MATCHING PURSUIT POUR LE CODAGE ET LA CLASSIFICATION DE LA PAROLE

Blaise Bertrac

Rapport de stage

MATCHING PURSUIT POUR LE CODAGE ET LA CLASSIFICATION DE LA PAROLE

Bertrac Blaise

Master Traitement des signaux et des images

Encadrant : Monsieur Khalid Daoudi
Stage réalisé du 3 juin au 31 juillet 2013

Remerciements

Je remercie mon maître de stage Monsieur Khalid DAOUDI pour la rigueur scientifique qu'il m'a communiqué, sa disponibilité pour assurer le suivi dont j'ai eu besoin afin de mener à bien mes travaux. Ces conseil précis m'ont été d'une aide précieuse.

Mes remerciements s'adressent également à :

- L'équipe GeoStat au sein de laquelle j'ai travaillé, notamment à Monsieur Hussein YAHIA pour la qualité de l'accueil qui m'a été réservée.

- Monsieur Charles DOSSAL qui m'a mis en contact avec Messieurs Hussein YAHIA et Khalid DAOUDI.

- Et à mon tuteur de stage Monsieur Jean-François AUJOL pour sa disponibilité et tous les conseils prodigués.

Table des matières

1 Introduction

Dans le cadre de mon Master 1 en Traitement des signaux et des images, j'ai effectué un stage du 3 juin au 31 juillet 2013 au sein de l'INRIA à Talence. Il s'agit d'un institut de recherche dans le domaine des sciences technologies. Monsieur Khalid Daoudi était mon maître de stage au sein de l'équipe GEOSTAT dirigée par Hussein Yahia.

La problématique posée au cours de la période de stage est la suivante : Matching Pursuit pour le codage et la classification de la parole. Le but est le suivant : À partir d'une base de données constituée d'enregistrement de voix, déterminer celles qui sont normales et celles qui sont pathologiques.
Après avoir présenté l'INRIA, deuxième point de mon rapport, je montrerai en quelques lignes comment fonctionne la voix humaine, et qu'est ce qui distingue une voix normale d'une voix pathologique, troisième point de mon rapport. Je présenterai en quatrième partie l'algorithme du maching pursuit, puis je m'intéresserai à la classification en précisant quelques méthodes qui ont été utilisées durant mon stage. Et pour terminer je montrerai comment j'ai appliqué toutes ces notions pour traiter la base de données constituée d'enregistrement de voix normales et pathologiques.

2 Présentation de l'INRIA

2.1 Qu'est que Inria ?

INRIA (Institut National de Recherche en Informatique et en Automatique) est le seul un institut public de recherche entièrement dédié aux sciences numériques. Placé depuis sa création, en 1967 sous la tutelle des ministres de la Recherche et de l'Industrie, cet établissement a pour missions de produire une recherche d'excellence dans les champs informatiques et mathématiques des sciences du numérique et de garantir l'impact de cette recherche, grâce au transfert de ses travaux vers le monde économique et la société.

2.2 Quelques chiffres

Les quelques chiffres suivants illustrent l'activité importante de l'Inria :

- 8 centres de recherche répartis dans toute la France (Rocquencourt, Rennes, Sophia Antipolis, Grenoble, Nancy, Bordeaux, Lille et Saclay) et un siège social à Rocquencourt près de Paris

- 4290 personnes

- 233 millions d'euros de budget en 2013

- 800 contrats de recherche en cours

- 70 équipes associées avec des laboratoires étrangers

- Publication de près de 5 000 articles en 2011 qui sont à l'origine de la création de 110 start-ups (jeunes sociétés à fort potentiel de croissance)

2.3 Les principaux partenaires industriels

L'Inria travaille en collaboration avec les principaux partenaires suivants :

- Microsolf Research

- Alcatel-Lucent Bell Labs

- France-Telecom-Orange

- Thomson-Technicolor

- ST Microelectronics

- Total

- EDF *R&D*

2.4 Lieu et cadre du stage

J'ai effectué mon stage dans le centre Inria Bordeaux-Sud-Ouest situé à Talence à proximité de l'Université de Bordeaux 1 et j'ai intégré l'équipe GeoStat (Géométrie et statistiques dans les données d'acquisition).

Cette équipe entreprend des recherches fondamentales et appliquées sur des nouvelles méthodes émergentes dans l'analyse non-linéaire des signaux et systèmes complexes, en utilisant des paradigmes liés aux notions d'invariance d'échelle, de predictabilité, ainsi que dans le développement du formalisme multiéchelles microcanonique.

Ainsi, les recherches théoriques dans l'équipe concernent les domaines suivants : méthodes multiéchelles issues de la physique pour l'analyse des systèmes complexes, prédictabilité dans les systèmes complexes, ondelettes optimales, analyse, classification et détection.

GeoStat s'intéresse en premier lieu à différents domaines applicatifs : analyse des signaux turbulents issus des observations satellitaires, signaux complexes en astronomie, optique adaptative, analyse du signal parole.

3 Caractérisation de la voix

La voix est le vecteur de la parole et nous permet la communication orale. Elle est possible grâce à la coordination des trois principales entités fonctionnelles de l'appareil phonatoire humain :

- Le larynx et plus précisément les deux cordes vocales sont la source de la vibration sonore. Elles se rapprochent en phonation et s'ouvrent en respiration pour laisser l'air passer dans la trachée et les poumons. La muqueuse qui représente la couche la plus superficielle des cordes vocales oscille sous l'influence de l'air sous pression provenant du souffle phonatoire.

- Le souffle phonatoire est la source d'énergie de la voix. Elle est délivrée par la mise en pression de l'air inspiré dans les poumons par l'ensemble des muscles expiratoires (comme le diaphragme et les abdominaux).

- Les résonateurs, cavité buccale, pharynx, fosses nasales, ont à la fois un rôle d'articulattion des phonèmes de la parole mais aussi d'élaboration du timbre vocalique propre à chaque individu.

Ainsi, la voix possède trois paramètres acoustiques :

- La hauteur est le paramètre qui définit le caractère grave ou aigu de la voix. Il correspond au mécanisme de vibration des cordes vocales. Plus les vibrations sont nombreuses et plus le son est aigu. La hauteur est aussi appelée fréquence fondamentale et est caractérisée par sa large variabilité.

- L'intensité est le paramètre qui permet de distinguer un son fort ou faible. Il dépend de l'énergie du signal vocal et de l'amplitude des vibrations.

- Le timbre est le paramètre qui permet d'identifier une personne juste en écoutant sa voix. Il dépend des caractéristiques anatomiques des cavités de résonance et des conditions de rapprochement des cordes vocales ainsi que de leurs épaisseurs.

La figure 1 suivante illustre le système phonatoire :

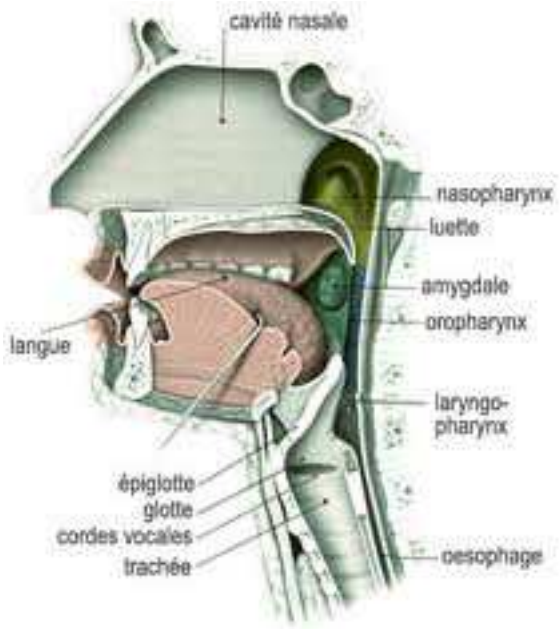

FIGURE 1 – Le système phonatoire

Une voix pathologique correspond à des troubles de la voix ou dysphonie (enrouement, voix éraillé ou soufflée). Dans ce cas, elle se traduit par l'atteinte isolée ou combinée des paramètres acoustiques de la voix. La dysphonie peut être d'origine organique caractérisée par des lésions des cordes vocales ou du larynx, ou d'origine neurologique en provoquant des troubles de la mobilités des cordes vocales.

4 Matching Pursuit

Les signaux sonores (parole, musique, ...) sont non-stationnaires. Ils contiennent des structures à différentes échelles (transitoires de très courte durée, parties soutenues et résonances de notes qui durent, etc...) et différentes fréquences à des instants variés. Pour de tels signaux, l'efficacité des approximations à un nombre de termes fixé dans une base orthogonale est donc limitée. Ainsi, il faut pouvoir puiser dans un ensemble où les éléments sont les mieux adaptés aux structures d'un signal donné.

4.1 Présentation générale de l'algorithme

Le matching pursuit est un algorithme itératif permettant d'approcher un signal par une décomposition en atomes temps-fréquence, c'est à dire en fonctions localisées à la fois dans le temps et en fréquence. Selon le choix de l'ensemble des atomes temps-fréquence appelé dictionnaire (fixé au départ), la décomposition peut avoir des propriétés très différentes. Les transformées de Fourier en fenêtre et transformées en ondelettes sont des exemples de décomposition de signaux temps-fréquence. Pour extraire des informations à partir de signaux complexes, il est souvent nécessaire d'adapter la décomposition atomique.

Une famille générale d'atomes peut être généré par dilation, translation et modulation d'une seule fonction fenêtre $g(t) \in \mathcal{L}^2(\mathbb{R})$.On suppose que cette fonction $g(t)$ est réelle, continument différentiable, de norme 1, d'intégrale non nulle avec $g(0) \neq 0$ et que l'on a $g(t) = O(\frac{1}{t^2+1})$ quand $t \to +\infty$.

On note $\gamma = (s, u, \xi)$ où $s > 0$ représente l'échelle (ou l'octave) d'un atome (c'est la longueur temporelle), u est la position temporelle et ξ la position fréquentielle (centrale).
L'indice γ est un élément de l'ensemble $\Gamma = \mathbb{R}^+ \times \mathbb{R}^2$; on définit la fonction suivante :

$$g_\gamma(t) = \frac{1}{\sqrt{s}} g(\frac{t-u}{s}) e^{i\xi t}.$$

Le facteur $\frac{1}{\sqrt{s}}$ normalise à 1 la norme de $g_\gamma(t)$. Si la fonction $g(t)$ est la même, ce qui est généralement le cas, $g_\gamma(t)$ est centré en u. L'essentiel de son énergie est localisée temporellement dans un voisinage de u et dans le domaine de Fourier, dans un voisinage de ξ qui est proportionnel à $1/s$.

La famille $\mathcal{D} = (g_\gamma(t))_{\gamma \in \Gamma}$ est extrêmement redondante. Pour représenter efficacement un signal $f(t)$, on doit sélectionner un sous ensemble dénombrable d'atomes $(g_{\gamma_n}(t))_{n \in \mathbb{N}}$ avec $\gamma_n = (s_n, u_n, \xi_n)$ tel que :

$$f(t) = \sum_{n=-\infty}^{+\infty} a_n g_{\gamma_n}(t).$$

Selon le choix des atomes $g_{\gamma_n}(t)$, les coefficients de décomposition a_n donnent une information explicite sur les structures du signal.

4.2 Familles particulières d'atomes

Les transformées de Fourier à Fenêtre et transformées en ondelettes sont des familles d'atomes temps-fréquences, qui sont des frames ou des bases de $\mathcal{L}^2(\mathbb{R})$.

Dans la transformée de Fourier à fenêtre, tous les atomes $g_{\gamma_n}(t)$ ont une échelle constante $s_n = s_0$ et sont donc principalement localisés sur un intervalle dont la taille est proportionnelle à s_0. Ainsi on aura une bonne précision sur la position temporelle et le contenu fréquentiel d'un signal avec cette transformée si ses principales structures sont localisées sur une échelle-temps de l'ordre de s_0. En revanche, la transformée de Fourier à fenêtre n'est pas bien adaptée pour décrire des structures qui sont plus petites ou plus grandes que s_0. Pour analyser des éléments de tailles variables, il faut utiliser des atomes temps-fréquences d'échelles s_n différentes.

Contrairement à la transformée de Fourier à fenêtre, la transformée en ondelettes décompose les signaux en atomes temps-fréquences de tailles s_n variables, appelés ondelettes. Une famille $(g_{\gamma_n}(t))_{n \in \mathbb{N}}$ est construite en définissant la fréquence ξ_n par rapport à l'échelle s_n tel que $\xi_n = \xi_0/s_n$, où ξ_0 est une constante. La famille obtenue est composée de dilations et

de translation d'une seule fonction, multipliée par le paramètre (complexe) phase. Ainsi, les coefficients de décomposition a_n caractérisent les comportement d'échelles. Cependant, elle ne fournit pas d'estimation précises sur contenu fréquentiel des ondelettes contrairement aux transformées de Fourier surtout dans les hautes fréquences (puisque la fréquence ξ_n est inversement proportionnelle à l'échelle s_n).

Ainsi, pour les signaux présentant à la fois des structures de tailles (d'échelles) variables et des structures de hautes fréquences, il est nécessaire de choisir de manière adaptative les éléments du dictionnaire $\mathcal{D} = (g_\gamma(t))_{\gamma\in\Gamma}$, selon les propriétés locales d'un signal $f(t)$ donné.

4.3 Description de l'algorithme

Soit H un espace de Hilbert. Soit $f \in H$ un signal. Tout d'abord, on définit un dictionnaire $\mathcal{D}$. Il s'agit d'un espace complet, c'est-à-dire que les combinaisons linéaires de ses atomes sont denses dans l'ensemble de définition du signal $\mathcal{L}^2(\mathbb{R})$.

Ainsi, on a $\mathcal{D} = \{g_\gamma\}_{\gamma\in\Gamma}$. On suppose qu'il contient $P > N$ vecteurs de norme 1. De plus, il contient N vecteurs linéairement indépendants définissant une base de l'espace $\mathbb{C}^N$ des signaux de taille N. Les atomes qui sont utilisés dans la décomposition linéaire sont choisis de manière adaptative par rapport au signal f.

L'algorithme matching pursuit choisit parmi les différents atomes ceux dont la taille est la mieux adaptée aux caractéristiques locales du signal. Le critère de sélection est la corrélation des atomes du dictionnaire avec le signal.

Au début de l'algorithme, on fait la projection de f sur un vecteur $g_{\gamma_0} \in \mathcal{D}$ et on calcule le résidu Rf :

$$f = \langle f, g_{\gamma_0}\rangle g_{\gamma_0} + Rf$$

Comme Rf est orthogonal à g_{γ_0}, on a :

$$\|f\|^2 = |\langle f, g_{\gamma_0}\rangle|^2 + \|Rf\|^2$$

Pour minimiser $\|Rf\|$, il faut choisir $g_{\gamma_0} \in \mathcal{D}$ tel que $\langle f, g_{\gamma_0}\rangle$ soit maximal. En pratique, dans certains cas, il est plus efficace numériquement de trouver un vecteur g_{γ_0} presque optimal tel que :

$$|\langle f, g_{\gamma_0}\rangle| \geq \alpha sup_{\gamma\in\Gamma}|\langle f, g_\gamma\rangle|,$$

où $\alpha \in]0, 1]$ est le facteur d'optimalité. L'algorithme du matching pursuit itère cette procédure en décomposant le résidu.

Récurrence :

Soit $R^0 f = f$. On suppose que le résidu $R^m f$ d'ordre m a été calculé pour $m \geq 0$. A l'itération suivante on choisit $g_{\gamma_m} \in \mathcal{D}$ tel que :

$$|\langle R^m f, g_{\gamma_m} \rangle| \geq \alpha sup_{\gamma \in \Gamma} |\langle R^m f, g_\gamma \rangle|,$$

et on projette $R^m f$ sur g_{γ_m} :

$$R^m f = \langle R^m f, g_{\gamma_m} \rangle g_{\gamma_m} + R^{m+1} f.$$

Comme $R^{m+1} f$ est orthogonal à g_{γ_m}, on a :

$$\|R^m f\|^2 = |\langle R^m f, g_{\gamma_m} \rangle|^2 + \|R^{m+1} f\|^2$$

Au bout de M itérations, f devient :

$$f = \sum_{m=0}^{M-1} \langle R^m f, g_{\gamma_m} \rangle g_{\gamma_m} + R^M f.$$

Et de même on en peut en déduire :

$$\|f\|^2 = \sum_{m=0}^{M-1} |\langle R^m f, g_{\gamma_m} \rangle|^2 + \|R^M f\|^2.$$

Le critère d'arrêt peut être le nombre d'itérations M ou un rapport signal sur bruit minimum (par exemple).

Si l'on continue à itérer, S. Mallat et Z. Zhang ont démontré que l'erreur $\|R^m f\|$ converge exponentiellement vers 0 quand m tend vers $+\infty$. On obtient donc :

$$f = \sum_{m=0}^{+\infty} \langle R^m f, g_{\gamma_m} \rangle g_{\gamma_m}.$$

4.4 Librairie MPTK

Dans la suite, pour la décomposition des signaux, on va utiliser principalement la librairie MPTK (Maching Pursuit Toolkit) à l'aide du logiciel Matlab. Il s'agit d'une implémentation efficace de l'algorithme Matching Pursuit, pour la décomposition parcimonieuse de signaux, développée à l'IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires).

4.5 Choix du dictionnaire

Nous verrons dans cette partie deux exemples de dictionnaires multi-échelles : le Dictionnaire de Gabor et le dictionnaire des Gammatones.

4.5.1 Dictionnaire de Gabor

Le dictionnaire de Gabor sera utilisé avec la librairie MPTK plus tard. Il est constitué d'atomes temps-fréquence obtenus par dilatation, translation et modulation d'une fonction $g(t)$ gaussienne (en raison de ses propriétés optimales de localisation combinée temps-fréquence, au sens du principe d'incertitude de Heisenberg).

On définit ce dictionnaire par : $\mathcal{D}_\alpha = (g_\gamma)_{\gamma \in \Gamma_\alpha}$, où l'ensemble Γ_α est constitué des γ de la forme $(2^j, p2^{j-1}, k\pi/2^j)$, tel que pour N représentant le nombre d'échantillons d'un signal f donné, on a $0 < j < \log_2 N$, $0 \leq p < N2^{-j+1}$ et $0 \leq k < 2^{j+1}$. Le nombre de vecteurs dans $\mathcal{D}_\alpha$ est $\mathcal{O}(N \log_2 N)$.

Dans la figure 2 suivante, nous pouvons voir une décomposition atomique avec le dictionnaire de Gabor (avec MPTK) d'un signal d'enregistrement du son 'ah'.

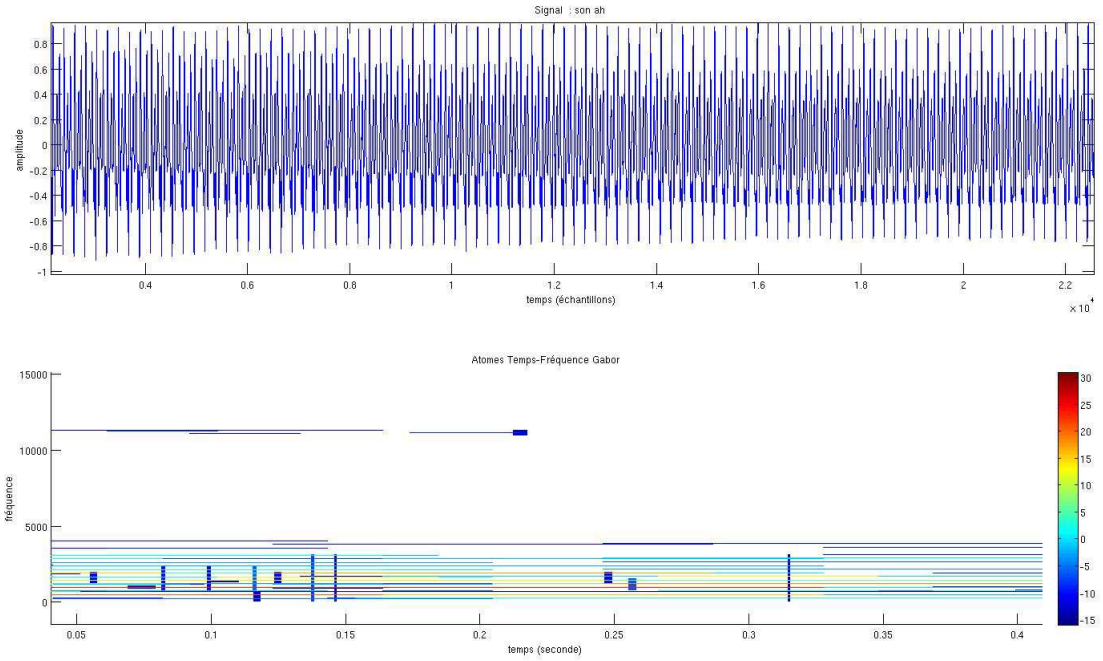

FIGURE 2 – Décomposition d'un signal avec le dictionnaire de Gabor

4.5.2 Dictionnaire Gammatone

Pour un signal donné $f(t)$, la décomposition en atomes temps-fréquence avec le dictionnaire Gammatone est la suivante :

$$f(t) = \sum_{m=1}^{M} \sum_{i=1}^{n_m} a_i^m g_m(t - \tau_i^m) + r_x(t),$$

où τ_i^m et a_i^m sont respectivement la position temporelle et l'amplitude de la fonction noyau g_m à la i-ème itération.

Pour faire l'analogie avec l'algorithme du matching pursuit défini précédemment, on considère les noyau g_m comme l'ensemble des fonctions fenêtres sur lesquelles le signal f est décomposé. On a $\|g_m = 1\|$ et $a^m = \langle f(t), g_m \rangle$. $r_x(t)$ est le résidu du signal.

$g_m(t) = t^{(l-1)} e^{-2\pi b ERB(f_{cm})t} cos(2\pi f_{cm} t)$, $t > 0$,

où $l = 4$ est l'ordre du filtre, $b = 1,019$ est une constante qui correspond à la bande passante et les f_{cm} sont les fréquences centrales du filtre. Elles sont distribuées linéairement dans l'échelle ERB (Equivalent Rectangular Bandwith).

Le système auditif humain ne perçoit pas les signaux selon une échelle uniforme. Notamment, il effectue une analyse plus précise dans les fréquences graves que dans les bandes hautes du spectre. Ainsi, réalisant l'analyse spectrale selon une échelle en fréquence conforme à notre perception auditive, l'échelle ERB est la largeur des bandes critiques et elle modélise la résolution fréquentielle des filtres auditifs. La relation moyenne pour plusieurs sujets "normaux" (à propos de l'audition) entre la fréquence centrale f en Hz et l'ERB en Hz est la suivante :

$$ERB(f) = 24,7.(\frac{4,37f}{1000} + 1)$$

La figure 3 ci dessous nous montre les réponses de fréquence d'un banc de filtres Gammatones avec dix filtres dont les fréquences centrales sont espacées de façon égale entre 50 Hz et 4 kHz sur l'échelle ERB.

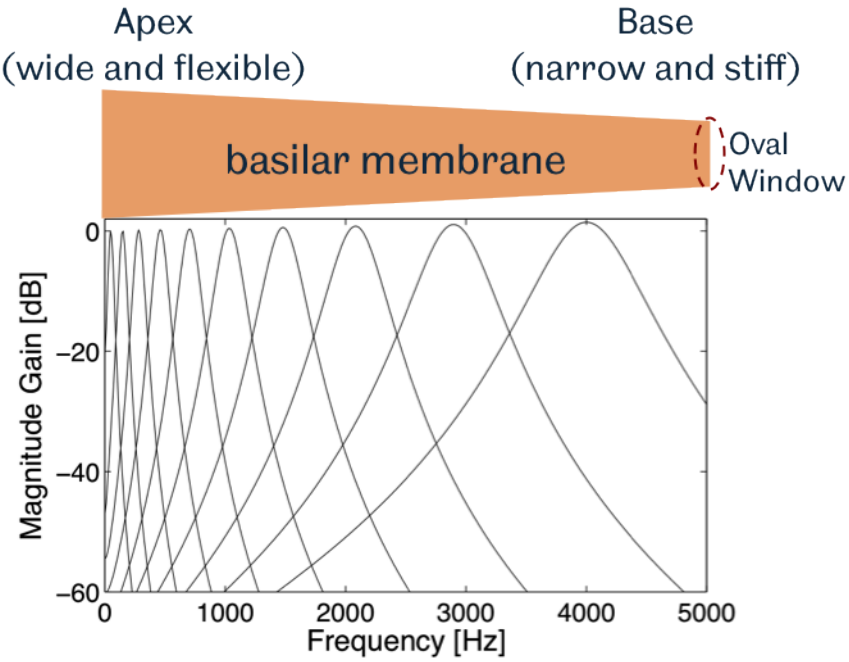

FIGURE 3 – Filtres Gammatones ERB

5 Méthodes de classification

Une classification ou système de classification est un système organisé et hiérarchisé de catégorisation d'objets appelées classes. Dans notre cas nous nous limiterons à deux classes pour organiser les signaux sonores : la première classe devra contenir les signaux de voix normales, et la seconde classe les signaux de voix pathologiques. Nous verrons trois méthodes utilisées pour la classification : les modèles de mélanges Gaussiens, l'analyse discriminante linéaire et la classification SVM (support vecteur machine).

5.1 Les modèles de mélanges Gaussiens (GMM)

Ces modèles statistiques GMM (Gaussian Mixture Model) sont utilisés pour estimer les paramètres d'une distribution de variables aléatoires en les modélisant par une densité mélange. Usuellement, ils sont employés en classification et on considère que chaque composant du mélange caractérise une classe. Ces modèles présentent deux avantages principaux :

• Il s'agit d'une méthode probabiliste permettant d'obtenir une classification des observations. Une probabilité d'appartenance à chacune des classes est calculée et une classification est généralement obtenue en affectant chacune des observations à la classe la plus probable. Ces probabilités permettent également d'interpréter certaines classifications suspectes.

• Ils offrent une grande flexibilité de modélisation et permettent donc de modéliser un grand nombre de phénomènes. Le but des modèles de mélange est de structurer un jeu de donnée en plusieurs classes en s'appuyant sur une modélisation par un mélange de distributions.

5.2 L'analyse discriminante linéaire (LDA)

La méthode par LDA est une méthode d'analyse numérique qui permet de chercher la combinaison linéaire des variables qui représentent au mieux les données, c'est à dire que l'on part de la connaissance de la partition en classes des individus d'une population et on cherche les combinaisons linéaires des variables décrivant les individus qui conduisent à la meilleure discrimination entre les classes. Cette analyse permet de maximiser l'éparpillement interclasses et de réduire l'éparpillement intra-classes. Ainsi, les combinaisons résultantes peuvent être employées comme classificateur linéaire, ou généralement dans la réduction de caractéristiques avant la classification postérieure. Cette technique cherche les directions qui sont efficaces pour la discrimination entre les données.

La LDA attribue à chaque classe une moyenne et une variance, et obtient ainsi une scatter-matrix (matrice de dispersion), représentant les distances séparant les classes les uns des autres autour de la moyenne de l'ensemble des points.

5.3 La classification SVM

Faisant partie du domaine de l'apprentissage automatique (machine learning), les SVM (Support Vector Machine) sont des modèles supervisés d'apprentissage avec des algorithmes d'apprentissage associés qui analysent les données et reconnaissent des modèles, utilisés pour la classification et l'analyse de régression (pour estimer la relation entre les variables). On parle

de modèle supervisé car les classes sont prédéterminées et les exemples connus ; le système apprend à classer selon un modèle de classement. Ainsi, étant donné un ensemble d'exemples d'entraînement, chacun marqué comme appartenant à l'une des deux catégories, un algorithme d'apprentissage SVM construit un modèle qui attribue de nouveaux exemples dans l'une ou l'autre catégorie, ce qui en fait un classificateur linéaire binaire non probabiliste. C'est en fait une représentation des exemples comme des points dans l'espace, cartographiés afin que les exemples de catégories distinctes soient séparés par un espace clair qui est aussi large que possible. De nouveaux exemples sont alors placés dans ce même espace et prédits d'appartenir à une catégorie selon le côté de l'écart sur lequel ils tombent. En plus d'effectuer classification linéaire, les SVM peuvent effectuer efficacement une classification non-linéaire.

6 Applications sur la base de données

Nous arrivons à la partie application sur une base de données constituée de signaux de voix normales et pathologiques. J'ai appliqué à chacun de ces signaux la décomposition matching pursuit. A partir de ces décompositions j'ai extrait des paramètres qui ont servi pour une classification.

6.1 Base de données

Le travail a été réalisé avec une base de données. Elle est décrite en Annexe 1 (section 8.1). Elle contient au total 1425 signaux. Parmi ces signaux il y a d'une part les enregistrements (des voix normales et pathologiques) du son "ah" qui durent de 1 à 3 secondes et d'autre part les enregistrement d'une même phrase par des voix normales et pathologiques qui durent tous 12 secondes. Il s'agit de la phrases suivantes : "When the sunlight strikes rain drops in the air, they act like a prism and form a rainbow ; the rainbow is a division of white light into many beautiful colors".

6.2 Paramètres pour la classification

C'est le dictionnaire de Gabor qui a été principalement utilisé pour les décompositions atomiques avec Matlab en utilisant la librairie MPTK. Les paramètres j'ai extraits à partir de ces décompositions proviennent de l'article : "Discrimination of Pathological Voices Using a Time-Frequency Approach" publié en Mars 2005 par Karthikeyan Umapathy, Sridhar Krishnan, Vijay Parsa et Donald G. Jamieson. La base de données (en section 6.1) est la même qui est utilisée dans l'article. Cela a permis de confirmer ou d'infirmer les résultats qui ont été obtenus dans cet article.

Pour bien comprendre la phase de l'extraction, je présente en quelques mots les octaves et fréquences résultant des décompositions atomiques des signaux.

6.2.1 Les octaves

Les octaves sont les échelles des atomes. Dans le cas du dictionnaire de Gabor, ce sont des puissances de 2 et varient de 2^2 à 2^{14}.

La distribution des paramètres octaves est un élément potentiel pour les discrimination des signaux de parole pathologique. En effet, les signaux de paroles pathologiques contiennent plus de structures irrégulières que les signaux de parole normale. Pendant le processus de décomposition, le paramètre octave est décidé de manière adaptative par la durée de les fonctions TF (Temps-Fréquence) gaussiennes qui sont utilisés pour approximer la structure locale du signal. Les octaves les plus élevées correspondent à des fenêtres de longues durées et les octaves les plus bas correspondent à des fenêtres de courtes durées. La combinaison de ces différents octaves de durées différentes génère l'enveloppe des signaux vocaux. Ainsi, les octaves peuvent être considérés comme une mesure indirecte de la composition phonétique du signal de parole.

La figure 4 illustre une comparaison des octaves pour un signal de voix normal et un signal de voix pathologique sur le même son 'ah'.

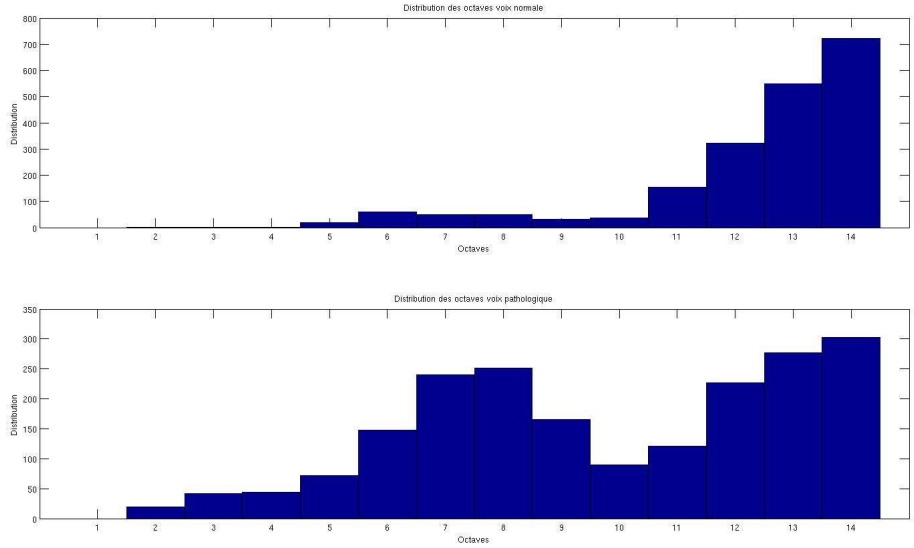

FIGURE 4 – Distribution des octaves : voix normale (en haut) et pathologique (en bas)

• Le premier paramètre que l'on considère est $Ocmax$. Il correspond au nombre maximum de fois qu'une taille d'atome est sélectionnée :

$$Ocmax = max[O_{i=1,2,...,14}],$$

où O_i est le nombre de fois où l'atome de taille 2^i est sélectionné.

• Le second paramètre est $Ocmean$. Il indique la contribution des plus petites structures. C'est la moyenne des tailles des octaves inférieures ou égales à 2^7 (appelées basses octaves) :

$$Ocmean = \frac{\sum_{i=1}^{7} O_i}{7}$$

6.2.2 Les fréquences

Avec MPTK (et le dictionnaire de Gabor), les fréquences sont distribuées de la façon suivante : Pour un signal de fréquence d'échantillonnage Fs (en Hz) et une octave de taille 2^j l'axe des fréquences est décomposé linéairement par pas de $Fs/2^j$.

La figure 5 montre une comparaison des fréquences pour un signal de voix normal et un de voix pathologique échantillonnés tous les deux à $50kHz$.

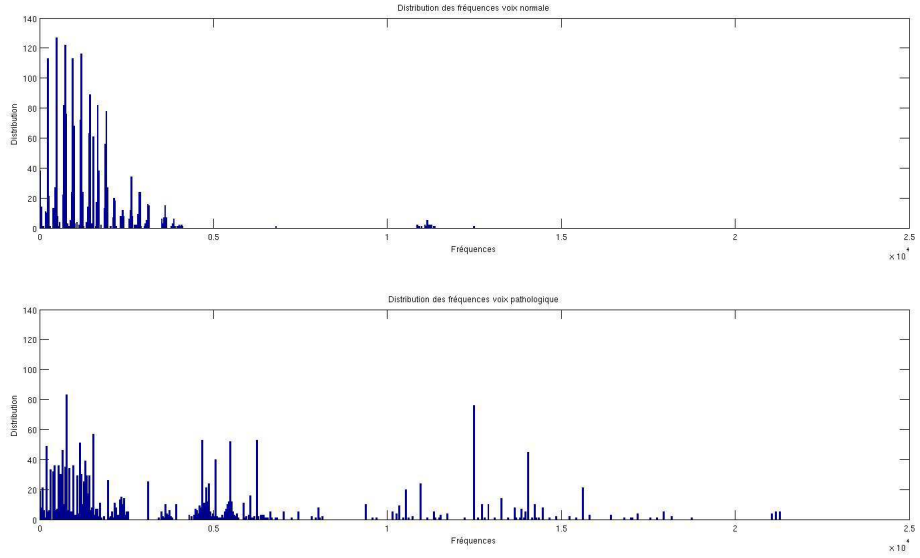

FIGURE 5 – Distribution des fréquences : voix normale (en haut) et pathologique (en bas)

• Le troisième paramètre est Fr. Soit un signal de fréquence Fs, et soit $nbIter$ le nombre d'atomes TF utilisé pour la décomposition, c'est aussi le nombre d'itérations de l'algorithme matching pursuit (voir section 4.3). Fr correspond aux nombre d'atomes inférieures à $Fs/4$ divisé par $nbIter$.
D'après l'article cité ci-dessus, le paramètre Fr donne une estimation précise du pourcentage de la contribution du nombre de fonctions TF utilisées dans l'approximation des composantes de fréquences inférieures et supérieures à la moitié de la fréquence maximale du signal pour un signal. Cette fonctionnalité fournit non seulement des informations au sujet de la distribution des fréquences du signal, mais donne surtout un indice indirect sur la concentration des structures non cohérentes dans le spectre.

6.3 Extraction des paramètres

Dans un premier temps, j'ai extrait les paramètres sur les signaux de phrases pour une classification conformément à l'article, puis j'ai renouvelé l'expérience sur les signaux de voyelles (son "ah"). Le détail des codes matlab que j'ai implantés pour extraire les paramètres est affiché en Annexe 2 (section 8.2). J'ai observé les distributions des paramètres et j'ai évalué leur pertinence pour la classification. Je me suis intéressé en premier lieu à la classification avec

la méthode LDA (voir section 5.2). J'ai fixé le nombre d'itération $nbIter$ à 2000 comme dans l'article dans la mesure où mon objectif était de retrouver les mêmes résultats.

6.3.1 Classification pour les phrases

J'ai choisi tous les signaux de phrases échantillonnés à 25kHz. J'ai disposé de 36 voix normales et 648 voix pathologiques.

• Classification du paramètre $Ocmax$

J'ai calculé le paramètre $Ocmax$ pour chaque signal à partir de l'algotihme MPTK sur matlab. Les résultats montrent que l'octave la plus utilisée pour les signaux normaux et pathologiques est l'octave 11. D'après la distribution des $Ocmax$ en figure 6 et 7, les taux de classification des voix normales sont moins satisfaisant que ceux des voix pathologiques ($69,44\%$ contre $95,99\%$). Cependant, le taux de classification total est de $94,59\%$. Cela s'explique par le fait qu'il y ait 18 fois plus de voix pathologiques que de voix normales.

Ocmax							
Distribution	maximum	minimum	moyenne	écart type	seuil	classe	pourcentage
Normale	745	476	591	65,61	554	25/36	69,44%
Pathologique	623	557	417	72,38		622/648	95,99%
					Total	647/684	94,59%

FIGURE 6 – Tableau des distribution des $Ocmax$

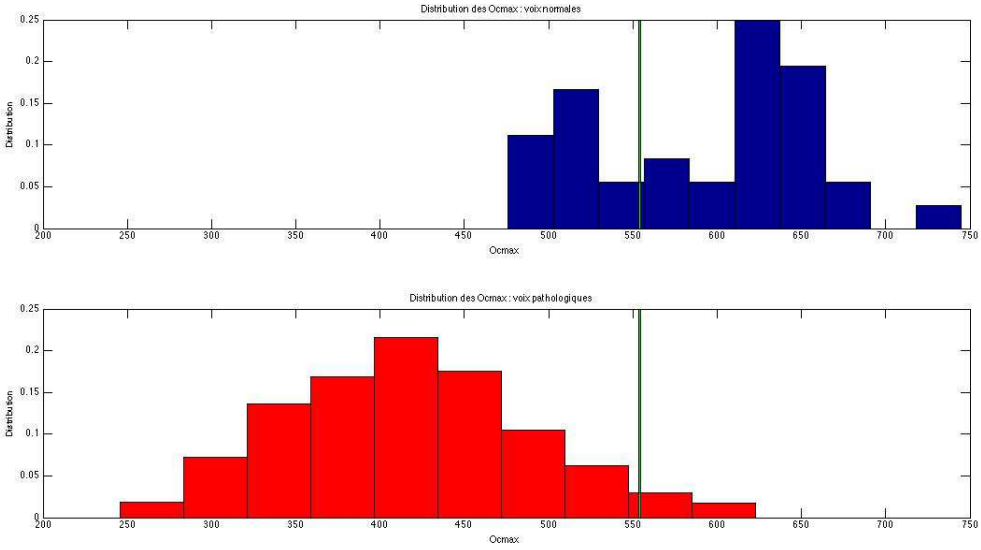

FIGURE 7 – Distribution (normalisée) des *Ocmax*

• Classification du paramètre *Ocmean*

En calculant le paramètre *Ocmean*, j'ai obtenu que le taux de classification total de $81,87\%$ des *Ocmean* est inférieur à celui des *Ocmax*.

Ocmean							
Distribution	maximum	minimum	moyenne	écart type	seuil	classe	pourcentage
Normale	53,43	9	27,14	11,84	35	28/36	77,78%
Pathologique	104,14	5	48,87	15,71		532/648	82,10%
					Total	560/684	81,87%

FIGURE 8 – Tableau des distribution des *Ocmean*

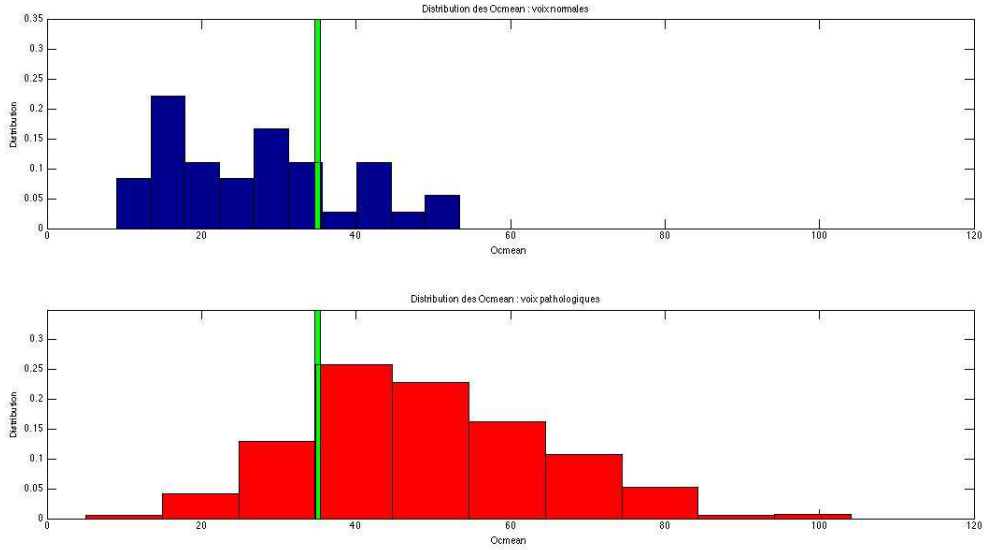

FIGURE 9 – Distribution (normalisée) des *Ocmean*

• Classification du paramètre Fr

Le calcul des paramètres Fr que j'ai effectué donne des résultats moins satisfaisant que ceux calculés précédemment. En effet, d'après les figures 10 et 11 la distribution du paramètre Fr des voix normales est "noyée" dans celle des voix pathologiques. Ainsi la classification est plus difficile. Étant donné que le nombre de signaux pathologiques est beaucoup plus important que celui des voix normales, le classifieur LDA va considérer tous les signaux comme pathologiques. (Si on diminue le seuil fixé à 1 par le classifieur, le taux de classification total est encore plus bas). Ainsi ce paramètre Fr ne permet pas de classer nos signaux bien que son ensemble de définition soit plus important chez les voix pathologiques que chez les voix normales.

Fr							
Distribution	maximum	minimum	moyenne	écart type	seuil	classe	pourcentage
Normale	1	0,95	0,99	0,012	1	0/36	0%
Pathologique	1	0,25	0,88	0,12		648/648	100%
					Total	648/684	94,74%

FIGURE 10 – Tableau des distribution des Fr

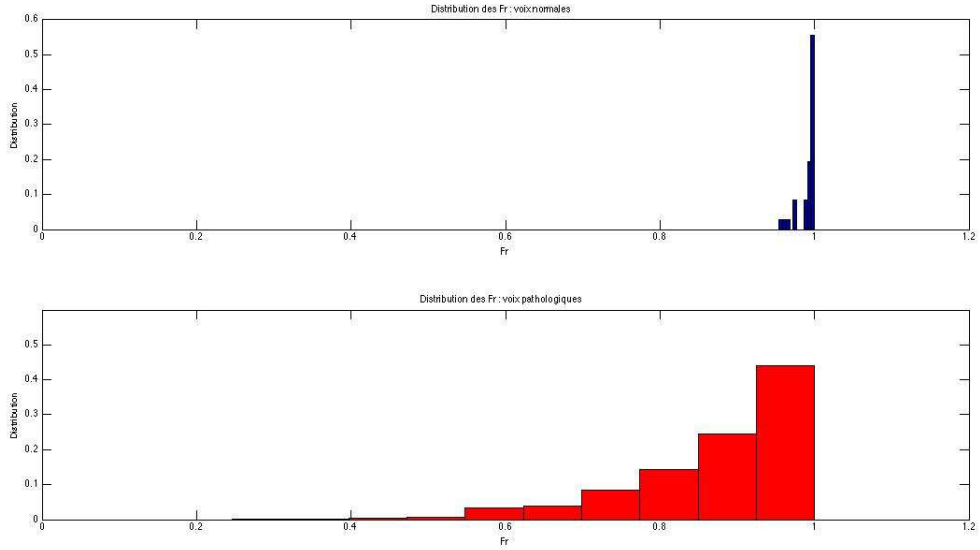

FIGURE 11 – Distribution (normalisée) des Fr

• Classification des signaux avec les 3 paramètres $Ocmax$, $Ocmean$ et Fr

La figure 12 suivante permet de visualiser tous les signaux de phrases normaux (en bleu) et pathologiques (en rouge) en fonction des 3 paramètres $Ocmax$, $Ocmean$ et Fr.

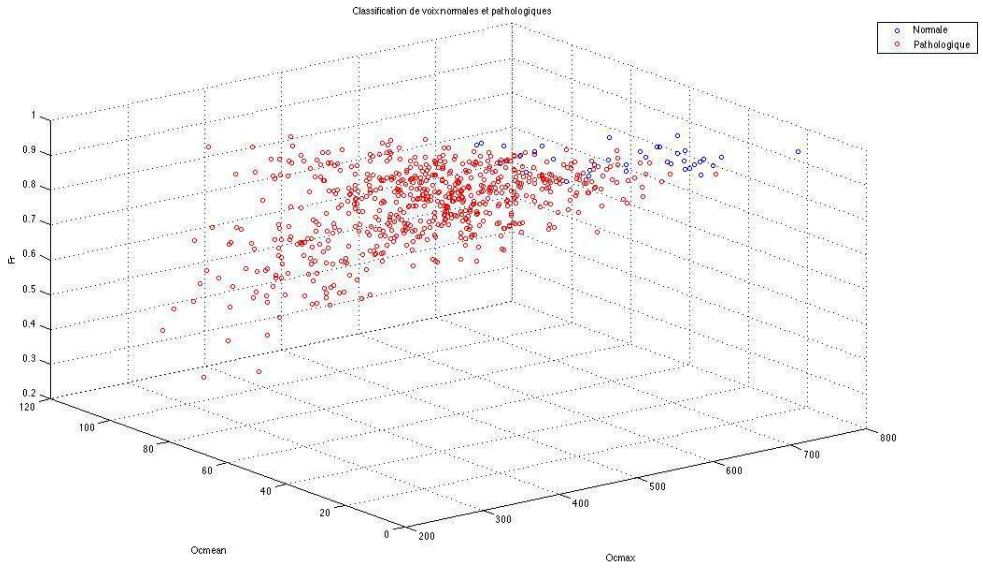

FIGURE 12 – Affichage des signaux en fonction de $Ocmax$, $Ocmean$ et Fr

Avec la classification LDA, on obtient les résultats suivants en figure 13 :

Classification 3D avec la méthode LDA		
	Voix bien classées	Taux
Voix normales	32 (sur 36)	88,89%
Voix pathologique	571 (sur 648)	88,12%
Total	603 (sur 648)	88,16%

FIGURE 13 – Résultats de la classification LDA avec les paramètres $Ocmax$, $Ocmean$ et Fr

Les taux de classification que j'obtiens sont inférieurs à ceux obtenus dans l'article cité en début de section 6.3. En effet, dans cet article, le taux de voix normales bien classées est de $96,1\%$ (contre $88,89\%$) et celui des voix pathologiques bien classées est de $92,6\%$ (contre $88,12\%$). Ces différences de taux s'expliquent par le fait que le nombre voix utilisées dans l'article est de 212, soit 51 voix normales et 161 voix pathologiques alors que j'en ai utilisées 684 afin d'avoir un maximum de données pour gagner en précision.

Compte tenu que le paramètre Fr n'est pas efficace pour la discrimination comme je l'ai démontré précédemment, j'ai fait la classification sans ce dernier, c'est à dire avec uniquement les paramètres $Ocmax$ et $Ocmean$. Les résultats de cette classification en figure 14 sont quasiment les mêmes qu'en figure 13 avec le paramètre Fr, ce qui confirme son inefficacité.

Classification 2D avec la méthode LDA		
	Voix bien classées	Taux
Voix normales	32 (sur 36)	88,89%
Voix pathologique	570 (sur 648)	87,96%
Total	602 (sur 648)	88,01%

FIGURE 14 – Résultats de la classification LDA avec les paramètres $Ocmax$ et $Ocmean$

J'ai procédé ensuite à la classification SVM (voir résultats en figure 15 ci dessous) afin de comparer cette méthode avec la classification LDA.

Classification 3D avec la méthode SVM		
	Voix bien classées	Taux
Voix normales	21 (sur 36)	58,33%
Voix pathologique	641 (sur 648)	98,92%
Total	662 (sur 648)	96,78%

FIGURE 15 – Résultats de la classification SVM avec les paramètres $Ocmax$, $Ocmean$ et Fr

En comparant ces résultats avec ceux obtenus en figure 13, j'ai constaté que le taux de classification total est nettement plus important avec la classification SVM qu'avec la classification LDA ($96,78\%$ contre $88,16\%$). Cependant, les voix normales sont moins bien classées avec un taux faible de $58,33\%$ (contre $88,89\%$ avec LDA). La méthode à privilégier est celle qui pénalise moins les voix normales, donc la méthode de classification LDA.

Remarques :

• La précision des classifications (LDA et SVM) a été estimée en utilisant la méthode leave-one-out (LOO). Avec cette méthode, un échantillon (une des voix normales ou pathologiques) est exclu de la base de données utilisée et le classifieur est formé avec les échantillons restants. Puis le signal exclu est utilisé en tant que donnée de test et la précision de la classification est déterminée. Cette opération est répétée pour tous les échantillons de la base de données. Ainsi, chaque signal est exclu de l'ensemble d'apprentissage à son tour, ce qui maintient l'indépendance entre l'ensemble de test et l'ensemble d'apprentissage.

• Pour la classification SVM, j'ai utilisé une fonction noyau de type linéaire.

6.3.2 Classification pour les voyelles

J'ai disposé de 53 voix normales et 652 voix pathologiques.

De même que pour les signaux de phrases, j'ai extrait d'abord le paramètre *Ocmax* pour chacun des signaux, ensuite le paramètre *Ocmean* puis le paramètre Fr.

• Classification du paramètre *Ocmax*

L'extraction des *Ocmax* a montré que sur les 53 signaux de voyelles normales, 51 ont pour Ocmax l'octave 14 et les deux autres ont pour Ocmax l'octave 13. Pour les signaux pathologiques, l'Ocmax la plus sélectionnée est l'octave 12 (273 fois sur 652).

Dans le cas présent les figures 16 et 17 montrent clairement que les distributions des *Ocmax* pour les voix normales et pathologiques sont à supports disjoints.

Ocmax					
Distribution	maximum	minimum	moyenne	seuil	pourcentage
Normale	939	558	747	500	100%
Pathologique	446	205	278		100%

FIGURE 16 – Tableau des distribution des *Ocmax*

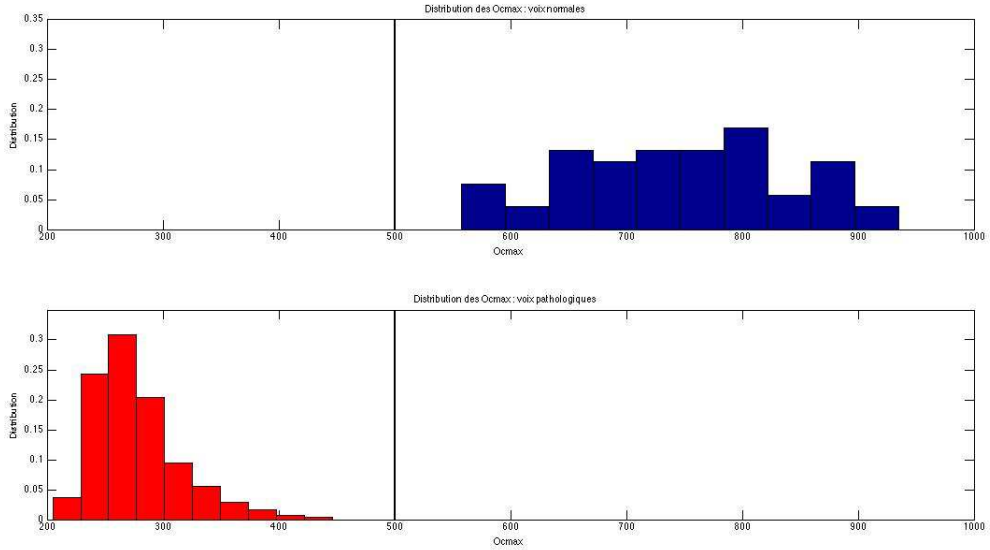

FIGURE 17 – Distribution (normalisée) des *Ocmax*

Ainsi, le seuil fixé à 500 permet sépare nettement ces deux distributions : en rouge nous avons les voix pathologiques et en bleu les voix normales. Le taux de classification est donc maximum (tous les signaux sont bien classés). Nous déduisons que le paramètre *Ocmax* est plus discriminant dans le cas des signaux de voyelles que dans le cas des signaux de phrases.

• Classification du paramètre *Ocmean*

Ici, nous obtenons également de meilleurs résultats que dans le cas de signaux de phrases : le paramètre *Ocmean* parvient à classer la quasi totalité les signaux : 702 signaux sur les 705 sont bien classées.

Ocmean							
Distribution	maximum	minimum	moyenne	écart type	seuil	classe	pourcentage
Normale	43,57	5,42	17,57	7,46	34	51/53	96,23%
Pathologique	131	33	82,98	17,36		651/652	99,55%
					Total	702/705	99,57%

FIGURE 18 – Tableau des distribution des *Ocmean*

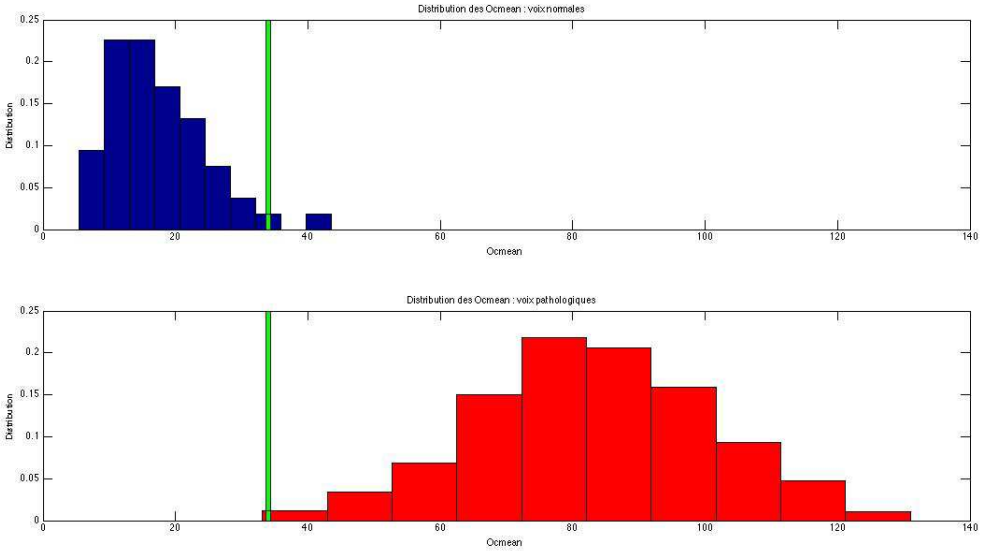

FIGURE 19 – Distribution (normalisée) des *Ocmean*

• Classification du paramètre Fr

La distribution des Fr pour les signaux de voyelle est quasiment la même que pour les signaux de phrases. Nous ne pouvons donc pas fixer de "bon" seuil (différent de 1) pour une classification.

Fr							
Distribution	maximum	minimum	moyenne	écart type	seuil	classe	pourcentage
Normale	1	0,94	0,99	0,0079	1	0/53	0%
Pathologique	1	0,34	0,88	0,11		652/652	100%
					Total	652/705	92,48%

FIGURE 20 – Tableau des distribution des Fr

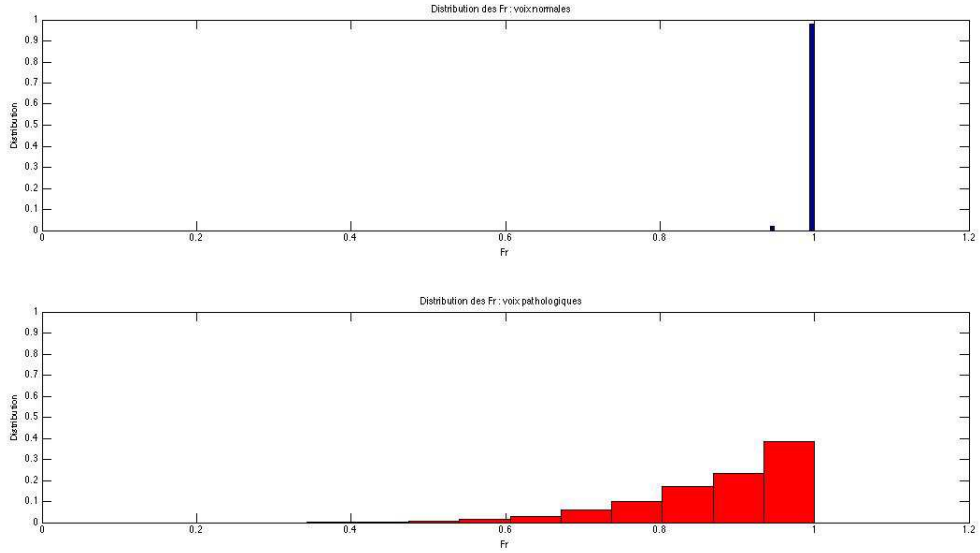

FIGURE 21 – Distribution (normalisée) des Fr

Ainsi, nous constatons que dans le cas des signaux des voyelles, le paramètre Ocmax seul suffit à discriminer totalement les voix normales des voix pathologiques.

6.3.3 Classification avec le paramètre $nbIter$

J'ai utilisé le dictionnaire Gammatone (voir section 4.5.2). J'ai décomposé les signaux en prenant comme critère d'arrêt un SNR fixé à $20dB$, et j'ai observé l'influence du paramètre $nbIter$ pour la classification (cf Annexe 2). Je rappelle que ce paramètre correspond au nombre d'atomes temps-fréquences utilisés lors de la décomposition matching pursuit d'un signal donné. J'ai utilisé ce paramètre pour les signaux de phrases car ils ont tous la même durée, à savoir 12 secondes, contrairement aux signaux de voyelles.

D'après les distributions affichées en figures 22 et 23, j'ai constaté que la distribution des signaux pathologiques a une variance beaucoup plus importante que celle des signaux normaux. Pour déterminer un seuil (de classification), on choisit de privilégier au maximum les voix normales. Ainsi, j'ai estimé que les signaux ayant des valeurs $nbIter$ supérieures à 15157 sont pathologiques, et que tous les autres sont normaux. Cette décision se fonde sur le fait que les signaux pathologiques contiennent généralement plus de structures irrégulières (et donc d'atomes temps-fréquences) que les signaux normaux.

nbIter							
Distribution	maximum	minimum	moyenne	écart type	seuil	classe	pourcentage
Normale	15157	8229	11969	1857	15157	36/36	100,00%
Pathologique	55379	3435	14311	8487		190/648	29,32%
					Total	226/684	33,04%

FIGURE 22 – Tableau des distribution des *nbIter*

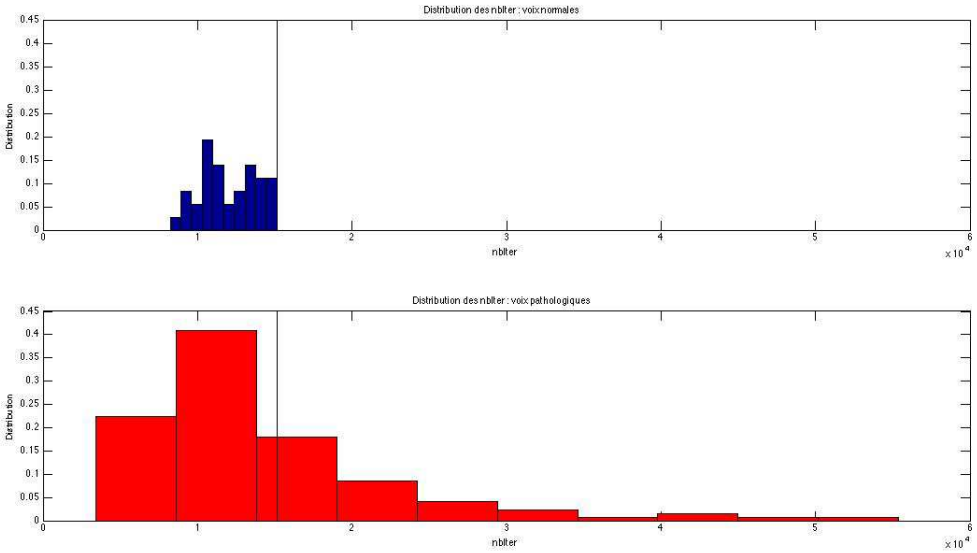

FIGURE 23 – Distribution (normalisée) des *nbIter*

Fixer le seuil de la sorte a défavorisé considérablement les signaux pathologiques. Cependant en couplant ces paramètres avec les paramètres précédents ($Ocmax$, $Ocmean$ et Fr), j'ai réussi à augmenter le taux de classification des voix pathologiques, ce qui souligne la qualité discriminante du paramètre *nbIter* (voir tableau ci-dessous en comparaison avec celui en figure 13).

Classification 4D avec la méthode LDA		
	Voix bien classées	Taux
Voix normales	32 (sur 36)	88,89%
Voix pathologique	585 (sur 648)	90,28%
Total	617 (sur 648)	90,20%

FIGURE 24 – Résultats de la classification LDA avec $Ocmax$, $Ocmean$, Fr et *nbIter*

7 Conclusion

Les résultats des classifications des signaux de paroles (phrases et voyelles) à partir des paramètres extraits du matching pursuit ont été révélateurs de la pertinence ou non des caractères discriminants de ces paramètres. Au final, les données les plus discriminantes se sont révélées être $Ocmax$, $Ocmean$ et $nbIter$. Les signaux de voyelles normales et pathologiques ont pu être classées avec un taux de classification maximum, soit 100%, et les signaux de phrases l'ont été avec un taux d'environ 90%.

L'extraction des différents paramètres a été possible grâce à des représentations adaptatives des signaux sonores, c'est-à-dire en utilisant pour un même signal plusieurs résolutions temps-fréquences. Par la suite, des améliorations sont envisageables, notamment sur la prise en compte des paramètres psychoacoustiques des signaux sonores dans l'algorithme du matching pursuit. Des algorithmes ont déjà été implémentés sur cette base, mais pour des raisons d'optimalité ils n'ont pas pu être exploités durant le stage.

Enfin, l'expérience acquise au sein de l'INRIA restera une étape majeure dans ma formation de master en traitement du signal. Pour un premier contact avec le monde de la recherche, j'en ressors très satisfait, d'autant plus que j'avais toujours souhaité travailler concrètement dans le domaine du traitement des signaux sonores. J'ai pu être confronté aux difficultés réelles du travail de recherche. En effet, il a fallu m'adapter à la problématique soulevée en étudiant ses éléments bibliographiques et les algorithmes déjà implémentés au sein de l'équipe sur le sujet. Tout cela, afin de mieux exploiter les codes (en matlab). La rigueur en organisant mon travail, l'efficacité face aux difficultés rencontrées ont été des qualités que j'ai développées tout au long du stage, qualités qui sont d'emblée acquises pour ma vie professionnnelle future.

8 Annexes

8.1 Annexe 1 : Description de la base de données

Description de la Base de Données

Enregistrements			
	Normale	**Pathologique**	**Total**
Voyelle	53	657	710
Phrase	53	662	715

Personnes		
	Normale	**Pathologique**
Masculin	21	200
Féminin	32	276
Sexe non précisé (snp)	0	247
Total	53	723

Remarque: Il n' y a pas d'enregistrements pour toutes les personnes
(dont la voix est pathologique) décrites dans la base.
C'est pour cela qu'au niveau de la table personnes il y a 723 personnes
dont la voix est pathologique alors qu'au niveau de la table Enregistrements
il y a 657 voyelles pathologiques et 662 phrases pathologiques.

F.d'E (Hz)				
	10000	**25000**	**50000**	**Total**
Voyelle normale	0	0	53(21M, 32F)	53
Voyelle pathologique	0	580(149M, 214F)	77(22M,25F)	657
Phrase normale	17 (0M, 17F)	36(21M, 15F)	0	53
Phrase pathologique	13(2M,4F)	648(160M,230F)	1	662

Remarque: Il y a des enregistrements pour lesquels le sexe n'est pas précisé.
Exemple: 13 phrases patholgiques échatillonnées à 10000 Hz dont 2 enregistrements masculins,
4 féminins, et le reste est non précisé.

8.2 Annexe 2 : Codes matlab utilisés pour l'extraction des paramètres en section 6.3

```
% Calcul des paramètres Ocmax, Ocmean et Fr
% pour un signal donnee signal_x.wav (avec mptk Gabor)


M = 2000; % nombre d'itérations pour la décomposition des signaux (nbIter)

dict_gabor = dictread('dic/dic_gabor.xml');

[ signal sampleRate ] = sigread( Path_to_signal_x.wav  );

[book residual decay ] = mpdecomp( signal, sampleRate, dict_gabor, M );

% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Relevé des paramètres

L_book = length( book.atom ); % Nombre de blocs dans book

% Calcul des octaves

% V_oct : vecteur contenant les octaves des atomes

V_oct = log2( book.atom( 1 ).params.len );
for b = 2 : L_book
    oct = book.atom( b ).params.len;
    V_oct = [ V_oct; log2( oct ) ];
end


% Occurences des octaves

Un = unique( V_oct ); % renvoie les différents octaves utilisées

% Max des octaves Ocmax

Ocmax = 0;
for i = 1 : length( Un )
    max_oct = length( find( V_oct ==  Un( i ) ) );
    if max_oct > Ocmax
       Ocmax = max_oct ;
       long_oct = Un( i ); % renvoie l'octave qui est le plus sélectionné
    end
```

```
end


% Moyenne des octaves Ocmean

Ocmean = 0;
for j = 1 : find( Un ==  7 )
    Ocmean = Ocmean +  length( find( V_oct ==  Un( j ) ) );
end
Ocmean = Ocmean/7;


% Calcul des fréquences

% V_fr : vecteur contenant les fréquences des atomes

 V_fr = sampleRate * book.atom( 1 ).params.freq;

 for k = 2 : L_book
    fr = sampleRate * book.atom( k ).params.freq;
    V_fr = [ V_fr; fr ];
 end

nb_fr = length( unique( V_fr ) ); % nombre des fréquences uniques

% Calcul du taux de fréquences Fr

Max_fr = max( V_fr );

 M_lf = 0;
 for l = 1 : M
     if V_fr( l ) <= ( sampleRate / 4 )
        M_lf = M_lf + 1;
     end
 end

Fr = M_lf / M;


% Vecteur des paramètres

M_param = [ Ocmax Ocmean Fr ];
```

```
% Calcul du parametre nbIter pour la classification (avec Gammatones_ERB)


M = 200000; % nombre d'iterations (pas atteint) pour la decomposition des signaux

len = 2048; % taille max des atomes
numchans = 64; % nombre d'atomes

exitIter = M;
exitSNR = 20; % critère d'arrêt

[ sig.y sig.Fs ] = wavread( signal_x.wav );

mincf = 0;
maxcf = sig.Fs / 2;

[ maxs, cfs, dict, duration ] = gammatones_ERB(sig.Fs, mincf, maxcf, numchans, len);

[ ws_CMP, r_CMP, snr_CMP, Vect_En_Res, iter_sortie ] =
CMP( sig.y, dict', exitIter, exitSNR );

snr =
10 * log10( ( ( 1/length( sig.y ) ) .* sum( sig.y.^2 ) ) ./
( ( 1/length( sig.y ) ) .* sum( ( sig.y-r_CMP ).^2 ) ) )


% Vecteur des parametres

M_param = [ iter_sortie ]; % iter_sortie = nbIter
```