



HAL
open science

Modèles et algorithmes pour la modélisation parcimonieuse de signaux de grande dimension

Boris Mailhé

► **To cite this version:**

Boris Mailhé. Modèles et algorithmes pour la modélisation parcimonieuse de signaux de grande dimension. Traitement du signal et de l'image [eess.SP]. Université Rennes 1, 2009. Français. NNT : . tel-00512559

HAL Id: tel-00512559

<https://theses.hal.science/tel-00512559v1>

Submitted on 30 Aug 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de
DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Traitement du Signal

Ecole doctorale Matisse

présentée par

Boris Mailhé

préparée à l'unité de recherche n° 6074 IRISA
(Institut de Recherche en Informatique et Systèmes Aléatoires)
(Composante universitaire)

**Modèles et algorithmes
pour la représentation
parcimonieuse de signaux
de grande dimension**

**Thèse soutenue à Rennes
le 2 décembre 2009**

devant le jury composé de :

Fabrice WENDLING

Directeur de recherche INSERM / président

Christian JUTTEN

Professeur à l'Université Joseph Fourier / rapporteur

Gabriel PEYRÉ

Chargé de recherche au CNRS / rapporteur

Maureen CLERC

Directrice de recherche INRIA / membre invité

Pierre VANDERGHEYNST

Professeur associé EPFL / co-directeur de thèse

Frédéric BIMBOT

Directeur de recherche CNRS / directeur de thèse

Rémi GRIBONVAL

Directeur de recherche INRIA / co-directeur de thèse

Why play so many notes instead of just choosing the most beautiful ?
Miles Davis

Remerciements

Merci Rémi, Pierre et Frédéric pour m'avoir encadré pendant ce travail.
Merci aux membres du jury qui vont consacrer un peu de leur temps à l'examen de cette thèse.

Merci à tous les collègues et amis grâce à qui ces années sont passées si vite.
Merci en particulier à Karine pour m'avoir permis de camper dans son salon quand je cherchais un appartement à Lausanne.

Merci à ma compagne Xiang pour m'avoir soutenu pendant l'écriture de ce manuscrit.

Table des matières

Remerciements	3
Table des matières	4
Introduction	11
1 Décomposition parcimonieuse rapide de signaux longs sur des dictionnaires locaux	21
1.1 Problème	21
1.2 État de l'art	22
1.2.1 Minimisation de norme ℓ_p	22
1.2.2 Algorithmes gloutons	23
1.2.2.1 Seuillage dur	23
1.2.2.2 Matching Pursuit (MP)	23
1.2.2.3 Orthogonal Matching Pursuit (OMP)	24
1.2.3 Complexité des algorithmes de poursuite	24
1.2.4 Dictionnaires rapides	25
1.2.5 Gradient Pursuit (GP)	26
1.3 Implémentation rapide de MP pour les dictionnaires invariants par translation	27
1.4 Conditions d'optimalité	27
1.4.1 Conclusion	28
1.5 Implémentation « rapide » d'OMP sur des dictionnaires locaux	28
1.5.1 Calcul des corrélations	28
1.5.2 Calcul de la matrice de Gram	29
1.5.2.1 Bilan	32
1.6 LocOMP	32
1.6.1 Description générale.	32
1.6.2 Complexité	32
1.6.3 Choix du sous-dictionnaire	33
1.7 Extension aux dictionnaires multi-échelles	34
1.8 Implémentation dans MPTK	36
1.8.1 Algorithme LocGP	38
1.8.2 Accès rapide aux atomes sélectionnés	38
1.9 Application aux signaux multicanaux	39
1.10 Résultats expérimentaux	40
1.10.1 En MATLAB	40
1.10.2 MPTK	42
1.11 Perspectives	42
1.11.1 Multidimensions	42

1.11.2	Quantification de l'erreur commise par rapport à OMP	44
1.12	Conclusion	45
2	Apprentissage de dictionnaires invariants par translation	47
2.1	État de l'art	48
2.1.1	Algorithme d'Olshausen-Field	49
2.1.2	Algorithme MOD	49
2.1.3	Algorithme K-SVD	50
2.1.4	Conclusion	51
2.2	Pas optimal pour l'algorithme d'Olshausen et Field	51
2.2.1	Problème	52
2.2.2	Calcul de l'atome optimal $\varphi_d^{(i)}$	52
2.2.3	Lien avec le pas de descente ρ dans l'algorithme d'Olshausen et Field	53
2.2.4	Complexité	54
2.2.5	Lien avec K-SVD	55
2.3	Résultats expérimentaux	56
2.3.1	Objectif et mesure de qualité de l'apprentissage	56
2.3.2	Synthèse de données de tests	56
2.3.3	Quel est le gain du pas adaptatif par rapport au pas fixe?	57
2.3.3.1	Comparaison à coefficients fixés	57
2.3.3.2	Comparaison à support de décomposition fixé	57
2.3.3.3	Ensemble de l'algorithme	59
2.3.4	Évaluation sur la base d'images d'Olshausen et Field	59
2.4	Apprentissage de dictionnaires invariants par translation	68
2.5	Apprentissage de dictionnaires multicanaux	69
2.5.1	Mélange sans déformation	69
2.5.2	Mélange avec déformations quelconques	70
2.6	Conclusion	71
3	Séparation de signaux cardiaques par apprentissage de dictionnaires	73
3.1	Introduction : la fibrillation auriculaire	73
3.1.1	Phénomène clinique	73
3.1.2	Acquisition des électrocardiogrammes	74
3.1.3	Problème : extraction du signal de fibrillation	74
3.2	État de l'art	75
3.2.1	Analyse en composantes indépendantes (ICA)	76
3.2.2	Soustraction de battement typique (ABS)	77
3.2.3	Méthodes basées sur la parcimonie	77
3.2.4	Conclusion	79
3.3	Analyse des limites de l'ABS	79
3.3.1	Un seul battement typique	80
3.3.2	Deux battements typiques	80
3.4	Apprentissage de dictionnaire pour la suppression de l'activité ventriculaire	81
3.4.1	Modèle génératif	81
3.4.2	Principe	82
3.4.3	Description du système	82
3.5	Résultats expérimentaux	84
3.5.1	Évaluation sur des données synthétiques	84
3.5.2	Expérimentation sur données cliniques	86

3.5.2.1	Caractéristiques des données cliniques	86
3.5.2.2	Analyse qualitative	88
3.5.2.3	Analyse quantitative	88
3.6	Perspectives	94
3.6.1	Apprentissage à taille de dictionnaire variable	94
3.6.2	Estimation automatique de la taille des dictionnaires à chaque itération	94
3.7	Modélisation <i>a priori</i> des motifs	96
3.8	Conclusion	96
Bibliographie		106
Table des figures		107
Liste des algorithmes		111

Table des notations

Vecteurs	φ atome s signal r résidu x coefficients de décomposition λ corrélations du résidu avec le dictionnaire m motif \hat{x} solution recherchée à un problème d'optimisation
Matrices	Φ dictionnaire Ψ sous-dictionnaire S signal multicanal ou signaux d'apprentissage R résidu X coefficients de décomposition A activité atriale V activité ventriculaire
Indexation	X_i $i^{\text{ème}}$ colonne de X $X_{i,:}$ $i^{\text{ème}}$ ligne de X x_i $i^{\text{ème}}$ coefficient du vecteur x $x^{(i)}$ variable x à l'itération i
Dimensions	N longueur du signal D nombre d'atomes dans le dictionnaire K nombre d'atomes dans la décomposition L longueur d'un motif T longueur de modification du résidu dans LocOMP B nombre de battements cardiaques dans un ECG I nombre d'itérations d'un algorithme C nombre de canaux S nombre de signaux d'apprentissage
Normes	$\ r\ _2$ norme euclidienne $\ x\ _1$ norme ℓ_1 des coefficients $\ x\ _0$ nombre de coefficients non nuls dans le vecteur $\ M\ _{FRO}$ norme de Frobenius d'une matrice
	$\langle \varphi_1, \varphi_2 \rangle$ produit scalaire de deux vecteurs $\operatorname{argmin}_E f(e)$ ensemble des éléments e de E qui minimise $f(e)$ \mathcal{M} ensemble de motifs \mathcal{T} ensemble de transformations M^* adjoint de M

Introduction

L'objet de cette thèse est d'apporter des réponses pratiques aux problèmes qu'on rencontre quand on souhaite utiliser des représentations parcimonieuses. Les représentations parcimonieuses consistent à approcher le signal comme une combinaison *linéaire* d'un faible nombre de vecteurs appelés *atomes* choisis dans un *dictionnaire*. Pour trouver une représentation basée sur peu d'atomes, on a besoin d'un large choix dans les atomes pouvant participer à la décomposition. On utilise donc des dictionnaires *redondants*, contenant plus d'atomes que la taille du signal.

On se trouve alors devant deux problèmes :

- comment choisir dans un dictionnaire donné les atomes qu'on va utiliser pour décomposer un signal donné ?
- comment trouver un dictionnaire qui permet la représentation parcimonieuse d'une certaine famille de signaux ?

Ces deux problèmes sont très complexes. Les algorithmes qui les résolvent exactement sont inutilisables en pratique. D'autres algorithmes sous-optimaux mais plus rapides existent, mais eux aussi montrent leurs limites quand on veut traiter de grands signaux de plusieurs millions d'échantillons. Cette thèse propose des algorithmes capables de traiter des signaux de grande taille sans sacrifier la qualité du résultat, ainsi qu'une application de la parcimonie pour la séparation de sources cardiaques.

Notions de base

Les physiciens, les mathématiciens, les informaticiens ne connaissent pas leur bonheur. Certes on a souvent une idée déformée de leur travail, certes il est souvent difficile d'expliquer leur sujet de recherche précis, mais au moins on sait que leur domaine existe ! Nous allons donc commencer par une introduction très générale au traitement du signal destinée aux nombreuses personnes pour qui le mot signal n'évoque rien d'autre qu'une marque de dentifrice.

Qu'est-ce que le traitement du signal ?

Un signal est une grandeur physique *mesurée* qui varie dans l'espace et/ou dans le temps : sons enregistrés, images capturées par un appareil photo, sismographes, électrocardiogrammes, tension électrique dans une ligne ADSL, ... Le traitement du signal est la discipline qui manipule ces signaux dans plusieurs buts :

- améliorer la qualité perçue par l'humain (exemples : un égaliseur de lecteur audio, la suppression des yeux rouges sur une photo avec flash,...)
- extraire automatiquement de l'information (exemples : reconnaître des mots dans une phrase, des objets dans une image, des bits dans un courant électrique,...)



FIG. 1 – Comparaison de la photographie d’une scène naturelle (portrait de Lena, à gauche) et d’une image aléatoire (pixels indépendants de loi uniforme sur l’intervalle $[0, 1]$) à droite.

- intégrer de l’information au signal de manière à ce qu’elle puisse être extraite ensuite (exemple : modems ADSL, radiocommunications)

Traitement numérique Avant l’avènement de l’informatique, l’essentiel de ces traitements étaient réalisés par des circuits électriques. De nos jours, le traitement du signal est presque entièrement effectué dans le monde numérique. Le signal physique commence par être *échantillonné* (on ne prend qu’un nombre fini de mesures) et *quantifié* (chaque mesure ne peut prendre qu’un nombre fini de valeurs distinctes), ce qui permet d’approcher le signal par une suite finie de bits manipulable par un ordinateur.

Modèles

Même finies, ces données sont souvent très grandes. Par exemple, pour représenter numériquement tous les sons audibles par l’oreille humaine, il faut prendre autour de 40 000 échantillons par seconde (un CD audio est échantillonné à 44 100 Hz). Donc pour une chanson de 5 minutes, il faut $44100 * 60 * 5 \approx 13$ millions d’échantillons. Pour réduire cette complexité on utilise des *modèles*.

La dimension N d’un espace est le nombre de paramètres qu’il faut fixer pour localiser un point de manière unique. Par exemple, 2 coordonnées suffisent pour se repérer sur une carte, donc une carte est de dimension 2. La chanson numérique précédemment mentionnée est de dimension $N = 13$ millions.

Pourtant, parmi tous les signaux existants dans cet espace, la très grande majorité ne pourraient jamais passer pour des chansons. Par exemple, la figure 1 montre côte à côte le portrait de Lena (image couramment utilisée pour faire des tests) et une image où chaque pixel a été généré aléatoirement et indépendamment des autres. On voit bien que l’image aléatoire ne pourrait jamais passer pour une photographie. Une scène naturelle est composée d’objets, souvent opaques, qui se cachent les uns aux autres à l’objectif de l’appareil photo. Sur la photographie obtenue, on voit donc de grandes zones homogènes qui correspondent à la surface d’un objet et de contours vifs qui marquent la frontière entre deux objets. Rien de tout ça dans la « neige » aléatoire. Pour la chanson, le phénomène qui a généré le signal est considérablement plus simple que les 13 millions de mesures : la chanson a été produite en jouant une partition qui contient de l’ordre du millier de notes.

Choisir un modèle des signaux, c'est séparer l'espace entre une petite zone qui contient les signaux intéressants et une grande zone qui contient tout le reste. Comme la zone d'intérêt est plus petite que l'espace, elle rend les signaux plus simples à représenter : au lieu d'identifier une chanson par ses 13 millions de coefficients, on la représente par :

- le fait qu'elle appartient à un modèle de chansons, de photographies, ...
- un jeu restreint de $K \ll N$ paramètres qui identifient la chanson à l'intérieur du modèle.

Il y a donc 2 dimensions : la *dimension extrinsèque* N de l'espace de mesure, grande, et la *dimension intrinsèque* K du modèle.

On utilise souvent des modèles « flous », à base de statistiques par exemple, afin de gommer d'éventuelles erreurs de modélisation : au lieu qu'un signal soit dans ou hors du modèle, on lui donne un score d'appartenance continu entre 0 et 1. Plus ce score est fort, plus la donnée appartient au modèle.

Parcimonie

On peut utiliser l'invariance des données étudiées par diverses transformations pour décrire le modèle simplement : si un signal appartient au modèle, alors sa transformée appartient aussi au modèle. Par exemple, dans le cas de la musique, le volume sonore d'une note peut varier et 2 notes différentes jouées en même temps s'ajoutent. On peut donc définir un modèle musical par :

- une note est de la musique,
- un morceau de musique amplifié, c'est-à-dire multiplié par un nombre, est de la musique,
- l'addition de deux morceaux de musique est de la musique.

Un modèle invariant par addition et multiplication par un nombre est dit *linéaire*. L'étendue de l'espace qu'il couvre dépend des éléments de base, ou *atomes* (les notes pour notre chanson). On sait que s'il y a au moins 13 millions de notes possibles, (c'est le cas si on compte tous les instants auxquels une note peut être jouée), alors le modèle englobe tout l'espace de mesure. Or l'intérêt d'un modèle est d'être plus petit que l'espace. Donc la linéarité seule ne suffit pas.

Parcimonie On ajoute une hypothèse au modèle pour en réduire l'étendue. Un morceau de musique contient $K \ll N$ notes. Le modèle devient alors une union de sous-espaces, comme le montre la figure 2 pour un espace de dimension 3.

Problèmes étudiés

Cette thèse s'attache à la résolution de 2 problèmes importants qu'on rencontre quand on travaille sur des modèles parcimonieux ainsi qu'à une application médicale des ces modèles.

Décomposition parcimonieuse Décomposer un signal sur un modèle parcimonieux, c'est trouver quels atomes sont actifs. Quand il y a plus d'atomes que N dans le modèle, il existe plusieurs décompositions. Il est facile d'en trouver une, mais très difficile de trouver celle qui contient le moins d'atomes : il faudrait essayer toutes les décompositions possibles et les comparer. Nous avons mis au point un algorithme rapide qui calcule non pas la meilleure décomposition, mais une avec relativement peu d'atomes.

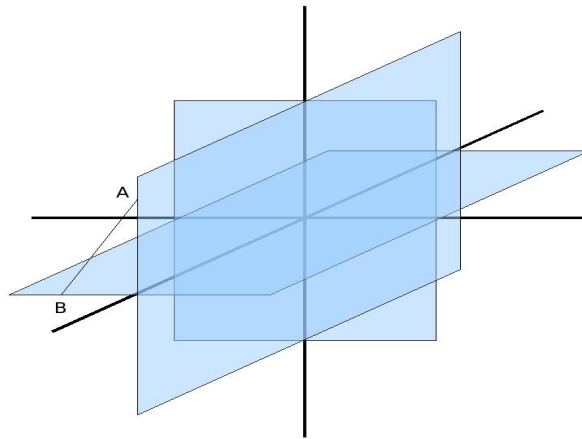


FIG. 2 – Sous-espaces parcimonieux avec $K = 2$ atomes d'un dictionnaire à 3 atomes dans un espace de dimension $N = 3$. Les axes sont la direction des atomes. Le modèle est l'union des plans bleus : les signaux recherchés peuvent se trouver dans n'importe lequel des plans, mais pas ailleurs. Ce modèle est non-convexe : les points A et B sont dedans, mais aucun point du segment $]A, B[$ n'y est.

Apprentissage de dictionnaires Un même modèle ne permet pas de représenter tous les signaux de l'espace. Quand on souhaite étudier des signaux, il faut donc choisir un bon modèle pour eux. Dans le cadre parcimonieux, le choix du modèle correspond au choix du dictionnaire. Apprendre un dictionnaire, c'est chercher un dictionnaire qui représente bien un ensemble de données d'apprentissage.

Nous avons proposé une amélioration d'un algorithme d'apprentissage de dictionnaires existant. Cette amélioration rend l'algorithme plus rapide et plus simple à utiliser.

Étude de la fibrillation atriale La fibrillation atriale est la plus courante des anomalies du rythme cardiaque. Au lieu de battre, les atria (ou oreillettes) se mettent à trembler. Nous avons utilisé l'apprentissage de dictionnaires pour isoler les traces de la fibrillation dans l'électrocardiogramme du patient.

Définitions

Parcimonie sur une base

La notion de parcimonie permet de formaliser le concept abstrait d'une base « bien adaptée » pour représenter une famille donnée de signaux numériques.

Parcimonie exacte Soit H un espace de Hilbert de dimension N . Un signal $s \in H$ est dit K -parcimonieux sur une base Φ si sa décomposition x telle que $s = \Phi x$ ne contient que $K \ll N$ coefficients non nuls. Le nombre de coefficients non nuls de x est habituellement noté $\|x\|_0 = \sum_{1 \leq n \leq N} |x_n|^0$. Ainsi, alors que l'algèbre linéaire montre que tout signal peut être décomposé sur toute base, la parcimonie établit une nuance entre les signaux qui ont effectivement besoin de toute la base et ceux qui peuvent se contenter d'un sous-ensemble. Géométriquement, l'ensemble des signaux K -parcimonieux sur Φ est l'union des C_N^K sous-espaces engendrés par toutes les combinaisons possibles de K vecteurs de Φ . Cet ensemble est non-convexe, comme le montre la figure 2.

Parcimonie approchée Les signaux numériques étant en général bruités, ne serait-ce qu'à cause de la quantification, on doit donc souvent se contenter d'une parcimonie approchée. Intuitivement, un signal s possède une *approximation parcimonieuse* sur un dictionnaire Φ s'il peut se décomposer en $s = \Phi x + r$, avec $\|r\|_2 \ll \|s\|_2$ et $\|x\|_0 \ll N$. Bien sûr, cette définition qualitative n'est pas utilisable directement. Elle peut être formulée de plusieurs façons comme un problème d'optimisation sous contrainte. Celle qui sera utilisée ici est la meilleure approximation à K termes :

$$\hat{x} = \operatorname{argmin}_{\|x\|_0 \leq K} \|s - \Phi x\|_2^2$$

Pour une base orthonormale, cette approximation se calcule simplement à partir de la décomposition $x = \Phi^* s$ de s sur Φ en mettant à 0 les coefficients les plus faibles. Le nombre de coefficients affectés dépend du compromis souhaité entre la compacité de l'approximation et l'erreur commise.

Usages La parcimonie procure une aide pour plusieurs applications. La première est la compression avec perte. Si un signal peut être approché parcimonieusement sur une base orthonormale, alors sa décomposition comporte quelques coefficients forts et de nombreux coefficients faibles. On peut se permettre de coder plus grossièrement les petits coefficients :

ils ont moins d'importance sur la qualité de reconstruction finale. Parmi les formats de compression répandus qui exploitent cette idée, on peut citer :

- JPEG qui exploite une transformée DCT
- JPEG-2000 à base d'ondelettes
- les formats audio mp3 et AAC basés sur des transformées MDCT

Comme tout autre modèle, la parcimonie peut également être exploitée en débruitage. Supposons que le signal s est généré à partir d'une composante parcimonieuse et d'un bruit additif blanc gaussien e .

$$s = \Phi x + e$$

Alors on peut débruiter s en calculant une approximation parcimonieuse sur Φ avec un SNR égal à celui de e . C'est la base des techniques de reconstruction d'images par seuillage des coefficients d'ondelettes.

Parcimonie sur un dictionnaire redondant

Toutes les classes de signaux ne sont pas représentables parcimonieusement sur une base orthonormale. C'est notamment impossible quand les signaux possèdent plusieurs composantes aux propriétés différentes : attaque et relaxation pour de la musique, contours et textures pour des images,... Dans ce cas, on peut trouver une bonne base pour représenter chaque composante (DCT pour les textures, curvelets pour les contours, MDCT à courte et longue échelle pour les attaques et relaxations des notes,...). On souhaiterait alors pouvoir représenter le signal sur l'union de ces bases, mais on y perdrait l'unicité de la décomposition.

Définition Un *dictionnaire* $\Phi = (\varphi_m)_{1 \leq m \leq M}$ est une famille de vecteurs unitaires de H appelés *atomes*, ou de manière équivalente une matrice dont les atomes sont les colonnes. Un dictionnaire est dit *redondant* s'il génère tout H et que $M > N$. Dans ce cas, tout signal possède une infinité de décompositions sur Φ . Néanmoins, la notion de parcimonie se généralise : un signal est dit parcimonieux sur un dictionnaire redondant Φ si une de ses décompositions l'est. On peut alors voir la parcimonie comme une contrainte de régularisation qui permet de choisir une décomposition parmi toutes celles possibles.

Complexité Les applications mentionnées précédemment (compression, débruitage, ...) restent envisageables dans le cas redondant, à condition de savoir calculer une approximation parcimonieuse de s . Malheureusement la redondance ajoute une complexité combinatoire à ce calcul. Formellement, on se donne un niveau de parcimonie autorisé K , et on cherche la meilleure approximation possible à ce niveau :

$$\hat{x} = \operatorname{argmin}_{\|x\|_0 \leq K} \|s - \Phi x\|_2^2 \quad (1)$$

Le problème consiste principalement à identifier le support de \hat{x} , c'est-à-dire l'ensemble des indices d pour lesquels les coefficients x_d sont non-nuls. Il suffit ensuite de projeter s sur les atomes de ce support. Ce problème a été prouvé NP-Complet ; on ne connaît pas de meilleur algorithme exact que l'essai exhaustif des C_N^K supports possibles [DMA97].

Toutes les applications qui exploitent la parcimonie utilisent donc des algorithmes sous-optimaux, mais de complexité polynomiale. En pratique, la distinction entre algorithmes polynomiaux et non polynomiaux n'est pas forcément pertinente en traitement du signal. En effet, les hautes fréquences d'échantillonnage amènent facilement à traiter des données de très grande taille : pour une seule image 1024*1024 ou une minute de musique échantillonnée à 16kHz, on est déjà à $N \approx 10^6$. Dans ce cas, on ne voit plus la différence entre un

algorithme exponentiel et un algorithme polynomial de degré élevé : les deux sont trop coûteux pour être utilisés. Sur de vraiment grandes données, même la complexité quadratique N^2 est inaccessible. Par exemple, pour traiter un CD de musique (1 heure, échantillonné à 44.1kHz) avec un programme de complexité N^2 sur un ordinateur de fréquence 1GHz, il faudrait presque 1 an ! Il y a donc un besoin pour des algorithmes qui permettent de traiter des données de grande dimension en un temps raisonnable.

Approximation parcimonieuse rapide de signaux longs sur des dictionnaires locaux

Puisque la recherche exhaustive de la meilleure approximation à K termes définie à l'équation (1) est en général trop coûteuse, de nombreux algorithmes sous-optimaux ont été proposés.

Minimisation de norme ℓ_p Une première approche est inspirée par l'optimisation numérique. Dans ce cadre, on sait qu'on peut chercher les minima locaux d'une fonction différentiable par descente de gradient. Si de plus cette fonction est strictement convexe, alors elle n'admet au maximum qu'un unique minimum local qui est aussi global. Malheureusement, la contrainte de parcimonie $\|x\|_0 \leq K$, n'est ni convexe, ni même continue. On cherche donc à remplacer la norme ℓ_0 par une fonction proche, mais plus simple à optimiser. Comme

$$\|x\|_0 = \lim_{p \rightarrow 0^+} \sum_{d=1}^D x^p \quad (2)$$

$$= \lim_{p \rightarrow 0^+} \|x\|_p^p \quad (3)$$

les candidats naturels sont les pseudo-normes $\|x\|_p$ avec p proche de 0. Ces pseudo-normes fournissent un résultat parcimonieux tant que $p \leq 1$. La norme ℓ_1 , bien que la plus éloignée de la pseudo-norme ℓ_0 , est la plus populaire car c'est une vraie norme, donc convexe. Différents algorithmes ont été proposés pour résoudre ces nouveaux problèmes.

Algorithmes gloutons Une autre vision plus informatique part de la combinatoire du problème. La NP-complétude vient de la complexité gigantesque de l'espace des solutions, qu'on ne peut pas se permettre d'explorer exhaustivement. Il faut donc trouver un moyen de naviguer à vue dans cet espace pour construire une solution petit à petit. C'est le principe des algorithmes gloutons (ou algorithmes de poursuite) : étendre itérativement le support de la décomposition. Un ou plusieurs atomes considérés comme bons sont sélectionnés et soustraits au signal, puis le processus est itéré sur le résidu ainsi obtenu. Plusieurs variantes de ces algorithmes gloutons ont été proposées pour le critère de sélection de l'atome à enlever et la manière de le soustraire du résidu.

Il est amusant de constater que les algorithmes les plus rapides existant actuellement sont également les plus anciens. Puisque ces algorithmes sont sous-optimaux, la communauté semble ensuite s'être plutôt orientée vers la recherche d'algorithmes commettant moins d'erreurs, ce qui a entraîné une inflation des coûts. Notre approche consistera donc plutôt à partir des algorithmes rapides et à chercher à les améliorer, soit en les accélérant, soit en diminuant l'erreur commise pour un coût de calcul négligeable.

Sans faire d'autre hypothèse, tous ces algorithmes sont de complexité au moins quadratique en N , ce qui suffit à empêcher leur utilisation sur des données de grande taille.

Néanmoins, on peut faire mieux si on dispose d'hypothèses supplémentaires sur la structure du dictionnaire.

Cas des dictionnaires locaux Le cas particulier qui nous intéresse est celui des dictionnaires locaux. On suppose que tous les atomes du dictionnaire sont nuls hors d'un intervalle de longueur $L \ll N$. C'est le cas pour des dictionnaires temps-fréquence comme les dictionnaires de Gabor ou MDCT (transformée en cosinus discrets modifiée), par exemple. On sait que le coût d'une itération de Matching Pursuit (MP) peut alors dépendre plus de L et moins de N car tous les traitements effectués restent locaux, ce qui n'est pas le cas pour les autres algorithmes existants. Néanmoins cet algorithme est l'un des plus grossiers existants.

La première partie de cette thèse 1 est consacrée à la présentation de LocOMP, un algorithme glouton capable de calculer de meilleures approximations que MP tout en restant dans la même classe de complexité.

Apprentissage de dictionnaires

Avant même le calcul de décomposition $s = \Phi x + r$ se pose le problème du choix du dictionnaire Φ : comment trouver un dictionnaire qui permet de représenter parcimonieusement un ensemble de signaux d'apprentissage $S = (s_1 \dots s_S)$?

Si ce problème fondamental peut souvent être laissé de côté, c'est parce qu'on peut souvent l'esquiver en faisant appel à une connaissance experte des signaux à traiter. Par exemple, on sait que des bases temps-fréquence sont généralement bien adaptées à la représentation de signaux sonores.

Problème Parfois, le dictionnaire à utiliser n'est pas connu. C'est le cas quand on souhaite traiter des signaux « exotiques » sur lesquels la connaissance *a priori* fait défaut. L'application visée peut aussi nécessiter des dictionnaires plus adaptés qu'un dictionnaire générique aux données particulières qu'on souhaite représenter. Dans ce cas, une autre voie est d'apprendre un dictionnaire à partir d'exemples. Formellement, on se donne un niveau de parcimonie K , une collection de données d'apprentissage rangées dans une matrice $S = (s_1 \dots s_S)$, et on cherche le dictionnaire

$$\hat{\Phi} = \operatorname{argmin}_{\forall \varphi \in \Phi, \|\varphi\|_2=1} \min_{\forall 1 \leq d \leq D, \|x_d\|_0 \leq K} \|S - \Phi X\|_{FRO}^2 \quad (4)$$

La minimisation sur X n'est rien d'autre que l'erreur obtenue par l'approximation parcimonieuse de S sur Φ .

Ce problème est encore plus complexe que le précédent. L'approche sous-optimale classique pour le résoudre estime conjointement les facteurs Φ et X par optimisation alternée. L'optimisation des coefficients à dictionnaire fixé étant elle-même une approximation parcimonieuse, l'essentiel du travail en apprentissage porte sur l'optimisation du dictionnaire.

La deuxième partie de cette thèse présente une extension de l'algorithme d'Olshausen et Field [OF97], dans lequel a été ajoutée l'estimation automatique d'un des paramètres qui devait auparavant être réglé à la main. Cet ajout permet une convergence plus rapide car notre paramètre optimal à chaque itération est meilleur qu'un paramètre fixé pour tout l'algorithme. Elle rend l'algorithme plus facile à prendre en main puisqu'il y a un paramètre de moins à régler. Son surcoût par itération est négligeable devant le temps de calcul d'une itération d'Olshausen et Field.

Séparation des parties auriculaires et ventriculaires d'électrocardiogrammes

Séparation de sources La parcimonie sur des dictionnaires redondants peut également être utilisée en séparation de sources. On observe un signal $s = s_1 + s_2$ mélange de 2 sources s_1 et s_2 . Le problème de séparation de sources consiste à estimer s_1 et s_2 à partir de s et de connaissances *a priori* sur les sources. Si on sait que les sources s_1 et s_2 sont parcimonieuses sur les dictionnaires respectifs Φ_1 et Φ_2 et que chaque source n'est pas parcimonieuse sur le dictionnaire de l'autre, alors on peut séparer les sources en décomposant s sur le dictionnaire $(\Phi_1 \Phi_2)$:

$$s = (\Phi_1 \Phi_2) \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} + r \quad (5)$$

puis en séparant les sources dans l'espace des coefficients :

$$\hat{s}_1 = \Phi_1 \hat{x}_1 \quad (6)$$

$$\hat{s}_2 = \Phi_2 \hat{x}_2 \quad (7)$$

Fibrillation atriale La dernière partie de cette thèse présente une application de cette idée à des signaux d'électrocardiogrammes (ECG). Le but de cette application est d'isoler le signal de fibrillation auriculaire. La fibrillation auriculaire est l'arythmie cardiaque la plus fréquente. Lors de la fibrillation, les oreillettes cessent de battre normalement et se mettent à osciller de manière désordonnée. Le signal ECG enregistré est donc la superposition d'une activité ventriculaire normale et de la fibrillation. On souhaite étudier le signal de fibrillation, mais il n'est jamais observable seul : sans activité ventriculaire, le patient est en arrêt cardiaque. Il faut donc isoler le signal auriculaire dans l'ECG. On considère ce problème comme un problème de séparation de sources avec les activités ventriculaire et auriculaire comme sources.

Apprentissage pour la séparation de sources C'est un exemple de cas où les dictionnaires doivent être appris. En effet, de précédentes tentatives avec des dictionnaires génériques ont donné des résultats décevants [DEGL⁺06]. C'est dû au fait que ces dictionnaires sont justement connus pour permettre de bonnes représentations pour de nombreux signaux. Ils ne sont donc pas assez discriminants pour cette application : le dictionnaire utilisé pour représenter l'activité ventriculaire capte également une partie non négligeable de la fibrillation.

Le principal obstacle rencontré lors de cette application est que contrairement à d'autres applications de séparation de sources, on n'a pas accès à des exemples d'apprentissage d'une source seule. Les deux dictionnaires doivent donc être appris à partir des mêmes données.

Organisation de la thèse

La partie 1 présente les travaux réalisés en décomposition. Après un état de l'art consacré à la description des principaux algorithmes existants, notamment la complexité des algorithmes gloutons et l'implémentation locale de MP, elle décrit les accélérations qui peuvent être apportées à OMP dans le cas des dictionnaires locaux. Le gain se révélant insuffisant, le nouvel algorithme LocOMP est présenté. LocOMP réalise une approximation du comportement d'OMP pour un coût proche de celui de MP. Nous comparerons les performances expérimentales de ces algorithmes et présenterons plusieurs extensions réalisées ou envisageables.

La partie 2 est consacrée à l'apprentissage de dictionnaires. L'état de l'art présente les différentes itérations d'optimisation utilisées pour la mise à jour du dictionnaire, avec pour chacune leurs limites. Nous présentons ensuite le calcul du pas d'apprentissage optimal pour l'algorithme d'Olshausen et Field et son lien avec d'autres approches. Une étude expérimentale sur des données synthétiques et réelles permet de comparer les différents algorithmes et de mettre en évidence leurs faiblesses communes. Enfin nous présenterons les extensions pour des dictionnaires multicanaux et invariants par translation qui seront utilisés dans la partie suivante.

La partie 3 développe l'application de ces outils à la séparation de signaux cardiaques. Après avoir décrit l'origine des signaux et le phénomène qu'on souhaite observer, la fibrillation atriale, nous passerons en revue les méthodes de débruitage ou de séparation de sources qui ont déjà été appliquées à ce problème, afin de montrer ce que notre système emprunte à chacune. Le système et ses idées sous-jacentes seront ensuite expliqués. Nous avons réalisé une étude expérimentale quantitative sur des données synthétiques et qualitatives sur des données cliniques.

Chapitre 1

Décomposition parcimonieuse rapide de signaux longs sur des dictionnaires locaux

1.1 Problème

Contexte Représenter un signal par une décomposition linéaire avec peu de coefficients non-nuls offre de nombreux avantages. On peut par exemple compresser le signal en ne conservant que les coefficients non-nuls ou le traiter plus rapidement. Malheureusement on ne peut pas toujours trouver de base bien adaptée à la décomposition d'une classe de signaux. C'est notamment le cas quand les signaux sont hétérogènes : la musique contient des notes longues et courtes, une photo contient des contours et des textures,... On peut trouver une bonne base pour chaque composante, mais on a besoin de toutes les bases pour décomposer tout le signal. En concaténant les bases on obtient un *dictionnaire redondant*. La décomposition n'est alors plus unique. La décomposition parcimonieuse consiste à rechercher la décomposition qui contient le moins de coefficients parmi celles possibles.

Formalisation Étant donné un signal s de taille N et un dictionnaire redondant Φ de taille $N \times D$ avec $D > N$ (D est le nombre d'atomes du dictionnaire, chaque atome étant une colonne), l'approximation parcimonieuse consiste à tenter d'approcher un signal s par une combinaison linéaire d'un faible nombre d'atomes pris dans le dictionnaire Φ . Si on note $\|x\|_0$ le nombre de coefficients non-nuls de x , on peut décrire le problème de manière floue :

$$s = \Phi x + r \tag{1.1}$$

$$\|r\|_2 \ll \|s\|_2 \tag{1.2}$$

$$\|x\|_0 \ll N \tag{1.3}$$

Il s'agit donc d'optimiser conjointement l'énergie de l'erreur $\|r\|_2^2$ et le nombre de coefficients non-nuls $\|x\|_0$. Ce problème d'optimisation sous contraintes admet plusieurs formulations. Dans un contexte de débruitage, on peut fixer l'énergie résiduelle \mathcal{E} qu'on souhaite atteindre. Le problème à résoudre est alors

$$\hat{x} = \operatorname{argmin}_{\|s - \Phi x\|_2^2 \leq \mathcal{E}} \|x\|_0 \tag{1.4}$$

Si on cherche plutôt à réduire la dimension du signal, on peut fixer la longueur maximale K de la décomposition :

$$\hat{x} = \operatorname{argmin}_{\|x\|_0 \leq K} \|s - \Phi x\|_2^2 \quad (1.5)$$

Enfin, comme tout problème d'optimisation sous contraintes, on peut faire entrer la contrainte dans la fonction de coût par pénalisation :

$$\hat{x} = \operatorname{argmin} \|s - \Phi x\|_2^2 + \eta \|x\|_0 \quad (1.6)$$

Dans tous les cas, un paramètre permet de régler l'importance relative des 2 fonctions. Cela peut être l'énergie résiduelle \mathcal{E} , la parcimonie autorisée K ou le coefficient de pénalisation η . Dans le reste de cette partie on utilisera surtout la formulation (1.5).

Dictionnaires locaux De plus, on se place dans le cas particulier où le dictionnaire est local. Ce cas est d'utilisation courante puisqu'il englobe en particulier toutes sortes d'analyses temps-fréquence.

Définition 1.1 *Un dictionnaire est dit local si tous ses atomes sont nuls hors d'un intervalle de longueur $L \ll N$. Pour chaque atome φ , on note $\operatorname{supp}(\varphi) = [t_{\min}(\varphi), t_{\max}(\varphi)]$ cet intervalle et ses bornes avec $t_{\max}(\varphi) = t_{\min}(\varphi) + L - 1$.*

Dans ce cadre, on sait qu'on peut implémenter Matching Pursuit (MP) beaucoup plus efficacement [KG06]. Grâce à cette faible complexité, on peut envisager de traiter de très grands signaux, de l'ordre du million d'échantillons voire plus. Néanmoins, MP fournit empiriquement des approximations de moins bonne qualité qu'Orthogonal Matching Pursuit (OMP), aussi on aimerait bien disposer d'un OMP capable lui aussi de passer à l'échelle sur de grands signaux, c'est à dire dont la complexité varie en fonction de L au lieu de N .

Dans cette partie nous détaillons les accélérations que l'hypothèse de localité permet d'apporter à OMP. Cette partie présente un algorithme nommé LocOMP qui exploite la propriété de localité du dictionnaire pour fournir une qualité d'approximation proche de celle d'OMP tout en restant dans le même ordre de complexité que MP.

1.2 État de l'art

Le problème 1.5 a été prouvé NP-Difficile [DMA97]. On ne connaît pas d'algorithme plus rapide que l'essai exhaustif des C_N^K supports possibles.

1.2.1 Minimisation de norme ℓ_p

La minimisation de norme ℓ_p est moins une famille d'algorithmes qu'une manière de ramener à des problèmes qu'on sait résoudre, aussi nous n'entrerons pas ici dans le détail des algorithmes. Il est impossible d'utiliser une approche d'optimisation locale pour diminuer une norme ℓ_0 parce que celle-ci est localement constante en presque tout point, ses seules variations étant discontinues. Pourquoi ne pas remplacer cette contrainte par une autre proche mais au moins continue et différentiable par morceaux? Les candidats naturels pour ce remplacement sont les pseudo-normes ℓ_p avec p proche de 0, définies par

$$\|x\|_p = \sqrt[p]{\sum_{d=1}^D x_d^p} \quad (1.7)$$

Il a été prouvé que si on choisit $0 < p \leq 1$, alors la solution au problème

$$\hat{x} = \operatorname{argmin}_{\|s - \Phi x\|_2^2 \leq \mathcal{E}} \|x\|_p \quad (1.8)$$

est parcimonieuse [KdRE⁺99]. La norme ℓ_1 a un statut particulier : c'est la seule fonction convexe qui permet d'obtenir un résultat parcimonieux. On a alors la garantie qu'il existe un unique ensemble convexe de minima locaux de même valeur, qui sont également globaux.

Dans le cas sans bruit ($\mathcal{E} = 0$), le problème de minimisation de norme ℓ_1 peut se formuler comme un problème de programmation linéaire [CDS96].

Dans le cas bruité, la minimisation de norme ℓ_1 admet les mêmes formulations que celles énoncées dans l'introduction de cette partie pour la norme ℓ_0 . Dans tous les cas, ces problèmes sont convexes. On connaît de nombreux algorithmes pour les résoudre depuis au moins la méthode de Newton.

Pour minimiser des normes ℓ_p avec $p < 1$, on utilise plutôt des méthodes itératives comme l'Iterative Reweighted Least Squares [DDFG09][BD06] ou le seuillage itératif [DDM04]. On n'a alors pas de garantie de converger vers le minimum global de la fonction de coût. Le seuillage itératif peut même être utilisé pour minimiser directement la norme ℓ_0 [BD09].

L'erreur commise par cette approche n'est pas tant dans l'algorithme de résolution que dans le changement de norme. Même si l'algorithme trouve le support du vecteur \hat{x} qui minimise la norme ℓ_p , ce support ne minimise pas forcément également la norme ℓ_0 .

1.2.2 Algorithmes gloutons

Le principal défaut des algorithmes de minimisation de norme ℓ_p est leur complexité. En effet, le résultat final est parcimonieux mais au cours de l'algorithme tous les coefficients de X sont en général non-nuls, ce qui engendre un coût important.

L'idée générale des algorithmes gloutons est de contraindre l'algorithme à n'utiliser que de l'ordre de K atomes à chaque itération.

1.2.2.1 Seuillage dur

Principe Il s'agit de l'algorithme le moins cher. Il consiste simplement à sélectionner les K atomes dont le produit scalaire avec le signal s en valeur absolue est le plus grand. On peut ensuite calculer l'amplitude des coefficients x par projection orthogonale sur ce sous-dictionnaire Φ_K .

Limites Cet algorithme fonctionne notoirement mal dès que le dictionnaire comporte des atomes qui se ressemblent. Prenons un cas limite. Soit $\Phi = (\varphi_1, \varphi_2, \varphi_3)$ avec $\varphi_1 = \varphi_2$ et $\varphi_1 \perp \varphi_3$. Soit $s = \varphi_1 + \frac{1}{2}\varphi_3$. Alors un seuillage à $K = 2$ sélectionne le couple (φ_1, φ_2) alors que les solutions optimales seraient (φ_1, φ_3) ou (φ_2, φ_3) . C'est dû au fait qu'on a sélectionné plusieurs fois les atomes qui modélisent la composante la plus importante au détriment de l'autre.

1.2.2.2 Matching Pursuit (MP)

Principe MP évite l'écueil du seuillage en remplaçant la sélection simultanée des K atomes les plus corrélés par une sélection séquentielle [MZ93]. A chaque itération, on ne sélectionne qu'un atome et on l'enlève du signal. Le fonctionnement est détaillé dans l'algorithme 1.

Algorithm 1 $x = \text{MP}(s, \Phi), K$

```

 $r^{(0)} = s$ 
 $\Phi^{(0)} = \emptyset$ 
 $x^{(0)} = 0$ 
for  $i = 1$  à  $I$  do
   $\varphi^{(i)} = \text{argmax}_{\varphi \in \Phi} |\langle r^{(i-1)}, \varphi \rangle|$  {recherche du meilleur atome}
   $x_{\varphi^{(i)}}^{(i)} = x_{\varphi^{(i)}}^{(i-1)} + \langle r^{(i-1)}, \varphi^{(i)} \rangle$  {ajout au coefficient de  $\varphi^{(i)}$ }
   $r^{(i)} = r^{(i-1)} - \langle r^{(i-1)}, \varphi^{(i)} \rangle \varphi^{(i)}$  {soustraction au résidu}
end for
return  $x^{(I)}$ 

```

Limites Le résidu $r^{(i)}$ est le projeté du résidu précédent $r^{(i-1)}$ orthogonalement à l'atome sélectionné $\varphi^{(i)}$. Le défaut principal de cet algorithme est que si Φ n'est pas orthogonal, alors en soustrayant un atome on peut rajouter une composante sur un atome enlevé précédemment. On peut donc devoir sélectionner plusieurs fois un même atome, voire même effectuer un nombre infini d'itérations pour annuler le résidu même si on a déjà identifié le bon support.

1.2.2.3 Orthogonal Matching Pursuit (OMP)

Pour assurer qu'un atome n'est sélectionné qu'une fois, on peut s'assurer que le résidu $r^{(i)}$ est toujours orthogonal à *tous* les atomes sélectionnés précédemment. On définit le sous-dictionnaire $\Phi^{(i)} = (\varphi_j)_{1 \leq j \leq i}$. Le résidu $r^{(i)}$ est alors calculé par projection de s (ou $r^{(i-1)}$, le résultat est le même) orthogonale à $\Phi^{(i)}$ [PRRK93]. Le fonctionnement est détaillé dans l'algorithme 2, avec Φ^* la matrice transconjugée de Φ .

Algorithm 2 $x = \text{OMP}(s, \Phi)$

```

 $r^{(0)} = s$ 
 $\Phi^{(0)} = \emptyset$ 
 $x^{(0)} = 0$ 
for  $i = 1$  to  $I$  do
   $\varphi^{(i)} = \text{argmax}_{\varphi \in \Phi} |\langle r^{(i-1)}, \varphi \rangle|$  {recherche du meilleur atome}
   $\Phi^{(i)} = \Phi^{(i-1)} \cup \varphi^{(i)}$ 
   $\chi^{(i)} = (\Phi^{(i)*} \Phi^{(i)})^{-1} \Phi^{(i)*} r^{(i-1)}$  {calcul des coefficients de la projection}
   $x^{(i)} = x^{(i-1)} + \chi^{(i)}$  {mise à jour des coefficients}
   $r^{(i)} = r^{(i-1)} - \Phi^{(i)} \chi^{(i)}$  {mise à jour du résidu}
end for
return  $x^{(I)}$ 

```

1.2.3 Complexité des algorithmes de poursuite

La complexité des différents algorithmes dépend à la fois de la taille du dictionnaire D , de la dimension de signaux N et du niveau de parcimonie K . De plus on suppose $D > N$ car le dictionnaire Φ est redondant et $K \ll N$ car la représentation est parcimonieuse.

Seuillage dur L'algorithme nécessite avant tout le calcul des corrélations $\Phi^* s$ en $O(DN)$, puis la recherche des K meilleurs atomes majorée par $O(D \log D)$ (c'est le coût pour trier

TAB. 1.1 – Complexité des différentes opérations utilisées par les algorithmes gloutons dans le cas général.

Algorithme	Seuillage	MP	OMP
$\lambda = \Phi^* r$	DN	DN	DN
$\operatorname{argmax} \lambda $	$D \log D$	D	D
$G = \Phi^{(i)*} \Phi^{(i)}$	$K^2 N$	\emptyset	iN
$\chi = G^{-1} \lambda^{(i)}$	K^2	\emptyset	i^2
$r^{(i)} = r^{(i-1)} - \Phi^{(i)} \chi$	KN	N	iN
Terme dominant pour 1 itération	$DN + K^2 N$	DN	DN
Nombre d'itérations	1	$\gtrsim K$	K
TOTAL	$DN + K^2 N$	DKN	DKN

toutes les corrélations), enfin la projection s sur les atomes sélectionnés en $O(NK)$. Le coût dominant est le coût des corrélations en $O(DN)$.

MP Une itération de MP est un seuillage à $K = 1$ atome, donc son coût est principalement celui du calcul des corrélations en $O(DN)$. Le nombre d'itérations nécessaires pour trouver K atomes différents varie selon les cas. Si on suppose que ce nombre reste de l'ordre de K , alors le coût total de l'algorithme est en $O(DKN)$.

OMP En plus de MP, OMP doit à chaque étape calculer une projection orthogonale. Pati propose une formule de récurrence pour calculer $(\Phi^{(i)*} \Phi^{(i)})^{-1}$ en fonction de la matrice précédente $(\Phi^{(i-1)*} \Phi^{(i-1)})^{-1}$ [PRRK93]. L'opération la plus coûteuse en $O(iN)$ est le calcul de la dernière ligne de la matrice de Gram $\Phi^{(i-1)*} \varphi^{(i)}$. Il faut ensuite appliquer une fois l'inverse précédent pour un coût en $O(i^2)$. Le coût total pour l'ensemble de l'algorithme est en $O(K^2 N)$, soit ce que coûterait le calcul direct de la matrice de Gram finale $\Phi^{(K)*} \Phi^{(K)}$.

Il peut sembler paradoxal que la matrice de Gram coûte plus cher à calculer qu'à inverser. C'est dû au fait que $K \ll N$. La matrice de Gram est de petite taille $K \times K$ mais son calcul implique des produits scalaires entre atomes de grande taille N .

Comme on suppose $K < N < D$, le coût principal du calcul reste le coût de calcul des corrélations en $O(DKN)$. Ces résultats sont résumés dans le tableau 1.1.

Dans ce cas, MP et OMP restent dans la même classe de complexité. Bien sûr, OMP nécessite plus d'opérations, mais leur coût ne dépasse pas le coût commun du calcul de $\Phi^* r$.

1.2.4 Dictionnaires rapides

Il existe des bases qui permettent une analyse (application de Φ^*) et une synthèse (application de Φ) rapides. Par exemple, sur une base de Fourier discrète, de taille N , ces deux opérations peuvent être effectuées en $O(N \log N)$ par transformée de Fourier rapide (FFT) : la FFT est l'opérateur d'analyse et la FFT inverse l'opérateur de synthèse.

Supposons que le dictionnaire redondant Φ soit une union de bases permettant un calcul en $O(N \log N)$. Soit α le nombre de bases. On a alors $D = \alpha N$ et le coût des opérateurs d'analyse et de synthèse est en $O(\alpha N \log N) = O(D \log N)$. On peut alors calculer plus rapidement les corrélations, la matrice de Gram et synthétiser le résidu à partir des coefficients.

Le coût de l'inversion de la matrice de Gram n'est alors plus négligeable pour OMP si $K > \sqrt{D \log N}$. Ces résultats sont résumés dans le tableau 1.2.

TAB. 1.2 – Coût des différentes opération utilisées par les algorithmes gloutons avec un dictionnaire rapide

Algorithmme	Seuillage	MP	OMP	GP
$\lambda = \Phi^* r$	$D \log N$	$D \log N$	$D \log N$	$D \log N$
$\operatorname{argmax} \lambda $	$D \log D$	D	D	D
$G = \Phi^{(i)*} \Phi^{(i)}$	$K \log N$	0	$i \log N$	$i \log N$
$\chi = G^{-1} \lambda^{(i)}$	K^2	0	i^2	i
$r^{(i)} = r^{(i-1)} - \Phi^{(i)} \chi$	$D \log N$	N	$D \log N$	$D \log N$
Terme dominant pour 1 itération	$D \log D + K^2$	$D \log N$	$D \log N + i^2$	$D \log N$
Nombre d'itérations	1	$\gtrsim K$	K	K
TOTAL	$D \log D + K^2$	$DK \log N$	$DK \log N + K^3$	$DK \log N$

1.2.5 Gradient Pursuit (GP)

GP est un algorithme qui remplace l'orthogonalisation exacte d'OMP par une étape de descente de gradient, plus rapide [BD08]. Les nouveaux coefficients sont donnés par l'expression

$$x^{(i)} = x^{(i-1)} - \rho^{(i)} \nabla x^{(i)} \quad (1.9)$$

avec $\nabla x^{(i)}$ le gradient par rapport aux coefficients qui s'exprime

$$\nabla x^{(i)} = -2\Phi^{(i)*} r^{(i-1)} \quad (1.10)$$

et $\rho^{(i)}$ le pas optimal de cette descente défini par

$$\rho^{(i)} = \operatorname{argmin}_{\rho \in \mathbf{R}} \left\| r^{(i-1)} - \rho \Phi^{(i)} \nabla x^{(i)} \right\|_2^2 \quad (1.11)$$

$$= - \frac{\left\| \Phi^{(i)*} r^{(i-1)} \right\|_2^2}{2 \left\| \Phi^{(i)} \Phi^{(i)*} r^{(i-1)} \right\|_2^2} \quad (1.12)$$

$$x^{(i)} = x^{(i-1)} + \frac{\left\| \Phi^{(i)*} r^{(i-1)} \right\|_2^2}{\left\| \Phi^{(i)} \Phi^{(i)*} r^{(i-1)} \right\|_2^2} \Phi^{(i)*} r^{(i-1)} \quad (1.13)$$

Complexité Les coefficients du gradient sont des corrélations déjà calculées. Le calcul du dénominateur du pas nécessite une synthèse en $O(D \log N)$. Ensuite l'ajout aux coefficients se fait en $O(i)$, qui reste négligeable par rapport aux autres coûts.

Ces opérations remplacent le calcul de la matrice de Gram et son inversion pour OMP. On a gardé la composante en $O(D \log N)$ mais diminué le coût de l'inversion de $O(i^2)$ à $O(i)$.

L'autre intérêt pratique de GP est d'éviter d'avoir à formuler la matrice de Gram. Si on dispose de dictionnaires rapides, on ne manipule plus les opérateurs Φ et Φ^* sous forme matricielle. Si on supprime la matrice de Gram, alors il ne reste plus aucun calcul matriciel dans l'algorithme. C'est un avantage qu'on ne voit pas tant qu'on code en MATLAB, mais cela permet d'éviter d'avoir à faire appel à une librairie matricielle extérieure pour une implémentation dans un langage de bas niveau comme C ou C++.

1.3 Implémentation rapide de MP pour les dictionnaires invariants par translation

Un dictionnaire invariant par translation est un dictionnaire composé de motifs de taille $L \ll N$ translétés à plusieurs endroits du signal. De tels dictionnaires sont couramment utilisés en analyse temps-fréquence, la transformée de Fourier à court terme par exemple.

Dans ce cas, un produit scalaire se calcule en $O(L)$. On peut même disposer de dictionnaires rapides qui calculent l'analyse et la synthèse d'une base de L atomes en $O(L \log L)$. Pour l'ensemble du dictionnaire, l'analyse/synthèse coûte de l'ordre de DL avec un dictionnaire lent et $D \log L$ avec un dictionnaire rapide.

De plus, pour MP, on remarque qu'entre deux itérations le résidu ne change que sur un support de taille L . Notons $\text{supp}(\varphi)$ le support d'un atome, l'intervalle de longueur L sur lequel il est non-nul. Alors

$$\forall t \notin \text{supp } \varphi^{(i)}, r^{(i-1)} = r^{(i)} \quad (1.14)$$

$$\Rightarrow \forall \varphi \in \Phi / \text{supp } \varphi \cap \text{supp } \varphi^{(i)} = \emptyset, \varphi^* r^{(i)} = \varphi^* r^{(i-1)} \quad (1.15)$$

Donc, quand on recalcule les corrélations à l'itération $i + 1$, on n'a besoin que de recalculer celles des atomes dont le support chevauche celui de l'atome précédent $\varphi^{(i)}$. Si les atomes de Φ sont répartis uniformément dans le temps, il n'y en a que de l'ordre de $D \frac{L}{N}$.

La recherche de la corrélation la plus forte peut également être accélérée : si 2 corrélations n'ont pas été modifiées entre les itérations, alors leur comparaison non plus. On peut organiser les comparaisons en un arbre de tournoi (arbre dans lequel chaque père est le plus grand de ses fils) pour trouver le meilleur atome en $O(\log D)$.

Ces accélérations ont été intégrées dans la librairie MPTK ¹ [KG06].

1.4 Conditions d'optimalité

Pour justifier l'emploi d'algorithmes sous-optimaux, il faut avoir une idée de l'erreur commise. Dans le cadre de la parcimonie, on s'est plutôt demandé dans quels cas particuliers un algorithme généralement sous-optimal calcule bien le meilleur support possible s'il n'y a pas de bruit ?

On sait que le problème de la décomposition parcimonieuse est trivial si le dictionnaire est une base orthonormale : dans ce cas un simple seuillage dur suffit à identifier le meilleur support, et les autres algorithmes donnent également le bon résultat.

Les zones d'optimalité connues sont basées sur une mesure de similarité du dictionnaire avec une base orthonormale.

Tropp a fourni des résultats pour les algorithmes gloutons basés sur la cohérence, i.e. le plus grand produit scalaire entre 2 atomes du dictionnaire [Tro04]. Son résultat est décomposé en 2 lemmes.

Tout d'abord, si le sous-dictionnaire optimal $\hat{\Phi}^{(K)}$ vérifie

$$\max_{\varphi \in \Phi - \hat{\Phi}^{(K)}} \left\| \hat{\Phi}^{(K)+} \varphi \right\|_1 < 1 \quad (1.16)$$

alors MP, OMP et BP convergent vers une solution de support inclus dans $\hat{\Phi}^{(K)}$. On peut retrouver les atomes du support optimal si ceux-ci sont faiblement corrélés avec les atomes du support optimal.

¹<http://mptk.irisa.fr>

Cette propriété dépend du support. Pour l'étendre aux dictionnaires quel que soit le support choisi, on définit ensuite la cohérence du dictionnaire

$$\mu(\Phi) = \operatorname{argmax}_{(i,j) \in [1,D]^2, i \neq j} |\langle \varphi^{(i)}, \varphi^{(j)} \rangle| \quad (1.17)$$

La cohérence est comprise entre 0 et 1 et vaut 0 pour une base orthonormale. Tropp montre que si la taille K du support vérifie

$$K < \frac{1}{2} \left(1 + \frac{1}{\mu(\Phi)} \right) \quad (1.18)$$

alors la condition 1.16 est vérifiée. *Plus le dictionnaire est incohérent, plus on peut retrouver des supports longs.*

Cette étude s'applique aussi bien à MP qu'à OMP. On remarque empiriquement qu'OMP calcule de meilleures approximations que MP, mais on n'en a pas la preuve.

1.4.1 Conclusion

Les algorithmes de décomposition parcimonieuse les plus rapides sont les algorithmes gloutons. Parmi eux, les 2 plus connus sont MP qui est un des plus rapides et OMP qui calcule de meilleures approximations. D'autres algorithmes comme GP s'insèrent entre les deux, essayant d'obtenir un comportement proche d'OMP pour un coût proche de MP.

Si on travaille avec des dictionnaires locaux, on peut implémenter MP très efficacement. Peut-on faire de même pour OMP ou d'autres algorithmes ?

1.5 Implémentation « rapide » d'OMP sur des dictionnaires locaux

Une première tentative consiste à tenter d'utiliser l'hypothèse de localité pour accélérer OMP sans apporter d'approximation, comme cela a été fait pour MPTK. Cette section détaille plusieurs accélérations qui peuvent être apportées sans changer le comportement de l'algorithme. Ces accélérations sont nettement insuffisantes et il reste un écart important avec MP. Néanmoins elles permettent d'indiquer l'étape coûteuse de l'algorithme et d'orienter la conception d'un algorithme approché plus rapide.

1.5.1 Calcul des corrélations

Comme pour MP, l'hypothèse de localité réduit le coût de calcul d'un produit scalaire à $O(L)$. Contrairement à MP, le changement du résidu $r^{(i)} - r^{(i-1)}$ n'est pas local. Comme détaillé dans l'algorithme 2, à l'itération $i - 1$ la variation des coefficients $\chi^{(i-1)} = x^{(i-1)} - x^{(i-2)}$ vaut

$$\chi^{(i-1)} = \left(\Phi^{(i-1)*} \Phi^{(i-1)} \right)^{-1} \Phi^{(i-1)*} r^{(i-2)} \quad (1.19)$$

Certains coefficients de $\chi^{(i-1)}$ peuvent être nuls. Tout d'abord, $r^{(i-2)}$ est par définition orthogonal à $\Phi^{(i-2)}$ donc $\Phi^{(i-1)*} r^{(i-2)} = \langle \varphi^{(i-1)}, r^{(i-2)} \rangle \delta_{\varphi, \varphi^{(i-1)}}$ avec δ le symbole de Kronecker qui vaut 1 si $\varphi = \varphi^{(i-1)}$, 0 sinon. Ensuite, si la matrice de Gram est de forme

$$\Phi^{(i-1)*} \Phi^{(i-1)} = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} \quad (1.20)$$

alors son inverse vaut

$$\left(\Phi^{(i-1)*}\Phi^{(i-1)}\right)^{-1} = \begin{pmatrix} A^{-1} & 0 \\ 0 & B^{-1} \end{pmatrix} \quad (1.21)$$

Dans ce cas, seul les coefficients $\chi^{(i-1)}$ du bloc contenant le dernier atome $\varphi^{(i-1)}$ peuvent être non-nuls.

Or, à cause de l'hypothèse de localité, la matrice de Gram peut posséder cette structure par blocs. Deux atomes dont les supports ne se chevauchent pas (i.e. dont les supports ont une intersection vide) ont un produit scalaire nul. Donc les composantes connexes du graphe dont les sommets sont les atomes et les arêtes la relation de chevauchement sont des blocs de la matrice de Gram : 2 atomes dans 2 composantes différentes ont un produit scalaire nul. Donc seuls les coefficients des atomes de la composante connexe de $\varphi^{(i-1)}$ peuvent changer, et $r^{(i-1)}$ ne peut être différent de $r^{(i-2)}$ que sur le support temporel de cette composante connexe.

Néanmoins, si on sélectionne suffisamment d'atomes, il est possible que les supports des atomes sélectionnés finissent par couvrir entièrement le signal. Dans ce cas, il n'y a plus qu'une seule composante connexe et la mise à jour du résidu perd toute localité. Ce comportement sera même souvent le comportement désiré d'OMP en fin d'algorithme : si on conserve plusieurs composantes connexes jusqu'au bout, peut-être aurait-il mieux valu commencer par segmenter le signal au lieu de le traiter d'un seul bloc. Il faut donc recalculer tous les produits scalaires à chaque itération, au moins à la fin de l'algorithme. Le coût de calcul des produits scalaires est alors en $O(DL)$ ou $O(D \log L)$ avec un dictionnaire rapide.

1.5.2 Calcul de la matrice de Gram

Principe L'hypothèse de localité des dictionnaires permet d'accélérer le calcul de la matrice de Gram $G^{(i)} = \Phi^{(i)*}\Phi^{(i)}$. Tout d'abord, comme les produits scalaires avec le résidu, les produits scalaires entre atomes sont moins chers puisque calculés sur des vecteurs moins longs. Il y a également moins de produits scalaires à calculer. En effet, deux atomes dont les supports ne se chevauchent pas ont forcément un produit scalaire nul, donc la matrice de Gram est creuse et on sait localiser les coefficients nuls sans les calculer.

Pour cela, on a besoin d'un test rapide pour décider si 2 atomes se chevauchent. La relation de chevauchement s'exprime sur les bornes des supports :

$$\begin{cases} t_{\max}(\varphi) \geq t_{\min}(\varphi^{(i)}) \\ t_{\min}(\varphi) \leq t_{\max}(\varphi^{(i)}) \end{cases} \quad (1.22)$$

$$\Leftrightarrow \begin{cases} t_{\min}(\varphi) + L - 1 \geq t_{\min}(\varphi^{(i)}) \\ t_{\min}(\varphi) \leq t_{\min}(\varphi^{(i)}) + L - 1 \end{cases} \quad (1.23)$$

$$\Leftrightarrow t_{\min}(\varphi^{(i)}) - L < t_{\min}(\varphi) < t_{\min}(\varphi^{(i)}) + L \quad (1.24)$$

$t_{\min}(\varphi)$ doit donc se trouver dans l'intervalle

$$I = [\max\{t_{\min}(\varphi^{(i)}) - L + 1, 0\}, \min\{t_{\min}(\varphi^{(i)}) + L - 1, N - L\}]$$

Pour estimer le nombre d'atomes se trouvant dans l'intervalle I , Supposons que les $t_{\min}(\varphi^{(i)})$ sont indépendants et suivent une loi uniforme sur l'intervalle $[0, N - L]$ de tous les instants admissibles sans déborder du signal.

Pour tout atome débutant dans l'intervalle $J = [L - 1, N - 2L + 1]$, on a

$$I = [t_{\min}(\varphi^{(i)}) - L + 1, t_{\min}(\varphi^{(i)}) + L - 1] \quad (1.25)$$

I est de longueur $2L - 1$. D'après le modèle uniforme, la probabilité p pour un atome de se trouver dans J est égale au rapport la longueur de J et le nombre d'instants de début possibles :

$$p = \frac{N - 3L + 3}{N - L + 1} = 1 - \frac{2L - 2}{N - L + 1}$$

Comme $L \ll N$, p est proche de 1. On peut donc négliger les effets de bord et considérer la formule (1.25) comme vraie partout. Toujours sous l'hypothèse uniforme, l'espérance $E^{(i)}$ du nombre d'atomes à sélectionner à chaque itération vaut donc

$$E^{(i)} = \frac{(i - 1)(2L - 1)}{N - L + 1} \quad (1.26)$$

Implémentation naïve Une implémentation naïve consisterait à parcourir $\Phi^{(i-1)}$ et à ne calculer le produit scalaire que si la contrainte (1.24) est vérifiée. Le coût serait alors en

$$O\left(|\Phi^{(i-1)}| + LE^{(i)}\right) = O\left((i - 1) + \frac{L(i - 1)(2L - 1)}{N - L + 1}\right) \quad (1.27)$$

$$= O\left(i + \frac{iL^2}{N}\right) \quad (1.28)$$

Ce coût comprend le temps de parcours en $O(i)$ et le temps de calcul pour les atomes pertinents. Il représente une réduction d'un facteur $\left(\frac{L}{N}\right)^2$ par rapport au cas général en $O(iN)$ mentionné dans la table 1.1. Le premier facteur $\frac{L}{N}$ vient du fait qu'on a moins de produits scalaires à calculer, le second de la baisse de coût du calcul d'un produit scalaire entre atomes. Néanmoins, on doit encore passer en revue tous les atomes de $\Phi^{(i)}$. Le simple coût de ce parcours est en $O(i)$. L'hypothèse de localité seule ne permet pas de négliger ce temps de parcours dans la complexité (1.28).

Recherche rapide de la composante connexe On peut se débarrasser du parcours de tous les atomes de $\Phi^{(i-1)}$ en utilisant une structure triée pour $\Phi^{(i)}$. En effet, si les atomes sont triés par t_{\min} croissants, il suffit alors de rechercher le premier et le dernier atome pour lesquels il faut calculer les produits scalaires. De plus, $\Phi^{(i)}$ est un objet dynamique qui grandit à chaque itération. On a donc besoin d'une structure de données capable de réaliser rapidement :

- l'insertion d'un nouvel élément,
- l'extraction de tous les atomes dont le support appartient à un intervalle,
- le parcours de la structure entre les 2 bornes trouvées

Tous ces critères sont remplis par les structures d'arbres binaires de recherche quasi-équilibrés, par exemple les arbres rouge-noir utilisés dans les implémentations actuelles d'ensembles triés par les bibliothèques standard Java ² et C++ ³. Dans ce cas, l'insertion et la recherche d'un élément se font en $O(\log i)$ et le parcours d'un ensemble à Q éléments se fait en $O(Q)$. La complexité du calcul de la matrice de Gram devient alors

$$O\left(\log i + \frac{iL^2}{N}\right) \quad (1.29)$$

²<http://java.sun.com/javase/6/docs/api/java/util/TreeSet.html>

³<http://gcc.gnu.org/onlinedocs/libstdc++/libstdc++-html-USERS-4.4/a00528.html>

Double indexation D'autre part, $G^{(i)}$ est une matrice dont la taille augmente à chaque itération. Il est plus simple de toujours ajouter la nouvelle ligne et la nouvelle colonne à la fin de la matrice, ce qui évite d'avoir à décaler les indices des autres éléments. Cela signifie que dans $G^{(i)}$, les atomes sont représentés par i croissants. On doit donc maintenir simultanément les 2 indices triés. Heureusement, le tri par i croissants peut être effectué plus simplement : puisque le dernier atome sélectionné est forcément le plus grand, il suffit de l'insérer à la fin d'un tableau.

Dictionnaires rapides Les dictionnaires rapides et locaux, comme par exemple les dictionnaires de Fourier à court terme, se combinent également bien avec ce calcul car les ensembles d'atomes qui peuvent être traités simultanément en $O(L \log L)$ sont locaux : les atomes traités ensemble ont le même support. Donc le calcul de la matrice de Gram par ce moyen ne nécessite pas le calcul des produits scalaires du dernier atome avec tous ces ensembles, ce qui prendrait de l'ordre de $O(D \log L)$ mais juste des quelques situés autour de l'atome $\varphi^{(i)}$, car seules ces bases peuvent contenir des atomes corrélés avec $\varphi^{(i)}$. Le calcul se fait donc en $O(L \log L)$. Cette implémentation n'est cependant pas tout le temps intéressante parce qu'on calcule ainsi plus de produits scalaires que nécessaire. Quand on utilise l'analyse rapide, on calcule les produits scalaires avec toute un ensemble d'atomes de Φ d'un coup, alors qu'on n'a besoin que des quelques-uns qui appartiennent à $\Phi^{(i-1)}$. Pour que l'analyse rapide soit plus efficace, on doit avoir

$$L \log L < \frac{iL^2}{N} \quad (1.30)$$

$$i > N \frac{\log L}{L} \quad (1.31)$$

Inversion de la matrice de Gram Comme nous l'avons vu, l'hypothèse de localité implique que $G^{(i)}$ est creuse. Le nombre d'atomes chevauchant un atome φ_m , dont l'espérance est donnée par l'expression (1.26), est aussi une borne supérieure du nombre de coefficients non-nuls de la $m^{\text{ème}}$ colonne de $G^{(i-1)}$. Le nombre moyen de coefficients non-nuls dans $G^{(i)}$ est de l'ordre de $iE^{(i)} = \frac{i^2(2L-1)}{N-L}$. La complexité de l'inversion de $G^{(i)}$, par exemple par la méthode Pati [PRRK93], étant de l'ordre du nombre de coefficients non-nuls, la complexité de l'inversion est en $O\left(\frac{i^2L}{N}\right)$

Au total, le coût d'un itération d'OMP est donc en

$$O\left(D \log L + \frac{L \log L}{N} \frac{i^2L}{N}\right) \quad (1.32)$$

$$= O\left(N \log L \left(\alpha + \frac{L}{N} + \frac{i^2L}{N^2 \log L}\right)\right) \quad (1.33)$$

On a choisi l'implémentation en $O(L \log L)$ pour le calcul de la matrice de Gram afin de simplifier la comparaison avec les autres coûts. L'hypothèse de localité suppose $L \ll N$ donc le coût de calcul de la matrice de Gram est négligeable devant le coût de calcul des produits scalaires. Le coût de l'inversion ne devient significatif que pour

$$i \gtrsim N \sqrt{\frac{\alpha \log L}{L}} \quad (1.34)$$

Cette inégalité est indéterminée sans hypothèse supplémentaire sur N et L . Numériquement, si on prend par exemple $N \approx 10^6$, $\alpha = 2$ et $L = 1024$, le coût de l'inversion

deviendrait significatif à partir de $i \gtrsim 140000$, ce qui semble proche de N pour une décomposition parcimonieuse. Donc le terme dominant de cette complexité est le calcul des produits scalaires en $O(D \log L)$, avec éventuellement une contribution de l'inversion pour de grandes itérations.

1.5.2.1 Bilan

Dans le cas général, la seule différence entre MP et OMP se situait au niveau du calcul du nouveau résidu. L'hypothèse de localité a réduit le coût de cette étape. Néanmoins, comme elle permet également de trouver le meilleur atome plus rapidement pour MP, on trouve encore un écart de coût important entre MP et OMP. Cet écart a changé de place, de la projection du résidu au calcul des corrélations..

Cette observation invalide toutes les approches précédentes pour combler le vide entre MP et OMP, comme GP. Ces approches se sont concentrées sur la réduction du coût de la projection. Or même si on arrivait à calculer $\Phi^{(i)+r^{(i-1)}}$ instantanément, OMP resterait beaucoup plus coûteux que MP, de l'ordre de $\frac{N}{L}$ fois plus coûteux.

Il faut donc chercher un moyen de réduire le coût du calcul des produits scalaires. C'est ce à quoi s'attache l'algorithme LocOMP.

1.6 LocOMP

1.6.1 Description générale.

L'écart de complexité entre MP et OMP vient du calcul de Φ^*r . Le gros gain de MP à cette étape est dû au fait que le résidu ne change que sur un intervalle de longueur L à chaque itération. Le coût d'une itération étant étroitement lié à la longueur de l'intervalle de modification du résidu, c'est cette longueur qu'il faut contrôler pour arriver à passer à l'échelle.

L'idée centrale de l'algorithme LocOMP est de maîtriser l'intervalle de mise à jour du résidu en limitant l'espace de projection du résidu à un sous-ensemble d'atomes proches dans le domaine temporel. L'algorithme 3 décrit le fonctionnement général simplifié.

On définit une fonction *neighbour* qui calcule le *sous-dictionnaire* $\Psi^{(i)} \subset \Phi^{(i)}$ sur lequel on va effectuer cette projection. Bien sûr, $\Psi^{(i)}$ contient le dernier atome sélectionné $\varphi^{(i)}$. Le choix de la fonction *neighbour* définit une progression entre MP et OMP. MP correspond au choix $\Psi^{(i)} = \varphi^{(i)}$ et OMP au choix $\Psi^{(i)} = \Phi^{(i)}$.

$$\varphi^{(i)} \in \Psi^{(i)} \subseteq \Phi^{(i)} \quad (1.35)$$

1.6.2 Complexité

Comme pour MPTK, à la première itération le produit Φ^*s doit être calculé exhaustivement pour coût de $O(D \log L)$ avec un dictionnaire rapide. Aux itérations suivantes, si on note $T \geq L$ la longueur de l'intervalle sur lequel le résidu a changé, il faut calculer le produit scalaire de $r^{(i-1)}$ avec $\alpha(T + 2L - 2)$ atomes, ce qui se fait en $O(\alpha T \log L)$. La recherche du meilleur atome $\varphi^{(i)}$ est effectuée comme dans MPTK et le calcul de la matrice de Gram $\Phi^{(i)*}\Phi^{(i)}$ comme pour OMP. On en tire ensuite la sous-matrice $\Psi^{(i)*}\Psi^{(i)}$ qui contient environ $i\frac{T}{N}$ atomes. La méthode d'inversion de Pati pour OMP [PRRK93] n'est pas applicable ici parce que les atomes choisis changent à chaque étape. Néanmoins, on peut toujours réaliser une inversion de complexité quadratique par une descente de gradient

Algorithm 3 $x = \text{LocOMP}(s, \Phi)$

```

 $r_0 = s$ 
 $\Phi_0 = \emptyset$ 
 $x_0 = 0$ 
for  $i = 1$  to  $I$  do
   $\varphi^{(i)} = \text{argmax}_{\varphi \in \Phi} |\langle r^{(i-1)}, \varphi \rangle|$  {recherche du meilleur atome}
   $\Phi^{(i)} = \Phi^{(i-1)} \cup \varphi^{(i)}$ 
   $\Psi^{(i)} = \text{neighbour}(\Phi^{(i)}, \varphi^{(i)})$  {sélection du sous-dictionnaire}
   $\chi^{(i)} = (\Psi^{(i)*}\Psi^{(i)})^{-1} \Psi^{(i)*}r^{(i-1)}$  {calcul des coefficients de la projection}
   $x^{(i)} = x^{(i-1)} + \chi^{(i)}$  {mise à jour des coefficients}
   $r^{(i)} = r^{(i-1)} - \Psi^{(i)}\chi^{(i)}$  {mise à jour du résidu}
end for
return  $x^{(i)}$ 

```

TAB. 1.3 – Complexité des différentes étapes pour MP, OMP et LocOMP.

Étape	MPTK	OMP	LocOMP
$\lambda(1) = \Phi^*s$	$D \log L$	$D \log L$	$D \log L$
$m^{(1)} = \text{argmax} \lambda^{(1)} $	D	D	D
$\lambda^{(i)} = \Phi^*r^{i-1}$	$\alpha L \log L$	$D \log L$	$\alpha T \log L$
$m^{(i)} = \text{argmax} \lambda^i $	$\log(D)$	D	$\log(D)$
$\Phi^{(i)} = \Phi^{(i-1)} \cup \{\varphi^{(i)}\}$	1	$\log i$	$\log i$
$G^{(i)} = \Phi^{(i)*}\Phi^{(i)}$	0	$L \log L$	$L \log L$
$\Psi^{(i)} = \text{neighbour}(\Phi^{(i)}, \varphi^{(i)})$	0	0	$\log i$
$\chi^{(i)} = \Psi^{(i)+}r^{(i-1)}$	0	$\frac{i^2 L}{N}$	$\frac{i^2 L^2}{N^2}$
$r^{(i)} = r^{(i-1)} - \Psi^{(i)}\chi^{(i)}$	L	$D \log L$	$T \log L$
Total sur une itération	$\alpha L \log L$	$D \log L + \frac{i^2 L}{N}$	$T \log L$
TOTAL	$\alpha(N + KL) \log L$	$DK \log L + \frac{K^3 L}{N}$	$\alpha(N + TK) \log L$

conjugué. Le coût de l'inversion est alors en $O\left(i^2 \frac{T^2}{N^2}\right)$. Enfin la mise à jour du résidu peut se faire en $O(T \log L)$ avec un dictionnaire rapide.

Sur un faible nombre d'itérations, le coût dominant reste celui de la première itération en $O(\alpha NL)$, comme pour MPTK. Pour les itérations suivantes, la partie dominante reste le calcul des produits scalaires en $O(T \log L)$. Pour assurer que LocOMP reste dans la même classe de complexité que MP, T doit rester de l'ordre de L .

1.6.3 Choix du sous-dictionnaire

Le choix du sous-dictionnaire $\Psi^{(i)}$ décrit une évolution entre MP et OMP. Si on décide à chaque itération i de ne conserver que le dernier atome sélectionné $\varphi^{(i)}$, alors le comportement de l'algorithme est celui de MP. À l'opposé, si on conserve à chaque étape toute la collection $\Phi^{(i)}$ des atomes précédemment sélectionnés, alors LocOMP se comporte comme OMP. La table 1.3 montre que si on souhaite conserver une complexité de l'ordre de celle de MP, alors l'union des supports des atomes de $\Psi^{(i)}$ doit rester de l'ordre de L .

D'autre part, on a toujours besoin de calculer la matrice $G^{(i)}$ comme pour OMP, ce qui nécessite la connaissance de l'ensemble des atomes de $\Phi^{(i-1)}$ qui chevauchent le dernier

atome $\varphi^{(i)}$, ensemble noté $V^{\Phi^{(i-1)}}(\varphi^{(i)})$. A chaque itération on ne travaille que sur la matrice de Gram du sous-dictionnaire $\Psi^{(i)*}\Psi^{(i)}$. Si $\forall 1 \leq i \leq I, \Psi^{(i)} \supset V^{\Phi^{(i-1)}}(\varphi^{(i)})$, tous les coefficients non-nuls de $\Phi^{(i)*}\varphi^{(i)}$ sont nécessaires à l'itération i , et seront donc calculés à ce moment. On peut en déduire par récurrence que tous les autres coefficients de $\Psi^{(i)*}\Psi^{(i)}$ ont déjà été calculés à une itération précédente. Il est alors possible de maintenir en mémoire la matrice de Gram $\Phi^{(i)*}\Phi^{(i)}$ de la collection comme on le fait pour OMP, puis d'en extraire la sous-matrice $\Psi^{(i)*}\Psi^{(i)}$.

Si on ne sélectionne pas au moins tous ces atomes, la gestion de la matrice de Gram devient plus délicate. Tous les coefficients de la $i^{\text{ème}}$ colonne ne sont plus nécessaires à l'itération i , mais ils pourraient être utilisés lors d'une itération ultérieure. Il y a alors 2 possibilités : on peut calculer toute la $i^{\text{ème}}$ colonne à l'itération i ou bien calculer les coefficients quand ils sont nécessaires. La première méthode implique le risque d'effectuer des calculs de corrélations inutiles, la seconde nécessite de vérifier à chaque itération l'état de chaque coefficient de la sous-matrice $\Psi^{(i)*}\Psi^{(i)}$. Il est donc moins intéressant de réduire le sous-dictionnaire en-dessous de ce seuil.

Dans ce travail, le sous-dictionnaire choisi a donc été l'ensemble de tous les atomes de $\Phi^{(i)}$ dont le support chevauche celui de $\varphi^{(i)}$, comme montré dans la figure 1.1. L'intervalle maximum de modification du résidu est alors $[t_{\min}(\varphi^{(i)}) - L + 1, t_{\max}(\varphi^{(i)}) + L - 1]$, de longueur $T = 3L - 2$. Avec ce choix de sous-dictionnaire, LocOMP devrait être $\frac{5}{3}$ plus coûteux que MP. Néanmoins ces ordres de grandeur sont à confirmer par l'expérience pour vérifier qu'on n'a pas négligé une constante multiplicative trop importante.

1.7 Extension aux dictionnaires multi-échelles

Jusqu'à présent nous avons supposé que tous les atomes avaient un support de même longueur L . Néanmoins ce n'est pas toujours le cas. Par exemple, un usage possible de dictionnaires redondants est l'analyse temps-fréquence multi-échelles d'un signal. Dans ce cas, le dictionnaire est une union des bases temps-fréquence (Gabor ou DCT) avec différentes longueurs de fenêtre. Ce modèle est couramment utilisé en traitement de la musique, afin de disposer à la fois d'un échantillonnage temporel fin pour modéliser les transitoires et d'un échantillonnage fréquentiel fin pour les parties stationnaires. Ce changement a une influence sur le choix du sous-dictionnaire en incitant à sélectionner les atomes de courtes échelles même s'ils ne chevauchent pas le dernier atome.

Supposons donc que chaque atome φ_m a un support de longueur L_m . La condition de chevauchement (1.24) avec le dernier atome $\varphi^{(i)}$ se réécrit donc :

$$\begin{cases} t_{\max}(\varphi) \geq t_{\min}(\varphi^{(i)}) \\ t_{\min}(\varphi) \leq t_{\max}(\varphi^{(i)}) \end{cases} \quad (1.36)$$

$$\Leftrightarrow \begin{cases} t_{\min}(\varphi_m) + L_m - 1 \geq t_{\min}(\varphi^{(i)}) \\ t_{\min}(\varphi) \leq t_{\min}(\varphi^{(i)}) + L_{m^{(i)}} - 1 \end{cases} \quad (1.37)$$

$$\Leftrightarrow t_{\min}(\varphi^{(i)}) - L_m < t_{\min}(\varphi) < t_{\min}(\varphi^{(i)}) + L_{m^{(i)}} \quad (1.38)$$

Si on partitionne le dictionnaire Φ en classes d'atomes de même échelle, alors pour chaque échelle L_m l'intervalle de modification du résidu est $[t_{\min}(\varphi^{(i)}) - L_m + 1, t_{\max}(\varphi^{(i)}) + L_m - 1]$, de longueur $L_{m^{(i)}} + 2L_m - 2$. Les atomes de grande échelle ont un impact plus large que les atomes à courte échelle sur le coût de calcul. Nous allons voir qu'on peut alors sélectionner plus d'atomes à courte échelle sans impact négatif sur la complexité, ce qui ne peut que diminuer l'erreur après la projection.

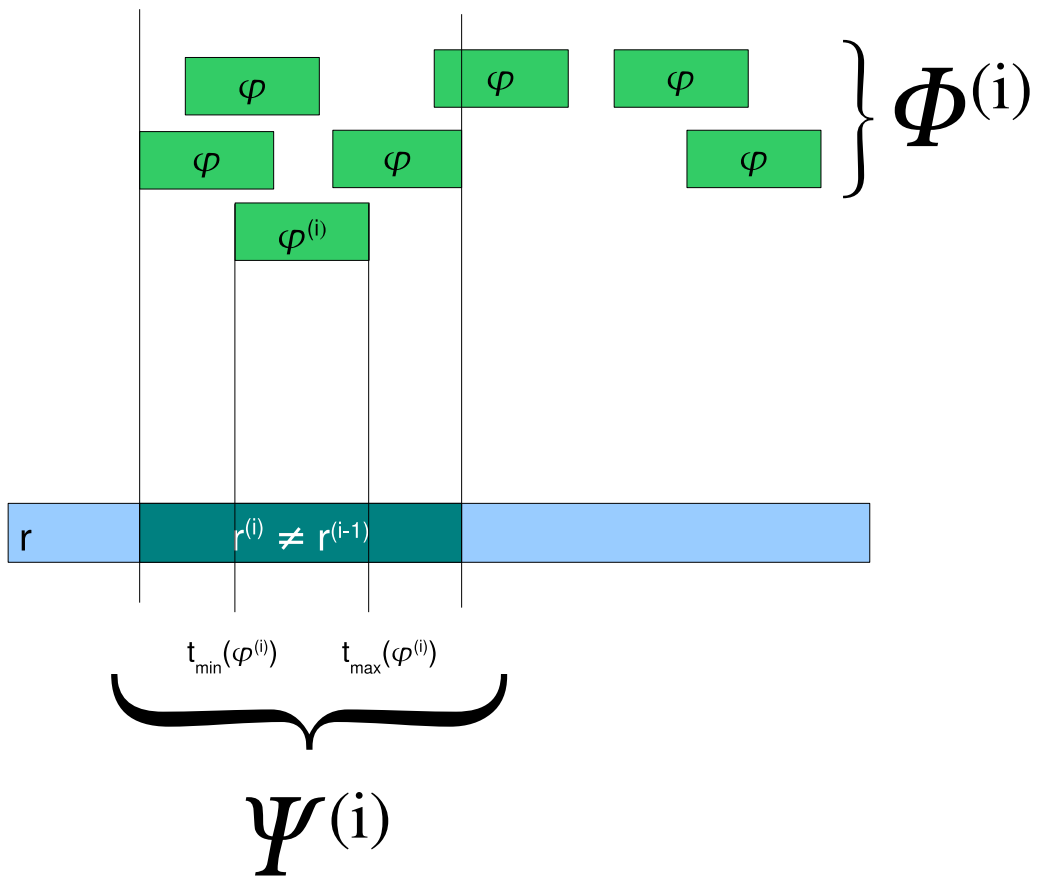


FIG. 1.1 – Sélection du sous-dictionnaire $\Psi^{(i)}$. On garde tous les atomes de $\Phi^{(i)}$ dont le support chevauche celui du dernier atome $\varphi^{(i)}$. Le résidu ne change au maximum que sur un intervalle de longueur inférieure à $3L - 2$.

Si on fixe les bornes de l'intervalle de modification du résidu $I = [t_{\min}^{\hat{}}, t_{\max}^{\hat{}}]$, le meilleur sous-dictionnaire possible est toujours le sous-dictionnaire exhaustif $\hat{\Psi}^{(i)}$ qui contient tous les atomes de $\Phi^{(i)}$ dont le support est inclus dans I . Tout autre sous-dictionnaire admissible Ψ est inclus dans $\hat{\Psi}^{(i)}$, donc l'espace engendré $\text{vect}(\Psi)$ est un sous-espace de $\text{vect}(\hat{\Psi}^{(i)})$. Si on note $\Pi_{\Psi}r$ la projection de r sur Ψ et $\Pi_{\Psi}^{\perp}r$ la projection de r orthogonalement à Ψ , on peut alors comparer les résidus obtenus :

$$\left\| \Pi_{\Psi}(r^{(i-1)}) \right\|_2 \leq \left\| \Pi_{\hat{\Psi}^{(i)}}(r^{(i-1)}) \right\|_2 \quad (1.39)$$

$$\left\| \Pi_{\Psi}^{\perp}(r^{(i-1)}) \right\|_2 \geq \left\| \Pi_{\hat{\Psi}^{(i)}}^{\perp}(r^{(i-1)}) \right\|_2 \quad (1.40)$$

Comme on l'a vu à la section 1.6.2, le coût de calcul de la projection est négligeable devant le coût de mise à jour des corrélations. Donc tout autre choix de sous-dictionnaire que $\hat{\Psi}^{(i)}$ fournit une approximation moins bonne pour un coût équivalent.

Pour sélectionner le plus petit intervalle qui permet de sélectionner tous les atomes de $\Phi^{(i)}$ corrélés avec $\varphi^{(i)}$, il faut :

- calculer la borne inférieure,

$$t_{\min}^{\hat{}} = \min\{t_{\min}(\varphi) \mid \varphi \in \Phi^{(i)} \wedge t_{\max}(\varphi) > t_{\min}(\varphi^{(i)})\}$$

- calculer la borne supérieure,

$$t_{\max}^{\hat{}} = \max\{t_{\max}(\varphi) \mid \varphi \in \Phi^{(i)} \wedge t_{\min}(\varphi) < t_{\max}(\varphi^{(i)})\}$$

- extraire le sous-dictionnaire $\Psi^{(i)}$ de la collection $\Phi^{(i)}$. $\Psi^{(i)}$ peut éventuellement contenir des atomes dont le support ne chevauche pas celui du dernier atome $\varphi^{(i)}$, comme le montre la figure 1.2.

Contrairement au cas mono-échelle, trier la collection ne suffit plus à extraire à la fois les bornes inférieure et supérieure. Cela fonctionnait quand tous les atomes étaient de longueur de support L car alors la collection était triée à la fois par t_{\min} croissants et par t_{\max} croissants. Ce n'est plus les cas ici, puisqu'un atome long peut commencer avant et terminer après un atome court.

En pratique, le dictionnaire comporte souvent peu d'échelles différentes. Il est par exemple courant d'utiliser les puissances de 2 successives, le nombre d'échelles est alors borné par $\log_2 N$ et souvent bien inférieur si on ne prend pas toutes les échelles. On peut alors maintenir un arbre de recherche par échelle, rechercher les bornes inférieures et supérieures dans chacun de ces arbres puis les comparer entre elles pour trouver les bornes globales. L'extraction du sous-dictionnaire est également effectuée séparément pour chaque échelle.

Pour un calcul plus simple, on peut également remplacer le calcul des bornes par la valeur fixe $I = [t_{\min}(\varphi^{(i)}) - L + 1, t_{\max}(\varphi^{(i)}) + L - 1]$. Dans le cas mono-échelle, ces deux opérations étaient équivalentes car tout atome inclus dans I chevauchait $\varphi^{(i)}$. Ce n'est plus le cas maintenant : un atome de courte échelle pourrait se situer dans l'espace entre l'intervalle minimal $[\hat{t}_{\min}, \hat{t}_{\max}]$ et l'intervalle fixe I . Un tel atome serait sélectionné par l'intervalle fixe mais pas par l'intervalle minimal.

1.8 Implémentation dans MPTK

L'implémentation d'une variante de l'algorithme LocOMP dans la librairie MPTK est en cours. Cependant il reste encore des problèmes de génie logiciel à résoudre pour obtenir un temps de calcul effectif aussi faible que la complexité théorique nous le laisse espérer.

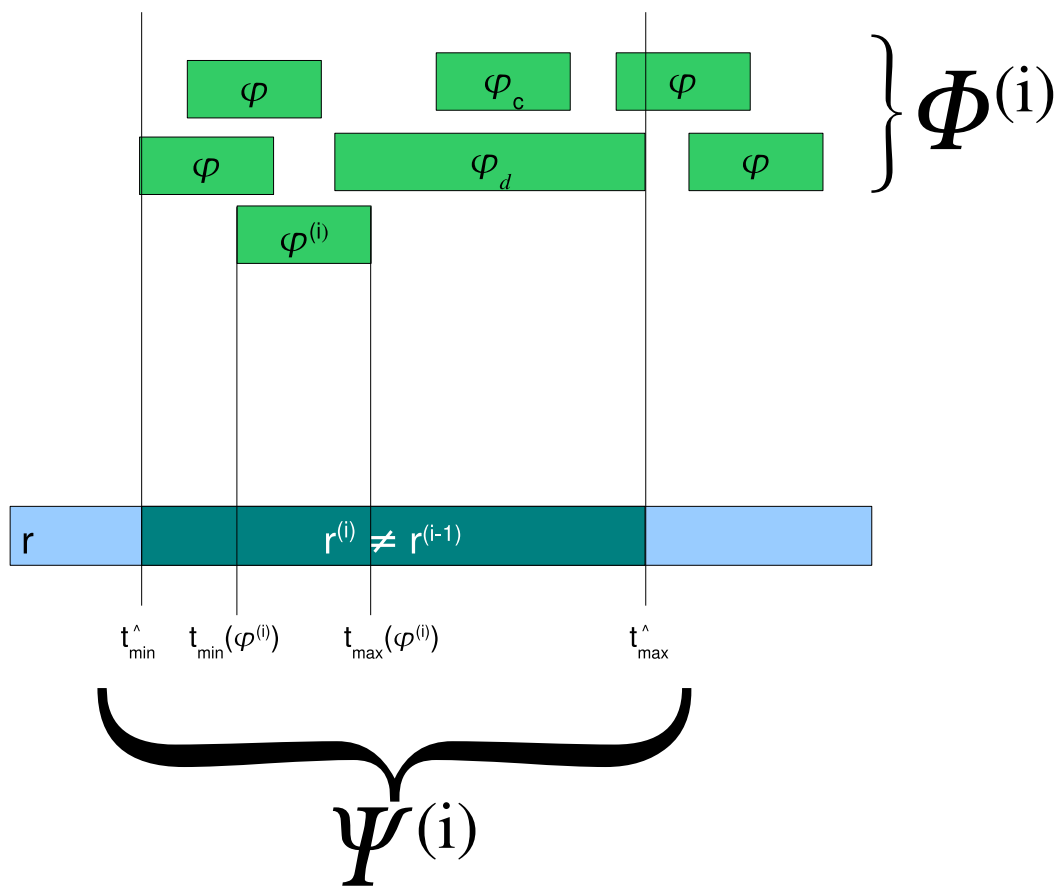


FIG. 1.2 – Sélection du sous-dictionnaire $\Psi^{(i)}$ dans le cas multi-échelles. L'atome court φ_c ne chevauche pas le dernier atome $\varphi^{(i)}$, mais son support est inclus dans celui de l'atome long φ_d . Il sera donc conservé.

1.8.1 Algorithme LocGP

L'algorithme choisi pour cette implémentation est LocGP, qui combine la sélection de sous-dictionnaire de LocOMP avec la mise à jour rapide du résidu de GP. Le fonctionnement est détaillé dans l'algorithme 4.

Algorithm 4 $x = \text{LocGP}(s, \Phi)$

```

 $r_0 = s$ 
 $\Phi_0 = \emptyset$ 
 $x_0 = 0$ 
for  $i = 1$  to  $I$  do
   $\varphi^{(i)} = \operatorname{argmax}_{\varphi \in \Phi} |\langle r^{(i-1)}, \varphi \rangle|$  {recherche du meilleur atome}
   $\Phi^{(i)} = \Phi^{(i-1)} \cup \varphi^{(i)}$ 
   $\Psi^{(i)} = \text{neighbour}(\Phi^{(i)}, \varphi^{(i)})$  {sélection du sous-dictionnaire}
   $\chi^{(i)} = \frac{\|\Psi^{(i)*} r^{(i-1)}\|^2}{\|\Psi^{(i)} \Psi^{(i)*} r^{(i-1)}\|^2} \Psi^{(i)*} r^{(i-1)}$  {calcul du gradient}
   $x^{(i)} = x^{(i-1)} + \chi^{(i)}$  {mise à jour des coefficients}
   $r^{(i)} = r^{(i-1)} - \Psi^{(i)} \chi^{(i)}$  {mise à jour du résidu}
end for
return  $x^{(i)}$ 

```

L'intérêt principal de GP pour cette application est la légèreté du code nécessaire à son implémentation en C++. En effet, le calcul exact de la projection nécessiterait :

- une structure de matrice creuse pour stocker $G^{(i)}$,
- un algorithme d'algèbre linéaire pour réaliser la projection.

Ces éléments nécessiteraient l'utilisation de bibliothèques extérieures, ce qui complique la procédure d'installation pour l'utilisateur. GP remplace tout ceci par un simple rapport d'énergies, bien plus simple à reprogrammer entièrement.

1.8.2 Accès rapide aux atomes sélectionnés

Nous avons vu dans la section 1.5.2 que la sélection de $\Psi^{(i)}$ pouvait être effectuée rapidement grâce à une structure d'ensemble trié. Néanmoins, l'implémentation dans MPTK pose des contraintes d'accès supplémentaires qui obligent à modifier légèrement cette structure.

L'algorithme LocGP nécessite la connaissance des corrélations $\Psi^{(i)*} r^{(i-1)}$. Ces corrélations ont été calculées précédemment lors de l'étape de sélection. Néanmoins, pour économiser la mémoire, elles ne sont pas toutes stockées par MPTK. Il faut donc modifier le calcul des corrélations pour détecter au vol les corrélations avec les atomes de $\Phi^{(i)}$ et les stocker à part.

Dans MPTK, le dictionnaire est subdivisé en *blocs*. Un bloc est un banc de filtres permettant le calcul rapide des corrélations du résidu avec tous les atomes du bloc. Par exemple, si on utilise un dictionnaire de STFT multi-échelles, on a un bloc par échelle, et chaque bloc d'échelle L utilise une FFT pour calculer les corrélations de tous les atomes du bloc à une position donnée en $O(L \log L)$. Pour chaque bloc, seule la corrélation la plus grande de chaque instant est conservée, puisqu'elle dominera toujours les autres jusqu'à ce qu'elles doivent toutes être mises à jour ensemble.

Or GP n'utilise pas les corrélations λ uniquement pour trouver le meilleur atome, mais également pour calculer le gradient. Il faut donc stocker en mémoire les corrélations des atomes de $\Phi^{(i)}$ au moment où elles sont calculées par les blocs. Il faut donc être capable d'extraire rapidement tous les atomes de $\Phi^{(i)}$ appartenant à un bloc et une position donnée pour savoir quelles corrélations de chaque trame doivent être sauvegardées. Pour cela, nous

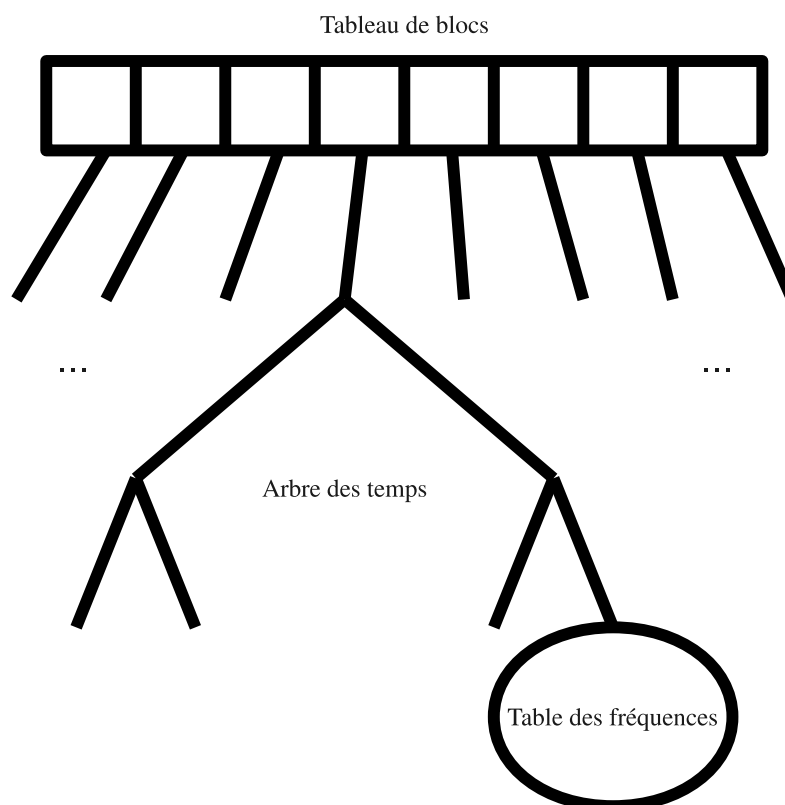


FIG. 1.3 – Structure de données pour l'accès rapide aux atomes. Les atomes de même bloc et t_{\min} forment un même tas, puis tous les tas d'un même bloc sont triés par t_{\min} croissants, puis tous ces ensembles triés sont rangés par bloc.

avons choisi une structure hiérarchique détaillée dans la figure 1.3. Les atomes sont d'abord triés par bloc, puis par t_{\min} , puis par autres paramètres d'identification (la fréquence pour le cas d'atomes STFT par exemple).

Le choix du bloc comme première subdivision entraîne un ralentissement lors de l'extraction du $\Psi^{(i)}$. S'il y a B blocs, l'extraction se fait en $O(B \log i)$ au lieu de $O(\log i)$ parce qu'il faut effectuer l'opération dans chaque arbre. Néanmoins, le nombre de blocs est en général faible car lié au nombre d'échelles. Par exemple, il suffit de 8 échelles de MDCT pour couvrir toute l'étendue utilisée en compression audio. Les temps d'insertion et de parcours, eux, ne changent pas.

1.9 Application aux signaux multicanaux

Un signal multicanal est obtenu par plusieurs observations simultanées de mêmes phénomènes. Mathématiquement, au lieu d'être un vecteur s , le signal est une matrice S dont chaque colonne est un canal. Supposons que le phénomène original observé q soit parcimonieux sur un dictionnaire Φ :

$$q = \Phi x + r \quad (1.41)$$

Supposons de plus que le processus d'acquisition de chaque canal soit linéaire : sur chaque canal c , il existe un opérateur linéaire A_c tel que

$$S_c = A_c q \quad (1.42)$$

$$= A_c \Phi x + A_c r \quad (1.43)$$

Si l'opérateur A_c n'amplifie pas trop le résidu r par rapport à la décomposition ΦX , on a $\|A_c \Phi x\|_2 \ll \|A_c r\|_2$ donc S_c est parcimonieux sur le dictionnaire $A_c \Phi$. Par contre, si l'opérateur A_c n'est pas orthogonal, rien n'assure que la décomposition x est optimale pour le signal S_c . Elle n'a même pas forcément le meilleur support. Si on souhaite identifier le support original de x d'après l'observation des canaux S , on cherche les atomes φ qui sont présents dans plusieurs canaux. Si on concatène les coefficients x_c sur les différents canaux en une matrice X , le critère de *parcimonie jointe* à minimiser est alors le nombre de lignes non-nulles de X . Formellement, on choisit une *norme intercanale* quelconque $\|\cdot\|_{interc}$, on calcule le vecteur

$$\|X\|_{interc} = \begin{pmatrix} \|x_{1,:}\|_{interc} \\ \vdots \\ \|x_{\alpha N,:}\|_{interc} \end{pmatrix} \quad (1.44)$$

et la contrainte de parcimonie qu'on cherche à minimiser est $\| \|X\|_{interc} \|_0$. Pour simplifier les calculs, on définit les dictionnaires renormalisés

$$\forall 1 \leq c \leq C, \forall 1 \leq d \leq D \varphi_{c,d} = \frac{A_c \varphi_d}{\|A_c \varphi_d\|_2} \quad (1.45)$$

Si on prend comme mesure de l'erreur l'énergie $\sum_{c=1}^C \|S_c - \Phi_c X_c\|_2^2$, alors le score de corrélation d'un atome φ_d avec le résidu R vaut $\sum_{c=1}^C (R_c^* \varphi_{c,d})^2$. On peut donc généraliser tous les algorithmes gloutons, y compris LocOMP.

1.10 Résultats expérimentaux

Le but de ces expériences est de valider les expressions théoriques de la complexité de LocOMP et de mesurer le gain en qualité d'approximation par rapport à MP. Nous avons travaillé sur des signaux réels, aussi la meilleure décomposition est inconnue.

1.10.1 En MATLAB

Cette expérience a été menée avec du code MATLAB expérimental. Le but est de pouvoir comparer LocOMP à la fois à MP et à OMP.

Protocole La forte complexité d'OMP limite la taille du signal et du dictionnaire utilisables. Nous avons choisi un extrait d'une minute de musique extrait de la base RWC [Mas04]. Cet extrait a été sous-échantillonné à 8000Hz pour le ramener à une longueur $N = 480000$ échantillons. Le dictionnaire est un dictionnaire MDCT de fenêtre $L = 32$ à chaque échantillon, donc il contient de l'ordre de $D = 15.10^6$ atomes.

Nous avons exécuté MP, OMP, GP, LocOMP et LocGP pendant $I = 20000$ itérations chacun.

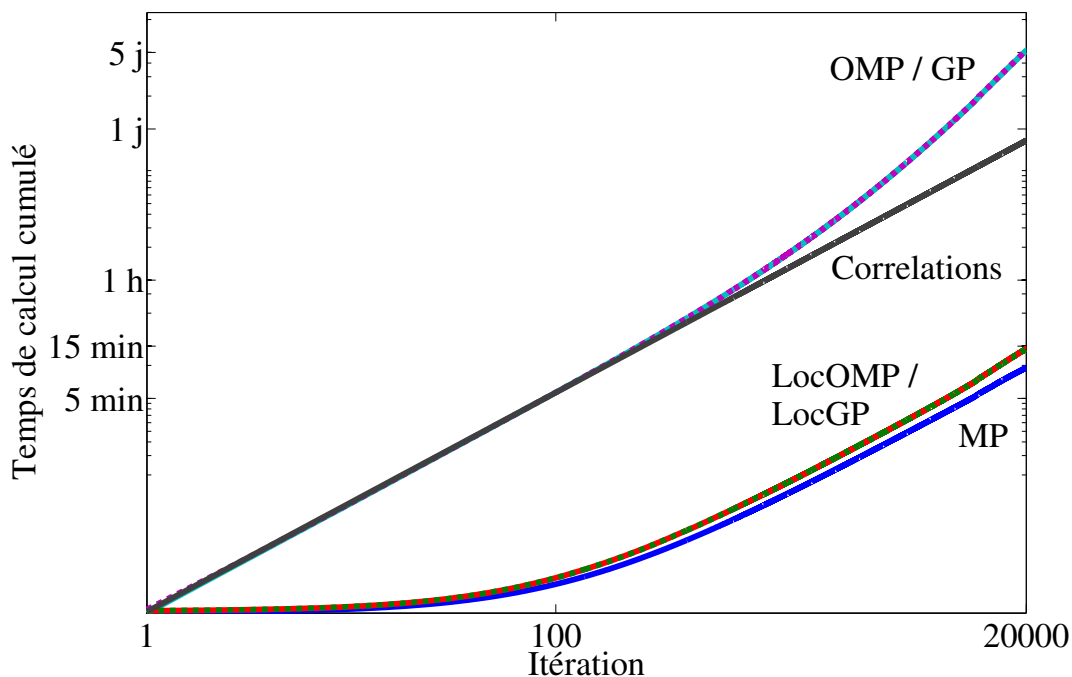


FIG. 1.4 – Temps de calcul nécessaire à différents algorithmes en fonction du nombre d'itérations total. Les algorithmes locaux (MP, LocOMP et LocGP) sont nettement plus rapides que les algorithmes globaux (OMP et GP). LocOMP et LocGP sont indissociables sur cette courbe, de même qu'OMP et GP. La tangente en pointillés est la limite inférieure du coût d'un algorithme global. Elle correspond au coût de calcul des corrélations $\lambda = \Phi^* r$ à chaque itération.

Temps de calcul Le graphe 1.4 trace le temps de calcul en fonction du nombre d'itérations pour chaque algorithme. On constate bien 2 types de comportement : OMP et GP d'un côté, MP, LocOMP et LocGP de l'autre. On a dû graduer en échelle log – log pour pouvoir représenter toutes les courbes sur le même graphe.

Les algorithmes globaux (OMP et GP) ont mis de l'ordre de 5 jours pour terminer le calcul alors que MP ne met que 10 minutes et LocOMP/LocGP 15 minutes. On a bien réussi à maintenir LocOMP dans le même ordre de coût que MP.

Pour les algorithmes locaux, la tangente à l'origine est horizontale parce que la première itération est beaucoup plus chère que les autres. Donc effectuer quelques itérations ou une seule prend le même temps.

Pour les algorithmes globaux, la tangente à l'origine traduit un comportement linéaire. C'est dû au fait que l'essentiel du coût de calcul est pris par le calcul des corrélations λ qui doit être effectué à l'identique à chaque itération. Quand i augmente, le coût de la projection prend de l'importance et la courbe s'écarte de la tangente. Il est important de noter que cette tangente constitue une limite inférieure du coût des algorithmes globaux : même si on arrivait à calculer la projection instantanément, on ne pourrait pas descendre plus bas.

Qualité d'approximation Les graphes 1.5 montrent le SNR d'approximation défini par

$$SNR^{(i)} = -10 \log \frac{\|r^{(i)}\|_2^2}{\|s\|_2^2} \quad (1.46)$$

Sur la courbe originale on ne peut pas distinguer OMP des autres algorithmes. Seul MP présente un comportement nettement moins bon avec une perte de 0.6dB au bout des 20000 itérations. Pour observer les différences on a tracé sur la courbe du bas la perte de SNR par rapport à OMP. LocOMP présente une perte finale de 0.01dB alors que GP fournit un résultat aussi bon que OMP. La perte de LocOMP est négligeable par rapport à celle de MP. Comme OMP et GP se comportent de la même manière, on s'attendrait à la même proximité pour LocOMP et LocGP, mais ce n'est étonnamment pas le cas. LocGP présente une perte finale de 0.09dB par rapport à OMP, soit 9 fois plus importante que LocOMP. Nous n'avons pas d'explication de ce phénomène à l'heure actuelle.

Sur cette expérience, LocOMP obtient la qualité d'approximation de OMP pour la complexité de MP. On dispose de garanties sur la complexité, pas sur la qualité d'approximation.

1.10.2 MPTK

Nous avons également évalué notre implémentation de LocGP sous MPTK. Dans ce cadre, on ne dispose plus des autres algorithmes, notamment OMP, mais on peut utiliser des dictionnaires plus gros. Les signaux musicaux utilisés font toujours 1 minute et sont toujours échantillonnés à 8000Hz, mais maintenant le dictionnaire est composé de 2 dictionnaires MDCT d'échelles $L_1 = 32$ et $L_2 = 1024$. Ces échelles ont été choisies parce qu'elles correspondent aux tailles de fenêtre utilisées en compression audio, notamment pour les formats mp3 et mp4.

Le graphe 1.6 montre l'évolution du SNR en fonction du nombre d'itérations.

Les temps de calcul de LocGP sont décevants. Alors que MP boucle le calcul en moins de 1 minute, il faut près d'1 jour pour LocGP. Après analyse du comportement du programme, il semble que cet écart vienne de la mise à jour du résidu.

MPTK utilise des dictionnaires rapides. Les corrélations sont calculées par FFT et la forme d'onde des atomes n'est utilisée que quand il faut soustraire l'atome du résidu. On a vu à la section 1.2.4 que cette soustraction peut également être effectuée par FFT inverse, mais c'est peu rentable quand on soustrait les atome 1 par 1, aussi ça n'a pas été implémenté dans MPTK. Actuellement, dans MPTK, la forme d'un atome est calculée à la demande quand il doit être soustrait et n'est pas conservée en mémoire. Or LocGP nécessite à chaque itération de soustraire tous les atomes du sous-dictionnaire $\Psi^{(i)}$, dont les formes ont été calculées aux itérations précédentes. On peut donc accélérer ce calcul en conservant en mémoire les formes des atomes ou en implémentant la synthèse rapide par FFT qui n'utilise pas du tout les formes d'onde. On s'oriente plutôt vers cette deuxième solution pour économiser la mémoire.

1.11 Perspectives

1.11.1 Multidimensions

Tel qu'il a été présenté, LocOMP s'applique à des signaux unidimensionnels, comme des séries temporelles. Néanmoins, on trouve également des dictionnaires locaux en traitement d'images. Peut-on étendre LocOMP à ce cas ?

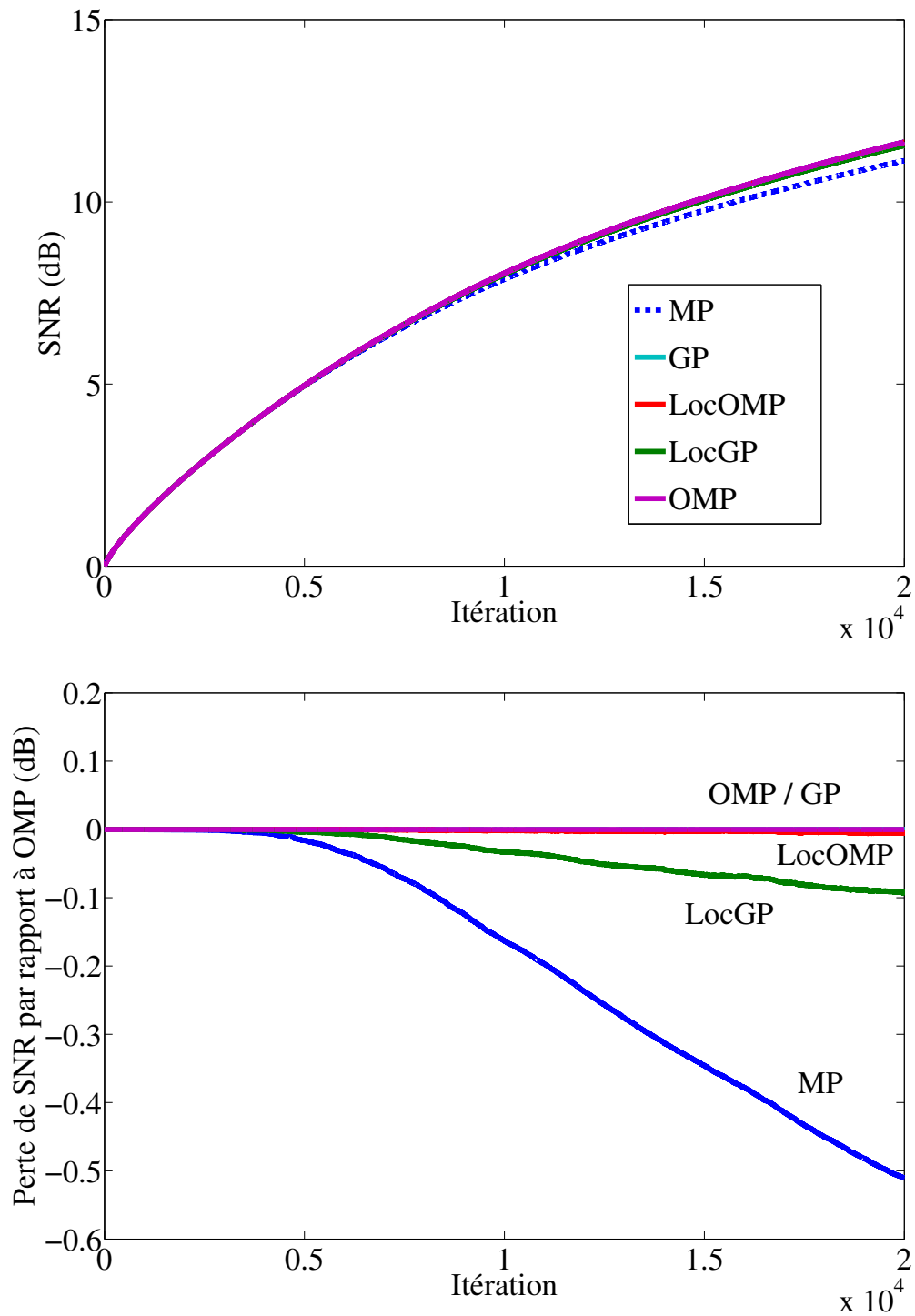


FIG. 1.5 – SNR obtenu par les différents algorithmes en fonction du nombre d'itérations total. MP mis à part, tous les autres algorithmes offrent une qualité équivalente. La courbe du bas montre la perte de SNR par rapport à OMP. On voit que LocOMP subit une légère perte de 0.01dB et LocGP une perte un peu plus importante de 0.09dB. La tangente en pointillés est la limite inférieure du coût d'un algorithme global. Elle correspond au coût de calcul des corrélations $\lambda = \Phi^*r$ à chaque itération.

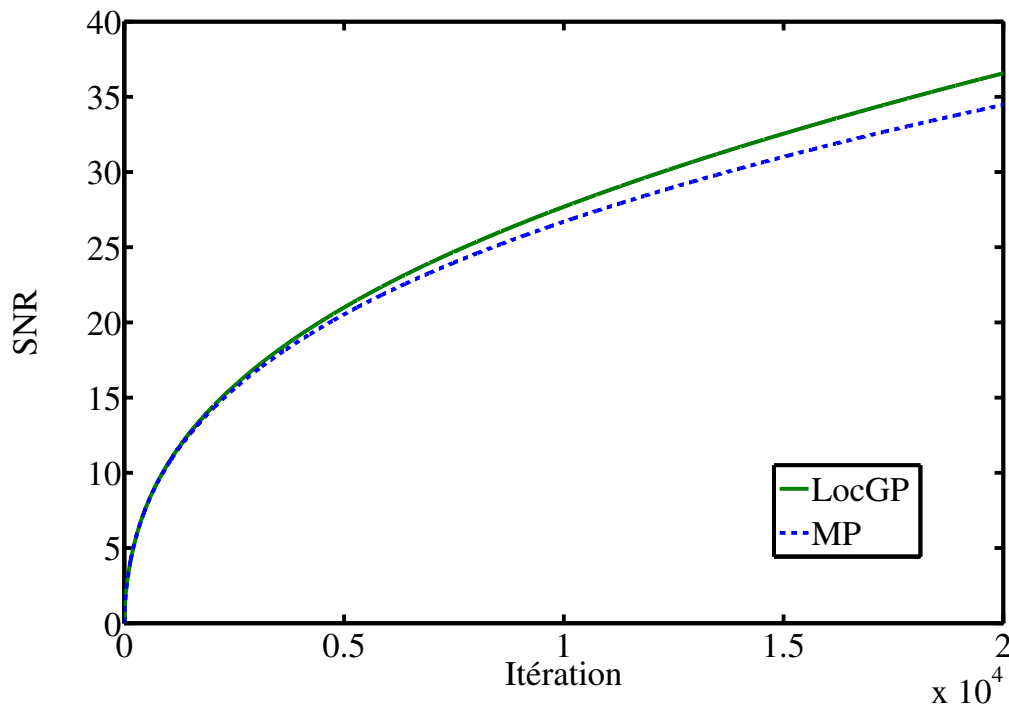


FIG. 1.6 – SNR moyen obtenu par MPTK et LocGP en fonction du nombre d’itérations total sur 10 signaux de piano échantillonnés à 8 kHz avec des dictionnaires MDCT d’échelles 32 et 1024. LocGP permet un gain moyen de 2dB.

La spécificité de LocOMP est la sélection du sous-dictionnaire $\Psi^{(i)}$. La notion de chevauchement des supports est définie quelle que soit la dimension des atomes, donc on peut conserver la même définition pour $\Psi^{(i)}$.

Par contre, l’extraction rapide de $\Psi^{(i)}$ hors de $\Phi^{(i)}$ utilise le tri des atomes par instants de début t_{\min} croissants. Or quand la dimension des atomes augmente, on n’a plus un instant de début mais des coordonnées. Il va donc falloir trouver une autre structure de données pour remplacer ce tri.

Dans le pire des cas, on peut toujours effectuer l’extraction lentement en testant pour chaque atome de $\Phi^{(i)}$ son appartenance à $\Psi^{(i)}$. Cela ajoute un coût en $O(i)$ à chaque itération, donc en $O(K^2)$ pour l’ensemble de l’algorithme. Ce coût peut devenir prédominant sur le calcul des corrélations pour $K > \alpha L \log L$.

1.11.2 Quantification de l’erreur commise par rapport à OMP

Idéalement, on souhaiterait obtenir la qualité d’approximation d’OMP pour la complexité de MP. Nous avons vu à la section 1.6.2 que l’objectif était atteint pour la complexité. Mais peut-on garantir la qualité de l’approximation obtenue?

De nombreuses garanties théoriques s’appliquent à la fois à MP et à OMP. Par exemple, les conditions de Tropp [Tro04] qui donnent une condition suffisante pour que l’algorithme ne sélectionne que des atomes appartenant au meilleur support, sont valables pour n’importe quel algorithme glouton qui :

- sélectionne l’atome le plus corrélé au résidu,
- calcule le nouveau résidu en lui soustrayant une combinaison linéaire d’atomes sélectionnés.

Elles sont donc également valables pour LocOMP, mais cela ne nous donne aucune information sur le comportement relatif à OMP.

Une différence importante entre MP et OMP est que OMP ne sélectionne jamais 2 fois le même atome et annule le résidu en N itérations. Mais LocOMP n'offre pas cette garantie.

1.12 Conclusion

Le traitement de signaux à l'aide de dictionnaires locaux nécessite des algorithmes spécialement conçus pour pouvoir traiter des signaux de grande taille $N > 10^6$. On savait que MP pouvait être implémenté efficacement dans ce cas.

Contributions Nous avons proposé plusieurs accélérations pour OMP. Ces accélérations sont insuffisantes parce qu'elles ne dispensent pas de devoir recalculer toutes les corrélations $\lambda = \Phi^*r$ à chaque itération.

Nous avons proposé un nouvel algorithme, LocOMP, dont le comportement approche celui d'OMP pour un coût proche de celui de MP. Cet algorithme s'appuie sur le choix d'un sous-dictionnaire $\Psi^{(i)}$ d'atomes proches temporellement du dernier atome sélectionné et sur une orthogonalisation locale de la décomposition. Nous avons également proposé le moyen de réaliser l'extraction de $\Psi^{(i)}$ rapidement en maintenant une structure de données triée des atomes sélectionnés $\Phi^{(i)}$. On peut maîtriser la complexité de l'algorithme en choisissant $\Psi^{(i)}$ suffisamment petit.

On constate expérimentalement que LocOMP peut calculer des approximations aussi bonnes qu'OMP. Par contre, l'implémentation dans la librairie MPTK destinée à être distribuée aux utilisateurs de décompositions parcimonieuses laisse encore à désirer.

Perspectives Il reste encore de nombreuses questions à résoudre sur l'analyse de LocOMP. Tout d'abord, quel est le lien entre le choix de $\Psi^{(i)}$ et la qualité de l'approximation? Plus le sous-dictionnaire est grand, meilleure est l'approximation sur une itération. On peut supposer que ce résultat s'étend à l'ensemble de l'algorithme, mais peut-on le quantifier? On a constaté expérimentalement que LocOMP était déjà très proche d'OMP en prenant $\Psi^{(i)}$ aussi petit que possible, donc ça ne vaut peut-être pas la peine de chercher un sous-dictionnaire plus grand.

D'autre part, peut-on obtenir des garanties sur la qualité du résultat final? Cette question dépasse LocOMP et englobe tous les algorithmes gloutons pour le cas des dictionnaires locaux. On dispose de résultats [Tro04] pour le cas de dictionnaires à faible cohérence, mais les dictionnaires locaux ne sont pas des dictionnaires à faible cohérence. Pour minimiser la cohérence d'un dictionnaire, les atomes doivent être répartis aussi uniformément que possible sur la sphère unité. Chaque atome est alors corrélé à tous les autres, mais très faiblement. Inversement, on a vu à la section 1.5.2 que la matrice de Gram d'un dictionnaire local est creuse. Donc un atome est strictement orthogonal à presque tous les autres et doit être fortement corrélé à quelques autres. Il doit y avoir des choses intéressantes à observer sur cette géométrie. Peut-être même pourrait-on en tirer partie pour améliorer les résultats. Nous avons proposé des algorithmes spécialement conçus pour aller vite sur des dictionnaires locaux. Est-ce qu'on pourrait imaginer des algorithmes spécialement conçus pour retrouver le meilleur support sur des dictionnaires locaux?

Chapitre 2

Apprentissage de dictionnaires invariants par translation

Avant même de pouvoir décomposer un signal, il faut savoir quel dictionnaire utiliser. On peut parfois connaître un bon dictionnaire à utiliser. Par exemple on sait que des bases DCT à 2 dimensions correspondent bien aux scènes naturelles [BS97] et que des atomes temps-fréquences correspondent bien aux sons [AP01]. Mais on se trouve parfois face à de nouveaux signaux pour lesquels on ne dispose pas de cette connaissance. On peut également ne pas souhaiter utiliser de dictionnaires génériques, justement parce qu'ils sont trop génériques. C'est le cas de l'application de séparation de sources que nous verrons à la partie 3. Dans ce cas, il est nécessaire d'obtenir le dictionnaire à partir des données qu'on souhaite représenter.

Problème Le problème de l'apprentissage de dictionnaire consiste à déterminer automatiquement un dictionnaire Φ capable d'approcher parcimonieusement un ensemble de signaux d'apprentissage. Soit S une matrice de taille $N \times \mathcal{S}$ contenant les signaux d'apprentissage rangés en colonnes, et Φ le dictionnaire de taille $N \times D$ qu'on souhaite apprendre. On doit avoir au moins $\mathcal{S} > D$ signaux d'apprentissage, faute de quoi le problème est trivial : on pourrait prendre le dictionnaire dont les atomes sont les signaux eux-mêmes.

La fonction à optimiser est dérivée des critères utilisés en approximation parcimonieuse. Considérons un signal s . Pour un dictionnaire Φ et un vecteur de coefficients x donnés, supposons qu'on connaît une fonction $\epsilon(s, \Phi, x)$ qui mesure l'erreur de l'approximation de s par Φx . On peut par exemple prendre l'énergie du résidu

$$\epsilon(s, \Phi, x) = \|s - \Phi x\|_2^2 \quad (2.1)$$

L'erreur totale $\mathcal{E}(S, \Phi, X)$ commise lors de l'approximation des signaux d'apprentissage S par une matrice de coefficients X vaut alors

$$\mathcal{E}(S, \Phi, X) = \sum_{i=1}^{\mathcal{S}} \epsilon(s_i, \Phi, x_i) \quad (2.2)$$

$$= \sum_{i=1}^{\mathcal{S}} \|s_i - \Phi x_i\|_2^2 \quad (2.3)$$

$$= \|S - \Phi X\|_{FRO}^2 \quad (2.4)$$

avec $\|\cdot\|_{FRO}$ la norme de Frobenius définie par $\|A\|_{FRO} = \sqrt{\text{Tr}(A^*A)}$ et $\text{Tr}(A)$ la somme des éléments diagonaux de A . On définit alors l'erreur $\mathcal{E}(S, \Phi)$ commise avec le dictionnaire

Φ comme la plus petite erreur atteignable avec des coefficients X parcimonieux.

$$\mathcal{E}(S, \Phi) = \min_{X \in \mathbf{R}^{D \times S}} \mathcal{E}(S, \Phi, X) \quad (2.5)$$

sous la contrainte

$$\forall 1 \leq s \leq S, \|x_s\|_0 \leq K \quad (2.6)$$

Apprendre un dictionnaire revient alors à chercher le dictionnaire $\hat{\Phi}$ qui minimise l'erreur $\mathcal{E}(S, \Phi)$. Comme on l'a vu dans l'introduction de cette thèse, on peut se restreindre aux dictionnaires composés d'atomes unitaires.

$$\hat{\Phi} = \operatorname{argmin}_{\Phi \in \mathbf{R}^{N \times D}} \mathcal{E}(S, \Phi) \quad (2.7)$$

sous la contrainte

$$\forall 1 \leq d \leq D, \|\varphi_d\|_2 = 1 \quad (2.8)$$

En pratique, on ne sait pas calculer directement $\hat{\Phi}$ sans exprimer les coefficients \hat{X} des approximations correspondantes. Le problème se réécrit alors en :

$$\left(\hat{\Phi}, \hat{X} \right) = \operatorname{argmin}_{\Phi \in \mathbf{R}^{N \times D}, X \in \mathbf{R}^{D \times S}} \|S - \Phi X\|_{FRO}^2 \quad (2.9)$$

sous les contraintes

$$\forall 1 \leq s \leq S, \|x_s\|_0 \leq K \quad (2.10)$$

$$\forall 1 \leq d \leq D, \|\varphi_d\|_2 = 1 \quad (2.11)$$

Les différentes contraintes sont non convexes. De plus, on sait qu'il existe plusieurs minima globaux à ce problème : l'erreur d'approximation est notamment invariante par permutation des atomes, colonnes de Φ (à condition d'appliquer la même permutation aux lignes des coefficients X).

Il ne faut pas confondre la matrice des signaux d'apprentissage S avec un signal multicanal : les supports de la décomposition de chaque signal d'apprentissage sur Φ sont indépendants. Dans le problème (2.9) on a choisi d'autoriser le même nombre de coefficients K dans chaque signal pour simplifier l'écriture.

Plan Après avoir présenté les principaux algorithmes utilisés et notamment vu leurs limites de complexité, nous présenterons une évolution de l'algorithme d'Olshausen et Field [OF97]. Cet algorithme sera validé expérimentalement puis étendu au cas des dictionnaires multicanaux et invariants par translation qui nous seront utiles à la partie 3

2.1 État de l'art

La plupart des algorithmes existants fonctionnent par optimisation alternée :

- chercher les meilleurs coefficients X à dictionnaire Φ fixé,
- chercher le meilleur dictionnaire à coefficients fixés.

On reconnaît dans la première étape le problème d'approximation parcimonieuse développé dans la partie 1. Ce problème étant largement étudié en lui-même, les travaux en apprentissage de dictionnaires se concentrent sur la manière de mettre à jour le dictionnaire.

2.1.1 Algorithme d'Olshausen-Field

Un des premiers algorithmes proposés est celui mis en œuvre par Olshausen et Field [OF97]. Plutôt que de faire varier simultanément l'ensemble du dictionnaire, chaque atome est optimisé séparément. L'optimisation se fait par descente de gradient projetée à pas fixe. Soit $R = S - \Phi X$ le résidu de la décomposition. On peut différencier l'erreur par rapport à l'atome φ_d pour obtenir le gradient $\nabla_d \mathcal{E}(S, \Phi, X) = \frac{\partial}{\partial \varphi_d} \mathcal{E}(S, \Phi, X)$. Pour calculer le gradient, calculons la variation de l'erreur en fonction d'une variation élémentaire $\partial \varphi_d$ de l'atome φ_d .

$$\mathcal{E}(S, \Phi + \partial \varphi_d, X) = \|S - (\Phi X + \partial \varphi_d X_{d,:})\|_{FRO}^2 \quad (2.12)$$

$$\mathcal{E}(S, \Phi + \partial \varphi_d, X) = \|R - \partial \varphi_d X_{d,:}\|_2^2 \quad (2.13)$$

$$\mathcal{E}(S, \Phi + \partial \varphi_d, X) = \langle R, R \rangle - 2 \langle R, \partial \varphi_d X_{d,:} \rangle + o(\|\partial \varphi_d\|_2) \quad (2.14)$$

$$\nabla_d \mathcal{E}(S, \Phi, X) = -2RX_{d,:}^* \quad (2.15)$$

avec $X_{d,:}$ la $d^{\text{ième}}$ ligne de X qui contient tous les coefficients de l'atome φ_d . *Le gradient pour l'atome φ_d est égal à la moyenne des résidus pondérée par l'amplitude des coefficients de l'atome dans la décomposition.* On effectue ensuite une descente de gradient projetée de pas fixe $\frac{\rho}{2}$; le nouvel atome est donné par la formule :

$$\varphi_d^{(i)} = \varphi_d^{(i-1)} - \frac{\rho}{2} \nabla_d \mathcal{E}(S, \Phi, X) \quad (2.16)$$

$$= \varphi_d^{(i-1)} + \rho RX_{d,:}^* \quad (2.17)$$

$$\varphi_d^{(i)} \leftarrow \frac{\varphi_d^{(i)}}{\|\varphi_d^{(i)}\|_2} \quad (2.18)$$

La complexité d'une mise à jour du dictionnaire entier est faible. Pour mettre à jour l'ensemble du dictionnaire, il faut :

- calculer le produit RX^* (en plusieurs fois, une ligne de X par atome)
- le multiplier par ρ
- l'ajouter à Φ
- normaliser les atomes
- recalculer le résidu $R = S - \Phi X$ (atome par atome)

La matrice de coefficients X est creuse, elle ne contient que KS coefficients non nuls. Pour chacun de ces coefficients $x_{d,s}$, le calcul du produit RX^* nécessite la multiplication de la colonne correspondante r_s par $x_{d,s}$, puis son addition à la $m^{\text{ième}}$ colonne du résultat. Au total, cela prend $O(KSN)$ opérations. Toutes les autres opérations se font en $O(D \times N)$ qui est le nombre de coefficients du dictionnaire.

Les principaux atouts de cette itération sont sa faible complexité et la simplicité de son code. Son gros défaut est la présence du paramètre ρ qui nécessite un réglage à la main.

2.1.2 Algorithme MOD

La Méthode des Directions Optimales (MOD) [EAHH99] se distingue du précédent algorithme par l'optimisation simultanée de tous les atomes du dictionnaire. L'algorithme cherche la matrice $\delta \Phi$ à ajouter à Φ pour minimiser l'erreur $\mathcal{E}(S, \Phi, X)$. Sans contrainte de normalisation, la réponse est donnée par un calcul de pseudo-inverse :

$$\delta \Phi = (XX^*)^{-1} XR^* \quad (2.19)$$

Par rapport à l'itération d'Olshausen et Field, on retrouve le calcul de RX^* et l'ajout d'une matrice à Φ . L'opération supplémentaire est l'inversion de l'opérateur XX^* de taille (D^2) . Son inversion peut se faire en $O(D^3)$ par méthode de Choleski par exemple. Cette étape supplémentaire ajoute un coût important quand le dictionnaire grandit, que ce soit parce que la dimension N augmente ou parce que le dictionnaire est fortement redondant.

2.1.3 Algorithme K-SVD

L'algorithme K-SVD [AEB05], nommé en référence à l'algorithme de clustering K-means¹, relâche le principe d'algorithme alterné. On a vu dans l'introduction de cette thèse que la partie intéressante (et difficile) de l'approximation parcimonieuse est le calcul du support de la décomposition. L'idée de K-SVD est donc de ne conserver que ce support lors de la mise à jour du dictionnaire, et d'optimiser conjointement les atomes et les coefficients non nuls, ce qui correspond à la recherche d'une composante principale.

Pour cela, K-SVD optimise chaque atome indépendamment comme l'algorithme d'Olshausen et Field. Pour chaque atome φ_d , on définit la matrice d'*extraits* $P^{(d)}$ obtenue en retirant de S la contribution des autres atomes :

$$P^{(d)} = S - \Phi X + \varphi_d X_{d,:} \quad (2.20)$$

Puis on cherche conjointement l'atome $\hat{\varphi}_d$ et les coefficients associés $\hat{X}_{d,:}$ (mais les coefficients des autres atomes sont fixés) qui minimisent l'erreur $\mathcal{E}(S, \Phi, X)$.

$$(\hat{\varphi}_d, \hat{X}_{d,:}) = \operatorname{argmin} \mathcal{E}(S, \Phi, X) \quad (2.21)$$

sous la contrainte

$$\|\varphi_d\|_2 = 1 \quad (2.22)$$

$$\begin{aligned} \mathcal{E}(S, \Phi, X) &= \|S - \Phi X\|_{FRO}^2 \\ &= \left\| P^{(d)} - \varphi_d X_{d,:} \right\|_{FRO}^2 \\ &= \sum_{s=1}^S \left\| P_s^{(d)} - \varphi_d X_{d,s} \right\|_2^2 \end{aligned} \quad (2.23)$$

Quel que soit l'atome φ_d , on sait que pour chaque extrait $P_s^{(d)}$ le meilleur coefficient correspondant est $X_{d,s} = \langle P_s^{(d)}, \varphi_d \rangle$ qui réalise la projection orthogonale de $P_s^{(d)}$ sur φ_d . Le problème se ramène donc à

$$\hat{X}_{d,:} = \varphi_d^* P^{(d)} \quad (2.24)$$

$$\hat{\varphi}_d = \operatorname{argmin} \left\| P^{(d)} - \varphi_d \varphi_d^* P^{(d)} \right\|_{FRO}^2 \quad (2.25)$$

$$\begin{aligned} \left\| P^{(d)} - \varphi_d \varphi_d^* P^{(d)} \right\|_{FRO}^2 &= \operatorname{Tr} \left(P^{(d)*} P^{(d)} - 2P^{(d)*} \varphi_d \varphi_d^* P^{(d)} + P^{(d)*} \varphi_d \varphi_d^* \varphi_d \varphi_d^* P^{(d)} \right) \\ &= \operatorname{Tr} \left(P^{(d)*} P^{(d)} - 2P^{(d)*} \varphi_d \varphi_d^* P^{(d)} + P^{(d)*} \varphi_d \varphi_d^* P^{(d)} \right) \end{aligned}$$

car $\varphi_d^* \varphi_d = 1$

$$= \operatorname{Tr}(P^{(d)*} P^{(d)}) - \operatorname{Tr}(P^{(d)*} \varphi_d \varphi_d^* P^{(d)})$$

$$= \operatorname{Tr}(P^{(d)*} P^{(d)}) - \operatorname{Tr}(\varphi_d^* P^{(d)} P^{(d)*} \varphi_d)$$

¹Ce choix de nom n'est pas très heureux : on peut tout à fait utiliser la SVD dans K-means sans pour autant apprendre un dictionnaire. Si dans K-means on prend comme distance l'angle entre deux vecteurs au lieu de leur distance euclidienne, alors le calcul des centroïdes se fait par SVD.

TAB. 2.1 – Récapitulatif des spécificités et limites des algorithmes de l'état de l'art

Algorithme	Olshausen-Field	MOD	K-SVD
Coefficients pendant la mise à jour	fixes	fixes	libres
Mise à jour des différents atomes	séquentielle	jointe	séquentielle
Utilisable pour de grands $K\mathcal{S}$	oui	non	oui
Utilisable pour de grands N	oui	oui	non
Paramètres de l'algorithme	ρ	\emptyset	\emptyset

Or $\varphi_d^* P^{(d)} P^{(d)*} \varphi_d$ est un scalaire, donc

$$\hat{\varphi}_d = \operatorname{argmax} \varphi_d^* P^{(d)} P^{(d)*} \varphi_d \quad (2.26)$$

L'équation (2.26) est connue comme l'objectif minimisé par l'analyse en composantes principales (ACP). En général, l'inconnue de l'ACP est une matrice et la solution recherchée une base orthonormale d'un sous-espace qui minimise ce coût. Cette base est donnée par la SVD de $P^{(d)}$, ou de manière équivalente par la diagonalisation orthogonale de $P^{(d)} P^{(d)*}$, en choisissant les vecteurs propres associés aux valeurs propres les plus grandes. Dans notre cas, il n'y a qu'un seul vecteur φ_d , donc on se contente de la première composante principale.

D'après Golub et Van Loan [GVL96], la SVD peut se calculer en $O(N^2 \frac{K\mathcal{S}}{D})$. La dépendance en N^2 rend le calcul coûteux quand on travaille sur de grands atomes.

2.1.4 Conclusion

Différents algorithmes d'apprentissage ont été rappelés. Le tableau 2.1 récapitule les points forts et faibles de chaque algorithme. L'algorithme d'Olshausen et Field est le seul qui passe à l'échelle à la fois pour de grandes décompositions et pour de grands atomes. Sa simplicité semble également en faire l'algorithme le plus utilisé actuellement pour des applications de l'apprentissage de dictionnaires. Malheureusement la présence du pas de descente ρ complique son utilisation. Il semble qu'un pas $\rho = 0.1$ donne de bons résultats [BD06], mais on n'a pas de garantie à ce sujet. C'est pourquoi notre travail s'est orienté vers la recherche d'une estimation automatique de ce pas.

2.2 Pas optimal pour l'algorithme d'Olshausen et Field

L'algorithme d'Olshausen et Field est le seul algorithme itératif de mise à jour du dictionnaire qui assure une complexité linéaire en fonction de la dimension de l'espace N et du nombre d'atomes de la décomposition $\|X\|_0 = K\mathcal{S}$. Néanmoins il possède un paramètre à régler, le pas de la descente ρ . On souhaite se débarrasser de ce paramètre pour faciliter l'utilisation de l'algorithme et pour accélérer la vitesse de convergence.

On connaît l'influence du pas d'apprentissage sur les algorithmes de descente de gradient :

- si le pas choisi est trop petit ou trop grand, l'algorithme ne converge pas,
- on peut assurer la convergence en choisissant un pas décroissant lentement en fonction de l'itération (une décroissance en $\frac{1}{\sqrt{i}}$ convient, par exemple)
- même dans ce cas, la vitesse de convergence est très inférieure à celle d'une descente de gradient à pas optimal.

L'entrelacement avec l'étape de décomposition ne permet pas de transposer directement ces résultats à notre problème, mais on peut tout de même espérer une convergence plus rapide si on parvient à déterminer le pas optimal de la descente.

2.2.1 Problème

On cherche à minimiser la fonction décrite par l'équation (2.9). A l'itération i , on connaît le dictionnaire calculé à l'itération précédente $\Phi^{(i-1)}$. Les coefficients $X^{(i)}$ sont calculés en décomposant les signaux d'apprentissage S sur le dictionnaire. Pour diminuer la complexité on utilise le même cadre simplificateur qu'Olshausen et Field :

- on calcule le nouveau dictionnaire $\Phi^{(i)}$ à coefficients X fixés, ce qui simplifie par rapport à K-SVD,
- les atomes du dictionnaires sont calculés les uns après les autres, ce qui simplifie par rapport à MOD.

On peut supposer sans perte de généralité que les atomes sont mis à jour par ordre d'indices croissants. Donc quand on calcule l'atome $\varphi_d^{(i)}$, on connaît déjà tous les atomes $\varphi_{d'}^{(i)}$ avec $d' < d$. On note $X_{d,:}$ la $d^{\text{ième}}$ ligne de X qui contient l'amplitude de l'atome φ_d dans chaque signal. Le nouvel atome $\varphi_d^{(i)}$ est défini par :

$$\varphi_d^{(i)} = \operatorname{argmin}_{\{\varphi \in \mathbf{R}^N \mid \|\varphi\|_2=1\}} \left\| S - \sum_{d'=1}^{d-1} \varphi_{d'}^{(i)} X_{d',:}^{(i)} - \sum_{d'=d+1}^D \varphi_{d'}^{(i-1)} X_{d',:}^{(i)} - \varphi_d X_{d,:}^{(i)} \right\|_2^2 \quad (2.27)$$

$$= \operatorname{argmin}_{\{\varphi \in \mathbf{R}^N \mid \|\varphi\|_2=1\}} \left\| P^{(i,d)} - \varphi_d X_{d,:}^{(i)} \right\|_{FRO}^2 \quad (2.28)$$

Ce problème est le même que celui d'Olshausen et Field. C'est maintenant que notre méthode diverge. Olshausen et Field se contentent d'une étape de descente de gradient pour aller vers le meilleur atome. Nous allons montrer que cet optimum peut être calculé directement.

2.2.2 Calcul de l'atome optimal $\varphi_d^{(i)}$

Développons la fonction à minimiser dans le problème (2.28) :

$$\left\| P^{(i,d)} - \varphi_d X_{d,:}^{(i)} \right\|_2^2 = \left\| P^{(i,d)} \right\|_2^2 - 2 \left\langle P^{(i,d)}, \varphi_d X_{d,:}^{(i)} \right\rangle + \left\| \varphi_d X_{d,:}^{(i)} \right\|_2^2 \quad (2.29)$$

Le premier terme de cette somme ne dépend pas de φ_d , et le dernier est égal à :

$$\left\| \varphi_d X_{d,:}^{(i)} \right\|_2^2 = \operatorname{Tr} (X_{d,:}^* \varphi_d^* \varphi_d X_{d,:}) \quad (2.30)$$

$$= \operatorname{Tr} (X_{d,:}^* X_{d,:}) \quad (2.31)$$

$$= \|X_{d,:}\|_2^2 \quad (2.32)$$

A cause de la contrainte de normalisation des atomes, le troisième terme ne dépend pas non plus de φ_d . Il ne reste donc plus que le double produit :

$$\varphi_d^{(i)} = \operatorname{argmin}_{\{\varphi \mid \|\varphi\|_2=1\}} -2 \left\langle P^{(i,d)}, \varphi_d X_{d,:} \right\rangle \quad (2.33)$$

$$= \operatorname{argmax}_{\{\varphi \mid \|\varphi\|_2=1\}} \left\langle P^{(i,d)} X_{d,:}^*, \varphi_d \right\rangle \quad (2.34)$$

On cherche le vecteur unitaire le plus corrélé à un vecteur donné. La solution est simple : c'est le vecteur lui-même, normalisé. On aboutit donc à la règle de mise à jour suivante :

$$\varphi_d^{(i)} = \frac{P^{(i,d)} X_{d,:}^{(i)*}}{\|P^{(i,d)} X_{d,:}^{(i)*}\|_2} \quad (2.35)$$

2.2.3 Lien avec le pas de descente ρ dans l'algorithme d'Olshausen et Field

La règle de mise à jour (2.35) peut se mettre sous la forme d'une descente de gradient. On peut alors considérer notre méthode comme un calcul du pas optimal pour la descente de gradient.

Soit $R^{(i,d)}$ le résidu courant défini par :

$$R^{(i,d)} = S - \sum_{d'=1}^d \varphi_{d'}^{(i)} X_{d',:}^{(i)} - \sum_{d'=d+1}^D \varphi_{d'}^{(i-1)} X_{d',:}^{(i)}$$

Alors les extraits peuvent s'exprimer en fonction du résidu :

$$P^{(i,d)} = R^{(i,d)} + \varphi_d^{(i-1)} X_{d,:}^{(i)}$$

Si on injecte cette expression dans l'équation (2.35), on obtient :

$$\varphi_d^{(i)} \propto \left(R^{(i,d)} + \varphi_d^{(i-1)} X_{d,:}^{(i)} \right) X_{d,:}^{(i)*} \quad (2.36)$$

$$\propto \left\| X_{d,:}^{(i)} \right\|_2^2 \varphi_d^{(i-1)} + R^{(i,d)} X_{d,:}^{(i)*} \quad (2.37)$$

$$\propto \varphi_d^{(i-1)} + \frac{1}{\left\| X_{d,:}^{(i)} \right\|_2^2} R^{(i,d)} X_{d,:}^{(i)*} \quad (2.38)$$

Or le gradient de la fonction (2.28) au point $\varphi_d^{(i-1)}$, utilisé par Olshausen et Field, est justement égal à $-2R^{(i,d)} X_{d,:}^{(i)*}$. On a donc montré que la règle (2.35) peut aussi se mettre sous la forme d'une descente de gradient projeté. Cela nous apprend 2 choses :

- le *minimum global de la fonction se trouve bien dans la direction du gradient*,
- le pas optimal $\hat{\rho}$ de cette descente vaut

$$\hat{\rho} = \frac{-1}{\left\| X_{d,:}^{(i)} \right\|_2} \quad (2.39)$$

Ce pas est optimal en tenant compte de la normalisation qui aura lieu ensuite. Géométriquement, l'atome $\varphi_d^{(i)}$ vit sur une portion de cercle définie par :

$$\exists(\alpha, \beta) \in (\mathbf{R}^{+*})^2, \varphi_d^{(i)} = \alpha \varphi_d^{(i-1)} + \beta \nabla_d \mathcal{E}(S, \Phi, X) \quad (2.40)$$

$$\left\| \varphi_d^{(i)} \right\|_2 = 1 \quad (2.41)$$

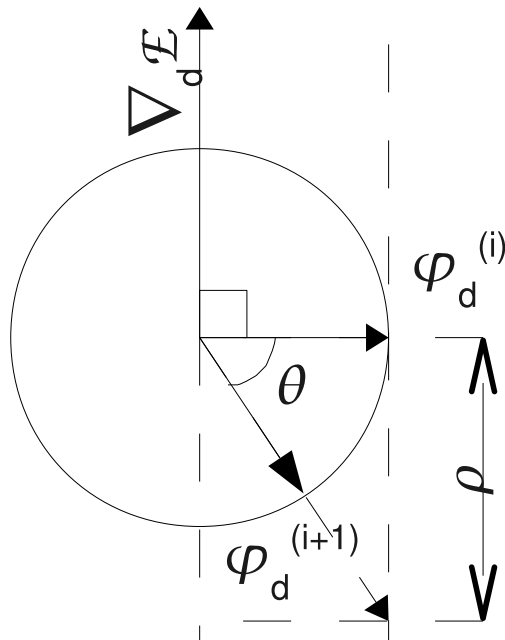


FIG. 2.1 – Interprétation géométrique du pas d'apprentissage ρ . Le nouvel atome $\varphi_d^{(i+1)}$ se trouve sur un quart du cercle trigonométrique défini par l'ancien atome $\varphi_d^{(i)}$ et le gradient $\nabla_d \mathcal{E}$. Le pas ρ est la tangente de l'angle θ entre l'ancien atome et le nouveau.

Si la décomposition est orthogonale (les coefficients donnent le projeté orthogonal du signal sur le support de la décomposition), alors tous les résidus sont orthogonaux à tous les atomes qui participent à la décomposition, donc le gradient est aussi orthogonal à l'atome :

$$\nabla_d \mathcal{E}(S, \Phi, X) \perp \varphi_d^{(i-1)} \quad (2.42)$$

Le nouvel atome se trouve donc sur le quart de cercle qui va de l'ancien atome à l'opposé du gradient. Dans ce cas, il existe un angle θ entre 0 et $\frac{\pi}{2}$ tel que

$$\varphi_d^{(i)} = \cos \theta \varphi_d^{(i-1)} - \sin \theta \frac{\nabla \mathcal{E}(S, \Phi, X)}{\|\nabla \mathcal{E}(S, \Phi, X)\|_2} \quad (2.43)$$

On a alors $\rho = \|\nabla_m \mathcal{E}(S, \Phi, X)\|_2 \tan \theta$. Le passage de ρ à θ est bijectif, ce sont 2 paramétrages différents du même problème. Quand on parle de pas optimal, il s'agit en fait de l'angle θ optimal. Cette interprétation géométrique est résumée dans la figure 2.1.

2.2.4 Complexité

La réécriture sous la forme (2.38) garantit également une faible complexité : la seule opération supplémentaire par rapport à l'algorithme d'Olshausen et Field est le calcul de la norme d'un vecteur creux. Pour l'ensemble de atomes, le coût total de cette opération est en $O(KS)$, ce qui est négligeable par rapport au coût de calcul du gradient en $O(KSN)$

Le surcoût qu'on a introduit est négligeable par rapport au coût original de l'algorithme d'Olshausen et Field, donc on reste dans la même classe de complexité.

2.2.5 Lien avec K-SVD

On peut tracer un parallèle entre notre algorithme et K-SVD. La règle (2.35) est la première itération d'un algorithme itératif qui converge vers la composante principale. Dans l'algorithme K-SVD, le nouvel atome $\varphi_d^{(i)}$ est défini comme la composante principale des extraits $P^{(i,d)}$:

$$\varphi_d^{(i)} = \operatorname{argmax}_{\varphi \in \mathbf{R}^N \mid \|\varphi\|_2=1} \varphi^* P^{(i,d)} P^{(i,d)*} \varphi \quad (2.44)$$

Les problèmes de composantes principales sont habituellement résolus par décomposition en valeurs singulières. Néanmoins, dans notre cas, on ne cherche que la première composante principale. Cette composante est le vecteur propre de $P^{(i,d)} P^{(i,d)*}$ associé à la plus grande valeur propre. L'algorithme des puissances est un algorithme itératif qui converge vers ce vecteur. Le fonctionnement général est décrit par l'Algorithme 5.

Algorithm 5 Calcul du vecteur propre principal v d'une matrice M par l'algorithme des puissances

Initialisation : v_0 aléatoire

for $i=1$ à I **do**

$v_i = M v_{i-1}$

$v_i = \frac{v_i}{\|v_i\|_2}$

end for

$v = v_I$

Dans notre cas, on peut supposer que le nouvel atome $\varphi_d^{(i)}$ devrait être proche du précédent. Donc le précédent atome $\varphi_d^{(i-1)}$ semble être un bon choix d'initialisation. Écrivons alors la première étape de l'algorithme :

$$v_0 = \varphi_d^{(i-1)} \quad (2.45)$$

$$v_1 = P^{(i,d)} P^{(i,d)*} \varphi_d^{(i-1)} \quad (2.46)$$

$$= \frac{v_1}{\|v_1\|_2} \quad (2.47)$$

Cette première étape est très proche de la règle (2.35). L'expression (2.46) est la moyenne des extraits pondérée par leur corrélation avec l'atome précédent, alors que la règle (2.35) utilise les coefficients de la décomposition comme pondérations. Or, si la décomposition calculée est la projection orthogonale de S sur les atomes de la décomposition, ces coefficients sont égaux. En effet, dans ce cas, on a

$$P^{(i,d)*} \varphi_d^{(i-1)} = \left(R^{(i,d)} + \varphi_d^{(i-1)} X_{d,:}^{(i)} \right)^* \varphi_d^{(i-1)} \quad (2.48)$$

$$= R^{(i,d)*} \varphi_d^{(i-1)} + X_{d,:}^{(i)*} \varphi_d^{(i-1)*} \varphi_d^{(i-1)} \quad (2.49)$$

$$= 0 + X_{d,:}^{(i)*} \left\| \varphi_d^{(i-1)} \right\|_2^2 \quad (2.50)$$

$$= X_{d,:}^{(i)*} \quad (2.51)$$

On a vu dans la partie 1 qu'il existe des algorithmes de décomposition parcimonieuse qui calculent cette projection, OMP par exemple. Dans ce cas, la règle (2.35) correspond donc à effectuer la première itération d'un calcul de composante principale par l'algorithme des

puissances. Ce lien est analogue à celui entre OMP et GP décrit à la section 1.2.5h . Dans les deux cas, on dispose d'un algorithme d'optimisation itératif et on n'en effectue que la première itération.

La décomposition calculée par OMP à l'itération i n'est orthogonale que par rapport au dictionnaire $\Phi^{(i-1)}$. Quand on commence à mettre à jour les atomes et à modifier le résidu en conséquence, la décomposition n'est plus orthogonale et le lien devient moins évident. Ce ne serait pas le cas si on mettait les atomes à jour en parallèle, chaque atome étant mis à jour en fonction des autres atomes $\Phi^{(i-1)}$ à l'itération précédente et du résidu calculé par OMP $R^{(i)} = S - \Phi^{(i-1)}X^{(i)}$ au lieu de tenir compte des autres atomes mis à jour dans la même itération.

2.3 Résultats expérimentaux

Cette étude expérimentale sert à évaluer les qualités de notre algorithme, à le comparer aux algorithmes de l'état de l'art et à approcher plusieurs autres questions pour lesquelles on ne possède pas pour l'instant de réponse théorique.

2.3.1 Objectif et mesure de qualité de l'apprentissage

L'évaluation du succès de l'apprentissage dépend du but poursuivi par l'application. Dans un contexte abstrait, on peut proposer plusieurs mesures.

Tout d'abord, puisque l'apprentissage est vu comme la minimisation de la fonction d'erreur $\mathcal{E}(S, \Phi, X)$, on peut mesurer la valeur atteinte. Ensuite, si on travaille sur des données de synthèse pour lesquelles le dictionnaire génératif est connu, on peut regarder si on a retrouvé exactement ce dictionnaire. Un dictionnaire est dit *identifiable* s'il est le seul dictionnaire pouvant avoir généré les données d'apprentissage S aux invariances par permutation et changement de signe près. Si le dictionnaire n'est pas identifiable, tous les dictionnaires pouvant générer les données produisent une erreur nulle. Dans ce cas, les algorithmes étudiés, qui sont basés sur une minimisation de l'erreur, n'ont aucun moyen de favoriser un dictionnaire par rapport à l'autre. La comparaison directe des dictionnaires est donc autant une mesure des données utilisées que des algorithmes.

Pour cette raison, on s'attachera surtout au SNR obtenu pour l'approximation défini par

$$SNR = -10 \log_{10} \frac{\mathcal{E}(S, \Phi, X)}{\|S\|_2^2} \quad (2.52)$$

$$= -10 \log_{10} \frac{\|S - \Phi X\|_2^2}{\|S\|_2^2} \quad (2.53)$$

2.3.2 Synthèse de données de tests

Toutes les expériences synthétiques ont été réalisées sur les mêmes données. Il s'agit de 100 dictionnaires aléatoires Φ dont les $D = 128$ atomes de taille $N = 64$ sont indépendants identiquement distribués (i.i.d.) de loi uniforme sur la sphère unité. On a ensuite généré $\mathcal{S} = 256$ signaux parcimonieux aléatoirement. Pour chaque signal, un support de décomposition de taille $K = 8$ est tiré uniformément. Les coefficients X non-nuls sont i.i.d. Nous avons effectué les mêmes essais avec une loi normale centrée et une loi de Rademacher (-1 ou 1 avec probabilité $\frac{1}{2}$). Puis les signaux sont synthétisés par le produit $S = \Phi X$.

2.3.3 Quel est le gain du pas adaptatif par rapport au pas fixe ?

Notre méthode possède moins de paramètres à régler que l'algorithme d'Olshausen et Field. Elle est également moins complexe que K-SVD et MOD. Mais quelle est la qualité des résultats obtenus par rapport à l'état de l'art ?

Nous avons évalué les différentes méthodes dans l'algorithme complet et en effectuant une part plus ou moins importante de l'algorithme complet afin d'isoler la mise à jour du résidu et de simuler le comportement de l'algorithme en fin de calcul, lorsque le support de la décomposition ne change plus. Nous avons d'abord fixé les coefficients et itéré des mises à jour du dictionnaire, puis fixé le support des coefficients et alterné la mise à jour du dictionnaire avec le calcul des coefficients par projection orthogonale, et enfin exécuté l'algorithme entier.

2.3.3.1 Comparaison à coefficients fixés

Pour cette première expérience, seule la mise à jour du dictionnaire a été considérée. Les coefficients X sont fixés à la valeur X_0 utilisée pour la synthèse des signaux $S = \Phi X_0$, et on itère des mises à jour du dictionnaire à partir d'un dictionnaire initial aléatoire. Dans ce cas, il ne reste plus qu'un problème d'optimisation quadratique sous contraintes. Les graphes 2.2 montrent le SNR atteint par les différents algorithmes en fonction de l'itération. Ces graphes sont des quantiles : ils montrent pour chaque algorithme le SNR atteint le plus bas, le plus haut, et celui atteint par chaque tranche de 10% des 100 données de test (soit un faisceau de 11 courbes pour chaque algorithme).

On voit que notre algorithme et K-SVD convergent tous les deux vers une solution optimale. Le maximum observé de 300dB correspond à la précision numérique. Il est un peu plus haut pour notre algorithme que pour K-SVD, ce qui semble indiquer que nos calculs sont plus simples et génèrent moins de propagations d'erreurs. Mais il est difficile de détailler ce point puisque le calcul de la SVD a été effectué par une fonction de la bibliothèque MATLAB. L'algorithme d'Olshausen et Field ne converge pas vers un dictionnaire optimal. Nous avons choisi un pas fixe $\rho = 0.1$.

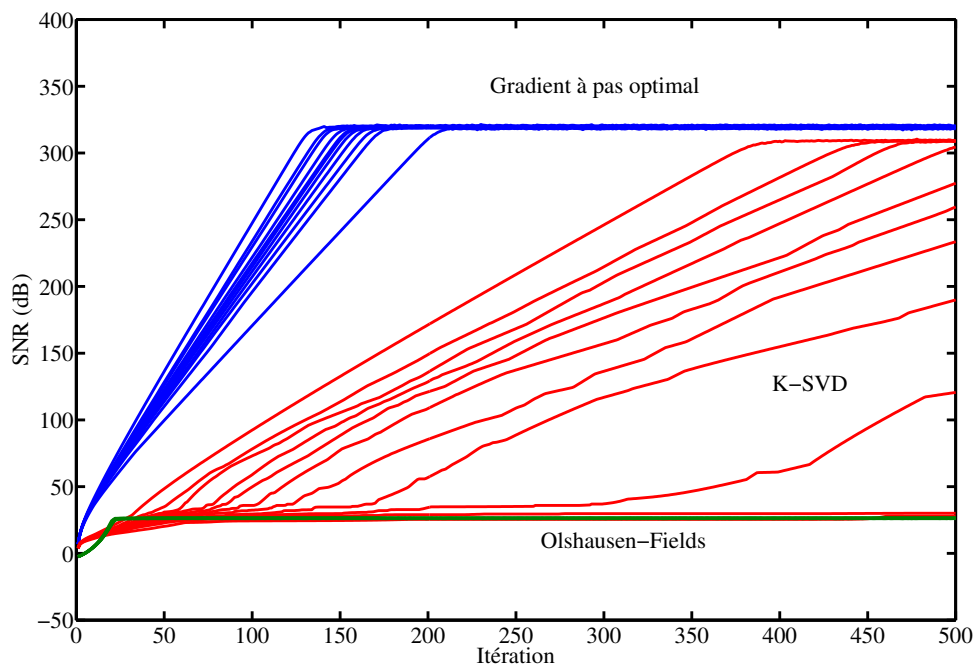
Si les coefficients non-nuls de X suivent une loi de Rademacher, alors les performances de K-SVD sont régulières. Par contre, avec une loi gaussienne sur les coefficients, on observe une énorme variabilité dans le comportement de K-SVD. Cela est dû au fait que la courbe du SNR présente un palier et c'est le nombre d'itérations que met K-SVD à dépasser ce palier qui varie fortement.

Pour cette expérience, l'algorithme MOD n'a pas été utilisé. Par définition, si on connaît les bons coefficients X , alors MOD converge vers le dictionnaire initial en 1 itération.

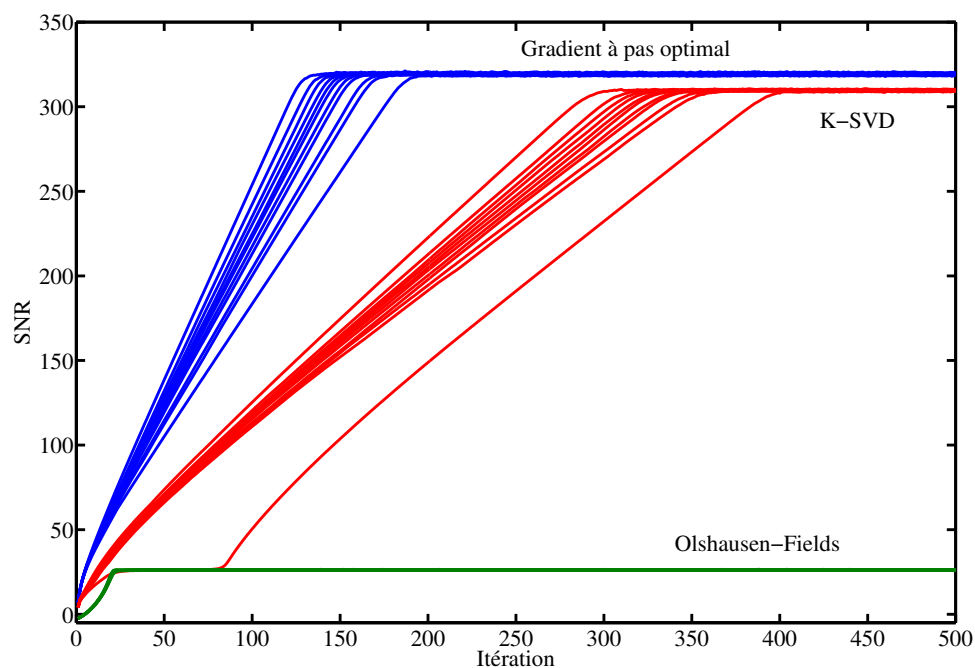
2.3.3.2 Comparaison à support de décomposition fixé

On peut trouver l'expérience précédente injuste pour K-SVD. En effet, le principal point fort de cet algorithme est d'estimer simultanément un atome et ses coefficients. Si on débute l'algorithme avec les coefficients cherchés, on peut s'attendre à ce que l'approche de K-SVD soit moins efficace puisque cette information n'est pas utilisée.

Protocole C'est pourquoi dans cette deuxième expérience, seul le support de la décomposition est fixé à la valeur $\text{supp}(X_0)$. Initialement, l'amplitude des coefficients non-nuls est aléatoire ainsi que le dictionnaire. Entre deux mises à jour du dictionnaire, au lieu d'effectuer une décomposition parcimonieuse pour estimer le support, on effectue juste une projection orthogonale de chaque signal sur le support de sa décomposition.



(a) Gauss



(b) Rademacher

FIG. 2.2 – SNR en fonction de l'itération pour les différentes règles de mise à jour du dictionnaire à coefficients fixés. Les coefficients à gauche ont été tirés selon une loi gaussienne à gauche et de Rademacher à droite. Notre descente de gradient à pas optimal assure la convergence la plus rapide. L'algorithme d'Olshausen et Field n'atteint pas de maximum global. L'algorithme K-SVD présente une très forte variabilité de comportement dans le cas des coefficients gaussiens. C'est dû à la présence d'un palier dans la progression. On peut observer ce palier sur le premier quantile de la courbe de droite. Avec des données gaussiennes, ce palier est tout le temps présent et c'est sa durée qui fait varier la vitesse de convergence de l'algorithme.

Résultats Les graphes 2.3 montrent les quantiles obtenus de la même manière qu'à la section précédente. La conclusion la plus frappante est que tous les algorithmes échouent la plupart du temps. Comme pour l'expérience précédente, les performances de K-SVD sont bien meilleures si les coefficients suivent une loi de Rademacher.

Ces résultats montrent à quel point on maîtrise peu l'apprentissage de dictionnaires. On sait depuis l'algorithme K-means que l'approche alternée n'est pas optimale. On sait également que le problème de la décomposition parcimonieuse est NP-complet. Ce que montre cette expérience, c'est qu'on ne sait pas non plus résoudre la deuxième étape de l'alternance. On peut donc se demander si l'alternance était une bonne idée : qu'a-t-on gagné à remplacer un problème qu'on ne sait pas résoudre par deux problèmes qu'on ne sait pas résoudre ?

2.3.3.3 Ensemble de l'algorithme

Pour cette dernière expérience, on a exécuté les algorithmes entiers (décomposition + mise à jour) à partir d'un dictionnaire aléatoire. Les résultats sont beaucoup plus réguliers, aussi l'observation du SNR moyen obtenu par chaque algorithme suffit. Il est tracé dans les figures 2.4 et 2.5 en utilisant MP et OMP pour la décomposition, avec des coefficients gaussiens ou de Rademacher.

La descente de gradient à pas optimal présente des performances décevantes, étant systématiquement dernier ex-aequo avec l'un ou l'autre des autres algorithmes. Le résultat surprenant est le bon résultat de l'algorithme d'Olshausen-Field. Néanmoins nous verrons à la section 2.3.4 que ces résultats sont à nuancer sur des données réelles.

K-SVD vaut-il encore la peine quand on utilise un algorithme de décomposition orthogonal ? La différence entre K-SVD et notre algorithme est que K-SVD optimise conjointement le dictionnaire Φ et la valeur des coefficients non-nuls de la décomposition X alors que notre méthode les fixe.

Il y a 2 causes d'erreur dans l'estimation des coefficients X par la décomposition. Tout d'abord, les coefficients $X^{(n)}$ sont calculés en fonction du dictionnaire précédent $\Phi^{(n-1)}$. Ensuite, certains algorithmes de décomposition ne calculent pas la projection orthogonale des signaux S . C'est par exemple le cas de MP et des algorithmes de pénalisation.

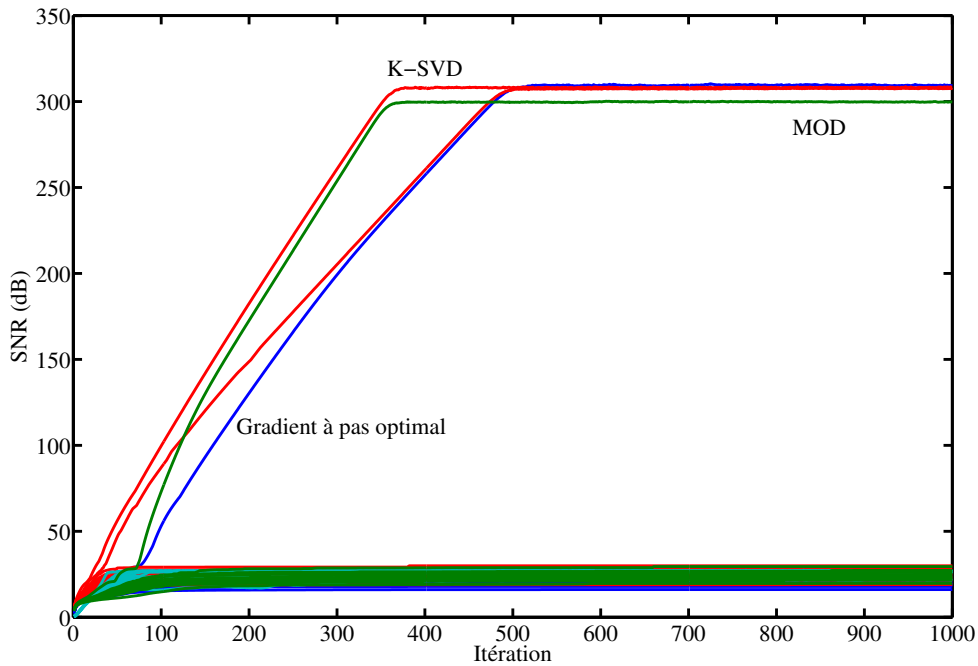
Néanmoins, il existe également des algorithmes qui décomposent bien S orthogonalement, notamment OMP. Si on utilise un tel algorithme pour la décomposition, la deuxième cause d'erreur sur les coefficients est annulée.

En comparant les figures 2.4 et 2.5, on constate que l'essentiel de l'avantage de K-SVD sur les algorithmes de gradient s'annule quand on utilise OMP, ce qui confirme notre intuition : *sur ces données, l'optimisation jointe des coefficients et des atomes ne sert pas tant à adapter les coefficients au nouveau dictionnaire qu'à corriger les erreurs commise lors de la décomposition.* On observe également que MOD présente de bien meilleures performances quand il est combiné avec OMP. Néanmoins ces résultats n'apparaissent pas aussi nettement sur des données réelles.

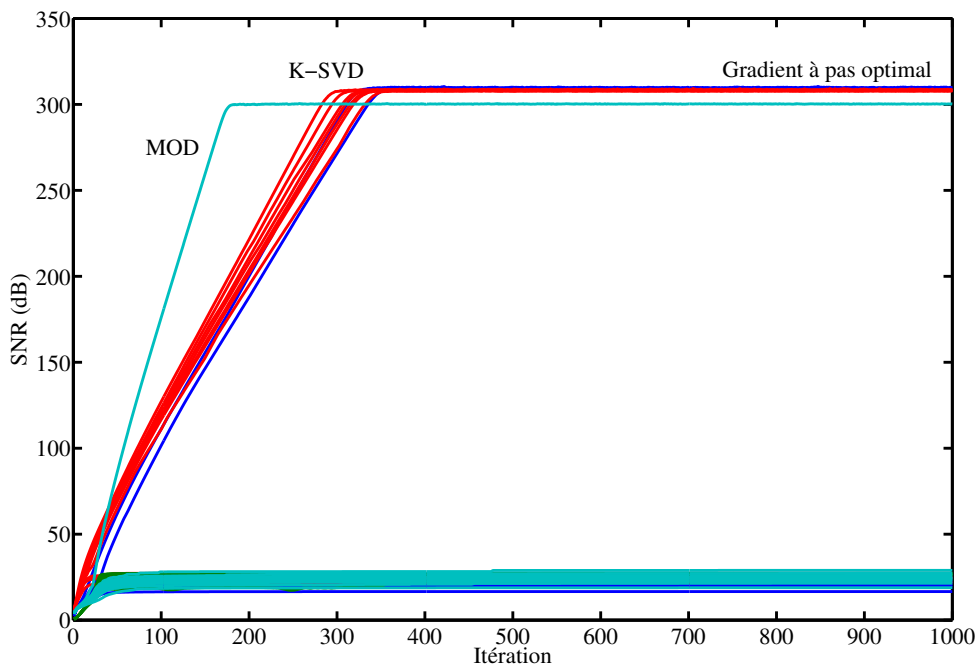
2.3.4 Évaluation sur la base d'images d'Olshausen et Field

Les images utilisées par Olshausen et Field sont 10 photographies de paysages sauvages en niveaux de gris de taille 512×512 . Ces images ont été filtrées afin d'égaliser la densité spectrale de puissance sur les fréquences spatiales.

Nous avons découpé chaque image en 1024 imagerettes de taille 16. Une fois vectorisées, ce sont ces imagerettes qui servent de données d'apprentissage. Pour chaque image, on a

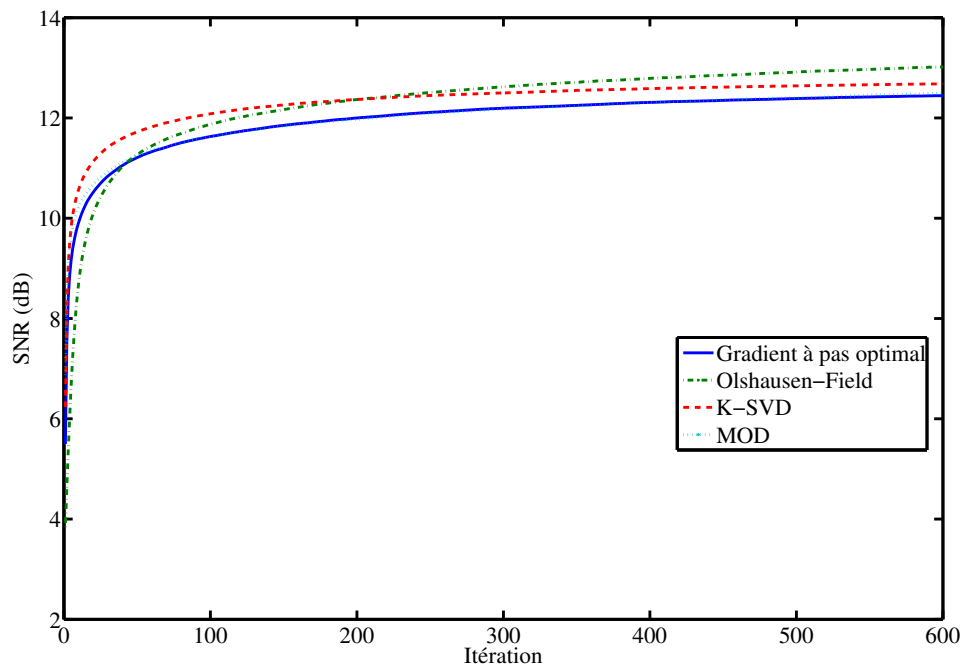


(a) Gauss

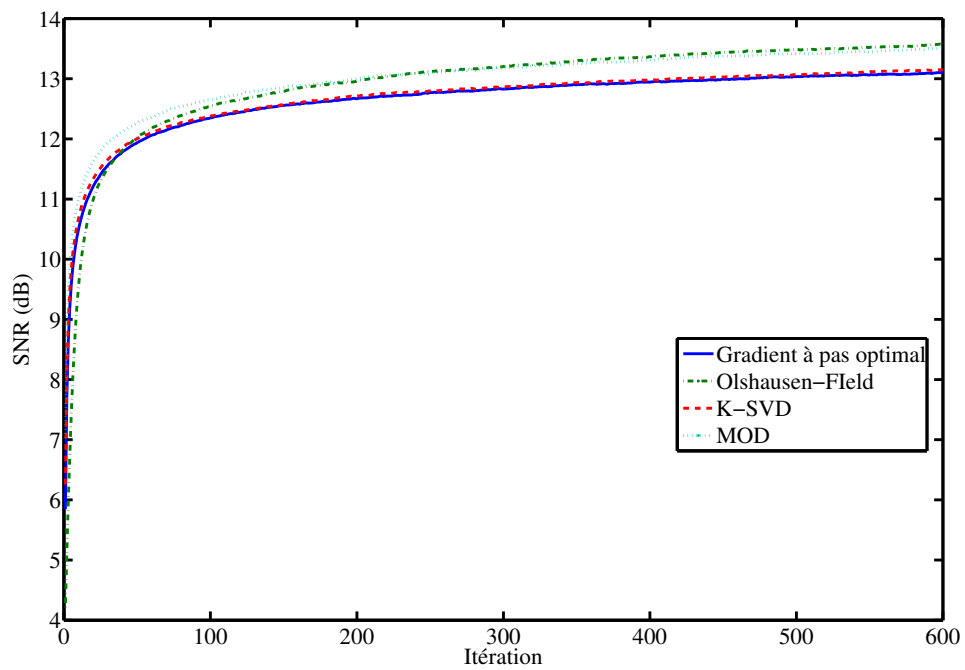


(b) Rademacher

FIG. 2.3 – SNR en fonction de l'itération pour les différents algorithmes à support de décomposition fixé. Les coefficients ont été tirés selon une loi gaussienne en haut et de Rademacher en bas. Tous les algorithmes atteignent parfois le minimum global et restent le plus souvent bloqués sur un minimum local sous-optimal. K-SVD atteint le minimum global dans plus de la moitié des cas pour les données de Rademacher.

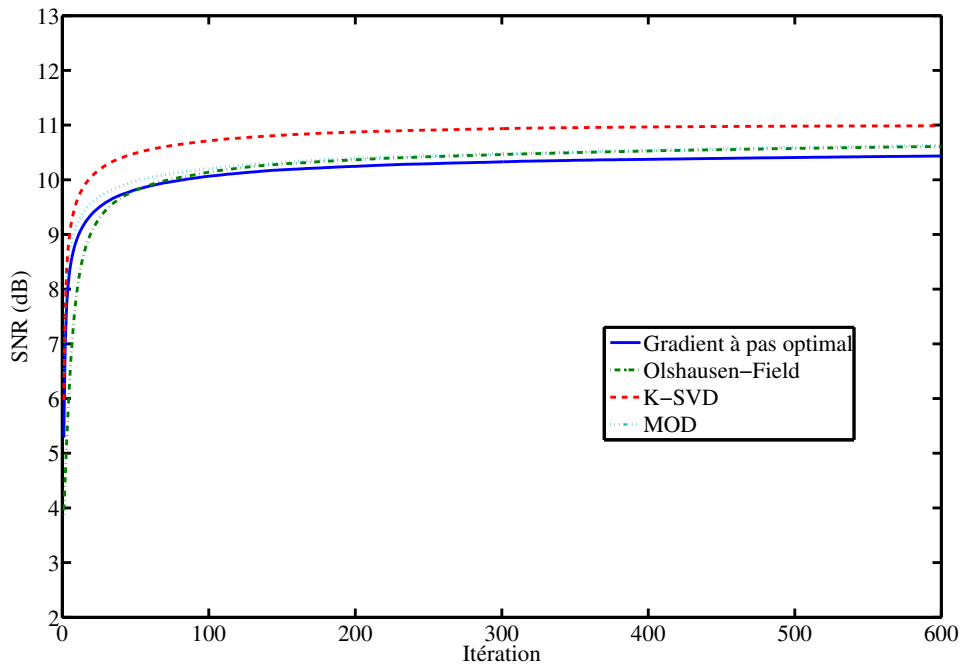


(a) MP

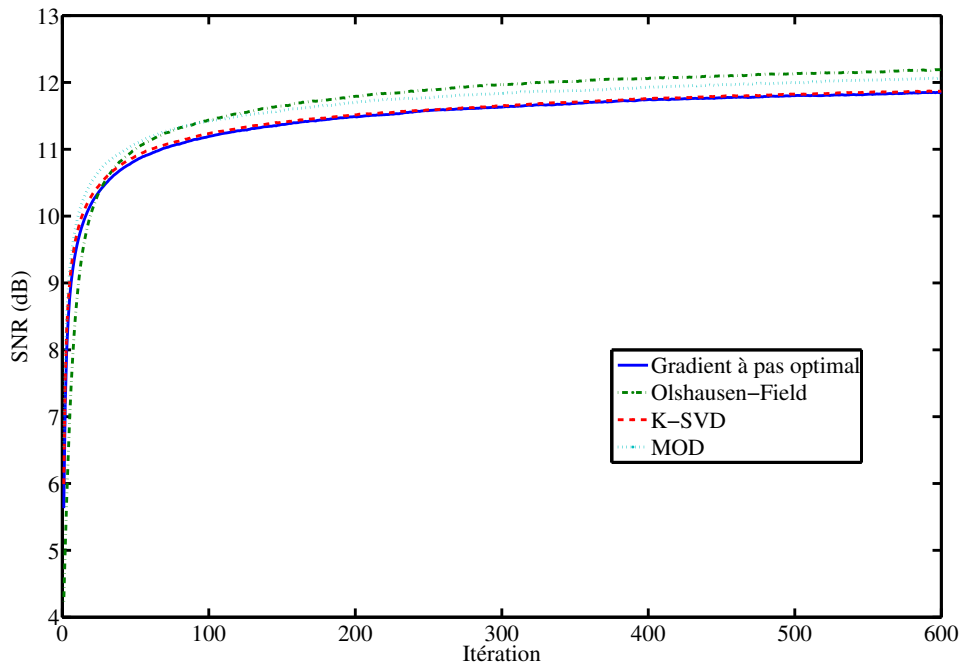


(b) OMP

FIG. 2.4 – SNR obtenu en fonction de l'itération par les différents algorithmes complets. Les coefficients suivent une loi gaussienne. On a utilisé MP ou OMP pour la décomposition.



(a) MP



(b) OMP

FIG. 2.5 – SNR obtenu en fonction de l'itération par les différents algorithmes complets. Les coefficients suivent une loi de Rademacher. On a utilisé MP ou OMP pour la décomposition.

donc $S = 1024$ signaux d'apprentissage de taille $N = 256$. On apprend sur ces données un dictionnaire de $D = 512$ atomes. Chaque image est décomposée sur $K = 16$ atomes.

Le dictionnaire a été appris avec les algorithmes d'Olshausen et Field, K-SVD, MOD et notre descente de gradient à pas optimal. Nous avons utilisé MP et OMP pour la décomposition.

La figure 2.6 montre le SNR atteint par chaque algorithme au cours de son exécution, en utilisant MP et OMP pour la décomposition. Notre algorithme et l'algorithme d'Olshausen et Field ont des comportements proches combinés avec MP. Par contre la descente de gradient à pas optimal profite plus de l'utilisation d'OMP que l'algorithme d'Olshausen et Field. Ce n'est pas surprenant : le calcul du pas optimal fait intervenir les coefficients de la décomposition qui sont donc plus utilisés que dans la descente à pas fixe.

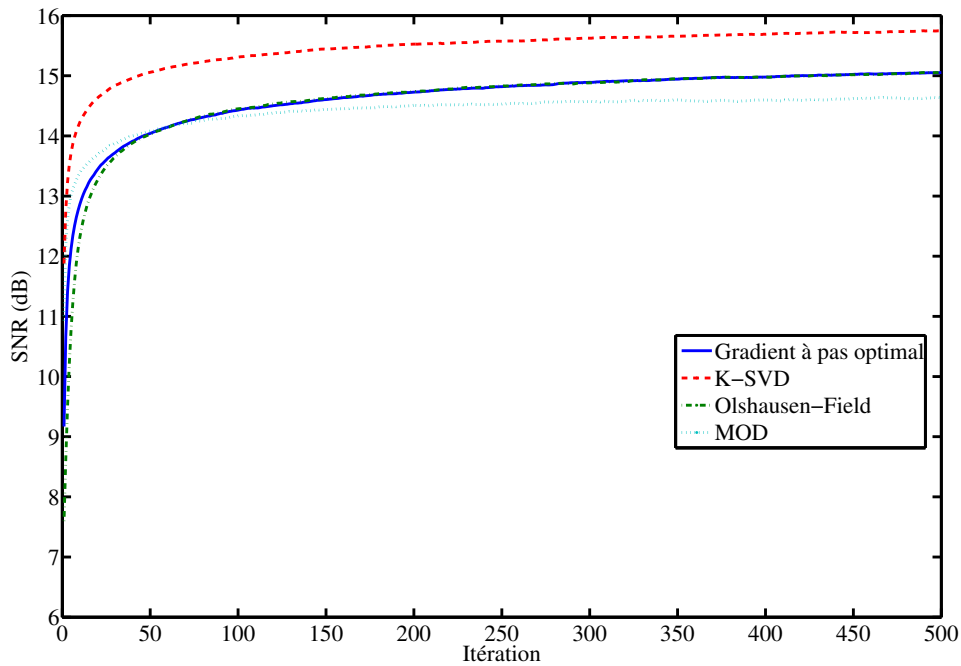
K-SVD permet une meilleure approximation des signaux, y compris quand on utilise OMP. Étonnamment, MOD présente des performances bien inférieures aux autres algorithmes, alors qu'on s'attendrait à ce qu'il dépasse au moins l'algorithme d'Olshausen et Field.

Ces résultats diffèrent de ceux observés sur des données synthétiques. Les propriétés des données semblent donc avoir une grande influence sur le comportement relatif des algorithmes. On ne peut pas en déclarer un meilleur que les autres sur tous types de signaux.

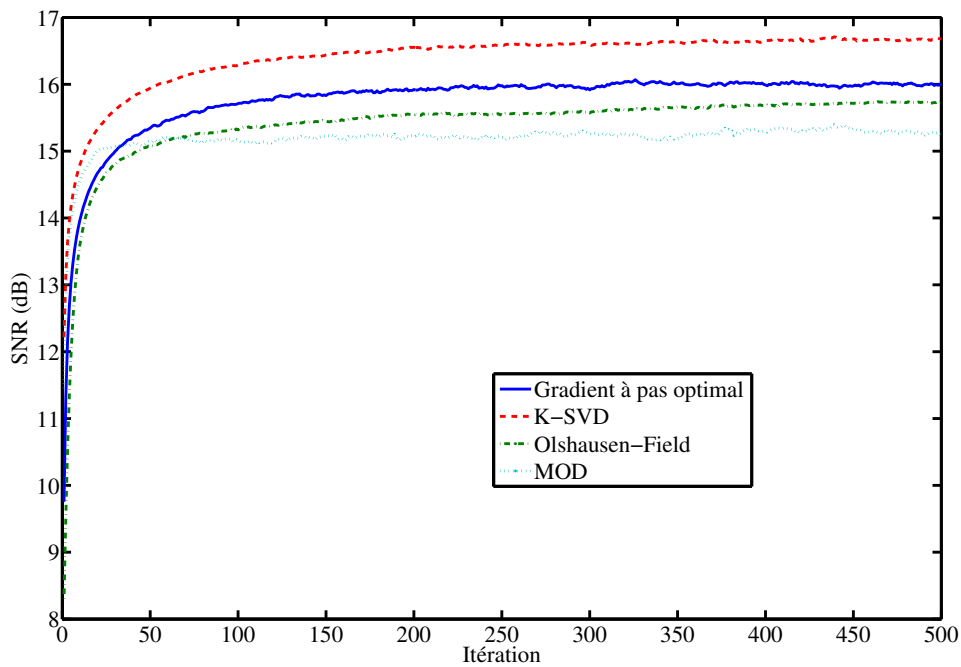
La figure 2.7 montre une des images originales, et son approximation sur les dictionnaires appris. On ne voit pas de différence significative entre les différents algorithmes. L'image semble bien approchée. Le résidu ne contient plus que des textures à très hautes fréquences, proches du bruit blanc. Les figures 2.8, 2.9, 2.10 et 2.11 montrent les 16 atomes de plus grande énergie des dictionnaires appris par les différents algorithmes sur cette image. On ne constate pas de différences importantes entre les dictionnaires. Ils sont composés de textures et d'atomes et de contours obliques montants. C'est cohérent avec l'image : à cause des strates du rocher au centre, l'image présente beaucoup de lignes dans la même direction. On voit également quelques atomes qui représentent un contour au bord de l'atome. C'est un effet de bord dû au filtre appliqué par Olshausen et Field sur les images : on peut voir une rupture sur les bords de l'image originale. Les dictionnaires appris par les autres algorithmes présentent les mêmes caractéristiques morphologiques.

Par rapport aux résultats obtenus par Olshausen et Field [OF97] ou Bell et Sejnowski [BS97], on trouve des atomes beaucoup plus texturés et moins d'éléments de contours. C'est dû au fait qu'on a appris plus d'atomes qu'eux sur moins de données. Olshausen et Field cherchaient des éléments caractéristiques des images naturelles. Ils ont donc appris une base sur toutes leurs images d'apprentissage. Contrairement à eux, nous avons appris un dictionnaire redondant sur une seule des images. On apprend donc des atomes plus spécialisés. Les dictionnaires que nous avons appris seraient sûrement mauvais pour représenter d'autres images.

Alors que le travail d'Olshausen et Field correspond à une étude fondamentale des propriétés d'une classe de signaux, notre approche se situe plutôt dans l'utilisation de l'apprentissage pour une application. Dans ce cas, on souhaite voir des atomes qui ne ressemblent pas à des formes analytiques connues. Si on intègre de l'apprentissage dans un système et que le résultat est une base DCT, on peut se dire qu'on ferait mieux d'imposer cette base dès le début plutôt que de perdre son temps à la réapprendre.

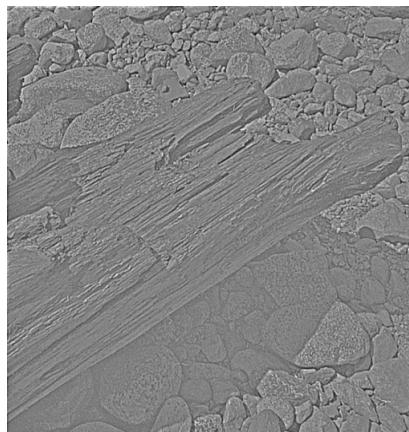


(a) MP

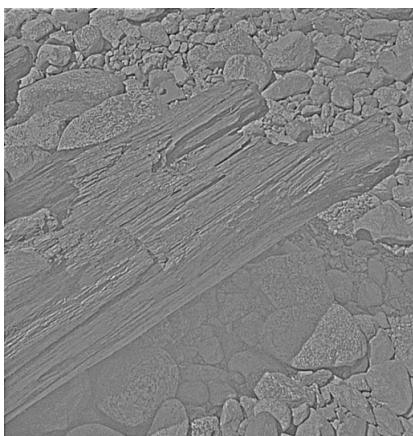


(b) OMP

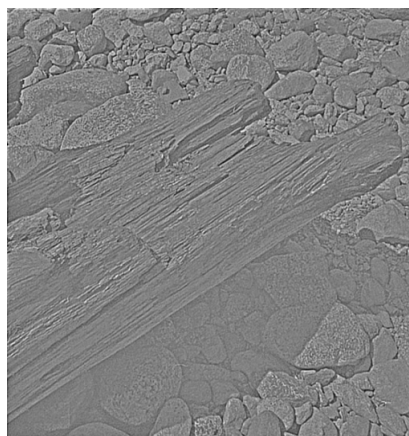
FIG. 2.6 – SNR moyen atteint par les différents algorithmes en fonction de l'itération sur les 10 images d'Olshausen et Field. La descente de gradient à pas optimal arrive fournir la deuxième meilleure approximation après K-SVD. MOD fonctionne étonnamment moins bien que l'algorithme d'Olshausen et Field.



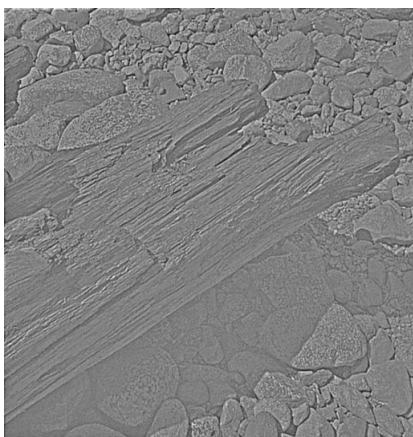
(a) Image originale



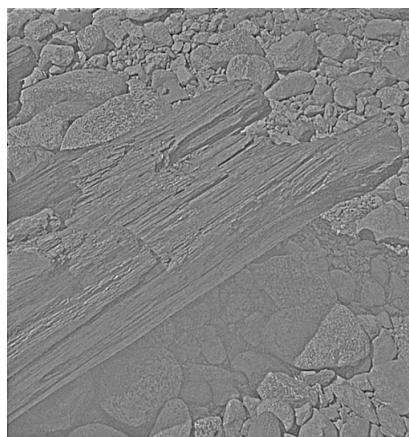
(b) Gradient à pas optimal



(c) Olshausen-Field



(d) K-SVD



(e) MOD

FIG. 2.7 – Première image de la base d’Olshausen et Field et son approximation sur des dictionnaires appris par les différents algorithmes.

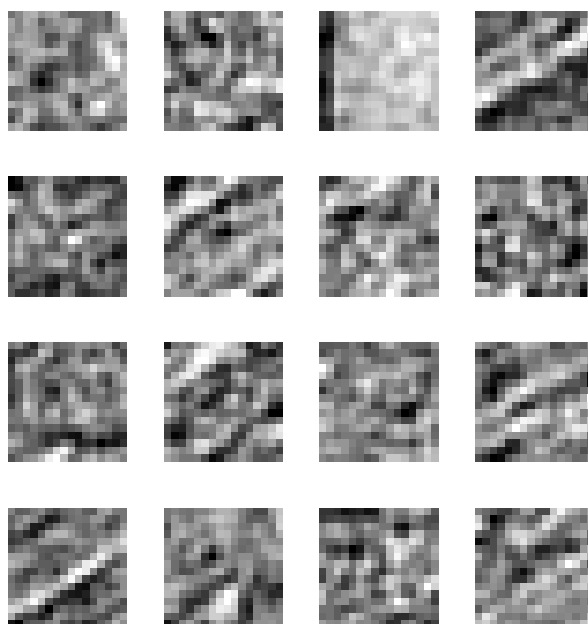


FIG. 2.8 – Dictionnaire appris sur l'image montrée à la figure 2.7 par descente de gradient à pas optimal.

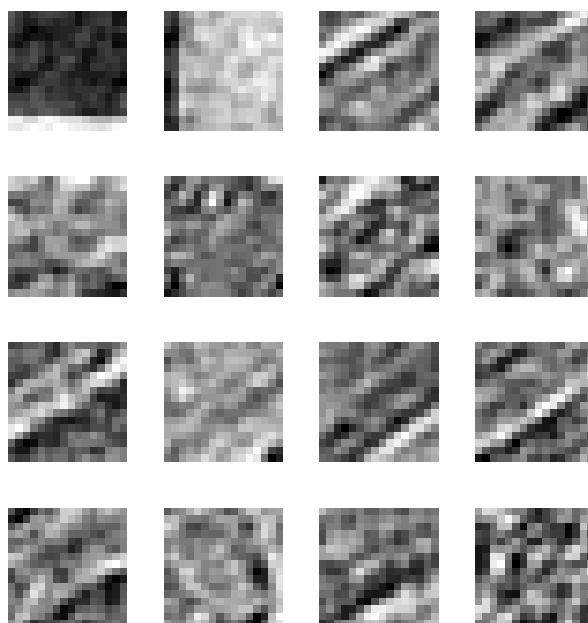


FIG. 2.9 – Dictionnaire appris sur l'image montrée à la figure 2.7 par l'algorithme d'Olshausen et Field.

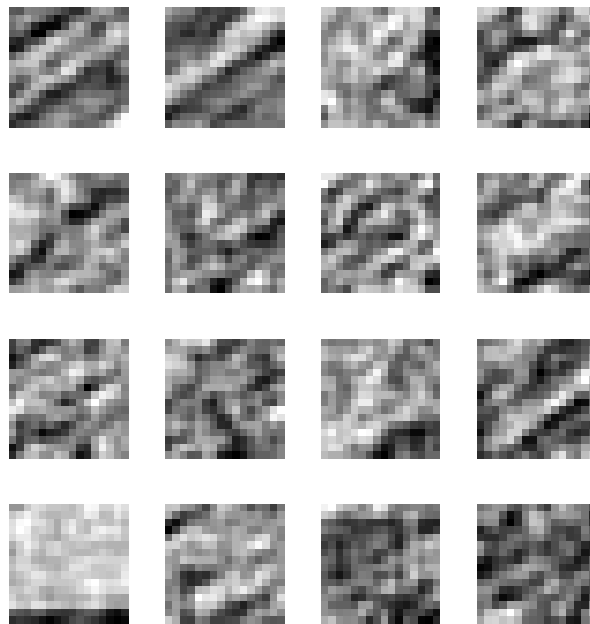


FIG. 2.10 – Dictionnaire appris sur l'image montrée à la figure 2.7 par l'algorithme K-SVD.

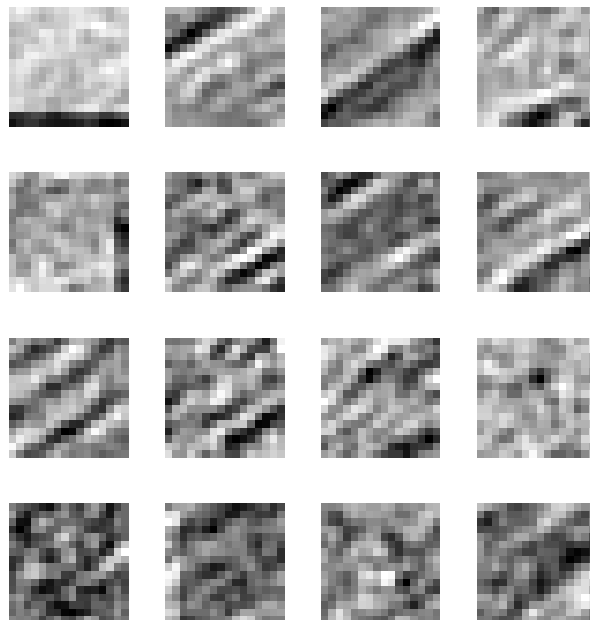


FIG. 2.11 – Dictionnaire appris sur l'image montrée à la figure 2.7 par l'algorithme MOD.

2.4 Apprentissage de dictionnaires invariants par translation

Nous avons vu au chapitre 1 que les signaux étudiés sont souvent constitués de répétitions d'événements de longueur L courte par rapport à durée N des signaux étudiés. Le dictionnaire Φ est alors une concaténation de matrices circulantes. On peut le représenter par un ensemble \mathcal{M} de *motifs* de dimension L et un ensemble \mathcal{T} de *translations* autorisées de dimension $N \times L$ qui prennent un motif de dimension L et le plongent dans l'espace signal de dimension N . L'atome $T_\tau m$ est l'atome nul partout sauf sur l'intervalle $[\tau, \tau + L - 1]$ qui contient le motif m . Le dictionnaire peut alors se factoriser en

$$\Phi = ((Tm)_{T \in \mathcal{T}})_{m \in \mathcal{M}} \quad (2.54)$$

Donc si on parvient à identifier \mathcal{M} et \mathcal{T} , on identifie Φ . De plus, l'ensemble \mathcal{T} est en général fixé (par exemple : toutes les translations d'un nombre entier d'échantillons). Le problème de l'apprentissage consiste alors à estimer \mathcal{M} .

Tous les algorithmes de l'état de l'art ont été étendus au cas invariant par translation : Olshausen et Field [BD06], MOD [SyA06][EAHH99] et K-SVD [Les07][MGVB08]

Soit s un signal d'apprentissage. On peut écrire sa décomposition sur un dictionnaire invariant par translation Φ :

$$s = \Phi x + r \quad (2.55)$$

$$= \sum_{m \in \mathcal{M}} \sum_{T \in \mathcal{T}} x_{T,m} Tm + r \quad (2.56)$$

$$\mathcal{E}(s, \Phi, x) = \left\| s - \sum_{m \in \mathcal{M}} \sum_{T \in \mathcal{T}} x_{T,m} Tm \right\|_2^2 \quad (2.57)$$

On remarque qu'un seul signal s peut contenir plusieurs occurrences de chaque motif m , donc peut suffire comme donnée d'apprentissage. Le problème d'apprentissage s'écrit alors :

$$(\hat{\mathcal{M}}, x) = \operatorname{argmin}_{\mathcal{M} \in \mathbb{R}^{L \times \alpha}, x \in \mathbb{R}^D} \left\| s - \sum_{m \in \mathcal{M}} \sum_{T \in \mathcal{T}} x_{T,m} Tm \right\|_2^2 \quad (2.58)$$

sous les contraintes

$$\forall m \in \mathcal{M}, \|m\|_2 = 1 \quad (2.59)$$

$$\|x\|_0 \leq K \quad (2.60)$$

A dictionnaire fixé, on retombe sur le problème des décomposition parcimonieuse sur un dictionnaire invariant par translation traité dans le chapitre 1.

Pour la mise à jour du dictionnaire, on cherche à minimiser la fonction $\mathcal{E}(s, \Phi, x)$ en fonction de chaque motif m_d séparément.

$$\hat{m}_d = \operatorname{argmin}_{m_d \in \mathbb{R}^L, \|m_d\|_2=1} \left\| s - \sum_{m \in \mathcal{M}} \sum_{T \in \mathcal{T}} x_{T,m} Tm \right\|_2^2 \quad (2.61)$$

$$= \operatorname{argmin}_{m_d \in \mathbb{R}^L, \|m_d\|_2=1} \left\| s - \sum_{d' \neq d} \sum_{T \in \mathcal{T}} x_{T,m_{d'}} Tm_{d'} - \sum_{t \in \mathcal{T}} x_{T,m_d} Tm_d \right\|_2^2 \quad (2.62)$$

On définit le résidu partiel $r_d = s - \sum_{m \neq m_d} \sum_{T \in \mathcal{T}} x_{T,m} Tm$.

Cas sans chevauchement Si les occurrences du motif m_d ne se superposent pas, on a alors

$$\hat{m}_d = \operatorname{argmin}_{m_d \in \mathbb{R}^L, \|m_d\|_2=1} \left\| r_d - \sum_{T \in \mathcal{T}} x_{T,m_d} T m_d \right\|_2^2 \quad (2.63)$$

$$= \sum_{T \in \mathcal{T}} \|T^* r_d - x_{T,m_d} m_d\|_2^2 + cte \quad (2.64)$$

T^* est l'opérateur adjoint de T . C'est un opérateur d'*extraction* qui prélève un intervalle de signal de longueur L . La constante représente alors l'énergie de r_d aux instants non-couverts par une occurrence du motif m_d . On retrouve alors une expression analogue à l'équation (2.28), ce qui donne la solution :

$$\hat{m}_d = \frac{\sum_{T \in \mathcal{T}} x_{T,m_d} T^* r_d}{\left\| \sum_{T \in \mathcal{T}} x_{T,m_d} T^* r_d \right\|_2} \quad (2.65)$$

Cette solution est celle donnée la règle (2.35) en utilisant pour extraits les $T^* r_d$.

Cas général Dans le cas général, plusieurs occurrences d'un même motif peuvent se superposer, donc le calcul précédent n'est pas valide. Néanmoins il a été démontré [BD06] que le gradient de la fonction de coût par rapport à un motif m_d s'écrit

$$\nabla_d \mathcal{E}(s, \Phi, x) = \sum_{T \in \mathcal{T}} x_{T,m_d} T^* r \quad (2.66)$$

L'analogie développée dans le paragraphe 2.2.3 étant toujours valable, on peut quand même utiliser la règle (2.35) en tant que descente de gradient. Le pas de cette descente n'est plus optimal, mais on a au moins une estimation de son ordre de grandeur.

2.5 Apprentissage de dictionnaires multicanaux

L'apprentissage de dictionnaires peut être étendu aux signaux multicanaux. Nous en aurons besoin dans la partie 3.

Un signal multicanal est une collection de C signaux S de même durée. Intuitivement, chaque canal est l'enregistrement d'un capteur, et tous les capteurs observent la même scène simultanément. Si les phénomènes observés sont parcimonieux, alors chaque canal doit être parcimonieux.

L'apprentissage de dictionnaires peut se généraliser simplement pour 2 modèles simples : un modèle sans aucune déformation des atomes entre les canaux autre qu'un gain, et l'autre avec des déformations quelconques.

2.5.1 Mélange sans déformation

Dans ce modèle, on considère que tous les atomes du dictionnaire ont la même forme sur chaque canal. On peut donc décomposer tous les canaux sur le même dictionnaire Φ :

$$S = \Phi X + R \quad (2.67)$$

On retrouve ici l'expression de la décomposition des signaux d'apprentissage. La seule différence est que maintenant on souhaite imposer une structure de parcimonie jointe aux coefficients X : X doit comporter peu de lignes avec un coefficient non-nul.

Soit $\|X_d\|_{interc}$ le vecteur composé des normes de toutes les lignes de X , pour une norme quelconque. Le problème d'apprentissage du dictionnaire se formule :

$$\left(\hat{\Phi}, \hat{X}\right) = \operatorname{argmin}_{\Phi \in \mathbf{R}^{N \times D}, X \in \mathbf{R}^{D \times S}} \|S - \Phi X\|_{FRO}^2 \quad (2.68)$$

sous les contraintes

$$\| \|X_d\|_{interc} \|_0 \leq K \quad (2.69)$$

$$\forall 1 \leq d \leq D, \|\varphi_d\|_2 = 1 \quad (2.70)$$

A dictionnaire fixé, on retombe sur un problème connu de décomposition parcimonieuse multicanale. Plusieurs méthodes ont été proposées pour adapter les algorithmes de décomposition existants à ce formalisme [GRSV08][CREKD05].

A coefficients fixés, on n'a plus à se préoccuper de la contrainte de parcimonie. Le problème devient alors exactement équivalent au problème original (2.9). **Si la forme des atomes est la même sur tous les canaux, alors les différents canaux se comportent comme autant de données d'apprentissage indépendantes lors de la mise à jour du dictionnaire.**

Néanmoins, si on souhaite apprendre un dictionnaire redondant (ou au moins complet), il faut quand même disposer de plusieurs exemples multicanaux pour l'apprentissage. En effet, comme les décompositions sont parcimonieuses, tous les atomes du dictionnaire ne peuvent pas être présents dans une seule décomposition. Avec un seul signal multicanal d'apprentissage, les solutions au problème (2.68) sont toutes de la même formes : les K atomes utilisés dans la décomposition forment une base du sous-espace des K premières composantes principales des canaux, les autres atomes sont quelconques.

2.5.2 Mélange avec déformations quelconques

Supposons maintenant qu'il n'y a aucun lien connu entre les formes d'un atome sur les différents canaux. C'est le modèle le plus général qu'on puisse envisager. Dans ce cas, chaque atome est une matrice de taille $N \times C$ et le dictionnaire Φ devient un tenseur d'ordre 3 de taille $N \times C \times D$. Le problème d'apprentissage s'écrit alors :

$$\left(\hat{\Phi}, \hat{X}\right) = \operatorname{argmin}_{\Phi \in \mathbf{R}^{N \times D}, X \in \mathbf{R}^{D \times S}} \sum_{c=1}^C \|S_c - \Phi_c X_c\|_2^2 \quad (2.71)$$

sous les contraintes

$$\| \|X_d\|_{interc} \|_0 \leq K \quad (2.72)$$

$$\forall 1 \leq d \leq D, \forall 1 \leq c \leq C \|\varphi_{d,c}\|_2 = 1 \quad (2.73)$$

La décomposition multicanale s'effectue de la même manière, que les atomes aient la même forme sur chaque canal ou pas. Lors de la mise à jour du dictionnaire, le problème (2.71) est séparé en C problèmes indépendants

$$\hat{\Phi}_c = \operatorname{argmin}_{\{\Phi_c \in \mathbf{R}^{N \times D}\}} \|S_c - \Phi_c X_c\|_2^2 \quad (2.74)$$

sous les contraintes

$$\forall 1 \leq d \leq D, \|\varphi_{d,c}\|_2 = 1 \quad (2.75)$$

On peut donc appliquer les règles de mise à jour monocanales pour optimiser le dictionnaire indépendamment sur chaque canal. **Sans hypothèse sur la déformation intercanale des atomes, les dictionnaires sont mis à jour indépendamment sur chaque canal.** En particulier, cette structure peut permettre une implantation parallèle de la mise à jour. C'est ce modèle qui sera utilisé dans la partie 3.

2.6 Conclusion

Contributions Nous avons proposé une extension de l'algorithme d'Olshausen et Field qui rend l'algorithme plus simple à utiliser en estimant automatiquement un de ses paramètres. Nous avons également étendu cet algorithme aux cas multicanaux et invariants par translation. On ne peut pas dire qu'un algorithme se comporte mieux que les autres : on obtient des résultats différents entre les données synthétiques et réelles, et à l'intérieur des données synthétiques selon la loi de tirage des coefficients.

Néanmoins on peut retenir un certain nombre de bonnes pratiques d'apprentissage :

- K-SVD est plus indiqué quand la norme des coefficients non-nuls varie peu ;
- il est peu utile de combiner K-SVD avec un algorithme de décomposition parcimonieuse orthogonal ;
- inversement, MOD fonctionne nettement mieux quand on le combine avec un algorithme de décomposition orthogonal.

Idées noires Mais la leçon la plus importante à retenir est sans doute que le problème de l'apprentissage de dictionnaires est encore très loin d'être résolu. L'approche alternée comporte 3 difficultés :

- calculer une décomposition parcimonieuse
- optimiser le dictionnaire
- l'interaction entre les deux étapes

Le problème de la décomposition parcimonieuse est largement étudié. L'approche alternée peut profiter de toutes les avancées effectuées dans ce domaine.

On sait par l'expérience qu'on a de l'algorithme K-means en clustering que l'approche alternée elle-même n'est pas optimale. La comparaison est justifiée : quand $K = 1$, l'apprentissage de dictionnaire se réduit à un problème de clustering. Si une approche n'est pas optimale dans un cas particulier, elle ne peut pas l'être dans le cas général. Même si on maîtrisait parfaitement les deux étapes de l'algorithme, leur itération ne calculerait toujours pas le meilleur dictionnaire possible.

L'expérience 2.3.3.2 a mis en évidence que même la mise à jour du dictionnaire est loin d'être maîtrisée. Même lorsqu'on connaît le bon support de décomposition, si on n'a pas également l'amplitude des coefficients non-nuls, on n'arrive pas à retrouver le dictionnaire. Les expériences montrent également que le SNR reste bloqué autour de 50dB si un l'algorithme ne converge pas vers le dictionnaire optimal. Il y a peut-être un résultat à en tirer sur la profondeur du bassin d'attraction de la solution optimale. Un résultat du type : si on atteint un certain SNR, alors on sait qu'on va converger vers un dictionnaire optimal. De tels résultats existent pour K-means [Mei06]. Pour arriver à dépasser les algorithmes existants, il faudra sans doute explorer des méthodes d'optimisation non-monotone pour la mise à jour du dictionnaire.

Chapitre 3

Séparation de signaux cardiaques par apprentissage de dictionnaires

3.1 Introduction : la fibrillation auriculaire

3.1.1 Phénomène clinique

La fibrillation auriculaire est l'anomalie de rythme cardiaque la plus répandue. Elle correspond à la contraction désordonnée des oreillettes.

La figure 3.1 décrit le fonctionnement du cœur. Le sang entre le cœur par les oreillettes, puis passe dans les ventricules et ressort. Dans le cas d'un patient sain, (également nommé en *rythme sinusal*), les oreillettes se contractent un peu avant les ventricules pour finir de leur transférer le sang qu'elles contiennent, ce qui permet aux ventricules d'injecter plus de sang dans le système circulatoire lorsqu'ils se contractent. Pour le patient en fibrillation, les oreillettes se contractent de manière erratique et n'assurent plus leur fonction.

Les symptômes ne sont pas critiques (palpitations, fatigue, douleurs de poitrine), mais la fibrillation a tendance à devenir chronique, augmente les risques d'accident vasculaire cérébral et peut engendrer des complications dangereuses qui peuvent aller jusqu'à l'arrêt cardiaque.

La fibrillation auriculaire est souvent causée par un foyer de cellules malades sur la membrane auriculaire. En plus de traitements médicamenteux ou à base de chocs électriques

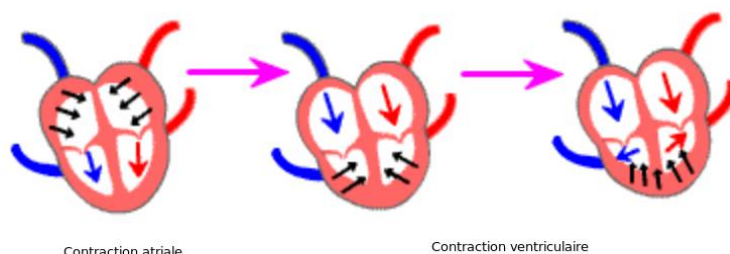


FIG. 3.1 – Fonctionnement d'une systole cardiaque. Les atria se contractent en premier pour expulser le sang qu'elles contiennent dans les ventricules. Ensuite ceux-ci se contractent pour propulser le sang hors du cœur. Chez le patient en fibrillation atriale, la contraction des atria n'a pas lieu.

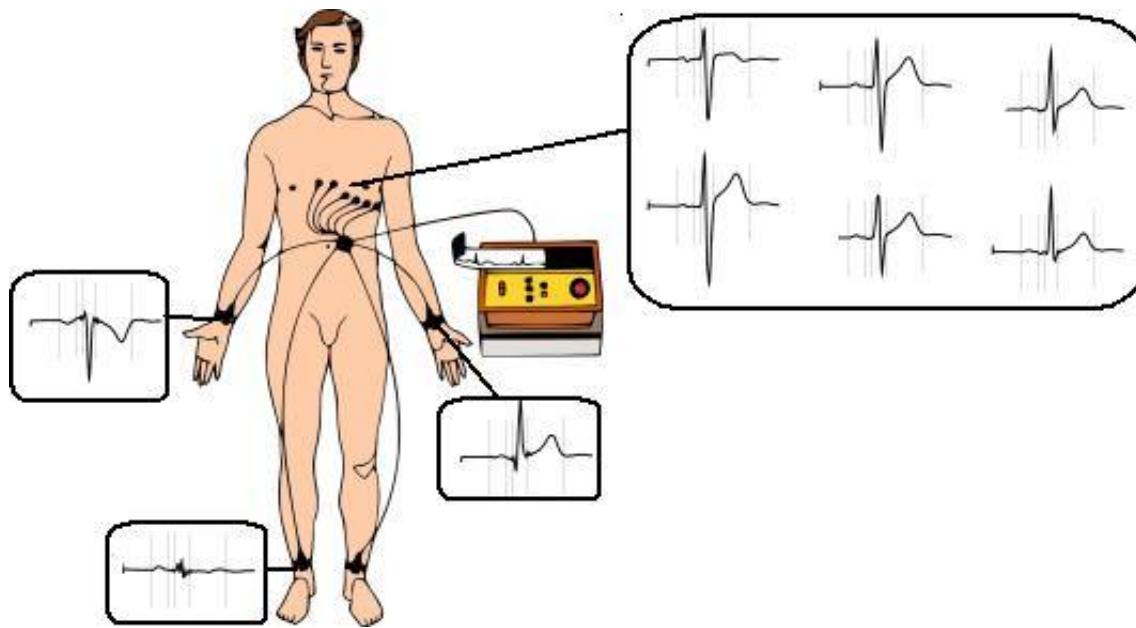


FIG. 3.2 – Placement de électrodes pour un électrocardiogramme à 12 canaux. On ne place que 9 électrodes [KGB⁺07]. Les 3 voies restantes sont des différences de potentiel entre membres. On voit que le signal est plus faible au niveau du pied.

destinés à réduire ou encadrer la fibrillation, les médecins s'intéressent depuis la fin des années 1990 à des méthodes chirurgicales visant à agir directement au niveau du foyer. Ils ont pour cela besoin d'outils de diagnostic avancés pour localiser le foyer et identifier son activité.

3.1.2 Acquisition des électrocardiogrammes

L'électrocardiographie (ECG) est un moyen non-invasif d'observer l'activité cardiaque, dont la fibrillation atriale. La contraction cardiaque est causée par une activité électrique des cellules membranaires. C'est ce qui permet l'utilisation de défibrillateurs et de stimulateurs cardiaques. L'ECG consiste à mesurer cette activité au moyen d'électrodes placées sur la peau du patient. On peut utiliser des électrodes placées sur les membres (*voie frontale*) ou sur la cage thoracique (*voie précordiale*). Le placement des électrodes est normalisé pour faciliter les comparaisons entre différents ECG [KGB⁺07]. Les protocoles de placement des électrodes ont été standardisés pour extraire 6 voies frontales et 6 voies précordiales. La figure 3.2 montre le placement des électrodes. Néanmoins, des dépendances linéaires existent entre certaines de ces voies, ce qui permet d'en supprimer 3 sans perdre d'information. De plus le signal mesuré au pied sert au calcul de la masse électrique, mais il semble moins intéressant en lui-même. Les signaux auxquels nous nous intéressons sont donc des signaux à 8 canaux : les 2 bras aVR et aVL et les 6 voies précordiales V1 à V6.

3.1.3 Problème : extraction du signal de fibrillation

Soit S un signal d'ECG multicanal de taille $N \times 8$ (un canal par colonne). S est un mélange de signaux induits par les activités atriale A et ventriculaire V :

$$S = A + V \quad (3.1)$$

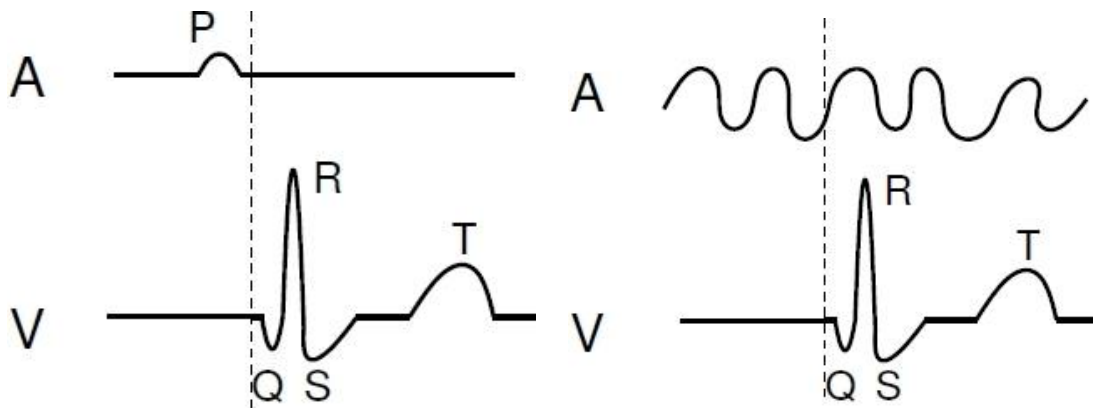


FIG. 3.3 – Trace des activités atriale et ventriculaire dans l'ECG. Chez le patient, la contraction de l'atria provoque l'onde P, suivie par le complexe QRS à la contraction des ventriculaire et l'onde T à leur relaxation. En cas de fibrillation, l'onde P disparaît pour être remplacée par des oscillation erratiques et indépendants du rythme cardiaque maintenu par les ventricules.

La figure 3.3 montre la forme attendue pour les différentes sources. En rythme sinusal, on peut observer l'onde P qui correspond à la contraction des oreillettes avant le complexe QRS dû à la contraction des ventricules. En cas de fibrillation, l'onde P est remplacée par une oscillation plus ou moins régulière et décorrélée de l'activité ventriculaire. Dans tous les cas, l'activité atriale est d'amplitude bien plus faible que l'activité ventriculaire, ce qui complique le diagnostic : avant d'étudier le signal atrial, il faut être capable de l'observer, ce qui nécessite de supprimer l'activité ventriculaire du signal S .

Le problème qu'on cherche à résoudre est donc l'estimation du signal atrial A à partir du signal observé $S = A + V$. Sous cette formulation, c'est un problème de débruitage. On connaît plusieurs propriétés caractéristiques des signaux pour nous aider :

- l'énergie ventriculaire est bien plus importante que l'énergie atriale,

$$\|A\|_{FRO} \ll \|V\|_{FRO} \quad (3.2)$$

- durant la fibrillation, les activités atriales et ventriculaires sont indépendantes [FRA⁺01],
- on sait que le signal ventriculaire V est composé d'une succession de complexes QRST, et les battements successifs d'un même patient ont des formes proches [Hur98],
- le signal de fibrillation est composé d'oscillations de spectre variable, avec un pic de fréquence à environ 300Hz.

3.2 État de l'art

Diverses approches ont été tentées pour la suppression de l'activité ventriculaire [ZNB97] [LVIK04] [LVvO⁺07] [DEGL⁺06]. On peut distinguer 2 grands types d'approches. D'une part, l'indépendance des activités atriale et ventriculaire ainsi que la mesure par de multiples électrodes permettent d'utiliser des méthodes de séparation de sources dans le cas surdéterminé, comme l'analyse en composantes indépendantes.

D'autre part, les différences morphologiques existant entre les 2 sources permettent d'envisager des méthodes issues du débruitage ou de la séparation de sources monocanale. Notre travail se situe dans ce second cadre. A l'intérieur de ce cadre, on peut distinguer des approches orientées débruitage qui n'exploitent qu'un modèle ventriculaire et des approches orientées séparation de sources qui utilisent des modèles pour les 2 sources.

3.2.1 Analyse en composantes indépendantes (ICA)

L'ICA est une famille qui regroupe plusieurs approches exploitant l'indépendance des sources pour les séparer [HKO01].

Séparation de sources aveugle On suppose que le signal observé S est le résultat de l'application d'un opérateur de mélange M appliqué à une matrice de sources Y (chaque colonne représente une source). On se restreindra ici au cas linéaire instantané; dans ce cas, M est une matrice.

$$S = YM \quad (3.3)$$

La séparation de sources consiste à estimer les sources Y à partir des observations S et d'hypothèses sur la distribution des sources.

Ce problème est légèrement différent de la suppression de l'activité ventriculaire décrite dans l'introduction de cette partie. En effet, en suppression on cherche des sources multicanales A et V (avec autant de colonnes que S). La séparation de sources demande en plus de factoriser chaque source multicanale comme le produit d'une ligne de l'opérateur de mélange M par une source monocanale. Si on arrive à séparer les sources, on peut estimer la matrice de mélange M par projection orthogonale de S sur Y et synthétiser les sources multicanales. Néanmoins nous allons voir que c'est l'inverse qui se produit pour l'ICA : pour estimer les sources, on estime la matrice de mélange M et son inverse M^{-1} .

Principe de l'ICA Supposons que le rang de la matrice de mélange M est au moins égal au nombre de sources. Dans ce cas, si on parvient à estimer M , alors on peut calculer sa pseudo-inverse $M^+ = M^*(MM^*)^{-1}$ et on a

$$Y = SM^+ \quad (3.4)$$

Pour l'ICA, on suppose que les sources Y sont indépendantes 2 à 2. L'idée est alors d'estimer M comme la matrice qui maximise l'indépendance des colonnes de SM^+ .

En mathématiques, l'indépendance est définie comme un critère binaire : deux variables aléatoires sont indépendantes où elles ne le sont pas. Pour obtenir un algorithme pratique d'ICA, on doit néanmoins en effectuer une mesure continue à optimiser, c'est-à-dire définir une fonction de SM^+ qui atteint sa valeur minimale quand les sources sont indépendantes. Plusieurs fonctions de coût ont été proposées, comme les cumulants croisés d'ordre strictement supérieur à 2 [JH91] qui sont nuls pour des variables indépendantes, et des mesures de non-gaussianité des sources [HKO01] : le théorème central-limite nous dit que la moyenne des sources converge vers une distribution gaussienne, donc les sources séparées doivent être moins gaussiennes que leur mélange.

Application aux signaux d'ECG Dans le domaine cardiaque, l'ICA a notamment été employée pour séparer les ECG de la mère et d'embryon [ZNB97]. Différents algorithmes d'ICA a également été utilisée pour l'identification de l'activité atriale [RCS⁺04][LVIK04].

On peut penser que l'ICA est plus adaptée à l'extraction de l'ECG fœtal qu'à l'extraction de l'activité ventriculaire. La différence entre les 2 cas est la localisation physique des sources dans le corps. Dans le premier cas, on cherche à séparer 2 cœurs distants de plusieurs dizaines de cm. Dans le second, on reste à l'intérieur d'un même cœur et les sources ne sont séparées que de quelques cm. On peut donc espérer que la matrice de mélange est mieux conditionnée dans le premier cas. D'autre part, l'indépendance même des sources n'est pas forcément assurée : selon la morphologie cardiaque du patient, l'activité électrique atriale peut se propager de façon limitée à la membrane ventriculaire [FRA⁺01].

La comparaison avec d'autres méthodes de l'état de l'art, notamment celles qui seront présentées à la section 3.2.2, semble montrer que l'ICA est moins efficace [LRS⁺06]. On sait qu'il existe une zone sans activité ventriculaire entre la fin de l'onde T et le début du complexe QRS suivant. On sait détecter automatiquement ces zones, même en cas de fibrillation. On peut alors comparer l'amplitude de l'activité atriale estimée dans les zones sans activité ventriculaire et dans les zones QRST. Si l'amplitude est plus importante dans les zones QRST, alors la différence est expliquée par l'activité ventriculaire résiduelle qu'on n'a pas réussi à supprimer. On constate qu'il reste plus d'activité ventriculaire résiduelle dans l'activité atriale estimée par ICA [LRS⁺06].

3.2.2 Soustraction de battement typique (ABS)

Ces approches apprennent des formes typiques de l'activité ventriculaire du patient [SS01]. Ensuite, lors de chaque battement, l'activité ventriculaire est supprimée par soustraction du battement typique le plus corrélé au signal S .

Principe A partir d'un signal d'ECG S , on commence par effectuer une détection de pics pour repérer les instants $(t_b)_{1 \leq b \leq B}$ de début de chaque onde Q, ce qui permet de segmenter S en une famille de trames $\mathcal{S} = (S_b)_{1 \leq b \leq B}$ avec $S_b = S[t_b, t_{b+1}]$ de longueur $L_b = t_{b+1} - t_b$. Chaque trame contient un battement. Ensuite, on apprend les motifs typiques par quantification vectorielle des battements.

Un modèle de dilatation temporelle a également été proposé pour les atomes [LVvO⁺07]. Ce modèle traite différemment les complexes QRS, considérés de durée constante, et les ondes T qui peuvent se dilater.

Limites Les limites de cette approche sont de 2 ordres. Tout d'abord, dans l'intervalle QRS, l'amplitude de l'activité ventriculaire dépasse très nettement celle de l'activité atriale. Donc une erreur d'approximation faible rapport à l'énergie du QRS peut être importante par rapport à la fibrillation qu'on souhaite observer. Ces erreurs apparaissent en général comme des pics dans la fibrillation estimée \hat{A} . Il a été proposé de contourner ce problème en estimant la fibrillation par soustraction hors des intervalles QRS, puis de synthétiser les intervalles manquants par un modèle sinusoïdal de la fibrillation [LVvO⁺07].

Ensuite, on constate que ces méthodes ne permettent d'apprendre qu'un seul battement typique. Si on en apprend plusieurs, on observe que les battements typiques comportent une composante oscillante en plus de la forme QRST attendue, signe de fibrillation.

3.2.3 Méthodes basées sur la parcimonie

Nous avons vu à la section 3.2.1 qu'on peut utiliser l'indépendance des sources pour les séparer. Une autre approche plus récente consiste à utiliser une hypothèse de parcimonie sur les sources. L'intérêt principal de ces méthodes est de pouvoir traiter des mélanges sous-déterminés, c'est-à-dire où le rang de la matrice de mélange M est inférieur au nombre de sources à estimer.

Degenerate Unmixing Estimation Technique (DUET) La méthode DUET, mise au point à l'origine pour le traitement de la parole, permet en théorie de séparer un nombre quelconque de source à partir de seulement 2 observations [JRY00]. Elle est basée sur une transformée temps-fréquence des sources S , une transformée de Fourier à court terme par exemple. L'hypothèse centrale, nommée orthogonalité W-disjointe, est qu'une seule source

est active dans chaque point temps-fréquence. Pour un mélange linéaire instantané, on calcule alors le rapport d'énergie entre les canaux pour chaque point temps-fréquence. Ce rapport ne dépend que de la matrice de mélange M et est constant pour chaque source. Donc les rapports pour tous les points temps-fréquence doivent se regrouper autour d'une valeur par source.

Un algorithme de clustering permet d'identifier ces points d'accumulation et de classer les points temps-fréquence entre les différentes sources. Il suffit ensuite de reconstruire les sources en inversant la transformée.

L'orthogonalité W -disjointe implique une hypothèse de parcimonie. En effet, si il n'y a qu'une seule source active par point temps-fréquence et qu'il y a P sources pour un signal de dimension N , alors chaque source n'est présente en moyenne que sur au plus $\frac{P}{N}$ points.

Analyse en composantes morphologiques (MCA) MCA est un principe de séparation de sources qui utilise une hypothèse de parcimonie avec dictionnaires connus pour séparer des sources à partir d'une seule observation [SED04]. Supposons que :

- chaque source y_p est parcimonieuse sur un dictionnaire Φ_p ,
- pour tout $q \neq p$, le source y_p n'est pas parcimonieuse sur le dictionnaire Φ_q .

Alors on peut séparer les sources et l'algorithme de séparation est le suivant :

- décomposer la source s sur le dictionnaire concaténé $\Phi = (\Phi_p)_{1 \leq p \leq P}$:

$$s = (\Phi_1 \dots \Phi_P) \begin{pmatrix} x_1 \\ \vdots \\ x_P \end{pmatrix} + r \quad (3.5)$$

- reconstituer chaque source avec les atomes de son dictionnaire :

$$\forall 1 \leq p \leq P, \hat{y}_p = \Phi_p x_p \quad (3.6)$$

On peut voir DUET comme un cas particulier de MCA. En effet, si on considère pour chaque source y_p le dictionnaire Φ_p des atomes temps-fréquence pour lequel cette source est active, alors ces dictionnaires suivent les hypothèses de MCA. La connaissance de 2 canaux dans DUET permet de ne connaître que le dictionnaire Φ et d'estimer automatiquement son découpage en Φ_p . MCA ne nécessite qu'un seul canal, mais en contrepartie on doit connaître les dictionnaires de chaque source. La difficulté est d'alors de trouver des dictionnaires permettant une bonne séparation.

Application aux signaux cardiaques L'approche MCA a été utilisée pour la suppression de l'activité ventriculaire [DEGL⁺06]. Les dictionnaires choisis étaient un dictionnaire de courbes gaussiennes pour l'activité ventriculaire (des cloches), et des atomes de Gabor pour l'activité atriale. Les résultats sont encourageants mais les auteurs notent néanmoins que les dictionnaires pourraient être mieux adaptés aux données. C'est notamment le cas pour l'onde T qui est en général asymétrique, donc mal modélisable par une cloche symétrique. Cette approche semble rester moins efficace que la soustraction de battement typique.

Apprentissage pour la séparation de sources Pour obtenir des dictionnaires plus discriminants, on peut vouloir les apprendre directement sur les données. Quand on ne possède pas de données d'apprentissage pour chaque source isolée, il faut apprendre directement à partir du mélange. Le problème est alors d'apprendre 2 modèles distincts à partir des mêmes données.

Les approches utilisées jusqu'à présent utilisent une certaine parcimonie dans le domaine temporel (ou spatial pour des images) : toutes les sources ne sont pas actives tout le temps. Par exemple, pour la séparation d'images en contours et textures, on peut sélectionner manuellement des extraits de texture qui ne contiennent pas de contour [PFS09]. On pourrait alors se demander comment détecter automatiquement ces bons extraits d'apprentissage.

Un autre exemple est la séparation de sources audiovisuelles [CMVG08]. On filme plusieurs sources sonores distinctes, par exemple deux personnes qui parlent. On enregistre simultanément ce qu'elles disent sur un seul canal, et on souhaite séparer les pistes sonores. L'idée de la séparation est de commencer par exploiter une détection du mouvement et une segmentation spatiale de la vidéo pour identifier les zones de l'image appartenant à chaque locuteur. On détecte ensuite l'activité de chaque locuteur dans le temps, et on apprend le modèle sonore de chaque locuteur sur les moments où il parle seul. L'exploitation de la vidéo permet de trouver les données d'apprentissage utilisables dans le signal.

Malheureusement cette méthode ne semble pas pouvoir s'appliquer à l'étude de la fibrillation atriale. Il existe bien une partie sans activité ventriculaire entre l'onde T et le complexe QRS suivant, mais ces extraits sont courts. D'autre part, la fibrillation peut être épisodique ou permanente selon les patients, mais si elle s'arrête elle est remplacée par l'activité ventriculaire saine corrélée avec l'activité ventriculaire. Donc en apprenant le signal ventriculaire sur des instants sans fibrillation, on n'apprendrait pas les bons atomes.

3.2.4 Conclusion

On peut distinguer 3 familles de méthodes pour la suppression de l'activité ventriculaire : l'analyse en composantes indépendantes (ICA), la soustraction de battements typiques (ABS) et les méthodes basées sur la parcimonie. Contrairement à l'ICA, les deux dernières méthodes sont monocanales : on pourrait les appliquer sur une seule observation. Si on dispose de plusieurs canaux, on peut néanmoins les utiliser pour renforcer les tâches de détection, comme on l'a vu par exemple à la section 1.9.

La différence essentielle entre l'ABS et la MCA est que l'ABS n'utilise aucun modèle de fibrillation. D'autre part, les deux approches ont leurs points faibles. L'ABS ne permet pas d'apprendre plusieurs battements typiques et la MCA nécessite des données d'apprentissage dont on ne dispose pas. L'approche que nous allons décrire dans la section suivante emprunte aux deux méthodes pour résoudre tous ces problèmes. Elle peut être vue comme l'ajout d'un modèle atrial à l'ABS pour apprendre plus de battements typiques ou comme un schéma d'apprentissage sur le mélange pour effectuer une MCA.

3.3 Analyse des limites de l'ABS

Cette section propose une interprétation de l'existence d'une composante de fibrillation dans les battements typiques en termes de niveau d'énergie. Cette interprétation permet de penser qu'on pourrait éliminer cette composante par l'ajout d'un modèle de fibrillation invariant par translation.

Puisqu'on souffre d'une faible tolérance aux erreurs d'approximation, on peut aussi être tenté d'apprendre plus de motifs différents pour affiner le modèle et réduire l'erreur. Malheureusement le nombre de battements typiques qu'on peut apprendre est limité par la présence de la fibrillation. Pour simplifier les notations, on considère dans la suite un seul canal. Un signal multicanal pourrait être traité de manière analogue en le ramenant

au préalable à un seul canal, par concaténation des canaux par exemple. Posons

$$\forall S_b \in \mathcal{S}, S_b = A_b + V_b \quad (3.7)$$

La quantification vectorielle des battements S_b consiste à chercher les battements typiques $(\phi_d)_{1 \leq d \leq D}$ qui minimisent l'erreur totale de quantification \mathcal{E} définie par

$$\mathcal{E} = \sum_{b=1}^B \min_{1 \leq d \leq D} \|S_b - \phi_d\|_2^2 \quad (3.8)$$

3.3.1 Un seul battement typique

Si on apprend un seul motif m , alors il est donné par l'expression

$$m = \frac{1}{B} \sum_{b=1}^B S_b \quad (3.9)$$

$$= \frac{1}{B} \sum_{b=1}^B A_b + \frac{1}{B} \sum_{b=1}^I V_b \quad (3.10)$$

On suppose les sources A et V indépendantes. Comme on a choisi le tramage pour maximiser la corrélation des trames ventriculaires V_b entre elles, les trames A_b sont décorrélées donc $\frac{1}{B} \sum_{b=1}^B A_b \rightarrow_{B \rightarrow \infty} E(A)$ avec $E(A)$ l'espérance de A . Si on suppose que la loi de A est centrée, la moyenne des A_b tend vers 0 quand I devient grand. m est donc un battement ventriculaire typique, bien qu'il ait été appris sur les extraits de mélange S_b . *L'opération de moyenne sur les battements fait disparaître la composante atriale.*

3.3.2 Deux battements typiques

Si on souhaite apprendre 2 battements typiques, alors on sait que les solutions s'expriment comme combinaison linéaire de la moyenne m et de la composante principale δ des battements centrés $R_b = S_b - m$ [DH04]. Soit P la matrice dont les colonnes sont les battements centrés, P_A la matrice dont les colonnes sont les A_b et P_V la matrice dont les colonnes sont les $V_b - m$. On a alors

$$\delta = \operatorname{argmax}_{\|\delta\|_2=1} \delta^* P P^* \delta \quad (3.11)$$

$$= \operatorname{argmax}_{\|\delta\|_2=1} \delta^* (P_A + P_V)(P_A + P_V)^* \delta \quad (3.12)$$

$$= \operatorname{argmax}_{\|\delta\|_2=1} \delta^* (P_A P_A^* + P_V P_V^* + P_A P_V^* + P_V P_A^*) \delta \quad (3.13)$$

Si on suppose que les lois marginales des éléments de P_A et P_V sont centrées en 0 et indépendantes, les termes croisés deviennent négligeables quand I devient grand. On a alors

$$\delta = \operatorname{argmax}_{\|\delta\|_2=1} \delta^* (P_A P_A^* + P_V P_V^*) \delta \quad (3.14)$$

δ est donc un compromis entre la composante principale atriale et la composante principale ventriculaire. Pour pouvoir apprendre 2 battements ventriculaires typiques à partir des mélanges S_i , il faudrait que le terme ventriculaire soit dominant dans la fonction de coût (3.14), donc que la variabilité ventriculaire soit plus importante ou plus concentrée sur une dimension. On observe empiriquement que ce n'est pas toujours le cas : l'activité ventriculaire est certes bien plus importante que l'activité atriale, mais une bonne partie de son énergie a déjà été capturée par la moyenne m . On peut alors voir apparaître

des motifs composés d'une partie ventriculaire et de parties fibrillatoires en opposition de phase. Inversement, si c'était l'énergie atriale qui dominait, on pourrait supprimer cette composante.

En pratique, les contributions atriale et ventriculaire semblent toutes les deux significatives. C'est le plus mauvais cas, puisque le motif qu'on apprend est lui-même un mélange. Néanmoins on dispose d'un moyen simple d'augmenter l'influence de l'activité atriale et de diminuer celle de l'activité ventriculaire : changer le tramage.

Le choix de trames alignées sur les complexes QRS maximise la corrélation des trames ventriculaires V_b et décorrèle les trames atriales A_b . En remplaçant ce tramage par un autre, on ne peut qu'augmenter l'énergie de la composante principale atriale au détriment de la ventriculaire. L'idéal serait de savoir choisir un tramage qui maximise la corrélation des A_b . La méthode qui sera présentée à la section suivante se base sur l'invariance par translation : on cherche des trames S_b corrélées, et on espère que cette corrélation soit due autant que possible aux A_b .

3.4 Apprentissage de dictionnaire pour la suppression de l'activité ventriculaire

Cette section détaille le schéma d'apprentissage développé pendant cette thèse qui tente d'apprendre plusieurs atomes ventriculaires dépourvus de composante fibrillaire.

3.4.1 Modèle génératif

On se place dans le cadre multicanal décrit à la section 2.5.2, avec un dictionnaire invariant par translation. Un motif générateur M_d est une matrice de taille $L \times 8$ dont chaque colonne $m_{c,d}$ est de norme euclidienne 1. Le dictionnaire est composé de tous les atomes de forme $T_\tau M_d$ avec T_τ l'opérateur de translation d'un motif à l'instant τ . Un signal multicanal S se décompose parcimonieusement sur l'ensemble des motifs $\mathcal{M} = (M_d)_{1 \leq d \leq D}$ si il existe des coefficients $x_{c,d,\tau}$ tels que

$$\forall 1 \leq c \leq 8, S_c = \sum_{d=1}^D \sum_{\tau=1}^T x_{c,d,\tau} T_\tau m_{c,d} + R_c \quad (3.15)$$

sous les contraintes

$$\|R\|_{FRO} \ll \|S\|_{FRO} \quad (3.16)$$

$$\|x\|_{2,0} \ll N \quad (3.17)$$

avec $\|x\|_{2,0}$ la mesure de parcimonie des coefficients calculée comme ceci :

- pour chaque motif m_d à chaque instant τ , calculer la norme l_2 du vecteur $x_{:,d,\tau}$
- calculer la norme l_0 de ces normes l_2 sur tous les instants τ et les motifs m_d .

Cette mesure compte le nombre d'atomes qui ont un coefficient non-nul sur au moins un canal.

Le but de notre approche est, à partir d'une ECG S , d'apprendre deux ensembles de motifs \mathcal{M}_A et \mathcal{M}_V qui représentent respectivement les composantes atriale et ventriculaire de S . Pour cela, on alterne des étapes d'apprentissage et de suppression afin d'apprendre le dictionnaire associé à chaque source sur un signal où l'autre source a été atténuée.

3.4.2 Principe

Le principe de notre méthode s'inspire de l'analyse effectuée à la section 3.3. On peut apprendre 1 motif ventriculaire m_V en segmentant les battements S_b de façon synchrone aux complexes QRS puis en en prenant une moyenne. Les résidus R_b présentent alors une composante fibrillatoire non-négligeable, ce qui empêche d'apprendre plusieurs motifs ventriculaires. Néanmoins la segmentation en battements déphase la fibrillation aléatoirement, elle est donc particulièrement mauvaise pour apprendre un motif fibrillatoire. Si on remplace cette segmentation par une recherche non-supervisée de tranches de résidu mutuellement corrélées, on sélectionne des composantes fibrillatoires plus en phase, donc on augmente l'importance de la fibrillation par rapport au résidu ventriculaire dans le motif appris. Or c'est exactement ce que fait un algorithme d'apprentissage de dictionnaire invariant par translation. On peut donc tenter d'apprendre un motif atrial m_A sur le résidu R par apprentissage invariant par translation.

L'apprentissage de dictionnaire ne calcule pas uniquement un dictionnaire, mais aussi une décomposition des données d'apprentissage (ici le résidu R) sur le dictionnaire appris (ici motif m_A). donc une première estimation de l'activité atriale. Si on soustrait cette estimation de l'ECG original S , on obtient alors un signal dont l'activité atriale a été atténuée. On peut alors espérer que l'activité ventriculaire est suffisamment dominante pour apprendre plusieurs motifs ventriculaires.

3.4.3 Description du système

Architecture La généralisation du principe ci-dessus fournit un algorithme itératif d'apprentissage et de séparation. Le fonctionnement d'une itération est résumé par la figure 3.4 et l'algorithme 6 décrit le fonctionnement complet.

Algorithm 6 $(\tilde{A}, \tilde{V}) = \text{separation}(S)$

```

 $R = S$ 
 $\tilde{A} = 0$ 
 $\tilde{V} = 0$ 
for  $i = 1$  à  $I$  do
   $(\Phi_V, X_V) = \text{apprentissage}(S - \tilde{A}, i)$ 
   $\tilde{V} = \Phi_V X_V$ 
   $(\Phi_A, X_A) = \text{apprentissage}(S - \tilde{V}, i)$ 
   $\tilde{A} = \Phi_A X_A$ 
end for
return  $(\tilde{A}, \tilde{V})$ 

```

Initialement, on $\tilde{A} = \tilde{V} = 0$ et $R = S$. On apprend un dictionnaire ventriculaire sur le signal $S - \tilde{A} = \tilde{V} + R$. Cet apprentissage met également à jour l'estimation \tilde{V} de l'activité ventriculaire et le résidu R . On apprend ensuite un dictionnaire atrial sur le signal $S - \tilde{V} = \tilde{A} + R$, qui met à jour l'activité atriale estimée \tilde{A} et le résidu R . On itère le procédé en augmentant le nombre de motifs des dictionnaires : à l'itération i , on apprend i motifs différents par source.

Apprentissage des dictionnaires Pour chaque dictionnaire, l'algorithme d'apprentissage est un algorithme d'apprentissage multicanal et invariant par translation décrit aux paragraphes 2.4 et 2.5.2. On alterne :

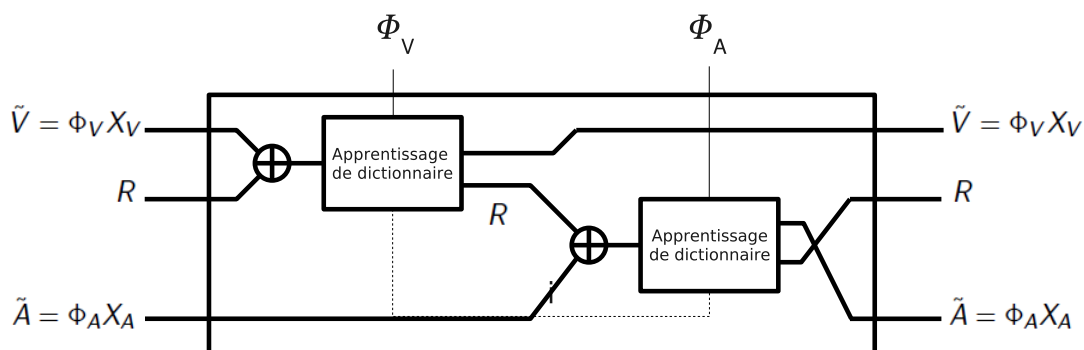


FIG. 3.4 – Fonctionnement d'une itération de séparation.

- une décomposition parcimonieuse multicanale invariante par translation, effectuée avec MP (qui sera remplacé par LocOMP quand on disposera d'une implémentation en MPTK),
 - une mise à jour du dictionnaire par canal. La mise à jour invariante par translation est effectuée par descente de gradient à pas adaptatif comme décrit à la section 2.4.
- Contrairement à la méthode de soustraction de battement typique, ici l'apprentissage du dictionnaire ventriculaire est également invariant par translation. En réglant le critère de parcimonie pour utiliser 1 atome par battement, ceux-ci se placent automatiquement autour de complexes QRS.

Initialisations Contrairement au cas général de l'apprentissage de dictionnaires, on dispose ici de connaissances expertes qui permettent d'initialiser les algorithmes d'apprentissage avec des atomes déjà proches de la solution qu'on cherche.

Pour le dictionnaire ventriculaire, on peut initialiser les instants τ de la décomposition à l'aide du même détecteur de pics utilisé pour localiser les complexes QRS pour la soustraction de battement typique [IvOH06]. Quand on souhaite apprendre plusieurs motifs, on peut ensuite répartir les battements aléatoirement entre les différents motifs.

Pour le dictionnaire atrial, on sait qu'on cherche des atomes oscillants. Après la suppression ventriculaire, on calcule une transformée de Fourier à court terme du signal d'apprentissage atrial $S - \tilde{V}$ qui permet de sélectionner les fréquences dominantes. Le dictionnaire initial est composé d'atomes de MDCT aux i fréquences de plus forte énergie. Initialement ces atomes sont identiques sur tous les canaux.

Filtrage des atomes atriaux On n'a malheureusement pas de garantie que l'apprentissage invariant par translation permet de supprimer l'activité ventriculaire résiduelle des motifs atriaux appris \mathcal{M}_A . Les formes d'erreur ventriculaire les plus courantes sont des pics localisés à l'endroit du complexe QRS, alors que l'activité ventriculaire est plutôt oscillante.

Empiriquement, on observe souvent la présence d'un unique grand pic dans les motifs atriaux appris : c'est un reste de l'onde R. Pour le supprimer, on applique une compression de dynamique douce à chaque itération pendant l'apprentissage atrial, après la mise à jour du dictionnaire. On détecte le maximum $\max_{1 \leq t \leq L} \|m_{d,t,c}\|$ de chaque motif sur chaque canal, puis les passages à 0 qui l'entourent (ou éventuellement les extrémités du motif) et toute cette portion est contractée linéairement jusqu'à ce que le pic se retrouve au niveau du maximum observé hors de la zone compressée.

C'est une méthode ad hoc pour réduire un unique pic sans avoir à fixer de seuil et en évitant de trop modifier la forme du motif.

3.5 Résultats expérimentaux

Notre système a été testé sur des données synthétiques et sur des données réelles. Le problème de l'évaluation sur données réelles est qu'on ne dispose pas des sources originales. Il est donc difficile de proposer des critères quantitatifs d'évaluation.

C'est pour cela qu'on a également eu recours à des données synthétiques. Contrairement à ce qu'on peut faire en traitement de signaux musicaux par exemple, il est impossible ici de réaliser un mélange synthétique de sources naturelles.

3.5.1 Évaluation sur des données synthétiques

Synthèse des sources Un modèle 3D aux éléments finis de la membrane de l'atrium humain a été réalisé à partir d'images obtenues par Imagerie à Résonance Magnétique (IRM) [Jac04]. Ce modèle comprend notamment les point d'entrée des vaisseaux sanguins et les valves de passage vers les ventricules. La fibrillation intracardiaque a été générée en modifiant localement les propriétés électriques de la membrane puis en apportant une stimulation électrique rapide. 21 différents types de fibrillation ont été générés pour différents types de membranes et d'excitation. Les signaux de surface (mesurés par les électrodes) ont été générés en propageant le signaux intracardiaques dans un modèle de poitrine comprenant le cœur et les poumons [vOJss].

L'activité ventriculaire provient de l'ECG de patients sains, dans lesquels on a mis à 0 les ondes P. Les fibrillations simulées ont été mélangées avec les ECG de 4 patients sains, soit un total de 84 signaux de 5 minutes.

Mesures de qualité L'intérêt de disposer des sources originales est qu'on peut mesurer leur proximité avec les sources estimées. Nous avons utilisé un ensemble de mesures spécialement conçues pour l'évaluation et l'étude d'algorithmes de séparation de sources [VGF06]. Ces mesures sont plus fines qu'une simple mesure de distortion parce qu'elles permettent d'analyser la provenance des erreurs.

Le principe est de calculer pour chaque canal c la projection orthogonale des sources estimées (\tilde{V}_c, \tilde{A}_c) sur les sources originales (V_c, A_c) :

$$\tilde{V} = \alpha_c^{(V)} V_c + \beta_c^{(V)} A_c + \Gamma_c^{(V)} \quad (3.18)$$

$$\tilde{A} = \beta_c^{(A)} V_c + \alpha_c^{(A)} A_c + \Gamma_c^{(A)} \quad (3.19)$$

La décomposition serait parfaite si on avait $\alpha_c^{(V)} = \alpha_c^{(A)} = 1$, $\beta_c^{(V)} = \beta_c^{(A)} = 0$ et $\Gamma_c^{(V)} = \Gamma_c^{(A)} = 0$. De plus, les quantités β_c et Γ_c jouent des rôles différents. Les coefficients β_c montrent la présence d'*interférences* résiduelles : les sources n'ont pas été parfaitement séparées. Les vecteurs Γ_c sont des *artefacts* de calcul, des composantes introduites par l'algorithme alors qu'elles n'existaient dans aucune des sources.

Les mesures de qualité évaluent l'importance de ces 2 sources d'erreur. Elles sont définies ainsi :

– les rapports source à interférence (SIR),

$$SIR_V = -10 \log_1 0 \frac{\|\beta_c^{(V)} A_c\|_2^2}{\|\alpha_c^{(V)} V_c\|_2^2} \quad (3.20)$$

$$SIR_A = -10 \log_1 0 \frac{\|\beta_c^{(A)} V_c\|_2^2}{\|\alpha_c^{(A)} A_c\|_2^2} \quad (3.21)$$

– les rapports sources à artefacts (SAR),

$$SAR_V = -10 \log_1 0 \frac{\|\Gamma_c^{(V)}\|_2^2}{\|\alpha_c^{(V)} V_c + \beta_c^{(V)} A_c\|_2^2} \quad (3.22)$$

$$SAR_A = -10 \log_1 0 \frac{\|\Gamma_c^{(A)}\|_2^2}{\|\beta_c^{(A)} V_c + \alpha_c^{(A)} A_c\|_2^2} \quad (3.23)$$

– les rapports source à distortions (SDR)

$$SDR_V = -10 \log_1 0 \frac{\|\beta_c^{(V)} A_c + \Gamma_c^{(V)}\|_2^2}{\|\alpha_c^{(V)} V_c\|_2^2} \quad (3.24)$$

$$SDR_A = -10 \log_1 0 \frac{\|\beta_c^{(A)} V_c + \Gamma_c^{(A)}\|_2^2}{\|\alpha_c^{(A)} A_c\|_2^2} \quad (3.25)$$

Le SIR mesure la quantité d'interférences présente dans chaque source par rapport à la source recherchée. Le SAR mesure la quantité d'artefacts créés par l'algorithme par rapport à toutes les sources. Le SDR est une mesure globale de distortion qui indique la proximité de la source estimée avec la source recherchée.

Dans notre cas, l'objectif de l'application est de retrouver l'activité atriale. Le critère de performance principal est donc le SDR atrial, les autres critères nous aidant à analyser le comportement des algorithmes.

Résultats Le tableau 3.1 compare les performances moyennes du système d'apprentissage de dictionnaires et de l'ABS de Lemay [LVvO⁺07]. On constate que notre système présente un gain d'1 dB en moyenne par rapport à l'ABS. Plus le canal contient d'activité atriale, plus le gain est fort. Ce n'est pas surprenant : plus l'activité atriale est forte, plus on gagne à la modéliser.

L'apprentissage de dictionnaires améliore tous les critères par rapport à l'ABS, sauf le SIR ventriculaire. Cela signifie qu'il reste plus d'activité atriale résiduelle dans la source ventriculaire estimée \tilde{V} . On pourrait tenter d'y remédier en supprimant plus d'activité atriale lors de l'apprentissage atrial. On dispose de deux moyens pour cela : apprendre plus de motifs atriaux ou utiliser plus d'atomes dans la décomposition atriale. Néanmoins si ces paramètres sont trop élevés on risque d'obtenir l'effet inverse : le dictionnaire atrial contiendra plus d'activité ventriculaire résiduelle, ce qui devrait faire diminuer le SIR atrial

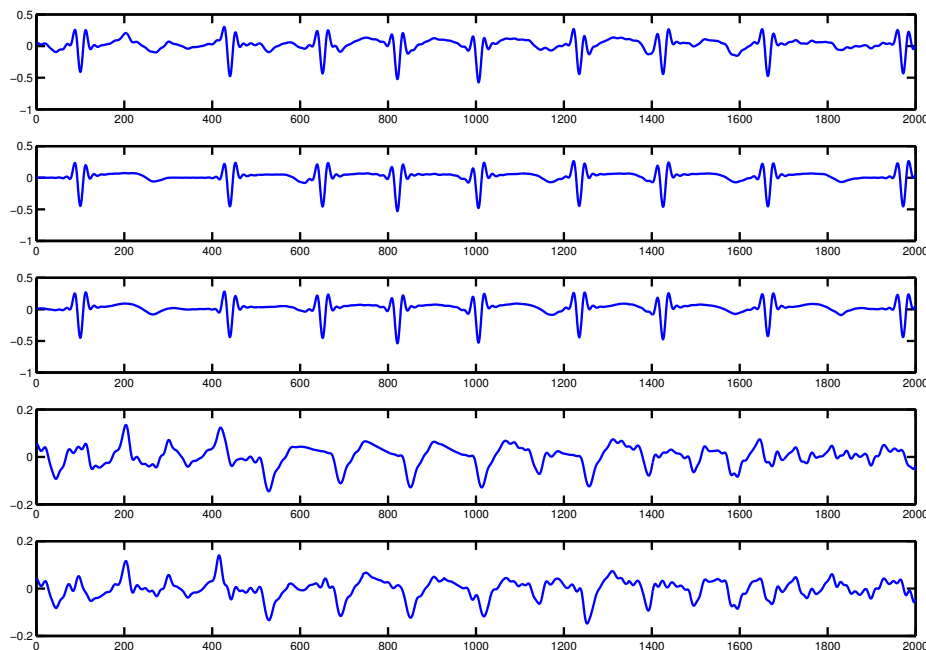


FIG. 3.5 – De haut en bas : le mélange synthétique S , les activités ventriculaires originale V et estimée \tilde{V} , les activités atriale originale A et estimée \tilde{A} , le tout sur la voie VR. L'estimation est bonne mais on peut voir des erreurs dans l'estimation atriale \tilde{A} au moment des complexes QRS.

et consisterait une erreur encore plus pénalisante pour l'application visée. On ne sait à l'heure actuelle pas estimer les paramètres qui permettent une séparation optimale. On pourrait également essayer de contraindre l'apprentissage ventriculaire comme on l'a fait à la section 3.4.3 pour l'apprentissage atrial.

La figure 3.5 montre le mélange, les sources et leurs estimations sur le canal VR. On peut voir des résidus ventriculaires dans l'estimation atriale \tilde{A} : ils forment de petites ondulations parasites à l'instant du QRS. À part ça, l'activité atriale semble bien estimée. La figure 3.6 montre le dictionnaire appris. Les atomes ventriculaires ont l'air semblables, mais il ne faut pas oublier qu'une faible différence entre eux peut avoir un grand impact sur l'activité atriale estimée.

3.5.2 Expérimentation sur données cliniques

Nous avons également testé l'apprentissage de dictionnaires sur les ECG de 184 patients souffrant de fibrillation atriale. On ne peut alors plus mesurer de proximité avec les sources originales, mais on peut toujours observer les résultats.

3.5.2.1 Caractéristiques des données cliniques

La figure 3.7 montre le canal V1 l'ECG d'un patient malade. On peut voir quelques pics de très forte amplitude.

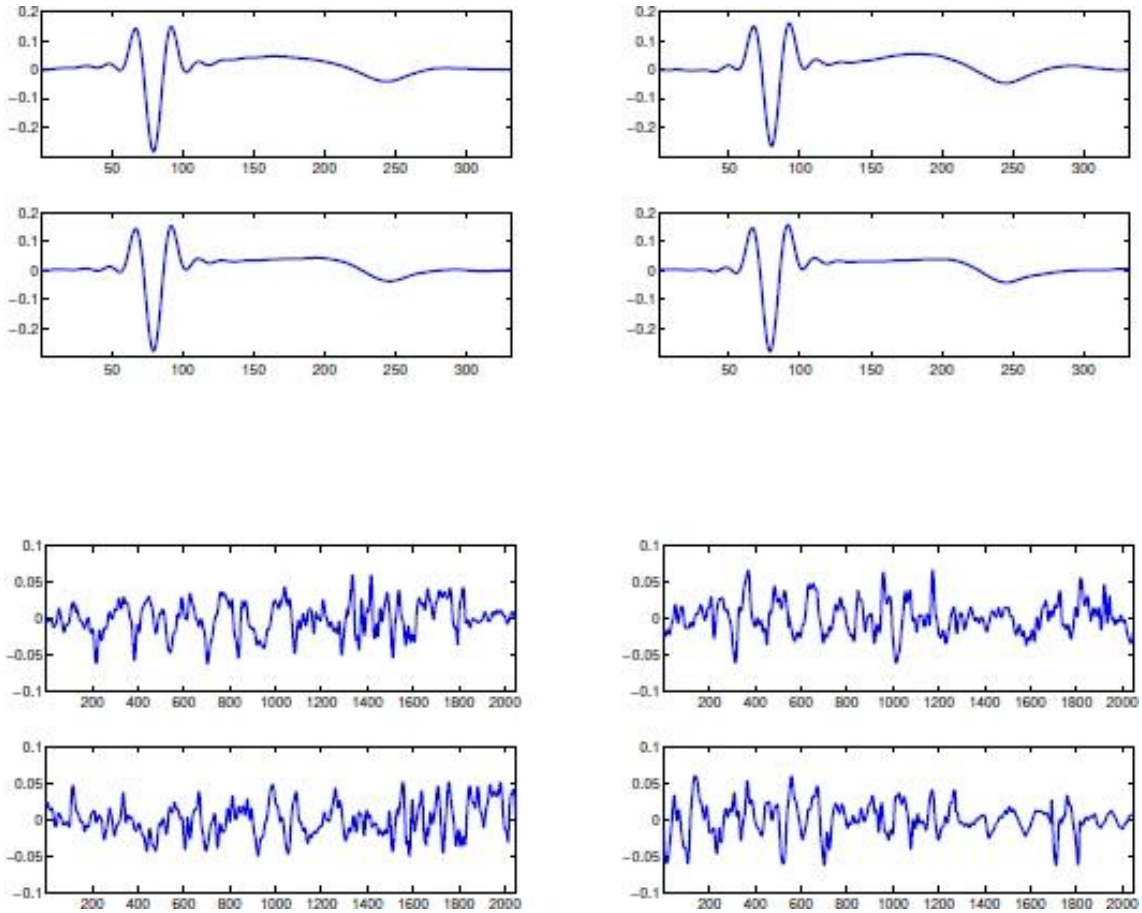


FIG. 3.6 – Dictionnaire appris sur un mélange synthétique. Le dictionnaire ventriculaire est en haut, le dictionnaire atrial en bas. On observe une forte ressemblance des atomes ventriculaires.

TAB. 3.1 – Comparaison des performance de séparation sur données synthétiques pour notre système et l’ABS. Tous les nombres sont en dB. On n’a noté ici que 3 canaux représentatifs et indiqué le rapport d’énergies original AVR entre les composantes atriales et ventriculaires dans le mélange. Le canal V1 présente l’activité atriale la plus fort de tous les canaux, le canal V4 la plus faible et le canal VR est au milieu.

Canal	VAR	dictionnaires			ABS		
		SDR	SIR	SAR	SDR	SIR	SAR
VR	V	15.6	24.1	16.7	15.1	24.3	16.1
	A	-12.3	1.2	23.0	1.4	-0.5	19.2
V1	V	16.4	23.3	17.7	16.8	24.6	17.9
	A	-11.7	3.0	28.4	3.1	1.5	27.9
V4	V	20.3	28.9	21.3	19.8	31.5	20.2
	A	-17.9	-1.4	22.2	-1.3	-1.9	21.1

Artefacts Certains pics correspondent à des artefacts. Même si ces ECG sont enregistrés sur des personnes au repos, il peut arriver au patient de bouger, d'éternuer, d'accrocher une électrode, ... Ces perturbations physiques entraînent d'importantes erreurs de mesure au niveau des électrodes. Ces erreurs sont de forte amplitude, mais durent assez peu de temps.

Palpitations D'autres anomalies proviennent d'extrasystoles, ou palpitations. Parfois les ventricules se contractent trop tôt, ce qui augmente la durée du QRS et modifie sa forme. Les extrasystoles peuvent survenir chez tout le monde, mais la fibrillation atriale favorise certains types de palpitations. Les palpitations sont mal corrigées par le premier apprentissage ventriculaire. Si elles sont trop nombreuses, elles risquent donc de perturber l'apprentissage atrial suivant. Sur les autres canaux, les extrasystoles présentent également une forme caractéristique mais leur amplitude ne dépasse pas celle des autres battements.

3.5.2.2 Analyse qualitative

En V1 La voie V1 est celle qui contient le plus de fibrillation. La figure 3.8 montre un extrait de l'ECG en V1 : on voit bien la fibrillation entre les QRS. Dans ce cas, la difficulté n'est pas tant de supprimer l'activité ventriculaire que d'éviter de supprimer la fibrillation avec.

La figure 3.9 montre le dictionnaire ventriculaire appris sur le canal V1. On reconnaît un troisième atome ventriculaire, et on remarque une forte oscillation dans le sixième atome. Les autres atomes ont l'air moins perturbés. C'est peut-être le signe qu'on a appris trop de motifs différents.

En V4 La voie V4 est celle qui contient le moins de fibrillation. Le signal est montré dans la figure 3.10, à comparer avec celui de la figure 3.8. Dans ce cas le risque est plutôt de ne pas réussir à supprimer assez d'activité ventriculaire pour pouvoir apprendre le dictionnaire atrial.

La figure 3.11 montre le dictionnaire atrial appris en V4. On n'y trouve pas de motif ventriculaire flagrant (pic isolé), ce qui est plutôt bon signe. C'est le filtrage des atomes atriaux qui a permis d'obtenir ce résultat.

La figure 3.12 montre l'activité atriale estimée \tilde{A} . La courbe du haut montre à la fois S et \tilde{A} , la courbe du bas ne montre que \tilde{A} en plus gros au même instant. On observe de grosses perturbations au moment du QRS, mais ces perturbations sont difficiles à identifier en tant que telles à la seule vue de l'estimation \tilde{A} . Elles deviennent flagrantes quand on observe l'activité ventriculaire simultanément et qu'on sait qu'elles arrivent au moment du QRS.

En VR La figure 3.13 montre le dictionnaire ventriculaire appris en VR. On voit un atome de palpitation, qui est bien le même que celui appris en V1 (il s'agit de la projection du même atome multicanal sur chaque voie), et le quatrième atome est un atome d'artefact. C'est plutôt chanceux : si on le retrouve dans le dictionnaire ventriculaire, c'est qu'il sera absent de la fibrillation estimée \tilde{A} . Malheureusement nous n'avons pas encore de moyen de l'assurer.

3.5.2.3 Analyse quantitative

Contrairement au cas des données synthétiques, on ne dispose pas ici des sources originales. On ne peut donc pas mesurer directement la qualité de la séparation. On peut

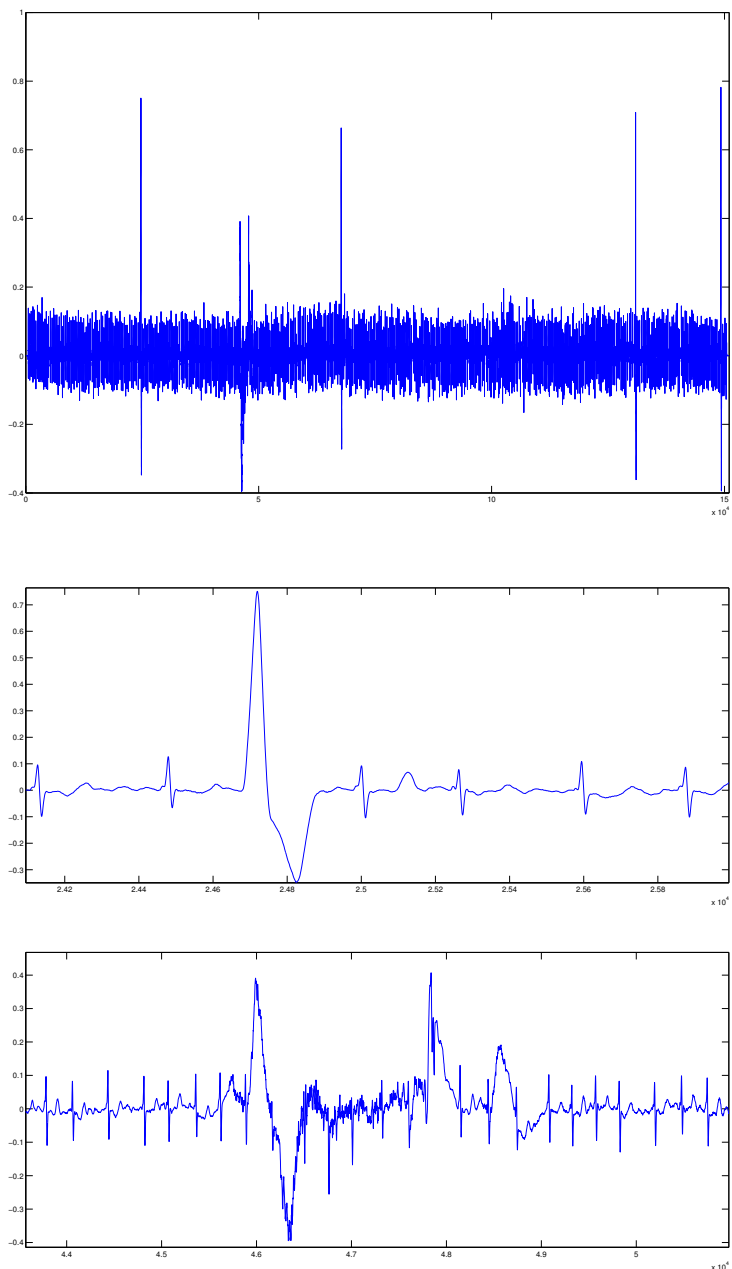


FIG. 3.7 – ECG d'un patient malade sur le canal VR. On voit de grands pics sur la vue d'ensemble (en haut). Le premier pic est causé par une palpitation (au milieu : vue en gros plan). Le second est dû à un artefact de mesure (en bas : vue en gros plan).

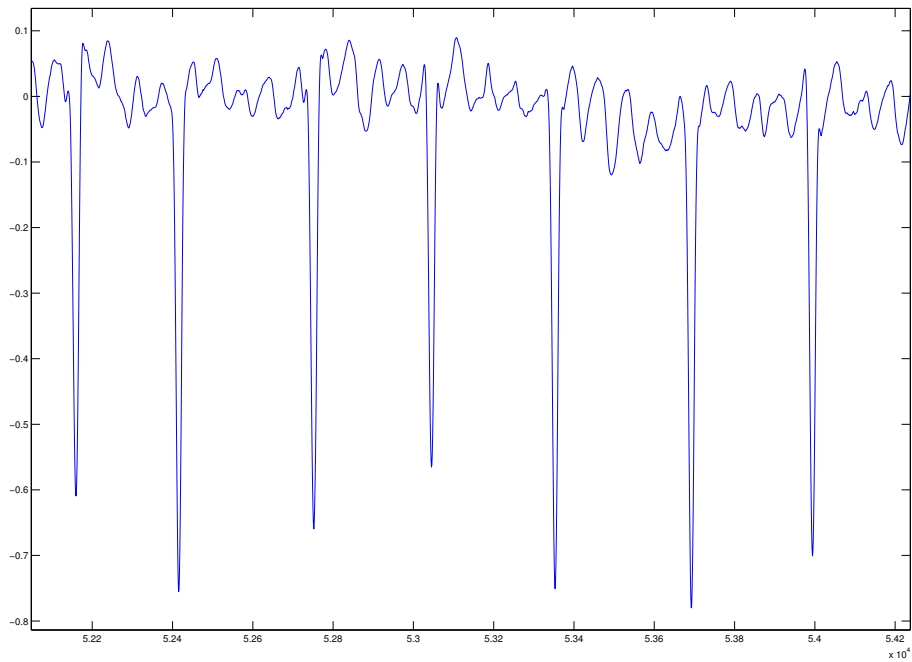


FIG. 3.8 – Fibrillation observable sur ECG d'un patient malade en V1.

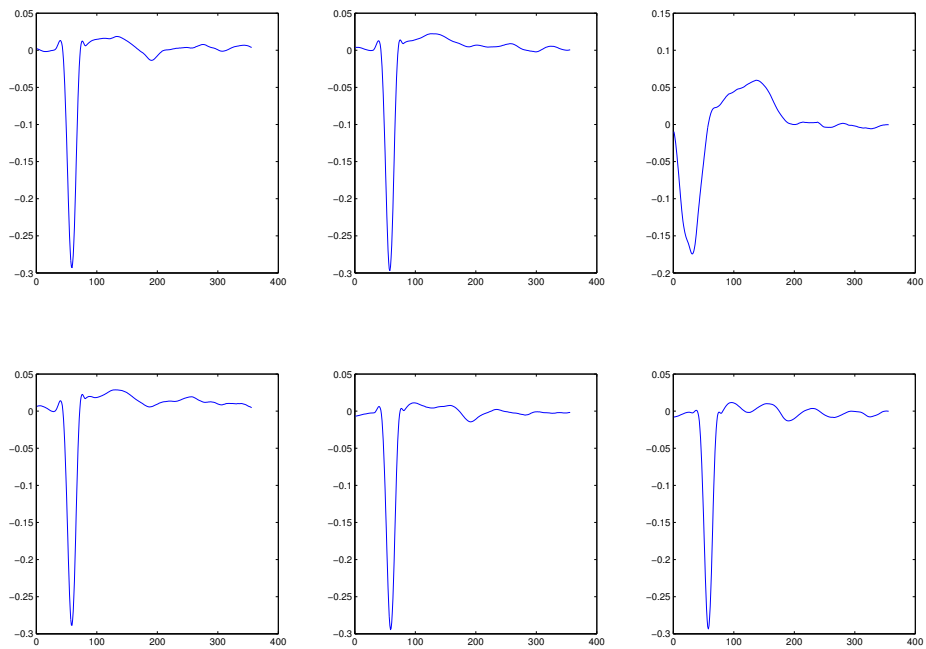


FIG. 3.9 – Dictionnaire ventriculaire appris sur le canal V1. On voit des traces de fibrillation sur le dernier atome.

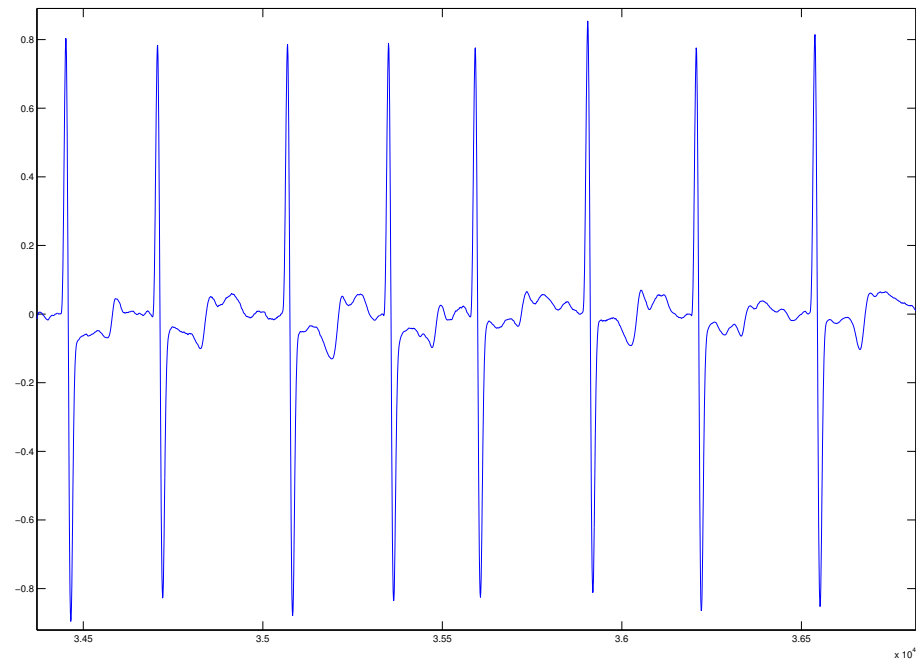


FIG. 3.10 – ECG d'un patient malade en V4. On voit des irrégularités, mais il est difficile de déterminer ce qui provient de la fibrillation.

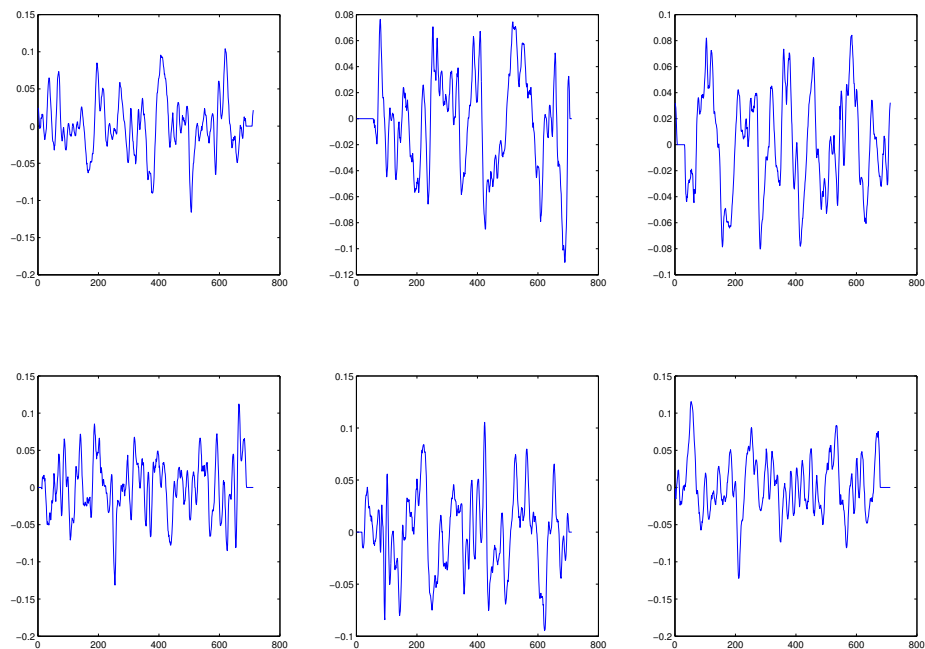


FIG. 3.11 – Dictionnaire atrial d'un patient malade en V4. Les atomes ne semblent pas présenter de trace d'activité ventriculaire résiduelle.

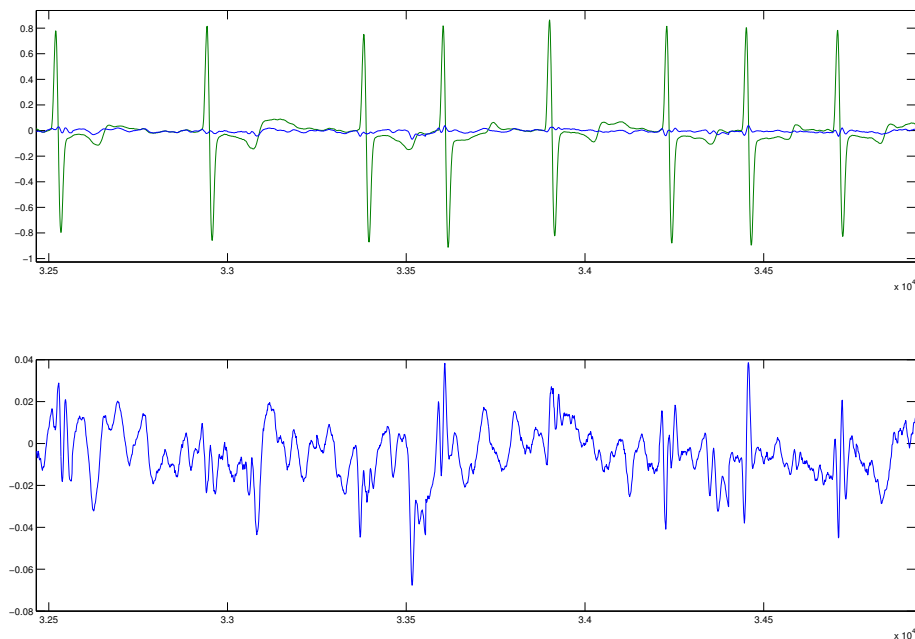


FIG. 3.12 – Estimation atriale \tilde{A} d'un patient malade en V4, avec le signal S (en haut) ou seule (en bas). L'amplitude de \tilde{A} est très faible par rapport à celle de S . Il reste d'importantes interférences ventriculaires dans \tilde{A} .

néanmoins utiliser des outils quantitatifs basés sur les modèles utilisés pour se faire une idée du comportement des différents systèmes. Au lieu de mesurer si les sources estimées sont proches des sources obtenues, on va mesurer si les sources estimées se comportent bien conformément au modèle. Cette adéquation peut être considérée comme une mesure de qualité de la séparation sous réserve que le modèle soit correct (les sources originales se conforment bien au modèle) et suffisamment informatif (la seule séparation qui se conforme au modèles est la séparation en sources originales).

Principe Le protocole d'évaluation utilisé a été mis au point par [LRS⁺06]. Il se base sur l'indépendance des sources et les caractéristiques morphologiques de l'activité ventriculaire. On peut détecter les ondes QRST dans le mélange grâce à la forte amplitude de l'activité ventriculaire. De plus, entre la fin de l'onde T et le début de l'onde Q suivante, il existe un intervalle plus ou moins long dépourvu d'activité ventriculaire. On peut donc segmenter le mélange et les sources estimées en segments QT qui contiennent toute l'activité ventriculaire et en segments TQ sans activité ventriculaire.

D'autre part, les sources originales sont supposées indépendantes, donc les sources estimées le sont aussi si la séparation a été bien faite. Dans ce cas, l'activité atriale devrait être indépendante des points de segmentation estimés à partir de l'activité ventriculaire, et les activités atriales estimées sur les intervalles QT et TQ devraient posséder les mêmes propriétés statistiques.

Réciproquement, on a vu sur les données synthétiques que les erreurs d'estimation avaient tendance à former des pics autour du complexe QRS. Donc ces erreurs devraient engendrer des différences entre les intervalles QT qui contiennent les complexes QRS et

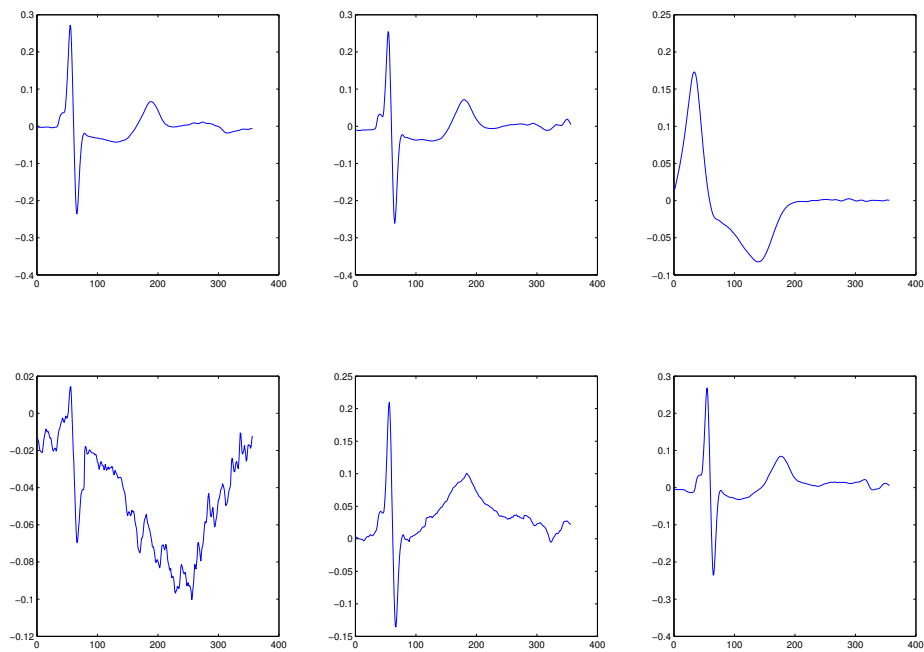


FIG. 3.13 – Dictionnaire ventriculaire appris sur le canal VR. On reconnaît dans les troisième et quatrième atome les formes de palpitation et d’artefact observées dans la figure 3.7.

les intervalles TQ qui ne les contiennent pas. Pour mesurer ces différences on choisit une statistique adaptée à la mesure de tels pics, par exemple l'amplitude maximale de la source atriale estimée sur un segment.

Protocole Nous avons évalué notre système d'apprentissage de dictionnaires et l'ABS de Lemay [SS01] sur 184 ECG de patients malades. Chaque signal est long de 5 minutes et comprend donc entre 300 et 500 battements. Tous ces signaux ont été séparés par les deux systèmes étudiés, puis segmentés en battements et chaque battement en intervalles QT et TQ. On a ensuite mesuré l'amplitude maximale de l'activité atriale estimée \tilde{A} sur chaque intervalle et observé la distribution des mesures obtenues sur le canal V1.

Résultats La figure 3.14 montre la distribution simplifiée des maxima d'amplitude mesurés sur les intervalles QT et TQ, pour les sources séparées par ABS ou apprentissage de dictionnaires. Si la séparation était parfaite, on devrait observer des distributions similaires sur les intervalles QT et TQ. Ce n'est pas tout à fait le cas.

Pour l'apprentissage de dictionnaires, on observe que l'amplitude maximale est légèrement plus élevée sur les intervalles QT que sur les intervalles TQ. Ce constat est cohérent avec les pics observés sur les données synthétiques au niveau des complexes QRS.

On observe le comportement inverse pour l'ABS : l'amplitude maximale est moins élevée sur les intervalles QT que TQ. Nous n'avons pour le moment pas d'explication à ce comportement. Il semble indiquer l'existence d'un autre type d'erreurs non encore identifié.

On peut remarquer par ailleurs que les distributions sont plus concentrées avec l'apprentissage de dictionnaires qu'avec l'ABS : il y a moins de points aberrants et leur amplitude est plus faible. On a vu lors de l'analyse qualitative que les amplitudes les plus fortes dans le mélange sont dues aux artefacts. Ce résultat montre donc que le choix d'un modèle de dictionnaire plus riche qu'un seul battement typique permet bien de mieux supprimer les artefacts.

3.6 Perspectives

L'approche décrite propose un schéma d'apprentissage alterné qui permet d'apprendre deux dictionnaires séparés à partir de la seule observation d'un mélange. Néanmoins il reste encore beaucoup de parties du système qui mériteraient d'être étudiées plus en profondeur.

3.6.1 Apprentissage à taille de dictionnaire variable

La taille des dictionnaires appris augmente à chaque itération. Actuellement, le dictionnaire précédent est oublié et le nouveau dictionnaire plus grand est entièrement réappris. On pourrait certainement accélérer le système global si on possédait un moyen d'étendre le dictionnaire existant.

3.6.2 Estimation automatique de la taille des dictionnaires à chaque itération

La taille des dictionnaires est cruciale pour notre application puisque c'est le paramètre principal de discrimination des sources. À une itération, si on apprend un dictionnaire trop petit, on laisse trop d'erreurs dans le résidu, ce qui va gêner le raffinement de l'autre source. Si on apprend un dictionnaire trop gros, les motifs appris sont corrompus par les erreurs résiduelles de l'autre source.

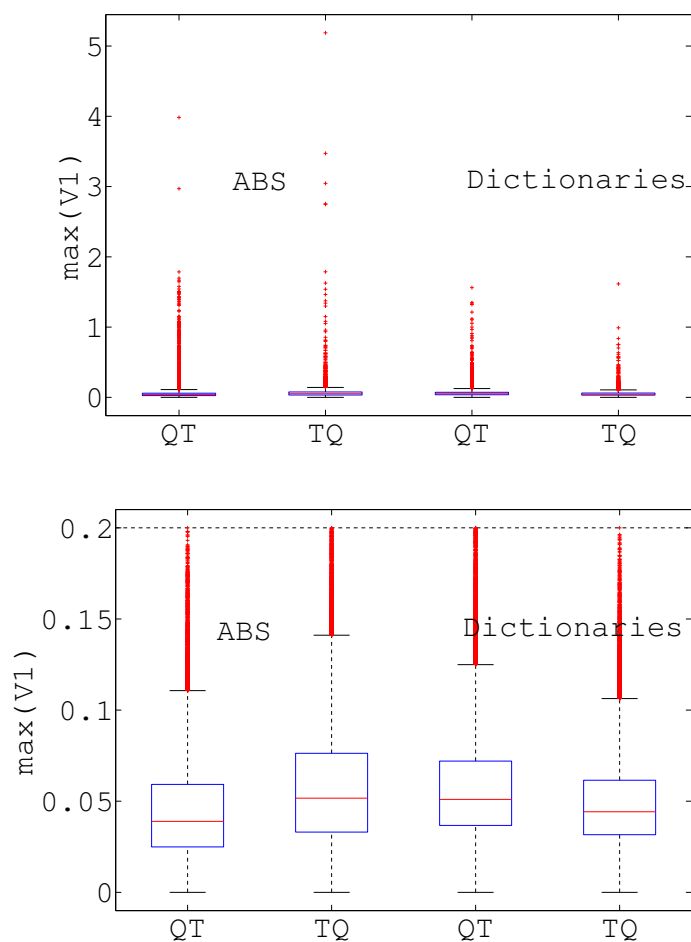


FIG. 3.14 – Distribution des amplitudes maximales de l'activité atriale estimée \tilde{A} sur chaque intervalle QT et TQ sur le canal $V1$ pour les méthodes d'apprentissage de dictionnaire et de soustraction de battement typique. La boîte contient 75% de la distribution avec la moyenne marquée d'un trait pointillé à l'intérieur. Les moustaches autour des boîtes contiennent 99% de la distribution. Les points aberrants hors des moustaches sont tracés individuellement. La figure du haut montre l'ensemble de la distribution et celle du bas uniquement le cœur.

Actuellement, on a retenu l'augmentation de 1 de la taille des dictionnaires à chaque itération. Nous n'avons pas de garantie que c'est un bon choix. Il est même possible qu'il n'existe pas de profil de raffinement convenable pour tous les signaux. On pourrait certainement améliorer grandement la qualité des algorithmes si on était capable de déterminer un nombre d'atomes optimal, l'optimum se définissant ici comme le plus grand nombre de motifs possible qui permet d'apprendre des motifs non-corrompus.

3.7 Modélisation *a priori* des motifs

Le filtrage des atomes atriaux décrit au paragraphe 3.4.3 correspond à l'utilisation d'un modèle *a priori* des motifs pour les débruiter de l'activité ventriculaire. L'approche choisie n'a pas de fondement théorique, mais on pourrait imaginer d'exploiter cette idée de manière plus rigoureuse.

Par exemple ; on sait que l'activité ventriculaire est constituée de pics et la fibrillation atriale d'oscillations. Ces observations n'étaient pas suffisantes pour permettre une analyse en composante morphologiques directe sur un dictionnaires d'ondelettes et de transformée de Fourier à court terme, mais on peut peut-être exploiter cette connaissance en forçant les motifs eux-mêmes à être parcimonieux sur une de ces bases [YBD09].

On peut également songer à utiliser l'apprentissage discriminant [MBSZ08], mais l'extension semble moins directe. L'apprentissage discriminant nécessite l'utilisation d'exemples des 2 sources séparées pour apprendre des dictionnaires qui représentent bien une source et mal l'autre. Or ici on ne possède que le mélange S . On pourrait bien sûr utiliser les sources estimées \tilde{A} et \tilde{V} , mais la discrimination apporte-t-elle quelque chose de plus ? A chaque itération le dictionnaire d'une source (\mathcal{M}_V par exemple) est appris à partir du signal $\tilde{V} + R$. Or \tilde{V} est l'estimation précédente de la source, supposée décorrélée de A si l'algorithme s'est bien comporté jusque là, et du résidu R trop faiblement corrélé pour être approché par \mathcal{M}_A . Ce schéma alternatif semble déjà introduire une contrainte de décorrélation entre les dictionnaires, il n'est pas sûr que l'apprentissage discriminant permettra d'aller plus loin.

3.8 Conclusion

Nous avons présenté une nouvelle approche pour la séparation de sources. Cette approche est basée sur la différence d'énergie entre les sources. En cela elle peut faire penser à une analyse en composantes principales, mais nous utilisons en plus l'invariance par translation pour détecter des occurrences corrélées d'une source. On peut également voir notre méthode comme un moyen d'apprendre plus de battements typiques pour l'ABS ou comme un moyen d'apprendre les dictionnaires sur le mélange pour la MCA.

On remarque que notre système exploite peu les dictionnaires. Ils sont appris, permettent de calculer une estimation de leur source, puis sont oubliés. En particulier, on n'exécute finalement pas la MCA proprement dite, puisque les sources ont déjà été décomposées pendant l'apprentissage. Cependant les dictionnaires contiennent eux-mêmes un information utile, notamment sur les spectre de la fibrillation qui devrait être identique dans les atomes et le signal. Un axe de poursuite de ces travaux peut être l'étude des dictionnaires en tant que résumé du signal. Quelles informations utiles peut-on en tirer sans avoir à se plonger dans un signal de grande dimension ?

D'autre part, justement parce qu'on n'utilise plus de MCA, le système présenté à la section 3.4.3 pourrait être envisagé en utilisant d'autres types de modèles que la parcimonie. Les modèles n'ont même pas besoin d'être du même type pour chaque source.

Pour la partie ventriculaire, on peut se demander si l'utilisation d'un dictionnaire invariant par translation n'est pas une perte de temps. On autorise 1 atome par battement et ceux-ci s'alignent naturellement sur les complexes QRS. Ce n'est pas la peine de faire tourner un algorithme invariant par translation pour ça : on connaissait l'emplacement des QRS dès l'initialisation. De plus, les atomes faisant la longueur d'un battement cardiaque et étant alignés sur les battements, ils ne se superposent pas. On n'utilise donc pas la possibilité d'effectuer des combinaisons linéaires d'atomes offerte par les modèles parcimonieux. Finalement, notre modèle complexe peut se ramener à un simple clustering sur des extraits définis par rapport à l'emplacement des complexes QRS, ce qui est exactement le fonctionnement de l'ABS. L'algorithme serait plus rapide en remplaçant l'apprentissage de dictionnaire ventriculaires par une simple ABS. Cela permettrait en plus de profiter des avancées dans ce domaine, notamment pour la dilatation temporelle des battements typiques.

Pour la partie atriale, par contre, le modèle choisi semble moins critiquable. On a vu à la section 3.3 que l'invariance par translation contribue à rejeter l'activité ventriculaire résiduelle. On n'a pas de raison de remettre en cause le choix d'un modèle parcimonieux à ce moment. Une des motivations initiales pour supprimer l'activité ventriculaire est justement de disposer de signaux de fibrillation à étudier et à modéliser.

Conclusion

Cette thèse a apporté des réponses pratiques aux principaux problèmes rencontrés dans le domaine des représentations parcimonieuses : la décomposition et l'apprentissage du dictionnaire. Nous avons également mis au point un système de séparation de sources par parcimonie pour l'étude de la fibrillation atriale.

On semble avoir atteint un palier dans le domaine des décompositions puisqu'on a réussi à conjuguer les avantages de MP et d'OMP. Par contre, tous les algorithmes étudiés pour l'apprentissage de dictionnaire, y compris celui proposé, restent bloqués sur un minimum local de la fonction de coût même lorsqu'on leur fournit le support des décompositions. Les résultats obtenus en séparation de sources cardiaques sont encourageants et ouvrent la porte à de nombreuses perspectives, qu'il s'agisse d'améliorer l'algorithme, de le porter à d'autres applications ou d'exploiter les sources obtenues.

Rappel des contributions

Décompositions parcimonieuses rapides L'algorithme proposé LocOMP permet un calcul rapide de décomposition parcimonieuse presque orthogonale. Il peut s'appliquer quand le dictionnaire est composé d'atomes de support court par rapport à la longueur du signal. Dans ce cas, sa complexité est semblable à celle de MP. Expérimentalement, on constate que LocOMP parvient à approcher le signal d'aussi près qu'OMP en un temps de calcul à peine 50% plus élevé que pour MP : on a presque réussi à exécuter OMP pour le coût de MP.

On dispose de garanties théoriques sur la complexité de l'algorithme, mais pas sur la qualité de l'approximation calculée. Ça ne doit pas nous empêcher d'utiliser LocOMP à la place de MP : on a pu vérifier expérimentalement que LocOMP calcule toujours une approximation significativement meilleure. Le dernier obstacle au remplacement de MP par LocOMP est l'absence d'une implémentation efficace de LocOMP, notamment dans la librairie MPTK.

Apprentissage de dictionnaires Nous avons proposé un calcul automatique du pas d'apprentissage pour l'algorithme d'Olshausen et Field. Cela simplifie la mise en œuvre de l'algorithme puisqu'il n'y a plus de paramètre à régler. On constate également une amélioration de la qualité du dictionnaire obtenu sur des images naturelles, mais ces résultats ne concordent pas avec ceux obtenus sur des données synthétiques. On ne peut donc pas affirmer qu'un algorithme calcule de meilleurs dictionnaires qu'un autre sans tenir compte des caractéristiques des données d'apprentissage.

Tous les algorithmes utilisés en comparaison présentent le même défaut : ils alternent des décompositions parcimonieuses et des mises à jour du dictionnaire à décomposition fixée. Cette approche est inspirée de l'algorithme de clustering K-means, qui alterne la projection des données sur le centroïde le plus proche (décomposition) et le calcul des centroïdes (mise à jour du dictionnaire). Dans K-means, on remplace ainsi un problème

complexe par deux problèmes simples. Pour l'apprentissage de dictionnaire, on a replacé un problème qu'on ne sait pas résoudre par deux problèmes qu'on ne sait pas résoudre non plus. Cela peut inciter à chercher d'autres approches que l'optimisation alternée. Si on préfère rester dans ce cadre, il faudra s'intéresser de plus près à la mise à jour du dictionnaire quand on connaît le support des décompositions des données d'apprentissage mais pas l'amplitude des coefficients.

Séparation des activités atriale et ventriculaire Nous avons également mis au point un système de séparation de sources pour les signaux d'ECG. Ce système utilise les différences d'énergie entre les 2 sources pour apprendre 2 dictionnaires, un par source, directement à partir de mélange. Sur des données synthétiques, on a pu mesurer une amélioration dans l'estimation de la fibrillation atriale par rapport aux algorithmes de l'état de l'art. Les résultats sur des données cliniques semblent également probants, même s'ils sont plus délicats à analyser faute de connaissance des sources originales.

Perspectives

Décompositions parcimonieuses rapides

Implémentation dans MPTK Nous avons pu vérifier la faible complexité théorique de LocOMP sur du code MATLAB, mais ce code expérimental reste peu intéressant pour des applications réelles. L'implémentation de LocGP sur MPTK est encore très loin de la vitesse qu'on attend d'elle. On a localisé le problème (les calculs répétés de forme d'onde des atomes), mais il reste à le résoudre, et un autre frein peut apparaître quand celui-ci sera résolu. D'autre part, les expériences sur le code MATLAB indiquent que LocGP calcule de moins bonnes approximations que LocOMP. Il faudra donc songer à également implémenter LocOMP dans MPTK, donc inclure un module de calcul matriciel pour inverser la matrice de Gram.

Autres algorithmes Sous l'hypothèse de dictionnaires locaux, LocOMP peut remplacer avantageusement tous les algorithmes se situant entre MP et OMP en termes de complexité et de qualité d'approximation. Pour chercher d'autres algorithmes pratiques, il faudra donc regarder hors de ce spectre : soit des algorithmes moins coûteux que MP, soit calculant une meilleure approximation qu'OMP.

Parmi les algorithmes moins coûteux que MP on peut citer les approches comme StOMP [DTS06]. StOMP se place entre le seuillage dur décrit à la section 1.2.2.1 et MP. Le seuillage dur choisit en une seule itération tous les atomes de la décomposition. MP choisit un nouvel atome par itération. StOMP choisit plusieurs atomes par itération. Ce principe peut s'appliquer à tout algorithme glouton (d'ailleurs les auteurs de StOMP ont préféré l'utiliser avec OMP que MP). On pourrait imaginer un algorithme StLocOMP. La sélection simultanée de plusieurs atomes peut rendre la mise à jour du résidu moins locale. Pour éviter ça, on pourrait se restreindre à ne sélectionner que des atomes proches les uns des autres à une itération donnée.

Les algorithmes plus complexes qu'OMP ont tendance à sortir du cadre glouton pur. L'idée est de sélectionner plus que les K atomes souhaités pour pouvoir revenir en arrière et supprimer a posteriori les atomes les moins influents. Par exemple l'algorithme CoSaMP [NT09] sélectionne à chaque étape $2K$ atomes et en supprime K . Là encore, appliquer CoSaMP tel quel brise la localité de mise à jour du résidu. On pourrait choisir de ne retirer que des atomes appartenant au sous-dictionnaire $\Psi^{(i)}$ à chaque itération.

Analyse théorique Les performances pratiques de LocOMP sont établies, mais on ne dispose d'aucune garantie théorique à ce propos. De telles propriétés semblent difficiles à obtenir.

L'algorithme a été conçu comme un intermédiaire entre MP et OMP, et c'est le comportement qu'on observe en pratique. On souhaiterait donc une propriété qui garantit que LocOMP se comporte plus comme OMP que comme MP. Or on dispose de peu de résultats qui établissent une différence entre MP et OMP. Les conditions de Tropp énoncées à la section 1.4 sont valables pour MP comme pour OMP. On peut facilement vérifier qu'elles sont également valables pour LocOMP, mais cela ne nous permet pas de situer l'algorithme par rapport à ses deux parents. On sait par sa définition qu'OMP annule le résidu au bout de N itérations. Ce n'est pas le cas pour MP, ni pour LocOMP. De ce point de vue, LocOMP se comporte donc plutôt comme MP que comme OMP.

Apprentissage de dictionnaires

L'apprentissage de dictionnaires est encore très loin d'être maîtrisé. L'algorithme proposé offre des performances au moins égales aux algorithmes de l'état de l'art pour moins cher, mais ils sont tous incapables de retrouver le dictionnaire optimal quand on leur fournit le support de la décomposition.

Au-delà de l'approche alternée Cela nous conduit à nous interroger sur le bien fondé de l'approche alternée. Cette approche puise son inspiration dans l'algorithme de clustering K-means qui alterne la segmentation de l'ensemble d'apprentissage en clusters et le calcul des centroïdes des clusters. Dans K-means, chaque étape est très simple : la segmentation est un simple maximum de corrélation entre les données d'apprentissage et chaque centroïde, tandis que le calcul des centroïdes est un calcul de moyenne. On remplace un problème complexe par deux problèmes simples, en sachant qu'on ne peut pas toujours trouver le meilleur clustering ainsi.

Dans l'apprentissage de dictionnaire, l'approche alternée remplace un problème complexe par deux problèmes complexes. Elle ne semble donc pas vraiment satisfaisante, mais il existe peu d'autres approches à l'heure actuelle. L'utilisation d'une minimisation de norme ℓ_1 des coefficients X sous contrainte de reconstruction parfaite $S = \Phi X$ a été proposée [Plu07][GS09], mais elle ne semble pas encore avoir donné lieu à des applications pratiques.

Optimisation non-monotone Même si on reste dans le cadre de l'approche alternée, il y a encore beaucoup de choses à faire pour améliorer chaque étape intermédiaire. La décomposition parcimonieuse étant déjà un sujet de recherche en soi, on peut se concentrer sur la mise à jour du dictionnaire. L'expérience décrite à la section 2.3.3.2 indique qu'un problème intéressant à considérer est celui du calcul du meilleur dictionnaire à support de décomposition fixé. Si on cherche à minimiser l'énergie du résidu $\|S - \Phi X\|_2^2$, alors c'est un problème de degré 4 car Φ et X sont tous les deux inconnus. Tous les algorithmes étudiés effectuent cette optimisation de manière itérative, un atome et ses coefficients après l'autre pour K-SVD, le dictionnaire Φ puis les coefficients X pour MOD, chaque atome l'un après l'autre puis les coefficients pour les descentes de gradient. Peut-être faudrait-il optimiser tous les paramètres ensemble ? On pourrait songer à adapter des algorithmes d'optimisation non-monotone comme le recuit simulé ou les algorithmes génétiques. On peut également se demander s'il y a une interprétation géométrique à l'existence de minima locaux : pourrait-il exister un lien simple entre le dictionnaire sous-optimal et le dictionnaire optimal ? Par

exemple (il ne s'agit que d'une intuition) le dictionnaire sous-optimal pourrait-il lui-même être parcimonieux sur le dictionnaire optimal ?

Isolation de la fibrillation atriale

Évolution du système Notre algorithme présente déjà des résultats encourageants, mais de nombreux éléments du système ont été choisis arbitrairement et mériteraient d'être étudiés plus en détail.

Tout d'abord le filtrage des motifs atriaux utilisé n'a aucune justification théorique. On pourrait le remplacer par un modèle a priori des motifs ou par un apprentissage discriminant qui viseraient au même but : éviter la présence d'activité ventriculaire résiduelle dans les motifs atriaux.

Ensuite, l'étude des signaux réels a montré d'autres problèmes qui n'ont pas été traités jusqu'à présent. Les battements ectopiques sont bien gérés par l'algorithme actuel à condition qu'ils restent exceptionnels, ce qui n'est malheureusement pas toujours le cas pour les patients malades. Dans ce cas, il faudrait être capable d'évaluer la quantité de battements ectopiques lors de l'apprentissage ventriculaire et d'apprendre un motif de plus pour les représenter si besoin est. On se trouve également en présence d'artefacts de calcul a priori sans structure. Comme on travaille sur des ECG mesurés au repos, ces artefacts restent heureusement rares. La meilleure solution est donc sans doute de les détecter et de mettre à 0 l'intervalle de signal correspondant, éventuellement seulement sur un canal si l'artefact est localisé.

D'autre part, on ne maîtrise pas actuellement les différents paramètres de l'algorithme. On peut jouer sur le nombre de motifs appris et la taille de la décomposition à chaque itération. Le but est d'annuler autant que possible la source qu'on cherche à modéliser tout en laissant l'autre intacte. Baser le nombre d'atomes sur le nombre de battements est naturel pour l'apprentissage ventriculaire, mais on ne dispose pas d'une telle indication pour l'apprentissage atrial. On a vu que le nombre de motifs nécessaires pour l'apprentissage ventriculaire varie en fonction de l'importance des battements ectopiques, mais encore une fois on ne dispose pas d'autant d'informations sur l'activité atriale.

Généralisation à d'autres problèmes Notre système de séparation pourrait être utilisé dans d'autres cadres. Pour qu'il fonctionne, on a besoin que les sources soient décorréelées et d'amplitudes déséquilibrées pour amorcer l'apprentissage. Avec le recul, on constate que la parcimonie des sources n'est pas absolument nécessaire. On doit être capable de proposer un modèle des sources qui permette d'en calculer une approximation, mais ce modèle n'a pas besoin d'être basé sur la parcimonie. Les activités des sources doivent par contre être locales : la séparation repose sur l'utilisation d'un tramage différent pour chaque source qui décorréle les trames de l'autre source. Puisqu'on observe des phénomènes locaux qui se répètent dans le temps, il y aura toujours une basse dimension intrinsèque du modèle, mais celui-ci n'a pas besoin d'être linéaire. D'autre part, notre système est applicable à des signaux monocanaux. En cas de signaux multicanaux, c'est au modèle de sources de tirer bénéfice de l'information supplémentaire pour mieux approcher les sources.

Les hypothèses nécessaires de décorrélation et de localité des sources semblent bien convenir aux signaux biologiques. L'analyse en composantes morphologiques a déjà été utilisée pour le traitement d'électro-encéphalogrammes (EEG) [YWB09]. L'apprentissage pourrait améliorer les résultats sur les EEG comme il l'a fait pour les ECG. Néanmoins, si on souhaite traiter des patients observés en activité, les signaux comporteront beaucoup plus d'artefacts qu'il faudra supprimer.

Bibliographie

- [AEB05] M. Aharon, M. Elad, and A. Bruckstein. K-SVD : design of dictionaries for sparse representation. In *In Proc. SPARS'05, IRISA, Rennes, France*, November 2005. <http://spars05.irisa.fr>.
- [AP01] S.A. Abdallah and M.D. Plumbley. If edges are the independent components of natural images, what are the independent components of natural sounds? In *Proc. of the Int'l Workshop on ICA and Blind Signal Separation (ICA 2001)*, pages 534–539, December 2001.
- [BD06] T. Blumensath and M. Davies. Sparse and shift-invariant representations of music. *Audio, Speech, and Language Processing, IEEE Transactions on [see also Speech and Audio Processing, IEEE Transactions on]*, 14(1) :50–57, Jan. 2006.
- [BD08] T. Blumensath and M.E. Davies. In greedy pursuit of new directions : (nearly) orthogonal matching pursuit by directional optimisation. In *Proc. EUSIP-CO'08*, 2008.
- [BD09] T. Blumensath and M.E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 2009.
- [BS97] A.J. Bell and T.J. Sejnowski. The "independent components" of natural scenes are edge filters. *Vision Research*, 1997.
- [CDS96] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20 :33–61, 1996.
- [CMVG08] A.L. Casanovas, G. Monaci, P. Vandergheynst, and R. Gribonval. Blind audiovisual separation based on redundant representations. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 1841–1844, 31 2008-April 4 2008.
- [CREKD05] Shane F. Cotter, Bhaskar D. Rao, Kjersti Engan, and Kenneth Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on signal processing*, 2005.
- [DDFG09] I. Daubechies, R. DeVore, M. Fornasier, and S. Gunturk. Iteratively re-weighted least squares minimization for sparse recovery. *Commun. Pure Appl. Math*, 2009. to appear.
- [DDM04] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math*, 57(11) :1413–1457, 2004.
- [DEGL⁺06] O. Divorra Escoda, L. Granai, M. Lemay, J. Molinero Hernandez, P. Vandergheynst, and J. Vesin. Ventricular and Atrial Activity Estimation Through Sparse ECG Signal Decompositions. In *ICASSP'06*, 2006.

- [DH04] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *ICML '04 : Proceedings of the twenty-first international conference on Machine learning*, page 29, New York, NY, USA, 2004. ACM.
- [DMA97] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constr. Approx.*, 13(1) :57–98, 1997.
- [DTS06] D. Donoho, Y. Tsaig, and J.L. Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. A paraître, 2006.
- [EAHH99] K. Engan, S.O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. *Acoustics, Speech, and Signal Processing, 1999. ICASSP '99. Proceedings., 1999 IEEE International Conference on*, 5 :2443–2446 vol.5, 1999.
- [FRA⁺01] Valentin Fuster, Lars E. Rydén, Richard W. Asinger, David S. Cannom, Harry J. Crijns, Robert L. Frye, Jonathan L. Halperin, G. Neal Kay, Werner W. Klein, Samuel Lévy, Robert L. McNamara, Eric N. Prystowsky, L. Samuel Wann, D. George Wyse, Raymond J. Gibbons, Elliott M. Antman, Joseph S. Alpert, David P. Faxon, Valentin Fuster, Gabriel Gregoratos, Loren F. Hiratzka, Alice K. Jacobs, Richard O. Russell, Sidney C. Smith, Werner W. Klein, Angeles Alonso-Garcia, Carina Blomström-Lundqvist, Guy De Backer, Marcus Flather, Jaromir Hradec, Ali Oto, Alexander Parkhomenko, Sigmund Silber, and Adam Torbicki. Acc/aha/esc guidelines for the management of patients with atrial fibrillation : executive summary : A report of the american college of cardiology/ american heart association task force on practice guidelines and the european society of cardiology committee for practice guidelines and policy conferences (committee to develop guidelines for the management of patients with atrial fibrillation) developed in collaboration with the north american society of pacing and electrophysiology. *Journal of the American College of Cardiology*, 38(4) :1231 – 1265, 2001.
- [GRSV08] R. Gribonval, H. Rauhaut, K. Schnass, and P. Vandergheynst. Atoms of all channels, unite! average case analysis of multi-channel sparse recovery using greedy algorithms. *Journal of Fourier Analysis and Applications*, 2008.
- [GS09] R. Gribonval and K. Schnass. Basis identification from random sparse samples. In *Proc. SPARS'09*, 2009.
- [GVL96] Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.
- [HKO01] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.
- [Hur98] J. Willis Hurst. Naming of the waves in the eeg, with a brief account of their genesis. *Circulation*, 1998.
- [IvOH06] Zenichi Ihara, Adriaan van Oosterom, and Rudi Hoekema. Atrial repolarization as observable during the pq interval. *Journal of Electrocardiology*, 39(3) :290 – 297, 2006.
- [Jac04] Vincent Jacquemet. *A biophysical model of atrial fibrillation and electrograms : formulation, validation and applications*. PhD thesis, EPFL, 2004.
- [JH91] C. Jutten and H. Héroult. Blind separation of source, part I : an adaptive algorithm based on neuromimetic architecture. *Signal processing*, 24 :1–10, 1991.

- [JRY00] A. Jourjine, S. Rickard, and O. Yilmaz. Blind separation of disjoint orthogonal signals : demixing n sources from 2 mixtures. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, volume 5, pages 2985–2988 vol.5, 2000.
- [KdRE⁺99] K. Kreutz-delgado, B.D. Rao, K. Engan, T. w. Lee, and T. Sejnowski. Convex/schur-convex (csc) log-priors and sparse coding. In *in 6th Joint Symposium on Neural Computation*, pages 65–71, 1999.
- [KG06] Sacha Krstulovic and Rémi Gribonval. MPTK : Matching Pursuit made tractable. In *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP'06)*, volume 3, pages 496–499, Toulouse, France, May 2006.
- [KGB⁺07] Paul Kligfield, Leonard S. Gettes, James J. Bailey, Rory Childers, Barbara J. Deal, E. William Hancock, Gerard van Herpen, Jan A. Kors, Peter Macfarlane, David M. Mirvis, Olle Pahlm, Pentti Rautaharju, and Galen S. Wagner. Recommendations for the standardization and interpretation of the electrocardiogram : Part i : The electrocardiogram and its technology a scientific statement from the american heart association electrocardiography and arrhythmias committee, council on clinical cardiology ; the american college of cardiology foundation ; and the heart rhythm society endorsed by the international society for computerized electrocardiology. *Journal of the American College of Cardiology*, 49(10) :1109 – 1127, 2007.
- [Les07] Sylvain Lesage. *Apprentissage de dictionnaires structurés pour la modélisation parcimonieuse de signaux multicanaux*. PhD thesis, Université de Rennes 1, 2007.
- [LRS⁺06] P. Langley, J.J. Rieta, M. Stridh, J. Millet, L. Sörnmo, and A. Murray. Comparison of atrial signal extraction algorithms in 12-lead ecgs with atrial fibrillation. *IEEE Transaction on biomedical engineering*, 53(2) :343–346, 2006.
- [LVIK04] Mathieu Lemay, Jean-Marc Vesin, Zenichi Ihara, and Lukas Kappenberger. Suppression of ventricular activity in the surface electrocardiogram of atrial fibrillation. In *ICA*, pages 1095–1102, 2004.
- [LVvO⁺07] M. Lemay, J.-M. Vesin, A. van Oosterom, V. Jacquemet, and L. Kappenberger. Cancellation of ventricular activity in the ecg : Evaluation of novel and existing methods. *Biomedical Engineering, IEEE Transactions on*, 54(3) :542–546, March 2007.
- [Mas04] G. Masataka. Development of the rwc music database. In *Proceedings of the 18th International Congress on Acoustics (ICA 2004)*, pages I–553–556, April 2004.
- [MBSZ08] J. Mairal, F. Bach, J. Ponce G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [Mei06] Marina Meilă. The uniqueness of a good optimum for k-means. In *ICML '06 : Proceedings of the 23rd international conference on Machine learning*, pages 625–632, New York, NY, USA, 2006. ACM.
- [MGVB08] B. Mailhé, R. Gribonval, P. Vandergheynst, and F. Bimbot. Shift-invariant dictionary learning : extending K-SVD. In *Proc. EUSIPCO'08*, 2008.
- [MZ93] S.G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12) :3397–3415, Dec 1993.

- [NT09] D. Needell and J.A. Tropp. Cosamp : Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 2009.
- [OF97] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set : A strategy employed by v1? *Vision Research*, 37 :3311–3325, 1997.
- [PFS09] Gabriel Peyré, Jalal Fadili, and Jean-Luc Starck. Learning the morphological diversity. Preprint HAL-00415782, september 2009.
- [Plu07] Mark D. Plumbley. Dictionary learning for l1-exact sparse coding. In *Proc. ICA'07*, 2007.
- [PRRK93] Y. C. Pati, R. Rezaifar, Y. C. Pati R. Rezaifar, and P. S. Krishnaprasad. Orthogonal matching pursuit : Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers*, pages 40–44, 1993.
- [RCS⁺04] José Joaquín Rieta, Francisco Castells, César Sánchez, Vicente Zarzoso, and José Millet. Atrial activity extraction for atrial fibrillation analysis using blind source separation. *IEEE Transactions on Biomedical Engineering*, 51(7) :1176–1186, July 2004.
- [SED04] J.-L. Starck, M. Elad, and D.L. Donoho. Redundant multiscale transforms and their application for morphological component analysis. *Journal of Advances in Imaging and Electron Physics*, 132 :287–348, 2004.
- [SS01] M Stridh and L Sörnmo. Spatiotemporal QRST cancellation techniques for analysis of atrial fibrillation. *IEEE Transactions on Biomedical Engineering*, 48(1) :105–111, Jan, 2001.
- [SyA06] Karl Skretting, John Håkon Husøy, and Sven Ole Aase. General design algorithm for sparse frame expansions. *Signal Process.*, 86(1) :117–126, 2006.
- [Tro04] J. Tropp. Greed is good : algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 2004.
- [VGF06] E. Vincent, R. GRibonval, and C. Févotte. Performance measurement in blind audio source separation. *IEEE Trans. on Audio, Speech and Language*, 2006.
- [vOJss] A van Oosterom and V Jacquemet. Genesis of the P wave : atrial signals as generated by the equivalent double layer source model. *Europace*, page null, 2005. [in press].
- [YBD09] M. Yaghoobi, T. Blumensath, and M.E. Davies. Parsimonious dictionary learning. In *Proc. ICASSP'09*, 2009.
- [YWB09] Xinyi Yong, Rabab K. Ward, and Gary E. Birch. Artifact removal in EEG using morphological component analysis. In *Proc. ICASSP'09*, 2009.
- [ZNB97] V. Zarzoso, A.K. Nandi, and E. Bacharakis. Maternal and foetal ecg separation using blind source separation methods. *MA Journal of mathematics applied in medicine and biology*, 14(3), 1997.

Table des figures

1	Comparaison de la photographie d'une scène naturelle (portrait de Lena, à gauche) et d'une image aléatoire (pixels indépendants de loi uniforme sur l'intervalle $[0, 1]$) à droite.	12
2	Sous-espaces parcimonieux avec $K = 2$ atomes d'un dictionnaire à 3 atomes dans un espace de dimension $N = 3$. Les axes sont la direction des atomes. Le modèle est l'union des plans bleus : les signaux recherchés peuvent se trouver dans n'importe lequel des plans, mais pas ailleurs. Ce modèle est non-convexe : les points A et B sont dedans, mais aucun point du segment $]A, B[$ n'y est.	14
1.1	Sélection du sous-dictionnaire $\Psi^{(i)}$. On garde tous les atomes de $\Phi^{(i)}$ dont le support chevauche celui du dernier atome $\varphi^{(i)}$. Le résidu ne change au maximum que sur un intervalle de longueur inférieure à $3L - 2$	35
1.2	Sélection du sous-dictionnaire $\Psi^{(i)}$ dans le cas multi-échelles. L'atome court φ_c ne chevauche pas le dernier atome $\varphi^{(i)}$, mais son support est inclus dans celui de l'atome long φ_d . Il sera donc conservé.	37
1.3	Structure de données pour l'accès rapide aux atomes. Les atomes de même bloc et t_{\min} forment un même tas, puis tous les tas d'un même bloc sont triés par t_{\min} croissants, puis tous ces ensembles triés sont rangés par bloc.	39
1.4	Temps de calcul nécessaire à différents algorithmes en fonction du nombre d'itérations total. Les algorithmes locaux (MP, LocOMP et LocGP) sont nettement plus rapides que les algorithmes globaux (OMP et GP). LocOMP et LocGP sont indissociables sur cette courbe, de même qu'OMP et GP. La tangente en pointillés est la limite inférieure du coût d'un algorithme global. Elle correspond au coût de calcul des corrélations $\lambda = \Phi^*r$ à chaque itération.	41
1.5	SNR obtenu par les différents algorithmes en fonction du nombre d'itérations total. MP mis à part, tous les autres algorithmes offrent une qualité équivalente. La courbe du bas montre la perte de SNR par rapport à OMP. On voit que LocOMP subit une légère perte de 0.01dB et LocGP une perte un peu plus importante de 0.09dB. La tangente en pointillés est la limite inférieure du coût d'un algorithme global. Elle correspond au coût de calcul des corrélations $\lambda = \Phi^*r$ à chaque itération.	43
1.6	SNR moyen obtenu par MPTK et LocGP en fonction du nombre d'itérations total sur 10 signaux de piano échantillonnés à 8 kHz avec des dictionnaires MDCT d'échelles 32 et 1024. LocGP permet un gain moyen de 2dB.	44
2.1	Interprétation géométrique du pas d'apprentissage ρ . Le nouvel atome $\varphi_d^{(i+1)}$ se trouve sur un quart du cercle trigonométrique défini par l'ancien atome $\varphi_d^{(i)}$ et le gradient $\nabla_d \mathcal{E}$. Le pas ρ est la tangente de l'angle θ entre l'ancien atome et le nouveau.	54

2.2	SNR en fonction de l'itération pour les différentes règles de mise à jour du dictionnaire à coefficients fixés. Les coefficients à gauche ont été tirés selon une loi gaussienne à gauche et de Rademacher à droite. Notre descente de gradient à pas optimal assure la convergence la plus rapide. L'algorithme d'Olshausen et Field n'atteint pas de maximum global. L'algorithme K-SVD présente une très forte variabilité de comportement dans le cas des coefficients gaussiens. C'est dû à la présence d'un palier dans la progression. On peut observer ce palier sur le premier quantile de la courbe de droite. Avec des données gaussiennes, ce palier est tout le temps présent et c'est sa durée qui fait varier la vitesse de convergence de l'algorithme.	58
2.3	SNR en fonction de l'itération pour les différents algorithmes à support de décomposition fixé. Les coefficients ont été tirés selon une loi gaussienne en haut et de Rademacher en bas. Tous les algorithmes atteignent parfois le minimum global et restent le plus souvent bloqués sur un minimum local sous-optimal. K-SVD atteint le minimum global dans plus de la moitié des cas pour les données de Rademacher.	60
2.4	SNR obtenu en fonction de l'itération par les différents algorithmes complets. Les coefficients suivent une loi gaussienne. On a utilisé MP ou OMP pour la décomposition.	61
2.5	SNR obtenu en fonction de l'itération par les différents algorithmes complets. Les coefficients suivent une loi de Rademacher. On a utilisé MP ou OMP pour la décomposition.	62
2.6	SNR moyen atteint par les différents algorithmes en fonction de l'itération sur les 10 images d'Olshausen et Field. La descente de gradient à pas optimal arrive fournir la deuxième meilleure approximation après K-SVD. MOD fonctionne étonnamment moins bien que l'algorithme d'Olshausen et Field.	64
2.7	Première image de la base d'Olshausen et Field et son approximation sur des dictionnaires appris par les différents algorithmes.	65
2.8	Dictionnaire appris sur l'image montrée à la figure 2.7 par descente de gradient à pas optimal.	66
2.9	Dictionnaire appris sur l'image montrée à la figure 2.7 par l'algorithme d'Olshausen et Field.	66
2.10	Dictionnaire appris sur l'image montrée à la figure 2.7 par l'algorithme K-SVD.	67
2.11	Dictionnaire appris sur l'image montrée à la figure 2.7 par l'algorithme MOD.	67
3.1	Fonctionnement d'une systole cardiaque. Les atria se contractent en premier pour expulser le sang qu'elles contiennent dans les ventricules. Ensuite ceux-ci se contractent pour propulser le sang hors du cœur. Chez le patient en fibrillation atriale, la contraction des atria n'a pas lieu.	73
3.2	Placement de électrodes pour un électrocardiogramme à 12 canaux. On ne place que 9 électrodes [KGB ⁺ 07]. Les 3 voies restantes sont des différences de potentiel entre membres. On voit que le signal est plus faible au niveau du pied.	74
3.3	Trace des activités atriale et ventriculaire dans l'ECG. Chez le patient, la contraction de l'atria provoque l'onde P, suivie par le complexe QRS à la contraction des ventriculaire et l'onde T à leur relaxation. En cas de fibrillation, l'onde P disparaît pour être remplacée par des oscillation erratiques et indépendants du rythme cardiaque maintenu par les ventricules.	75
3.4	Fonctionnement d'une itération de séparation.	83

3.5	De haut en bas : le mélange synthétique S , les activités ventriculaires originale V et estimée \tilde{V} , les activités atriale originale A et estimée \tilde{A} , le tout sur la voie VR. L'estimation est bonne mais on peut voir des erreurs dans l'estimation atriale \tilde{A} au moment des complexes QRS.	86
3.6	Dictionnaire appris sur un mélange synthétique. Le dictionnaire ventriculaire est en haut, le dictionnaire atrial en bas. On observe une forte ressemblance des atomes ventriculaires.	87
3.7	ECG d'un patient malade sur le canal VR. On voit de grands pics sur la vue d'ensemble (en haut). Le premier pic est causé par une palpitation (au milieu : vue en gros plan). Le second est dû à un artefact de mesure (en bas : vue en gros plan).	89
3.8	Fibrillation observable sur ECG d'un patient malade en V1.	90
3.9	Dictionnaire ventriculaire appris sur le canal V1. On voit des traces de fibrillation sur le dernier atome.	90
3.10	ECG d'un patient malade en V4. On voit des irrégularités, mais il est difficile de déterminer ce qui provient de la fibrillation.	91
3.11	Dictionnaire atrial d'un patient malade en V4. Les atomes ne semblent pas présenter de trace d'activité ventriculaire résiduelle.	91
3.12	Estimation atriale \tilde{A} d'un patient malade en V4, avec le signal S (en haut) ou seule (en bas). L'amplitude de \tilde{A} est très faible par rapport à celle de S . Il reste d'importantes interférences ventriculaires dans \tilde{A}	92
3.13	Dictionnaire ventriculaire appris sur le canal VR. On reconnaît dans les troisième et quatrième atome les formes de palpitation et d'artefact observées dans la figure 3.7.	93
3.14	Distribution des amplitudes maximales de l'activité atriale estimée \tilde{A} sur chaque intervalle QT et TQ sur le canal V1 pour les méthodes d'apprentissage de dictionnaire et de soustraction de battement typique. La boîte contient 75% de la distribution avec la moyenne marquée d'un trait pointillé à l'intérieur. Les moustaches autour des boîtes contiennent 99% de la distribution. Les points aberrants hors des moustaches sont tracés individuellement. La figure du haut montre l'ensemble de la distribution et celle du bas uniquement le cœur.	95

List of Algorithms

1	$x = \text{MP}(s, \Phi), K$	24
2	$x = \text{OMP}(s, \Phi)$	24
3	$x = \text{LocOMP}(s, \Phi)$	33
4	$x = \text{LocGP}(s, \Phi)$	38
5	Calcul du vecteur propre principal v d'une matrice M par l'algorithme des puissances	55
6	$(\tilde{A}, \tilde{V}) = \text{separation}(S)$	82

Résumé

L'objet de cette thèse est d'apporter des réponses pratiques aux problèmes qu'on rencontre quand on souhaite utiliser des représentations parcimonieuses. Les représentations parcimonieuses consistent à approcher le signal comme une combinaison *linéaire* d'un faible nombre de vecteurs appelés *atomes* choisis dans un *dictionnaire*. Pour trouver une représentation basée sur peu d'atomes, on a besoin d'un large choix dans les atomes qu'on peut choisir. On utilise donc des dictionnaires *redondants*, contenant plus d'atomes que la dimension du signal.

On se trouve alors devant deux problèmes. Comment calculer une décomposition parcimonieuse d'un signal donné sur un dictionnaire donné? Comment trouver un dictionnaire qui permet la représentation parcimonieuse d'une certaine famille de signaux?

Le premier problème, le calcul d'une décomposition parcimonieuse, est NP-complet quand le dictionnaire est redondant. Plusieurs algorithmes approchés ont été proposés, ils sont encore presque tous trop coûteux pour être employés sur de grands signaux, et ceux qui passent à l'échelle calculent une approximation bien moins bonne que les algorithmes plus complexe. Nous proposons un nouvel algorithme, LocOMP, qui passe à l'échelle sans sacrifier la qualité d'approximation. LocOMP ne fonctionne que dans le cas où les dictionnaires sont *locaux*, c'est-à-dire que la longueur du support d'un atome est petite devant celle du signal.

Le second problème, l'apprentissage de dictionnaire, est encore plus complexe puisque la manière habituelle est le résoudre implique l'itération de plusieurs décompositions parcimonieuses successives. Nous proposons une amélioration d'un algorithme existant, l'algorithme d'Olshausen-Field, qui optimise le dictionnaire par descente de gradient projeté à pas fixe. Nous montrons comment alculer le pas optimal pour cette descente. Cette amélioration permet de converger plus vite vers un meilleur dictionnaire et simplifie également la prise en main de l'algorithme par l'utilisateur puisqu'il y a un paramètre de moins à régler.

Enfin, nous avons appliqué tous ces outils pour isoler les signaux de fibrillation atriale dans l'électrocardiogramme du patient. La fibrillation atriale est l'arythmie cardiaque la plus fréquente : les atria (ou oreillettes) tremblent au lieu de battre normalement. L'observation de ces tremblements dans l'électrocardiogramme (ECG) peut aider au diagnostic, pour détecter par exemple si la fibrillation peut être soignée par chirurgie ou pose d'un stimulateur cardiaque. Cependant la fibrillation est masquée dans l'ECG par l'activité ventriculaire qui assure le rythme cardiaque. Nous avons proposé un algorithme de séparation de sources qui extraie l'activité atriale de l'ECG. Cet algorithme est basé sur l'apprentissage d'un dictionnaire représentatif de l'activité atriale et un pour l'activité ventriculaire. La difficulté principale est de parvenir à apprendre les deux dictionnaires à partir du même ECG qui mélange les deux activités. Nous sommes parvenus à apprendre les dictionnaires alternativement.

Abstract

This thesis aims at providing practical solutions to the problems encountered in the application of sparse models. Sparse modelling consists in modelling the signal as a linear combination of a few atoms picked from a dictionary. One needs a wide choice of atoms to ensure only a few of them will be enough, so the dictionary is usually redundant, i.e. it contains more atoms than the dimension of the signal.

Two problems arise when using sparse representations. Given a signal and a dictionary, how to select the most relevant atoms? How to find a dictionary that is suitable for the

sparse representation of a given family of signals ?

The first problem is the sparse decomposition problem. It has been proven NP-Complete when the dictionary is redundant. Several suboptimal algorithms have been proposed, but most of them fail to treat large signals in a reasonable time, and the few scalable algorithms compute much worse approximations. Our first contribution is a new algorithm called LocOMP that achieves both scalability and good approximation quality. However, LocOMP requires the dictionaries to be local, i.e. the length of atoms has to be much smaller than the length of the signal.

Dictionary learning, the second problem, is even more complex as usual solving methods involve several sparse decompositions. We proposed an to improve an existing dictionary learning algorithm, the Olshasen-Field that optimizes the dictionary using fixed-step gradient descent. Our improvement consists in the computation of the optimal step of the descent. This speeds up the convergence of the algorithm and makes the tuning of the algorithm for any given application easier.

We also applied the proposed algorithms to a biomedical source separation application. Atrial fibrillation is the most common cardiac arrhythmia. In this condition the atria stop beating normally and start contracting erratically instead. Observing the fibrillation in the electrocardiogram (ECG) could help in the diagnosis and prescription, for example when deciding if and how surgery or pacemaking should be employed. However, the atrial activity is hidden by the ventricular activity in the ECG. We devised a source separation algorithm to extract the atrial activity from the ECG. This algorithm is based on dictionary learning. One dictionary is learnt for the ventricular activity and one for the atrial activity. The main obstacle with that approach was to learn both dictionaries from the ECG that only contains the mixture of the activities. We managed to overcome this by alternating the learning between the ventricular and atrial dictionaries.