Les algorithmes de mélange de gaussiennes (NBC, GMM) nous permettent d'obtenir jusqu'à 95% de classification réussie, avec un coût réduit (peu de temps de calculs et peu de données apprises en mémoire) et une phase d'apprentissage pas non plus très longue.

La recherche du nombre idéal de gaussiennes pour chaque classe allongerait la durée de la phase d'apprentissage mais améliorerait peut-être les résultats.

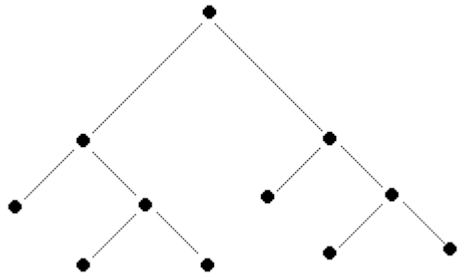
V) Classification supervisée : arbres de décision

A) Introduction

Les arbres de décision, ou encore arbres hiérarchiques, découpent l'espace des données en sous espaces auxquels ils attribuent à chacun une classe. Pour classer un nouvel individu, les arbres hiérarchiques trouvent le sous-espace auquel il appartient et lui attribuent sa classe.

Les arbres hiérarchiques sont composés de nœuds, de branches et de feuilles. A chaque nœud, les individus d'apprentissage sont séparés en plusieurs groupes, dirigés chacun par une branche vers un nœud enfant. Les feuilles sont les nœuds terminaux, où tous les individus (ou seulement la majorité s'il y a un élagage de l'arbre) appartiennent à la même classe. Les élagages des arbres permettent de ne pas trop s'attacher aux individus les plus

éloignés de leur classe, qui peuvent parasiter la classification s'il leur est accordé trop d'importance.



Les arbres peuvent être des arbres binaires (séparation en deux groupes à chaque nœud) ou *n*-aires (séparation en *n* groupes).

Au niveau des nœuds, les individus peuvent être séparés de plusieurs façons, c'est principalement ce qui différencie les algorithmes de création d'arbres hiérarchiques entre eux.

Structure d'un algorithme récursif de création d'un arbre :

Entrées : *N*, nœud racine de l'arbre, contenant tous les individus et toutes les classes,

Sortie : *N*, structure de l'arbre créé.

- Si le nœud est une feuille (s'il ne contient qu'une seule classe ou si les autres classes sont suffisamment minoritaires (<5 individus par exemple) pour être ignorées) :
  - Attribution de la classe à la feuille,
  - Fin (on retourne la feuille)
- Sinon :
  - Choix des séparations des individus en plusieurs groupe,
  - Application de l'algorithme à chaque branche créée,
- Fin.

B) Exemple d'arbre : CART

A chaque nœud, l'arbre binaire CART sépare les individus en deux uniquement à l'aide d'un des descripteurs. Un seuil est fixé et les individus dont la valeur du descripteur est supérieure au seuil sont dirigés vers un premier nœud enfant, les autres vers un deuxième.

Le choix du descripteur et du seuil sont faits en maximisant la mesure de qualité de la scission, calculée pour tous les descripteurs et toutes les séparations de classes possibles.

Mesure de la qualité d'une scission :

$$\Phi(s|t) = 2\, n_D n_G \sum_{c_k \in C} |p(c_k|t_G) - p(c_k|t_D)|$$

Avec : t_D et t_G les deux nœuds enfants du nœud t, et $p(c_k|t_D)$ la probabilité calculée empiriquement à l'aide des répartitions des individus : $p(c_k|t_D) = \frac{n_{k_D}}{n_k}$.

C) Exemples de séparation binaire

1) Choix des deux ensembles

Tester toutes les combinaisons possibles de classes n'est envisageable que pour un faible nombre de classes. Pour réduire le nombre de combinaisons, la méthode « un contre tous » peut être choisie : il s'agit de tester pour tout k : c_k contre $\{C \setminus c_k\}$.

La séparation en deux peut aussi être effectuée par des méthodes de classification non supervisée : cartes auto-organisatrices, algorithme EM, arbres ascendants, … Réduire l'ensemble de descripteurs sur lequel les

méthodes de classification non supervisée sont appliquées permet de les rendre moins sensibles au bruit. Leur application sur les axes principaux de l'ALD est particulièrement intéressante.

2) Séparation des deux ensembles choisis

Pour séparer de manière binaire, toutes les méthodes vues précédemment peuvent être utilisées et combinées.

Avec un seul descripteur :

Tous les descripteurs peuvent être testés, comme pour l'arbre CART, ou bien celui utilisé peut être choisi par un algorithme de sélection de descripteurs.

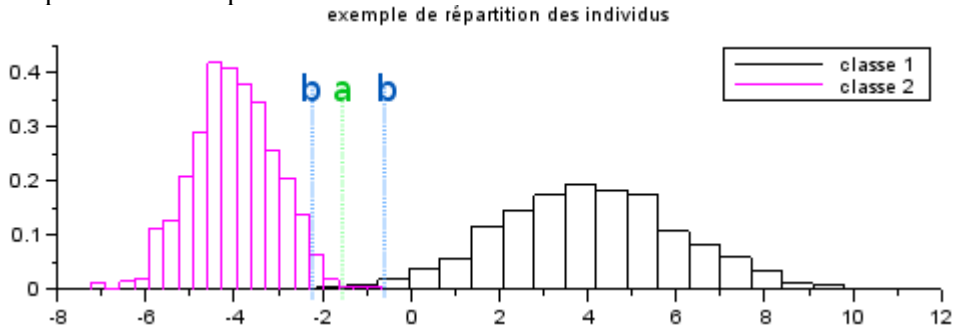
Avec plusieurs descripteurs :

Des classifieurs (SVMs, ALD binaire, …) peuvent être utilisés pour projeter les individus sur un axe. Ici, le calcul de plusieurs axes différents et l'optimisation par algorithmes génétiques est intéressant coûteux.

Choix des seuils :

Une fois les individus projetés sur un axe, il faut trouver le seuil de séparation en deux idéal. Lorsque la séparation ne peut pas être parfaite, plusieurs choix différents peuvent être faits.

Exemple de répartition non séparable de deux ensembles d'individus :



Le seuil peut être calculé de telle sorte que chacun des deux nœuds enfants aient le moins d'impuretés. Pour cela, les moyennes et variances peuvent être utiliser pour calculer la frontière probable entre les deux classes (**a**). Un histogramme peut aussi être utilisé, en recherchant le creux entre les deux ensembles.

Il peut s'avérer plus intéressant de séparer de manière à conserver une deux deux classes entière (**b**). Ainsi, un des deux nœuds enfant n'aura aucune impureté et les impuretés de l'autre nœud seront moins isolées.

Le seuil peut être calculé de nombreuses façons différentes, comme par exemple à l'aide des SVM qui définissent leur propre seuil.

Optimisation des descripteurs

Lorsqu'aucune séparation ne donne satisfaction (deux ensembles non séparables), la recherche de la classification parfaite peut pousser à appliquer l'algorithme de recherche automatique de descripteurs EDS aux deux ensembles d'individus que le nœud souhaite séparer. Pour des raisons de temps (le système EDS étant très coûteux en temps, comme nous l'avons déjà dit), nous n'avons pas testé. C'est néanmoins dans ce type de cas (classes non séparables avec les descripteurs analysés manuellement) que le système EDS est le plus intéressant.

3) Résultats

La première chose à remarquer est le temps d'apprentissage extrêmement long. Plus les données, sont mauvaises ou insuffisantes, plus la phase d'apprentissage est longue, de l'ordre de plusieurs heures. Les arbres hiérarchiques fonctionnant avec des algorithmes de sélection de variables embarqués, ils ne souffrent que très faiblement de la malédiction de la dimension et il n'est pas nécessaire de faire une première sélection des variables. Cela cause même du tord car les arbres ont alors moins de choix et sont moins efficaces

Les diverses projections calculées à chaque nœud de l'arbre coûtent très cher et calculer celles possibles à l'avance a permit d'améliorer notablement les résultats et donc le temps de création de l'arbre, tombé à une petite dizaine de minutes. Les projections calculées ont été toutes les projections multi-classes décrites dans la partie précédente : ACP, ALD, SVM multiclasses. Rajouter de nombreuses variables inutiles n'a pas posé de problème à l'arbre, toutes les séparations binaires des données d'apprentissage ont étés parfaites.

Les taux de classification avoisinent les 90%, peut-être à cause de sur-apprentissage (utilisation de l'ALD).

VI) Méthodes : comparaisons et mélanges

Tableau comparatif des méthodes, pour une base de données de taille m (descripteurs) x n (individus), avec K

classes :

	k-NN	GMM	Arbres Hiérarchiques
Meilleures classifications	98%	95%	88%
Nombre variables d'apprentissage	moyen	faible	élevé
Apprentissage : coûts calculatoires	0	O(m*n)	Arbre idéal : N^p complet
Poids données apprises	O(m*n)	O(K*m)	-
Attribution : coûts calculatoires	O(m*n)	O(K*m)	O(log(K)*m)

En conclusion, l'algorithme k-NN ne nécessite aucun apprentissage et a obtenu les meilleurs résultats durant notre stage, mais est coûteux à l'attribution des classes. Les arbres hiérarchiques, eux, ont une phase d'apprentissage très coûteuse et nécessitent beaucoup de variables d'apprentissage de bonne qualité pour donner de bons résultats. Les mélanges de gaussiennes sont un bon compromis : elles sont portables, rapides, et obtiennent de bons taux de classification.

Combiner les 3 méthodes pour améliorer les résultats peut être envisagé. Durant nos tests, cela a permis à coup sûr de se placer parmi les meilleurs résultats, mais jamais de dépasser le meilleur. Une étude des erreurs nous a montré que les algorithmes se trompaient généralement pour les mêmes individus.

VII) Conclusion

Les différentes méthodes de classification cherchent à attribuer la classe dont les caractéristiques sont les plus proches de l'individu à classer. Pour ce faire, elles peuvent comparer l'individu à classer avec tous les individus de la base d'apprentissage (k-NN), estimer à quelle gaussienne représentant une classe il appartient (GMM) ou encore le décortiquer de manière à réduire le nombre de classes auxquelles il peut appartenir, jusqu'à ce qu'il n'en reste qu'une (Arbres hiérarchiques).

Ces méthodes sont plus ou moins coûteuses, et le choix de la méthode à utiliser doit le prendre en considération. Les modèles de mélanges gaussiens semblent un bon compromis, avec un taux maximum obtenu de 95%. La méthode des plus proches voisins a obtenu les meilleurs résultats dans le cadre de notre stage, en frôlant les 98%.

Ces chiffres ont principalement été obtenus grâce à la prise en compte poussée de l'évolution temporelle des signaux, nous étions restés plafonnés à 93% avant leur analyse. Après tout, cela n'est pas dénué de logique : une seule valeur d'un signal sonore n'apporte aucune information si elle est isolée. Ce sont les valeurs alentours qui lui donnent un sens, ceci à petite comme à plus grande échelle.

Seules les méthodes de classification supervisée les plus connues ont été étudiées durant ce stage. Il en existe de nombreuses autres, mais toutes basées sur les trois principes de base évoqués plus haut : distance avec les individus de la base d'apprentissage, lois de probabilités ou arbres parfois aidés de classifieurs binaires (SVM, …).

Analyse-synthèse sonore

I) Introduction

L'analyse et la synthèse sonores sont très utilisées dans la vie quotidienne (en téléphonie, musique, …) et ont de nombreuses applications, telles que la compression, la reconnaissance vocale, …

L'analyse-synthèse peut être paramétrique (LPC, …), semi-paramétrique ou encore non paramétrique (Ondelettes, …). Dans le cadre de ce stage, nous nous sommes plus précisément intéressé à l'analyse-synthèse paramétrique, qui nous a permis d'envisager des applications telles que la reconnaissance de locuteur, à l'aide de notre programme de classification vu précédemment. De plus, les paramètres analysés, ajoutés aux descripteurs, nous ont permis de légèrement améliorer notre taux de classification.

La synthèse peut aussi nous permettre de vérifier la qualité des descripteurs audio analysés. En effet, si un ensemble de descripteurs nous permet de synthétiser des sons extrêmement proches des originaux, alors ils doivent aussi permettre de classifier les sons avec au moins autant de réussite que si la classification était réalisée par un humain.

II) Modèles de synthèse

A) Synthèse additive pour les sons d'instrument de musique

1) L'analyse

Pour simplifier le modèle, nous avons considéré que les propriétés sonores analysées restaient constantes sur toute la durée des notes jouées (sauf l'intensité). Nous avons donc analysé et synthétisé sans fenêtrage du signal.

Composition d'un son :

- Une durée,
- Dans le domaine temporel :
  - une enveloppe temporelle.
- Dans le domaine fréquentiel :
  - une fréquence fondamentale,
  - un nombre d'harmoniques,

- un coefficient d'inharmonicité,
- un rapport d'intensité entre les harmoniques paires et les impaires,
- une composante de bruit,
- une enveloppe spectrale.

La quantité d'informations et la complexité du modèle peuvent être réduites en ne conservant comme paramètres fréquentiels que la fréquence fondamentale, la composante de bruit et l'enveloppe spectrale.

2) La synthèse

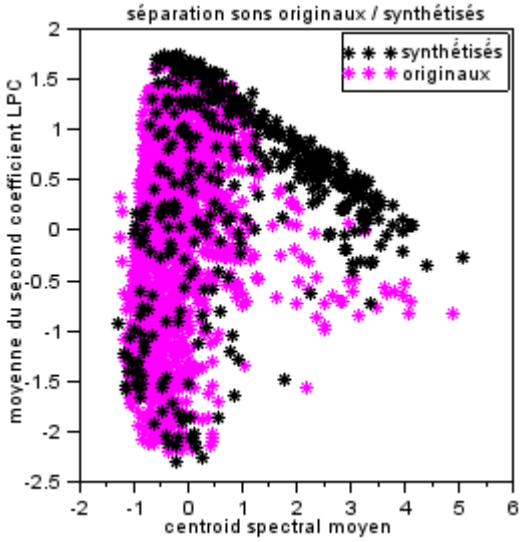
Structure de l'algorithme de synthèse :

- Initialisation : vecteur nul de la taille souhaitée
- Création du spectre :
  - Partie harmonique : Diracs aux emplacements de la fréquence fondamentale et de ses harmoniques
  - Partie bruitée : convolution (conservant la taille) avec un vecteur de bruit blanc modulé par une fenêtre de Hanning dont la taille dépend du taux de bruit.
  - Enveloppe spectrale : signal projeté dans le domaine temporel (à l'aide de la DCT) puis filtrage LPC.
- Enveloppe temporelle :
  - Création de l'enveloppe : $x^{a_1}.e^{-a_2.x}$
  - Modulation du son par enveloppe.

3) Résultats

A l'écoute, les sons synthétisés sont assez proches des sons originaux. Nous avons cherché à vérifier la qualités des sons synthétisés en demandant à notre programme de classification de séparer l'ensemble des sons synthétisés de l'ensemble des sons originaux. Ces deux classes (de plusieurs centaines d'individus) ont été séparées avec plus de 99,9% de réussite.

Nous avons donc cherché à comprendre en regardant les deux descripteurs les plus efficaces pour séparer les sons synthétisés des sons originaux (le centroïd spectral et le deuxième coefficient LPC) :



Le centroïd spectral est globalement plus élevé pour les sons synthétisés : nous savons que notre modèle surévalue légèrement le nombre d'harmoniques et fait l'impasse sur certaines très basses fréquences (variations temporelles du son) peu audibles ou inaudibles. Ce n'est donc pas étonnant.

Le deuxième coefficient LPC lui aussi est globalement plus élevé pour nos sons synthétisés. Des coefficients LPC plus élevés signifient des coefficients d'auto-corrélation plus élevés et donc un signal moins bruité. Nos sons synthétisés sont donc moins naturels que les sons originaux.

B) Synthèse LPC pour les sons vocaux

Le modèle de synthèse LPC est plus adapté à la synthèse vocale que le modèle présenté précédemment,

principalement grâce au découpage en fenêtres du signal. En plus des n premiers coefficients LPC ($2 \leq n \leq 20$), il a besoin de l'analyse de trois paramètres : un booléen indiquant si le signal est voisé ou non, la période (/fréquence) et l'intensité.

L'analyse-synthèse LPC d'une fenêtre n'est pas indépendante des fenêtres voisines.

Lors de l'analyse, pour éviter les erreurs, ou tout du moins réduire leur nombre, il faut rendre cohérents entre eux les paramètres analysés : si une période de 300 est trouvée au milieu d'un lot de périodes de 150, c'est que c'est une harmonique qui a été détectée.

Lors de la synthèse, la création du vecteur d'excitation (un peigne de Dirac ou un bruit blanc gaussien) d'une fenêtre ne peut pas être indépendant de celui de la fenêtre précédente : le recouvrement doit placer les Diracs des deux fenêtres aux mêmes emplacements pour qu'il n'y ait pas de superposition de fréquence.

Algorithme d'analyse LPC :

Entrée : y, une fenêtre du signal,

Sorties : a, les coefficients LPC, v, le booléen voisé/non voisé, f, la période, et I, l'intensité.

- Soit n=10 le nombre de coefficients LPC analysés,
- v :
  - Calcul du taux de passage par zéro du signal (ZCR),
  - v=ZCR<1/10.
- a : calcul des coefficients LPC (coefficients du filtre auto-regressif d'ordre n),
- Soit e l'erreur obtenue après filtrage inverse du signal,
- I = norme(e),
- f :
  - calcul de l'auto-corrélation de e,
  - création d'un vecteur des sommets locaux,
  - La période correspond au premier sommet local de ce vecteur.
- fin.

Une fois la fréquence et le booléen v obtenus pour toutes les fenêtres, une correction des erreurs potentielles est conseillée. Pour cela, on peut pour chaque fenêtre prendre la médiane des valeurs analysées sur les fenêtres voisines (médiane de t-2 à t+2 par exemple).

Algorithme de synthèse LPC :

Entrées : a, les coefficients LPC, v, le booléen voisé/non voisé, f, la période, I, l'intensité, t, la taille de la fenêtre, i_d, l'emplacement du premier Dirac, et y_t, les n valeurs du signal précédant la fenêtre,

Sorties : y, la synthèse d'une fenêtre du signal, et i_d, l'emplacement du prochain Dirac.

- Création du vecteur d'excitation :
  - Si v (si le signal est voisé) :
    - Initialisation à un vecteur nul de taille t,
    - Écriture de Diracs de i_d à t par pas de f,
    - Soit d l'indice du dernier Dirac écrit, alors : $i_d = f - d$.
  - Sinon :
    - Initialisation à un vecteur de bruit blanc gaussien de taille t.

- Multiplication des moitiés touchant une fenêtre voisée par une demie fenêtre de Hanning (pour éviter de trop importantes variations pouvant décentrer le signal synthétisé),
- $i_d = 1$ (ou 0 selon les conventions utilisées pour le premier indice d'un tableau).

- Multiplication du signal d'excitation par I.

- Synthèse du signal y par un filtrage auto-regressif.

Les résultats sont plutôt bons mais pas parfaits. Nous pouvons remarqué que les personnes semblent parler avec le nez bouché : nous avons utilisé un filtre AR, qui ne modélise qu'un seul conduit vocal (la bouche). Il faut s'intéresser aux modèles ARMA pour parer à ce problème.

C) Synthèse avec MFCC

La synthèse LPC est bien adaptée à la synthèse vocale mais moins à la modification vocale : la modification des coefficients LPC créé généralement un filtre à réponse impulsionnelle infinie. La correction basique de se défaut (du aux rayons>1 des racines du polynôme formé par les coefficients LPC) entraine la modification de la sonorité des sons synthétisés.

Les coefficients MFCC sont les premières valeurs du spectre d'un spectre modifié. Leur modification n'entrainera pas de problème d'explosion du signal et laisse plus de possibilités, mais nos résultats n'ont pas été concluants pour la modification, seule l'analyse-synthèse a été satisfaisante.

III) Conclusion

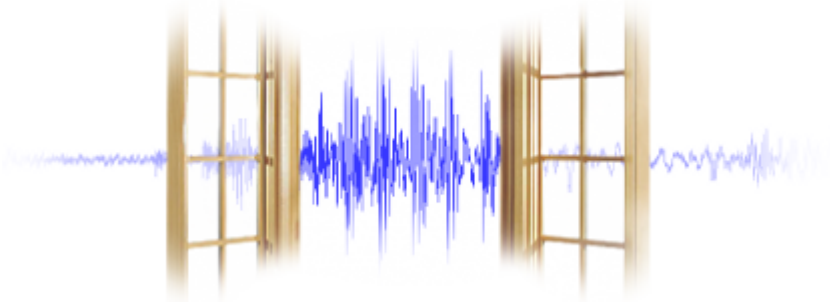
Par manque de temps, cette partie n'a pas été autant approfondie que nous aurions pu l'espérer. Néanmoins, elle nous a permit d'obtenir des descripteurs supplémentaires : paramètres de l'enveloppe temporelle, fréquence fondamentale, coefficients LPC et coefficients calculés sur la Transformée en Ondelettes (que nous n'avons pas développée ici car la Transformée en Ondelettes est une méthode d'analyse-synthèse non paramétrique). Tous ces descripteurs se sont montrés efficaces pour la classification.

De même, l'étude des descripteurs audio nous a permis d'enrichir cette partie, comme la synthèse à l'aide des coefficients MFCC.

Annexes

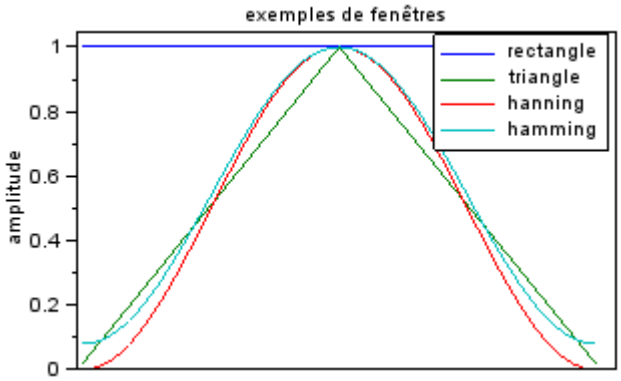
I) Fenêtrage

A) Les fenêtres



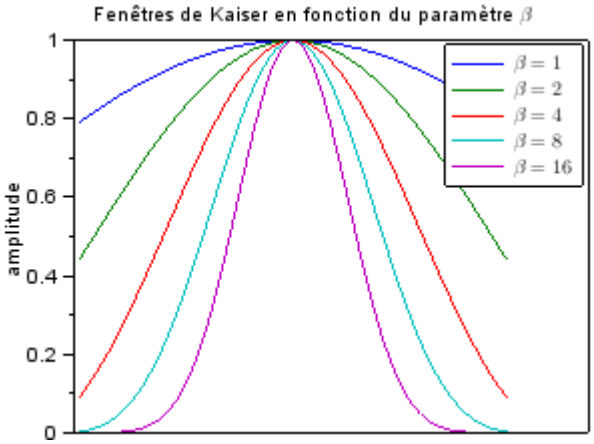
Le « fenêtrage » représente le découpage de signaux en morceaux, multipliés à la fenêtre choisie (cosinus², …). Elles répondent à trois besoins fréquents en traitement du signal : des informations plus locales, de la continuité aux bords et des coûts calculatoires plus faibles.

Selon les besoins, il existe différentes fenêtres. En voici quelques exemples :



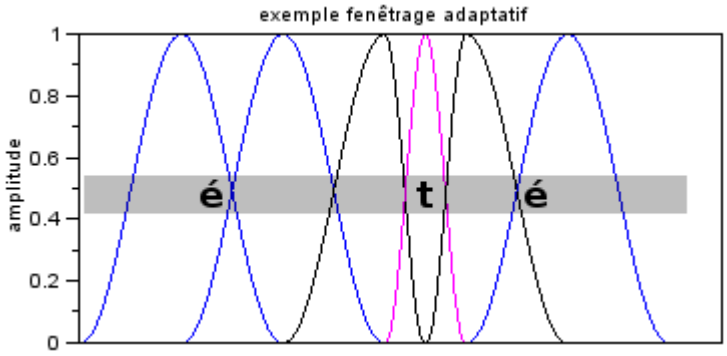
La fenêtre rectangulaire représente la seule sélection d'une partie du signal, sans modification. La fenêtre triangulaire permet de rajouter la continuité à ses bords et les fenêtres de Hamming et Hanning ont l'intérêt supplémentaire d'être dérivables. La différence entre ces deux dernières fenêtres se situe aux bords, où seule Hanning est nulle. La fenêtre de Hamming est plutôt conseillée pour l'analyse (aucune suppression d'information aux bords) et celle de Hanning pour la synthèse (continuité améliorée du signal).

En cas de besoin d'une fenêtre en particulier, les fenêtres de Kaiser (paramétrables) pourront être utilisées :



Le paramètre pourra être choisi selon les besoins : une forte conservation des informations du signal fera pencher vers un β petit alors qu'un β élevé permettra d'avoir l'assurance d'une bonne continuité aux bords.

Les fenêtres, encore, peuvent être asymétriques. Ce cas se produit par exemple si les tailles des fenêtres s'adaptent à la stationnarité du signal. L'utilisation d'un recouvrement, force la taille des moitiés de fenêtres à dépendre de la taille des fenêtres voisines, comme nous pouvons le voir sur cet exemple :



B) Le recouvrement

Le choix du recouvrement des fenêtres (de moitié, de trois quarts, …) doit se faire en fonction des différentes manipulations appliquées aux fenêtres. Voici quelques exemples :

- Sans recouvrement : jamais pour la synthèse, en analyse uniquement si une perte (minime) d'information est tolérée,
- Recouvrement ½ : fenêtrage par fenêtres en cosinus carré (Hamming, Hanning) : $\sin^2 x + \cos^2 x = 1$. A utiliser si uniquement une application de fenêtre (analyse ou synthèse seule par exemple).
- Recouvrement ¾ : fenêtrage par fenêtres en cosinus^4 (application d'une fenêtre pour l'analyse puis d'une nouvelle pour la synthèse) : $\cos^4 x+\cos^4(x+\frac{\pi}{4})+\cos^4(x+\frac{\pi}{2})+\cos^4(x+\frac{3.\pi}{4})=\frac{3}{2}$,
- Recouvrement supérieur à ¾ : besoin important de redondance.

C) Algorithmes

•**Algorithme de découpage en fenêtre du signal :**

•Entrées : *Y*, le signal, *t*, la taille des fenêtres, *r*, le recouvrement (r>½) ,

•Sortie : *S*, la matrice des fenêtres.

•Soit *S* une matrice vide,

•Soit *T* la taille de *Y*,

•Soit *H* la fenêtre de Hamming de taille *t*,

•Calcul du déplacement entre chaque fenêtre : $d=t.(1-r)$

•Nombre de fenêtres : $n = \text{ceil}(\frac{T-t}{d})$

•k allant de 1 à n :

- •Début de la fenêtre : (k-1).d+1,
- •Fin de la fenêtre : (k-1).d+t,
- •Ajout de la fenêtre à S.

•Fin

Chaque fenêtre de S peut maintenant être modifiée indépendamment des autres, grâce au recouvrement et aux fenêtres de Hamming qui permettent de conserver de la continuité lors de la reconstruction du signal.

•**Algorithme de reconstitution du signal à partir des fenêtres :**

•Entrées : S, la matrice des fenêtres, r, le recouvrement (r>½) ,

•Sortie : Y, le signal.

•Soit [t_1,t_2] la taille de S, avec t_1 le nombre de fenêtres et t_2 la taille des fenêtres,

•Soit le déplacement entre chaque fenêtre analysée : d=t_2.(1-r)

•Soit Y un vecteur nul de taille t_1.d+t_2,

•Soit H la fenêtre de Hanning de taille t_2,

•k allant de 1 à t_1 :

- •Début de l'emplacement de la fenêtre : (k-1).d+1,
- •Fin de l'emplacement de la fenêtre : (k-1).d+t_2,
- •Ajout de la $k^{ième}$ fenêtre de S à Y.

•Fin

II) Distances et similarités

La distance euclidienne est la plus connue et la plus utilisée, mais elle n'est pas la seule existante. Pour une introduction rapide mais assez complète aux diverses distances, on pourra regarder [JYB], puis s'intéresser à [BEL12] pour une ouverture sur les distances apprises.

A) Distances

1) Distances puissance

La distance (appelée de Minkowsky) entre deux vecteurs a et b (pour p strictement positif) :

$$\mathrm{d}(a,b)_p = \left(\sum_{1 \le k \le m} |a_k - b_k|^p \right)^{\frac{1}{p}}$$

appelée Euclidienne pour p=2, de Manhattan (ou métrique absolue) pour p=1.

$$\mathrm{d}(a,b)_\infty = \max_{1 \le k \le m} (|a_k - b_k|)$$

la distance infinie, appelée distance de Tchebichev (ou métrique maximum).
Enfin, pour plus de liberté, la distance de puissance :

$$\mathrm{d}(a,b)_{p,r} = \left(\sum_{1 \le k \le m} |a_k - b_k|^p \right)^{\frac{1}{r}}$$

2) Autres distances

Pour certaines distances, il est préférable de comparer des vecteurs positifs.

Distance de Canberra :

$$d(a,b)=\sum_{1\le k\le m}\frac{|a_k-b_k|}{|a_k+b_k|}$$

Distance des cordes carrées :

$$d(a,b)=\sum_{1\le k\le m}\left(\sqrt{a_k}-\sqrt{b_k}\right)^2$$

Distance du Khi carré :

$$d(a,b)=\sqrt{\sum_{1\le k\le m}\frac{|a_k-b_k|^2}{|a_k+b_k|}}$$

Distance de Mahalanobis :

$$d(a,b)=\sqrt{(a-b)S^T(a-b)^T}$$

avec S la matrice de covariance de a et b.

3) Corrélations

Le coefficient de corrélation permet de mesurer la ressemblance entre deux individus. Dans le cadre de la comparaison de similitudes d'individus appartenant à des lots d'individus, la corrélation peut aussi être calculée non pas à partir de leurs valeurs mais des rangs des valeurs. Ainsi, si un descripteur d'un individu est la $k^{\text{ième}}$ valeur la plus élevée, alors il prend la valeur k.

Pour mesurer les corrélations entre rangs de deux individus, on peut mesurer la distance entre leurs rangs (voir le coefficient de corrélation de Spearman (norme 2) ou encore le coefficient de Kendall (norme 1)).

B) Distances apprises

Le but des distances apprises sur une base d'apprentissage supervisée est de déformer l'espace de manière à augmenter le taux de classification, en rendant les distances entre individus d'une même classe faibles et les distances entre individus de classes différentes grandes. Diverses idées peuvent être utilisées, comme l'ALD (→ semie ALD) ou tout algorithme de sélection de descripteurs. A la différence de ces méthodes, les distances apprises ne réduisent pas la taille des données, elles modifient seulement leur importance lors de la classification. C'est une approche moins brutale mais beaucoup plus lourde (puisque tous les descripteurs sont conservés).

Bibliographie

[HER03] : **Introduction à la classification de sons d'instruments de musique :** ♥♥♥

Automatic Classification of Musical Instruments Sounds, Perfecto Herrera Boyer ; Geoffroy Peeters ; Shlomo Dubnov, 2003

[LEM06] : **Rapport technique sur des descripteurs :** ♥♥

Les descripteurs d'un son Librairie Matlab SPL et fichiers de descriptions".sig", Guillaume Lemaitre ; Emmanuel Gallo, 2006

[ESS05] : **Classification audio : descripteurs, méthodes de classification :** ♥♥♥

Classification automatique des signaux audio-fréquences : reconnaissance des instruments de musique, Slim Essid, 2005

[OBI05] : **Algorithmes d'estimation de la fréquence fondamentale :** ♥♥

Evaluation des algorithmes d'estimation de la fréquence fondamentale dans le cadre de signaux musicaux monophoniques, Nicolas Obin, 2005

[DOU02] : **Résumé LPC :**

Speech Processing: Theory of LPC Analysis and Synthesis, Douglas L. Jones ; Swaroop Appadwedula ; Matthew Berry ; Mark Haun ; Jake Janovetz ; Michael Kramer ; Dima Moussa ; Daniel Sachs ; Brian Wade, 2002

[HER02] : **Descripteurs + introduction sélection/classification :** ♥♥

Automatic Classification of Drum Sounds : a comparaison of feature selection methods and classification techniques, Perfecto Herrera ; Alexandre Yeterian ; Fabien Gouyon, 2002

[ZIL04] : **Système EDS de découverte d'algorithmes :**

Extraction de descripteurs musicaux: une approche évolutionniste, ymeric Zils, 2004

[ESC08] : **L'ACP, utilisation et explications :** ♥♥

Analyses factorielles simples et multiples, Brigitte Escofier ; Jérôme Pagès, 2008

[TOL06] : **Explications diverses méthodes (ACP, ALD, MMD, ...) :** ♥♥♥

Indexation et recherche d'images par fusion d'informations textuelles et visuelles, Sabrina TOLLARI, 2006

[RIC06] : **Sélection de descripteurs (reliefF) :**

Sélection de descripteurs pour modèles d'observation audio temps-réel, Olivier RICORDEAU, 2006

[CHO11] : **Diverses méthodes de sélection de descripteurs présentées :**

Sélection de caractéristiques: méthodes et applications, Haissan CHOUAIB, 2011

[WIK] : **SVM : explications et algorithme :** ♥♥

Machines à vecteurs supports, Wikistats,

[PIE09] : **Algorithme GMM :** ♥

De l'utilisation pratique des mélanges de gaussiennes, Bruneau Pierrick, 2009

[BOU09] : **Modèles paramétriques gaussiens :**

Classification supervisée et non supervisée des données de grande dimension, Charles Bouveyron ; Stéphane Girard, 2009

[JYB] : **Une introduction concise aux arbres ascendants, distances, ... :** ♥♥♥

jybaudot.fr,

[BEL12] : **L'apprentissage de distance de similarité :** ♥

Apprentissage de bonnes similarités pour la classification linéaire parcimonieuse, Aurélien Bellet ; Amaury Habrard ; Marc Sebban, 2012