

N° d'ordre: 3525

THÈSE

Présentée devant

devant l'Université de Rennes 1

pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITÉ DE RENNES 1
Mention TRAITEMENT DU SIGNAL ET TÉLÉCOMMUNICATIONS

par

Sylvain LESAGE

Équipe d'accueil : METISS/IRISA

École Doctorale : MATISSE

Composante universitaire : SPM

Titre de la thèse :

*Apprentissage de dictionnaires structurés pour la
modélisation parcimonieuse des signaux multicanaux*

soutenue le 3 avril 2007 devant la commission d'examen

MM. :	Christian	JUTTEN	LIS, Grenoble	Rapporteurs
	Bruno	TORRESANI	LATP, Marseille	
M. :	Jean-Jacques	FUCHS	IRISA, Rennes	Président
MM. :	Rémi	GRIBONVAL	IRISA, Rennes	Examineurs
	Pierre	VANDERGHEYNST	EPFL, Lausanne	
M. :	Frédéric	BIMBOT	IRISA, Rennes	Directeur de thèse

Remerciements

Merci en preMIER lieu à mes deux direCteurs de thèse, Frédéric Bimbot et Rémi Gribonval, chargés de recherche à l'IRISA, poUR la qualiTé de leur encadrement. J'ai du travailler ma détente verticale pOUr pouvoir aTtrAper au vol les idées que Frédéric a lancées, idées qui, une fois ramenées sur Terre, se sOnt avérées être les prIncipales contribuTions de la thèse. Frédéric a aussi été d'un soutien sans faille face à tous les tYPes de tracas qu'un thésArD peut rencontrer. RéMI m'a de soN côté toujours aidé à avancer dans cEt inconnu Peuplé de mathématiques, cOnservant coûte que coûte sa gentillesse et sa qUalité d'écoute. Notons que deux ou tRois auTres thèses seraient nécessaires pOUr suivre toutes les pistes qu'il a tracées.

Merci égalemeNt aux rapporteurs de cette thèse, Christian Jutten et Bruno Torrésani, pOUr l'aTtentIon très pointuE qu'ils oNt Porté à mon travail Et pour leurs remarques pertiNentes. Je suis aussi très reconnaissant à Jean-Jacques Fuchs D'Avoir accepté de présider le jury aiNsi qu'à Pierre Vanderghynts et JaLal Fadili qui ont biEn voulu examiner cette thèse.

Merci aux membres de L'équipe METISS de l'IRISA pOur tous les momeNts passés en leur compaGnie pendant leS trois dernières années. Merci donc, pêle-mêle, à Guillaume, AmadOu, Michael, Gilles, AhmaD, AlExey, LauREnt, DANiel, LorCan, Mikaël, MaThieu, SIMON, Sacha, Touria, Emmanuel, François, Pascal, Stéphanie, Sylvia, Sylvain, Anne, Benjamin, Sonia, Ewa, Carillo, Boris, Thomas, Robert, Marie-Noëlle et Olivier.

Enfin, merci à mes amis et à ma famille.

Table des matières

Table des matières	1
Introduction	9
I État de l’art	
-	
Décomposition parcimonieuse sur un dictionnaire redondant, apprentissage de dictionnaire et séparation de sources	25
1 Représentation parcimonieuse des signaux	27
1.1 Représenter des signaux grâce à un dictionnaire	28
1.1.1 Qu’est-ce qu’un signal ?	28
1.1.2 Synthèse de signaux à partir d’atomes	29
1.1.3 Analyse de signaux sur un dictionnaire	30
1.1.4 Cas particulier où le dictionnaire est une base	30
1.1.5 Cas d’un dictionnaire redondant	33
1.1.5.1 Exemple : le dictionnaire de Gabor	33
1.1.6 Représentation approchée des signaux	36
1.2 Principe de parcimonie de la représentation	37
1.2.1 Définition de la parcimonie	37
1.2.2 Pourquoi vouloir la parcimonie ?	37
1.2.3 Quelques notions d’optimisation	39
1.2.4 Critères de parcimonie	41
1.2.5 Résultats théoriques connus	43
1.3 Illustration expérimentale	44
1.3.1 Le dictionnaire doit être adapté à la nature des signaux	45
1.3.2 La redondance permet d’augmenter la parcimonie	46
1.3.3 Comparaison des critères de parcimonie	46
1.4 Conclusion et sommaire	48
1.4.1 Hypothèses de travail	48
1.4.2 Difficultés	49
1.4.3 Sommaire	50

2	Algorithmes de décomposition parcimonieuse	51
2.1	Algorithmes de poursuite	51
2.1.1	Matching Pursuit	52
2.1.2	Orthogonal Matching Pursuit	53
2.1.3	Résultats théoriques des algorithmes de poursuite	54
2.2	Programmation quadratique	54
2.2.1	Algorithme de point fixe : FOCUSS	54
2.2.2	Programmation quadratique	56
2.3	Seuillage alterné	57
2.3.1	Seuillage doux	57
2.3.2	Seuillage doux alterné	58
2.3.3	Seuillage itératif	59
2.4	Conclusion	60
3	Apprentissage de dictionnaire	61
3.1	Motivations	61
3.1.1	Espace de recherche du dictionnaire	62
3.1.2	Méthode d'optimisation du dictionnaire	62
3.1.3	Sommaire	63
3.2	Analyse en Composantes Principales	64
3.3	Analyse en Composantes Indépendantes	65
3.3.1	ACI par non-gaussianité	66
3.3.2	ACI par maximum de vraisemblance	67
3.4	Apprentissage de base pour la décomposition parcimonieuse exacte	68
3.4.1	Principe probabiliste : maximum de vraisemblance	69
3.4.2	Interprétation en tant que critère	69
3.4.3	Dérivation du critère	70
3.4.4	Algorithme de Bell-Sejnowski	70
3.4.5	Algorithme de Hyvärinen	70
3.4.6	Choix de la distribution des coefficients	71
3.5	Apprentissage de dictionnaire redondant pour l'approximation parcimonieuse	71
3.5.1	Interprétation probabiliste : maximum de vraisemblance	72
3.5.2	Mise à jour alternée des coefficients et du dictionnaire - calcul explicite des coefficients	73
3.5.3	Algorithme de Lewicki	76
3.5.4	Méthodes de Monte-Carlo	76
3.5.5	Algorithme EM	78
3.6	Conclusion	79
4	Décompositions parcimonieuses pour la séparation de sources sous-déterminée	81
4.1	Le problème de la séparation de sources	81
4.1.1	Configuration du mélange	82

4.1.1.1	Conditions favorables	82
4.1.1.2	Conditions difficiles	82
4.1.1.3	Autres difficultés	83
4.1.1.4	Configuration de notre étude	83
4.1.2	Modèle linéaire instantané	84
4.1.3	Connaissances sur les sources	85
4.2	Estimation des sources lorsque les modèles des sources sont connus	85
4.2.1	Estimation des paramètres de mélange	86
4.2.2	Parcimonie dans le plan temps-fréquence	86
4.2.2.1	Principe	86
4.2.2.2	Algorithme DUET	89
4.2.2.3	Extensions de l'algorithme DUET	89
4.2.3	Séparation de sources utilisant la parcimonie dans un dictionnaire	90
4.2.3.1	Mélange multicanal	90
4.2.3.2	Mélange monocanal	92
4.3	Conclusion	93
5	Analyse critique du modèle de dictionnaire non-structuré et proposition	95
5.1	Similarités et différences entre la séparation de sources et l'apprentissage de dictionnaire	95
5.1.1	Similarités	95
5.1.2	Différences	96
5.1.3	Complémentarité	97
5.2	Insuffisance de la modélisation	97
5.2.1	Les atomes appris sont significatifs...	98
5.2.2	... mais peuvent être factorisés	99
5.3	Proposition : la structuration du dictionnaire	99
 II Contributions		
-		
Modèle de dictionnaire structuré, décomposition parcimonieuse, apprentissage de dictionnaire structuré, séparation de sources		101
6	Exemple de structure : union de bases orthonormées	103
6.1	Mise à jour des bases par DVS	104
6.1.1	Apprentissage d'une seule base orthonormée	104
6.1.2	Apprentissage d'une union de bases orthonormées	106
6.1.3	Rôle du paramètre λ	107
6.2	Evaluation de la qualité de l'apprentissage du dictionnaire	108
6.2.1	Métriques	109
6.2.2	Intentions expérimentales	111
6.2.3	Influence de la taille de l'ensemble d'entraînement	111

6.2.3.1	Principe de l'expérience	111
6.2.3.2	Protocole détaillé	111
6.2.3.3	Résultats	112
6.2.4	Influence du dimensionnement du dictionnaire	114
6.2.4.1	Principe de l'expérience	114
6.2.4.2	Protocole détaillé	114
6.2.4.3	Résultats	115
6.2.5	Influence de la dimension des signaux	117
6.2.5.1	Principe de l'expérience	117
6.2.5.2	Protocole détaillé	117
6.2.5.3	Résultats	118
6.3	Analyse critique	120
7	Structure fondamentale : dictionnaire généré par déformation linéaire de motifs	123
7.1	Motivations	124
7.1.1	Problèmes du découpage en trames	124
7.1.2	Problème de la redondance au sein du dictionnaire appris	125
7.2	Modèle de signaux	126
7.2.1	Signaux temporels monocanaux	126
7.2.2	Signaux temporels multicanaux	127
7.2.3	Signaux image et vidéo	128
7.3	Modèle préliminaire : dictionnaire invariant par translation	129
7.3.1	Définition d'un atome et d'un motif	129
7.3.2	État de l'art	130
7.3.3	Schéma explicatif	130
7.3.4	Fonction de translation	131
7.4	Modèle structuré : dictionnaire généré par déformation linéaire de motifs	132
7.5	Déclinaison de diverses déformations	133
7.5.1	Modification d'amplitude	133
7.5.2	Translation	134
7.5.3	Convolution par un filtre RIF	134
7.5.4	Déformation d'un paramètre de phase	136
7.5.5	Expansion multicanale	137
7.5.6	Déformation par filtrage : interprétation pour la parole	139
7.6	Propriétés des fonctions de déformation	141
7.6.1	Paramètres continus ψ et paramètres discrets ϕ	142
7.6.2	Les déformations $F_{\psi,\phi}$ sont linéaires par rapport au motif	142
7.6.3	Simplifications lorsque la base de fonctions \mathbf{F}_ϕ est orthonormale	143
7.6.4	Mesure de la parcimonie	144
7.6.5	Propriété fondamentale des déformations linéaires : l'existence d'un adjoint	144
7.6.6	Composition de fonctions de déformation : stabilité et règle de construction	144

7.6.7	Composition de deux fonctions	144
7.6.8	Composition de N fonctions	146
7.6.9	Règle d'enchaînement de déformations	147
8	Algorithmes de décomposition sur un dictionnaire structuré	149
8.1	Pourquoi choisir Matching Pursuit ?	149
8.2	Matching Pursuit sur un dictionnaire structuré	151
8.3	Matching Pursuit invariant par translation	152
8.4	Matching Pursuit invariant par translation et déformation de phase . . .	153
8.5	Matching Pursuit multicanal et Demixing Pursuit	155
8.5.1	Matching Pursuit multicanal	156
8.5.2	Demixing Pursuit	158
8.6	Perspective : Matching Pursuit convolutif	159
8.7	Dictionnaires structuré et non-structuré : comparaison en termes de par- cimonie	161
8.7.1	Métrie	161
8.7.2	Intentions expérimentales	163
8.7.3	Principe de l'expérience	163
8.7.4	Protocole détaillé	164
8.7.5	Résultats attendus	165
8.7.6	Présentation des résultats	165
8.7.7	Conclusion	167
9	Apprentissage de dictionnaire structuré	169
9.1	Algorithmes existants pour l'apprentissage de dictionnaire structuré . .	170
9.1.1	Modèle de translation des motifs	170
9.1.1.1	Calcul alterné des coefficients et des motifs	171
9.1.1.2	Échantillonnage de Gibbs	172
9.1.2	Modèle de déformation des motifs	172
9.2	MoTIF : notre première approche dans le cas invariant par translation .	173
9.2.1	Principe	174
9.2.2	Algorithme	175
9.2.3	Avantages et limites	175
9.3	Algorithme de type K-SVD : notre principale contribution	177
9.3.1	Mise à jour alternée des paramètres et des motifs	177
9.3.2	Algorithme voisin proposé par Mailhé et Gribonval	178
9.3.3	Contribution majeure : algorithme de type K-SVD pour l'appren- tissage de dictionnaire invariant par déformation linéaire	179
9.3.3.1	Détails de la résolution	180
9.3.3.2	Conditions sur la matrice \mathbf{G}_{k,ϕ_k}	181
9.3.3.3	Intérêt de la linéarité de la déformation	181
9.3.3.4	Description de l'algorithme K-SVD pour l'apprentissage de dictionnaire invariant par déformation	181
9.3.4	Détail pour la translation et la déformation de phase	183

9.3.4.1	Dictionnaire invariant par translation	183
9.3.4.2	Dictionnaire invariant par translation et déphasage	184
9.4	Evaluation de la capacité à favoriser la parcimonie	185
9.4.1	Métrique	186
9.4.2	Intentions expérimentales	187
9.4.3	Principe de l'expérience	188
9.4.4	Protocole détaillé	188
9.4.5	Résultats attendus	189
9.4.6	Présentation des résultats	190
9.4.7	Conclusion	193
9.5	Evaluation de la qualité de l'apprentissage selon les déformations utilisées	193
9.5.1	Principe de l'expérience	194
9.5.2	Protocole détaillé	195
9.5.3	Présentation des résultats	195
9.5.3.1	Analyse du signal	196
9.5.3.2	L'invariance par translation est primordiale pour retrouver les notes	197
9.5.3.3	L'invariance par translation permet d'apprendre des motifs localisés en temps	200
9.5.3.4	Signaux MIDI : invariance par modulation ?	201
9.5.3.5	La déformation de phase est surtout utile pour de grands écarts entre atomes	201
9.5.3.6	L'apprentissage est invariant à la fréquence d'échantillonnage	203
9.6	Influence de la taille de l'ensemble d'apprentissage sur la qualité de l'apprentissage	206
9.6.1	Métrique	206
9.6.2	Principe de l'expérience	206
9.6.3	Protocole détaillé	207
9.6.4	Résultats	207
9.6.4.1	L'algorithme apprend les notes du signal d'apprentissage, indépendamment de la taille du signal	208
9.6.4.2	Seuls les longs signaux d'apprentissage permettent d'apprendre des dictionnaires généralisables	210
10	Séparation de sources : estimation des sources par dictionnaire structuré	215
10.1	Cas particulier de la séparation d'un mélange linéaire instantané	216
10.2	Perspective : formalisme global de séparation utilisant des dictionnaires structurés	218
10.3	Séparation multicanale sur des mélanges synthétiques	221
10.3.1	Métrique	221
10.3.2	Principe de l'expérience	222
10.3.3	Protocole détaillé	224

10.3.4	Résultats	225
10.3.5	Conclusion	229
10.4	Séparation monocanale sur des signaux synthétiques	230
10.4.1	Métrie	230
10.4.2	Principe de l'expérience	230
10.4.3	Protocole détaillé	231
10.4.4	Résultats	231
10.4.5	Conclusion et perspective	235
III Perspectives et conclusion		237
11 Problèmes ouverts et perspectives		239
11.1	Recouvrement des trames	240
11.2	Apprentissage d'une union de bases orthonormées : optimisation conjointe sur les coefficients et les atomes	242
11.3	Intérêt du modèle convolutif pour modéliser la parole	243
11.4	Apprentissage des dictionnaires adaptés aux sources sur le mélange	244
11.5	Apprentissage conjoint des conditions de mélange et des dictionnaires adaptés	246
11.6	Séparation de sources par Matching Pursuit sur un mélange convolutif	250
12 Perspectives pour d'autres applications		251
12.1	Parcimonie et codage	251
12.1.1	Considérations techniques sur le codage	251
12.1.2	Limites de l'approche par apprentissage de dictionnaire pour le codage	253
12.2	Débruitage par seuillage	255
12.3	Classification	256
12.4	Remix	256
Conclusion		259
A Annexes		261
A.1	Minimisation de l'erreur quadratique - Preuve du résultat (1.14)	261
A.2	Calcul du gradient de la vraisemblance par rapport au dictionnaire (équation (3.32))	262
A.3	Solution optimale du problème (3.41)	262
A.4	Apprentissage d'une base orthonormale - Preuve du résultat de 6.1.1	263
A.4.1	Annulation des gradients par rapport au dictionnaire et aux multiplicateurs de Lagrange	264
A.4.2	Introduction de la décomposition en valeurs singulières	264
A.4.3	Transformation en un problème équivalent	265
A.4.3.1	La matrice équivalente est bloc-diagonale	265

A.4.3.2	La matrice identité est la matrice équivalente qui minimise le critère	266
A.5	Condition d'identifiabilité des paramètres lors de la composition de N fonctions de déformation	266
A.6	Calcul du paramètre ψ optimal dans Matching Pursuit	267
A.7	Un vecteur et la partie imaginaire de sa transformée de Hilbert sont orthogonaux	268
Bibliographie		279

Introduction

Le traitement de signal numérique comporte schématiquement trois grandes familles d'outils, qui sont les outils d'analyse, de transformation et de synthèse pour manipuler les signaux. Les signaux peuvent être des signaux sonores, des signaux médicaux, de type électrocardiogramme, des signaux sismiques, mais aussi des images, des vidéos... Le point commun entre tous ces signaux est qu'ils ont été échantillonnés, c'est à dire qu'ils ne sont plus sous leur forme analogique initiale, mais sous une forme numérique. Ainsi, par exemple, un son peut être représenté par 44 100 échantillons par seconde, une image haute qualité par 2 millions de pixels (1 600 par 1 200), ou une vidéo par 24 images de 300 000 pixels (640 par 480) par seconde. Revenons aux outils de traitement de signal.

Les outils de la première famille servent à analyser les signaux, afin d'obtenir une **représentation** claire et compréhensible des données. L'analyse doit permettre de visualiser distinctement les caractéristiques principales du signal et de pouvoir les extraire et les manipuler facilement. Par exemple, pour un signal de musique, l'analyse peut consister à représenter les notes jouées sur une partition, à partir du son brut.

Les outils de la deuxième famille servent à transformer le signal. Ils suivent généralement une phase d'analyse, car ils manipulent une représentation du signal, et non le signal lui-même. Poursuivons avec l'exemple de la musique. Un outil de transformation peut par exemple consister à augmenter la hauteur des notes de la partition d'un ton. Le résultat de cette transformation est une autre partition, c'est-à-dire que la transformation se fait sur des représentations du signal et non sur le signal lui-même.

Les outils de la troisième famille servent à synthétiser les signaux, c'est-à-dire qu'ils génèrent des signaux à partir de leurs représentations. Dans l'exemple de la musique, si la représentation est une partition, alors la synthèse fournira en sortie le son correspondant à la partition. La synthèse est l'opération inverse de l'analyse.

En enchaînant les trois exemples, le son musical a donc été analysé sous la forme d'une partition, qui a ensuite été rehaussée d'un ton, puis synthétisée pour donner un nouveau son, plus aigu que le son initial.

Le cadre de ce travail est l'étude d'un modèle particulier de représentation des signaux, à savoir les **représentations parcimonieuses**. Le principe des représentations parcimonieuses est de décomposer un signal en un petit nombre d'éléments basiques. Ces éléments sont appelés des **atomes**, par analogie avec la matière qui peut être modélisée comme un assemblage d'atomes. Précisons qu'un atome est un signal, de la même nature que le signal original, mais représente uniquement une partie élémentaire d'un signal

complexe. Dans le cadre de la musique, les atomes peuvent par exemple être les notes, c'est-à-dire le son de chaque note, et pas la représentation symbolique. De façon plus imagée, le signal peut être vu comme un puzzle et les atomes comme les pièces du puzzle. Tout l'enjeu de la synthèse est d'assembler les pièces à la bonne place et dans le bon sens afin de reconstruire parfaitement le puzzle.

L'idée de parcimonie de la représentation signifie qu'à partir d'un ensemble conséquent d'atomes à disposition, le signal peut être reconstruit en utilisant uniquement quelques atomes, à savoir les plus appropriés au signal. L'ensemble des atomes est d'ailleurs appelé le **dictionnaire**, ce qui permet une autre analogie. En assimilant le signal à une phrase, et les atomes à des mots, la phrase peut être construite en piochant dans le dictionnaire les mots qui la composent.

Mais pour donner une idée de la difficulté, imaginons que quelqu'un prononce cette phrase, que le son est enregistré via un micro, et que l'objectif est de voir apparaître la phrase écrite sur l'écran d'un ordinateur. En suivant le principe des représentations parcimonieuses, la démarche consiste à décomposer le son de la phrase comme un ensemble de sons correspondant aux quelques mots prononcés.

Pour décomposer la phrase en mots, il faut tout d'abord posséder le dictionnaire qui contient la prononciation de tous les mots. Il faut ensuite pouvoir comparer chaque bout de la phrase à tous les mots du dictionnaire pour détecter les mots prononcés. Parmi toutes les instants dans la phrase et tous les mots du dictionnaire, il faut ensuite choisir lesquels ont le plus de chance de correspondre aux mots réellement prononcés. La phrase est enfin écrite à l'écran en rassemblant les mots sélectionnés. Notons que les techniques de reconnaissance automatique de la parole sont beaucoup plus sophistiquées que cette description rapide, et doivent surmonter de nombreux problèmes que nous n'évoquons pas ici.

Comment décomposer un signal avec un dictionnaire ?

Le premier grand problème des représentations parcimonieuses est de trouver la meilleure représentation d'un signal, c'est-à-dire le meilleur ensemble d'atomes et la meilleure façon de les combiner, pour un dictionnaire d'atomes donné. Si le nombre d'atomes est trop faible, il est impossible de décomposer le signal de façon exacte, car les atomes ne sont pas assez divers pour représenter toutes les facettes du signal. Une partie du signal ne peut donc pas être représentée, et la représentation utilisant les atomes n'est qu'une approximation, car elle contient une erreur.

Si le nombre d'atomes est exactement égal à une limite théorique que nous expliciterons par la suite, et s'ils sont linéairement indépendants, alors le signal se décompose de façon exacte et unique sur le dictionnaire, ce qui signifie qu'il existe un et un seul jeu d'atomes qui permet de représenter le signal sans erreur.

Enfin, si le nombre d'atomes est supérieur à la limite théorique, alors le dictionnaire est dit **redondant**, et il existe une infinité de combinaisons d'atomes différentes qui donne une décomposition exacte du signal. Il est donc nécessaire de définir un critère pour choisir parmi toutes les décompositions possibles. Le critère retenu dans cette thèse est la **parcimonie**, qui consiste à choisir la décomposition qui utilise le moins

d'atomes. La parcimonie est une caractéristique observée pour de nombreux signaux réels (ils se décomposent avec peu d'atomes pour un dictionnaire bien choisi). Le critère de parcimonie se justifie également au niveau conceptuel, puisqu'il promet la solution la plus "simple" à décrire.

Si l'objectif n'est plus de décrire exactement le signal, mais de l'approcher en acceptant une certaine erreur, c'est-à-dire de permettre qu'une partie du signal ne soit pas représentée, alors quel que soit le nombre d'atomes dans le dictionnaire, il existe une infinité de combinaisons d'atomes possibles. Pour choisir parmi ces combinaisons, il est encore une fois nécessaire de définir un critère de choix, qui tiendra compte cette fois-ci du niveau d'erreur souhaité, et sera un compromis entre une erreur faible et un nombre d'atomes faible. C'est un compromis car si aucun atome n'est utilisé, l'erreur est maximale et le signal n'est pas représenté du tout, tandis que plus le nombre d'atomes utilisés est grand, plus l'erreur peut être rendue petite.

Quel dictionnaire utiliser ?

Le deuxième grand problème des représentations parcimonieuses, qui est le sujet principal de ce travail, est le choix du dictionnaire d'atomes. Il faut en effet choisir un dictionnaire en adéquation avec le signal traité. Par exemple, en reprenant l'exemple de la reconnaissance de parole, si la phrase prononcée est en français, mieux vaut que le dictionnaire servant à la décomposition soit aussi en français, et pas en suédois. Un dictionnaire de suédois permettrait certainement de reconstruire la phrase, mais ne possédant pas les bons mots, il bricolerait une suite de mots suédois dont la prononciation est la plus proche de la phrase. Il utiliserait ainsi certainement plus de mots, pas au bon endroit, et le signal serait approché grossièrement.

Le dictionnaire peut être choisi parmi un ensemble de dictionnaires préexistants, ou alors être construit de toutes pièces à partir d'exemples des signaux à traiter. La construction automatique du dictionnaire à partir d'exemples de signaux est appelée **apprentissage** de dictionnaire. L'apprentissage peut dépendre uniquement des signaux, ou intégrer en plus des connaissances expertes sur la nature des signaux.

Par exemple, pour apprendre un dictionnaire de notes de piano (de sons de notes de piano, c'est-à-dire intégrant le timbre de l'instrument), la première méthode consiste à introduire des enregistrements de piano en entrée d'une boîte noire et de récupérer le dictionnaire en sortie, en espérant que les atomes correspondent aux sons des différents notes de la gamme. La deuxième méthode consiste à régler cette boîte noire pour que les atomes appris soient séparés d'un demi-ton. Le dictionnaire a alors plus de chances de correspondre à la gamme qu'avec la première méthode, puisque des informations sur la nature attendue des atomes a été intégrée à l'apprentissage. Bien sûr, si aucune information n'est disponible *a priori* sur la nature des signaux, il faudra utiliser la première méthode. L'élaboration de ces méthodes est au cœur de ce travail de thèse.

La **problématique** de la thèse présentée dans ce document est la suivante : "Comment utiliser l'information disponible *a priori* sur la nature des signaux à traiter afin de construire un dictionnaire bien adapté pour décomposer ces signaux de façon parcimonieuse ?"

Après cette phase introductive, présentons maintenant le cadre mathématique des représentations parcimonieuses.

Représentations parcimonieuses et algèbre linéaire

Nous avons expliqué que les outils d'analyse et de synthèse permettent de naviguer entre les signaux et leur représentation. La représentation sert à exprimer les signaux sous une autre forme, un autre point de vue, qui permet de les manipuler de façon plus efficace.

Introduisons quelques notations mathématiques. Un signal est un vecteur ligne, noté \underline{s} . Par la suite, toutes les lettres soulignées représenteront des signaux. La représentation, quant à elle, est un vecteur colonne, noté \mathbf{c} . Tous les vecteurs de coefficients seront notés en gras.

L'opérateur d'analyse est une fonction qui transforme \underline{s} en \mathbf{c} , et l'opérateur de synthèse transforme \mathbf{c} en \underline{s} .

Dans le cadre des représentations parcimonieuses, la synthèse consiste à exprimer le signal comme une combinaison d'atomes, que nous notons \underline{d}_k , où k indique le numéro de l'atome dans le dictionnaire. Nous notons \mathcal{D} le dictionnaire et K le nombre d'atomes dans le dictionnaire. L'opération de synthèse s'écrit :

$$\underline{s} = \sum_{k=1}^K c_k \cdot \underline{d}_k, \quad (1)$$

où c_k est le coefficient de la représentation \mathbf{c} associé à l'atome numéro k . Nous appellerons également \mathbf{c} jeu de coefficients ou vecteur de coefficients.

La synthèse consiste donc à faire la somme de tous les atomes du dictionnaire, pondérés par un coefficient associé. Si le coefficient est grand, l'atome est très présent dans le signal, s'il est faible alors l'atome compte peu dans le signal, et s'il est nul, alors l'atome ne contribue pas du tout à la construction du signal. Pour que les coefficients c_k soient comparables, les atomes \underline{d}_k sont contraints à être d'énergie 1. Cette convention est imposée jusqu'au chapitre 7, où nous présenterons un modèle d'atomes structurés incluant un paramètre d'amplitude. Jusqu'à ce chapitre, les atomes sont de norme unitaire.

La synthèse est une opération linéaire par rapport aux atomes, et par rapport aux coefficients. L'intérêt de la linéarité pour l'opération de synthèse est tout d'abord sa simplicité théorique et calculatoire, qui permet comme nous le verrons par la suite, d'élaborer des stratégies assez directes pour calculer les coefficients ou les atomes. La linéarité reflète ensuite assez fidèlement la réalité pour bon nombre de signaux. Ainsi, un son de piano est effectivement la somme des contributions de chacune des touches du piano. En enregistrant séparément chacune des notes du piano et en synthétisant ensuite un accord par la somme de quelques notes, alors le son obtenu sera très proche du son de l'accord joué réellement. La différence tiendra uniquement aux interférences physiques dues au fait que les cordes interagissent dans l'instrument. En première approximation, la linéarité est donc bien adaptée au traitement des signaux. Notons cependant que dans le cas des images, par exemple, un modèle de synthèse linéaire ne permet de traiter que

les images transparentes. En effet, l'opération d'occlusion, qui consiste à cacher un objet derrière un autre n'est pas linéaire. En utilisant le modèle linéaire, l'objet qui se trouve derrière est toujours visible.

L'opération d'analyse consiste quant à elle à trouver les coefficients \mathbf{c} à partir du signal. Si l'opération de synthèse est linéaire, en revanche, l'opération d'analyse peut être non-linéaire, ce qui est le cas dans le cadre de ce travail. En effet, nous l'avons déjà expliqué dans la phase d'introduction, le calcul des coefficients \mathbf{c} dépend du nombre d'atomes et du choix de décomposer exactement le signal ou seulement de l'approcher.

Pour expliciter ces différents cas, il faut introduire la notion de **dimension** des signaux. Par exemple, supposons que les signaux traités soient des signaux musicaux. Si un signal a une durée d'une seconde, ce qui correspond à 44 100 échantillons, alors la dimension de ce signal, notée L , est $L = 44\ 100$. Pour une image, la dimension sera le nombre de pixels. Pour une vidéo, la dimension sera le nombre de pixels dans chaque image multiplié par le nombre d'images.

Résolution d'un système linéaire

Nous avons évoqué dans la phase introductive une limite théorique autour de laquelle l'opérateur d'analyse prend des formes différents. Cette limite théorique correspond à $K = L$, c'est-à-dire au cas où le nombre d'atomes est égal à la dimension du signal à analyser.

En effet, si $K = L$, et à condition que l'ensemble des atomes forme une famille libre (chaque atome apporte une contribution originale, non présente dans les autres atomes), alors le dictionnaire peut être assimilé à une base de l'espace des signaux. L'opérateur d'analyse est alors linéaire et correspond à l'opérateur inverse de l'opérateur de synthèse. Les représentations sous la forme $\underline{\mathbf{s}}$ ou sous la forme \mathbf{c} deviennent équivalentes, puisqu'il est possible de passer de l'une à l'autre sans erreur et de façon unique.

Si $K < L$, alors dans le cas général, le signal ne peut pas être décomposé sans erreur sur le dictionnaire. Tous les atomes du dictionnaire sont utilisés, et les coefficients sont calculés de façon à minimiser l'énergie de l'erreur. Ce type d'opérateur d'analyse sert souvent pour la réduction de dimension, afin d'extraire uniquement les composantes les plus énergétiques du signal et de réduire la complexité, car les calculs sont plus légers en manipulant des données de taille K que des données de taille L . Notons que la minimisation de l'erreur quadratique donne un opérateur d'analyse linéaire.

Enfin, si $K > L$, nous avons dit que le dictionnaire prend le nom de dictionnaire redondant, et il existe une infinité de jeux de coefficients \mathbf{c} permettant de décomposer le signal de façon exacte. En effet, chaque sous-ensemble du dictionnaire de L atomes qui forme une base permet une décomposition exacte, comme nous l'avons vu dans les paragraphes précédents. Cela signifie notamment que l'opérateur de synthèse n'est pas inversible, et donc évidemment que l'opérateur d'analyse ne peut pas être obtenu en inversant l'opérateur de synthèse, comme c'était le cas lorsque le nombre d'atomes est égal à la dimension des signaux.

Pourquoi promouvoir la parcimonie ?

Afin de choisir parmi tous les jeux de coefficients un unique jeu de coefficients \mathbf{c} , il faut définir un critère permettant d'exprimer le type de coefficients que nous préférons. Parmi l'ensemble des critères possibles, nous nous intéressons à la famille des critères de parcimonie, qui consistent à sélectionner le jeu de coefficients qui reconstruit exactement le signal en utilisant le moins d'atomes, c'est-à-dire pour lequel le plus possible de coefficients c_k valent zéro.

Pourquoi vouloir utiliser le moins d'atomes possibles, et donc pourquoi choisir le critère de parcimonie ? Tout d'abord, comme pour la linéarité, une raison se trouve dans l'observation des signaux. Nous verrons par la suite que les signaux musicaux peuvent être décomposés en atomes correspondant à des fréquences pures de courte durée (assimilables en gros à des notes), et appelés atomes de Gabor. En construisant le dictionnaire d'atomes de Gabor pour qu'il soit une base, nous obtenons une décomposition exacte du signal. L'observation des coefficients obtenus fait apparaître que la plupart sont nuls, ou très faibles, et que seuls certains comportent beaucoup d'énergie. Ainsi, les signaux musicaux sont naturellement parcimonieux dans un dictionnaire d'atomes de Gabor. Les mêmes observations peuvent être effectuées sur d'autres types de signaux et d'atomes.

La deuxième raison est conceptuelle. En effet, pour un résultat équivalent, il est plus instinctif de vouloir représenter un objet par peu d'éléments que par beaucoup. Ce principe rappelle celui du rasoir d'Occam qui peut être exposé comme le fait de choisir, parmi deux explications équivalentes, la plus simple des deux.

Par exemple, pour représenter un signal musical, si le dictionnaire contient toutes les notes séparées, et en plus tous les accords possibles entre ces notes, il sera plus économique et plus logique de représenter le signal par la suite des accords joués que par la suite et la superposition de toutes les notes individuelles. Cette idée est reliée au principe de codage et de compression que nous décrirons dans la section suivante.

Enfin, rappelons que l'utilisation de la parcimonie est avant tout un moyen permettant de déterminer un unique jeu de coefficients parmi une infinité de possibles.

Quel outil d'analyse ?

L'opérateur d'analyse, dans le cas redondant, dépend du critère de parcimonie choisi, mais en général et plus particulièrement dans cette thèse, le critère choisi (qui dépend du nombre de coefficients non-nuls, ou d'une approximation de ce nombre) conduit à un opérateur d'analyse non-linéaire. Cet opérateur d'analyse est lourd à manipuler et trouver le meilleur jeu de coefficients est une tâche ardue, qui justifie les travaux menés dans le domaine des algorithmes de représentation parcimonieuse.

Rappelons enfin que lorsque l'objectif est d'approcher le signal en acceptant un certain niveau d'erreur, alors quel que soit le nombre d'atomes dans le dictionnaire, il est nécessaire de définir une stratégie pour choisir le jeu de coefficients \mathbf{c} (tous les jeux de coefficients donnent une approximation du signal, même si elle peut être extrêmement mauvaise). Pour ce problème d'approximation, la stratégie classique consiste à effectuer

un compromis entre le niveau d'erreur et le nombre d'atomes utilisés. Selon les stratégies, ce compromis peut être réglé de différentes façons, et selon les applications, une faible erreur sera privilégiée, tandis que dans un autre cas, l'erreur pourra être grande, à condition que seuls quelques atomes soient utilisés.

Ce problème d'approximation permet d'analyser un signal en imposant un certain niveau de parcimonie à sa représentation \mathbf{c} , quel que soit le dictionnaire utilisé, et même lorsque ce dictionnaire est une base. Cela signifie dans ce cas que la représentation exacte n'est pas satisfaisante et qu'il vaut mieux permettre une erreur faible si cela permet de diminuer sensiblement le nombre d'atomes utilisés. Cela rejoint notamment l'idée de compression de signaux avec perte, que nous exposons dans la section suivante, consacrée aux applications utilisant les représentations parcimonieuses.

Champs d'application des représentations parcimonieuses

Les représentations parcimonieuses sont précieuses pour traiter des tâches aussi diverses que la visualisation des données, la compression de signaux, la séparation de sources, ou la classification de signaux.

Visualisation des signaux

Premièrement, les représentations parcimonieuses sont utiles à des fins de visualisation. En effet, l'affichage des coefficients de la décomposition permet une analyse visuelle rapide quand seuls quelques coefficients sortent du lot. Ainsi, en reprenant l'exemple des atomes de Gabor, l'affichage des coefficients issus de l'analyse d'un son de violon fait apparaître au premier coup d'oeil une structure harmonique. En effet, nous verrons sur la figure 1.1 que les coefficients correspondant à une fréquence fondamentale, ainsi que les coefficients correspondant à toutes les fréquences multiples (octaves), comportent la majorité de l'énergie, et que les autres coefficients sont assez faibles. De façon générale, il est facile d'extraire une information si seul un petit nombre d'éléments a un comportement différent du reste des éléments.

Codage et compression

Comme nous l'avons mentionné précédemment, la parcimonie a un lien fort avec le codage. En effet, pour transmettre un signal d'un expéditeur à un destinataire, l'objectif du codage est généralement de décrire le signal de façon à ce que l'information qu'il contient (au sens de la théorie de l'information) soit transmise sans erreur, et de la façon la plus économique possible. En effet, les canaux de transmission ont un débit limité (l'information est mesurée en bits, et le débit est mesuré en bits par seconde) et plus les messages sont de poids faible, plus ils peuvent être transmis rapidement. D'un autre point de vue, plus les messages sont de poids faible, plus il est possible d'en transmettre un grand nombre simultanément.

Une technique classique de codage est de transformer le signal en une représentation équivalente, qui contient la même information, mais qui permet de la coder avec moins

de bits. Dans le cas des représentations parcimonieuses, le codage se fera alors sur les coefficients \mathbf{c} plutôt que sur le signal. Pour que cela soit intéressant, il faut que les coefficients “pèsent moins lourd” que le signal, ce qui est le cas quand la décomposition est parcimonieuse et que la plupart des coefficients sont nuls. Le nombre de coefficients non-nuls est assez corrélé avec la quantité d’information en nombre de bits. Si le dictionnaire d’atomes est connu par l’expéditeur et par le destinataire, alors le codage consiste uniquement à transmettre les numéros des atomes utilisés et les coefficients correspondants. Plus la décomposition est parcimonieuse, plus le codage est efficace.

En utilisant une approximation du signal, et non plus une décomposition, alors la compression sera une compression avec pertes, ce qui signifie que le destinataire ne pourra pas reconstruire exactement le signal. En revanche, la compression sera beaucoup plus forte, et le message envoyé de poids potentiellement beaucoup plus faible. Le compromis entre l’erreur de reconstruction et le nombre d’atomes utilisés est alors du même ordre que le compromis débit-distorsion classique en codage. Ce compromis vise à minimiser le débit (le nombre de bits par secondes) et la distorsion (le “taux” d’erreur dans le message transmis) selon un équilibre donné.

Séparation de sources

Le problème de la séparation de sources peut également être abordé avec le concept de parcimonie. Il consiste à traiter un signal que nous supposons généré par le mélange de plusieurs signaux généralement indépendants entre eux, appelés sources, afin d’extraire ces sources du mélange une à une. Il peut s’agir de traiter l’enregistrement d’un chanteur accompagné d’une guitare pour extraire d’un côté la guitare seule, et de l’autre la voix seule. Cette tâche est appelée “démixage” dans le cas sonore, par opposition au mixage effectué en studio d’enregistrement. La séparation consiste à discriminer les formes des sources, qui doivent être différentes si les sources sont différentes.

De façon analogue au dictionnaire redondant, connaître un seul signal de mélange et chercher deux sources est un problème redondant, au sens où il existe une infinité de façons de découper le mélange en deux sources. En revanche, une seule solution correspond aux sources originales. Le défi consiste à trouver cette solution parmi l’ensemble des possibles. Le critère de parcimonie peut encore une fois être utilisé à des fins d’unicité de la solution.

En supposant connaître un dictionnaire d’atomes modélisant fidèlement (*i.e.* avec parcimonie) les sons de guitare, et un autre dictionnaire d’atomes modélisant fidèlement la voix, et en supposant que ces deux types d’atomes soient suffisamment différents l’un de l’autre, alors le mélange de guitare et de voix peut être séparé efficacement. Il suffit pour cela de décomposer tout d’abord le signal sur les deux dictionnaires mis en commun, puis, parmi les atomes actifs, utiliser d’un côté les atomes de type guitare pour reconstruire la source de guitare, et de l’autre les atomes de type voix pour reconstruire le chant.

Classification de signaux

Enfin, la parcimonie peut servir également à classifier des signaux. Supposons qu'un ensemble de signaux a été découpé en sous-ensembles de signaux, appelés classes, qui regroupent les signaux les plus proches entre eux. Prenons l'exemple d'enregistrements d'instruments solo, où chaque classe correspond à un instrument. La classification consiste à choisir dans quelle classe placer un nouveau signal arrivant.

Grossièrement, selon le même principe que la séparation de sources, il suffit de connaître un dictionnaire d'atomes adapté à chaque classe de signaux. La classification se fait alors en décomposant le nouveau signal sur l'union de tous les dictionnaires, puis à attribuer ce signal à la classe qui a fourni le plus d'atomes dans la décomposition.

Problèmes et difficultés liés à l'utilisation de la parcimonie

Nous venons d'exposer quelques applications pour lesquelles l'appel aux représentations parcimonieuses est utile. La manipulation de ces représentations n'est cependant pas toujours aisée, en raison tout d'abord de la difficulté à implémenter l'opérateur d'analyse, comme nous l'avons vu auparavant. Cette difficulté est théorique et dépend du critère de parcimonie choisi, mais aussi pratique, car les stratégies de mise en oeuvre de l'opérateur d'analyse sont en général coûteuses.

Afin d'optimiser la parcimonie de la représentation, nous avons mentionné l'intérêt de bien choisir le dictionnaire. Faire ce choix est un problème à part entière.

Enfin, nous avons exposé le principe de séparation de sources basé sur la discrimination des formes des sources. Un autre type de séparation se base sur la disposition spatiale de chacune des sources dans le mélange (grossièrement, rapport d'intensités gauche-droite dans un mélange stéréo). Un défi est d'unifier les deux approches pour augmenter les chances de discriminer les sources.

Problème du calcul des décompositions parcimonieuses

Nous nous intéressons dans ce travail au problème de l'approximation d'un signal en utilisant un dictionnaire redondant. L'opérateur d'analyse est en général non-linéaire, et peut prendre des formes diverses selon la façon d'intégrer la contrainte de parcimonie. Pour chacune des formulations de la contrainte de parcimonie, diverses stratégies peuvent être utilisées pour obtenir les coefficients.

Dans certains cas simples, mais pas forcément efficaces en termes de parcimonie, l'opérateur d'analyse est une fonction linéaire et les coefficients peuvent être calculés facilement et de façon exacte.

En revanche, dans des cas plus réalistes où l'objectif de parcimonie est plus affirmé, l'opérateur est non-linéaire, et ne peut être appliqué directement. Il faut alors déployer des stratégies algorithmiques complexes et parfois sous-optimales pour estimer les coefficients.

Globalement, le problème peut être posé comme la minimisation d'un critère par

rapport aux coefficients :

$$\mathbf{c}^{\text{opt}} = \arg \min_{\mathbf{c}} \mathcal{C}(\underline{s}, \mathcal{D}, \mathbf{c}) \quad (2)$$

Le choix de la contrainte de parcimonie, l'expression de l'opérateur d'analyse, l'analyse théorique de l'existence et de l'unicité de coefficients optimaux, et l'élaboration de stratégies de calcul de ces coefficients sont des problèmes ouverts. De nombreuses solutions ont été proposées pour ces problèmes, mais ils restent des champs d'investigation pour les chercheurs du domaine.

Au delà de ces questions théoriques, la question pratique de la complexité algorithmique des méthodes servant à les résoudre est cruciale, car elle conditionne la possibilité même d'évaluer les solutions sur des exemples.

Ainsi, si la complexité n'est pas handicapante pour traiter des exemples jouets, en revanche, le problème devient critique pour manipuler des signaux réels. Sans entrer dans les détails des algorithmes, ni donner leur complexité algorithmique, nous assurons qu'il est crucial d'utiliser des algorithmes rapides, même s'ils sont sous-optimaux, uniquement afin de pouvoir expérimenter.

Dans ce document, nous exposons les principaux travaux consacrés au problème des décompositions parcimonieuses, mais ce n'est pas le centre du travail.

Le problème du choix du dictionnaire

Au delà des problèmes liés aux décompositions parcimonieuses dans un dictionnaire donné, un problème également largement traité par la communauté depuis environ 1995 est celui du choix du dictionnaire. Plus précisément, nous évoquerons le problème de la construction du dictionnaire le plus adapté aux signaux à décomposer, c'est-à-dire permettant la décomposition la plus parcimonieuse des signaux sur le dictionnaire.

Comme précédemment, chercher le dictionnaire le plus "adapté" à une classe de signaux n'est pas défini de façon unique, et dépend notamment de la façon dont la contrainte de parcimonie est exprimée. Le problème pourra s'écrire sous la forme d'une minimisation par rapport au dictionnaire :

$$(\mathcal{D}^{\text{opt}}, \mathbf{c}^{\text{opt}}) = \arg \min_{\mathcal{D}, \mathbf{c}} \mathcal{C}(\underline{s}, \mathcal{D}, \mathbf{c}) \quad (3)$$

Pour une formalisation donnée du problème d'adaptation, diverses stratégies peuvent être adoptées pour calculer le dictionnaire optimal.

Il faut noter que de nombreuses méthodes font appel à une étape de décomposition parcimonieuse. Cette étape peut revenir de nombreuses fois dans un algorithme d'apprentissage, et la complexité de cet algorithme d'apprentissage dépend donc directement de la complexité de la décomposition.

Prenons l'exemple des signaux de parole. Supposons que la taille des atomes est de l'ordre de la durée de stationnarité, c'est-à-dire environ 40 ms ou environ 1 800 échantillons, pour une fréquence d'échantillonnage de 44 100 Hz. Pour que le dictionnaire soit redondant, le nombre d'atomes doit être au moins égal à la dimension. Il faut donc plus de trois millions d'échantillons pour décrire les atomes. Il est nécessaire de disposer

de méthodes rapides pour traiter de telles données. Un allègement des calculs peut cependant intervenir en structurant le dictionnaire.

Le défi de l’incorporation de structure

De nombreuses méthodes très générales permettent d’apprendre un dictionnaire de façon automatique à partir de signaux d’entraînement. En revanche, elles sont peu nombreuses à prendre en compte les informations de plus haut niveau sur la nature intrinsèque des signaux. Pour des raisons d’utilisation de connaissances *a priori* sur les signaux et de coût de calcul, il est intéressant de structurer le dictionnaire comme :

$$\mathcal{D} = F(\mathcal{M}) , \quad (4)$$

où \mathcal{M} est un dictionnaire de motifs – des bouts élémentaires de signal – qui sont déformés selon des fonctions prédéfinies pour engendrer les atomes du dictionnaire \mathcal{D} .

Pourquoi la structure ?

Connaissances : Donnons un exemple des informations de haut niveau disponibles *a priori*, où les signaux traités sont des photos, et les atomes des objets élémentaires de la photo. Imaginons qu’un des atomes soit un visage. Si les photographies proviennent d’un photomaton, la position du visage dans l’image, sa taille et son inclinaison seront à peu près les mêmes pour toutes les photos. En revanche, si les photos ont été prises dehors, sans aucun dispositif particulier, alors la position, l’inclinaison et la taille du même visage dépendent de l’endroit où se trouve le photographe et de la façon dont il tient son appareil. Il faudra donc pouvoir décrire le visage, dans le dictionnaire, quelque soit sa position, sa taille et son inclinaison dans la photo à décomposer.

Savoir comment ont été générés les signaux et quels sont les propriétés d’invariance des atomes est donc une information utile, qui peut être disponible de façon experte. Ces propriétés peuvent être déduites également de données d’entraînement, mais de façon incomplète ou approximative, et sous forme brute. Il peut être souhaitable de les intégrer de façon plus formelle dans le modèle de création du dictionnaire. Les visages sont alors inclus dans le dictionnaire de motifs \mathcal{M} , et les déformations d’inclinaison, de taille et de position dans la fonction F .

Coût : L’utilisation de la structure permet de factoriser le dictionnaire et de réduire le nombre de paramètres libres. C’est une contrainte sur la forme des atomes, qui suivent le modèle de motifs déformés, mais elle permet de réduire le coût de manipulation du dictionnaire.

Le grand problème pour lequel peu de solutions ont été proposées, et qui est la motivation principale de notre travail de thèse, concerne l’utilisation de toutes les informations disponibles sur les sources, que ce soit des informations de haut niveau sur leur nature ou des informations brutes sous la forme de données d’entraînement, pour l’apprentissage de dictionnaire.

Utiliser les modèles parcimonieux pour la séparation de sources

Nous avons exposé le problème de la séparation de sources monocanale où un capteur enregistre plusieurs sources. La méthode classique utilisant la parcimonie repose sur la connaissance d'un dictionnaire "adapté" à chaque source, au sens de l'adaptation exposée dans la section précédente. Jusqu'à maintenant, les dictionnaires utilisés pour la séparation sont des dictionnaires analytiques qui ont été choisis *a priori* en connaissant les types de sources à séparer.

Utiliser des méthodes d'apprentissage de dictionnaire à partir d'exemples des sources pour construire les dictionnaires adaptés est un problème ouvert qui n'a pas encore été traité. Il est restreint, évidemment, au cas où les sources peuvent être captées séparément dans un premier temps pour effectuer l'apprentissage. Nous proposons d'étudier le problème dans cette thèse, en espérant améliorer la qualité de la séparation par rapport à l'utilisation de dictionnaires analytiques.

Au delà du problème de séparation de sources monocanale, que nous avons exposé plus haut, nous souhaitons utiliser la parcimonie pour améliorer la séparation de mélanges multicanaux. Un mélange multicanal correspond à un ensemble de plusieurs capteurs qui enregistrent chacun un mélange monocanal différent des sources.

Géométriquement, la différence de position et d'orientation des capteurs permet de disposer d'informations spatiales sur les sources. Prenons l'exemple simple d'un enregistrement musical stéréo. Si une source émet près du micro gauche, alors sa contribution sera plus forte sur le canal gauche que sur le canal droit. Si une autre source émet à mi-distance entre les deux micros, sa contribution sera similaire sur les deux canaux. Ainsi, en supposant que les sources sont immobiles, la séparation peut consister à se focaliser sur une partie de l'espace (dans l'exemple, de l'éventail gauche-droite) pour extraire une source, et à se focaliser sur une autre partie de l'espace pour une autre source.

Des techniques de séparation basées sur ce principe existent pour la séparation de mélanges multicanaux, mais utilisent des dictionnaires analytiques. Pour les mêmes raisons que dans le cas monocanal, nous supposons que l'utilisation de dictionnaires appris sur des exemples des sources, quand ils sont disponibles, peut aider à augmenter la qualité de la séparation multicanale.

Problématique

Nous venons de présenter les grands enjeux dans le domaine des décompositions parcimonieuses. Le premier concerne la conception et les méthodes d'application de l'opérateur d'analyse, à dictionnaire fixé. Le deuxième enjeu est celui du choix du dictionnaire, piloté par les informations connues *a priori* sur les signaux. Le troisième enjeu concerne les modalités d'application des décompositions parcimonieuses pour la séparation de sources. Enfin le dernier enjeu est celui de la complexité des méthodes de résolution. Parmi ces quatre enjeux majeurs, nous nous focalisons en priorité sur le deuxième.

La problématique de cette thèse est donc de savoir comment utiliser des informations expertes sur la nature des signaux conjointement avec des données d’entraînement afin d’élaborer le dictionnaire permettant la décomposition la plus parcimonieuse des signaux.

Plan et contributions de ce travail

Le document est divisé en trois parties. La première partie est consacrée à l’état de l’art concernant les problèmes mentionnés précédemment. Dans la deuxième partie, nous exposons les contributions de notre travail, qui apportent une réponse à la problématique posée. La dernière partie comprend un bilan des évaluations expérimentales présentées tout au long du document, ainsi qu’une présentation des perspectives de recherche, et une conclusion.

Au cours de ce travail, nous avons étudié les méthodes existantes pour les problèmes mentionnés précédemment, en nous focalisant sur les dictionnaires redondants pour l’approximation de signaux.

Après avoir pointé certaines lacunes du modèle classique de décomposition parcimonieuse sur un dictionnaire non-structuré, nous proposons une spécialisation de la définition d’atome, en **introduisant les concepts de motif et de déformation linéaire** pour donner aux atomes une structure semi-paramétrique qui permet d’inclure des hypothèses sur la nature des signaux.

En utilisant la propriété d’adjonction des déformations linéaires, nous montrons l’intérêt d’adapter les algorithmes Matching Pursuit pour la décomposition des signaux, et K-SVD pour l’apprentissage du dictionnaire.

Nous proposons également un nouvel éclairage sur la séparation de sources en utilisant l’algorithme Matching Pursuit et des déformations adaptées.

État de l’art

Nous exposons dans la première partie du document notre vision de l’état de l’art concernant le modèle de décomposition parcimonieuse (chapitre 1), les algorithmes de décomposition parcimonieuse pour un dictionnaire connu (chapitre 2), les algorithmes d’apprentissage de dictionnaire à partir des données (chapitre 3) et la séparation de sources (chapitre 4).

Pour conclure cet état de l’art, dans la section 5.1, nous proposons d’établir une mise en parallèle rétrospective entre le traitement des problèmes d’apprentissage de dictionnaire et de la séparation de sources multicanale et les algorithmes d’Analyse en Composantes Indépendantes (présentés en section 3.3).

Dans la section 5.2, nous pointons les insuffisances du modèle de dictionnaire non-structuré, qui proviennent notamment de l’approche trop générale de l’apprentissage de dictionnaire. En effet, prenons l’exemple de signaux temporels. Les méthodes classiques ne prennent pas en compte leur nature intrinsèque, selon laquelle un élément récurrent du signal se retrouvera à plusieurs instants, pas nécessairement réguliers. Pour modéliser

finement les signaux temporels, les atomes devraient donc pouvoir être placés à tout instant du signal, ce qui correspondrait à un dictionnaire invariant par translation.

Or, de façon classique, selon la technique choisie, si les atomes appris sont très longs, alors plusieurs d'entre eux représentent le même objet à des instants différents ce qui n'est pas économique, ou alors, si les atomes appris sont courts car appris sur des trames découpées arbitrairement dans le signal, alors ils ne parviennent pas à refléter fidèlement les formes typiques des signaux d'entraînement.

Dans la section 5.3 nous expliquons donc notre proposition d'un modèle de structuration des atomes pour spécialiser les approches d'apprentissage en fonction des signaux traités. Le modèle d'atomes structurés est développé dans la deuxième partie du document.

Nouveau modèle d'atomes structurés

Nous proposons dans le chapitre 7 un modèle d'atome structuré, qui est une restriction du modèle général d'atome, et qui est plus adapté au traitement des signaux temporels que le modèle non-structuré, dans le cadre de l'apprentissage de dictionnaire. La définition classique de l'atome est scindée en deux pour donner l'atome structuré : nous définissons le concept de motif, qui est un élément de base du signal, généralement de petite taille, et celui d'atome structuré qui est un signal formé à partir d'un motif et de paramètres divers.

Le modèle prend en compte l'invariance par translation permettant de placer les motifs à tout instant du signal. Au delà de l'invariance par translation, il permet d'inclure des connaissances *a priori* sur le type de signaux traités par l'application de fonctions de déformation sur le motif (phase, convolution par un filtre, rotation, ...), en vue de créer des atomes semi-paramétriques. Le dictionnaire peut être résumé par un dictionnaire de motifs et un jeu de déformations. Cette structure du dictionnaire diminue la redondance entre atomes, ce qui présente un intérêt pour la transmission du dictionnaire à des fins de codage.

Nous exposons les résultats expérimentaux qui montrent l'intérêt de structurer le dictionnaire pour améliorer la parcimonie de la décomposition.

Adaptation des algorithmes de décomposition et d'apprentissage

Dans le chapitre 8, nous expliquons notre choix d'étudier l'algorithme Matching Pursuit pour les décompositions parcimonieuses, et nous présentons son adaptation au modèle d'atomes structurés. En utilisant la propriété d'adjonction des déformations linéaires qui sont appliquées aux motifs, nous montrons que Matching Pursuit peut notamment traiter des atomes convolutifs et des atomes multicanaux.

Dans le chapitre 9, nous présentons deux algorithmes d'apprentissage de motifs. Le premier, appelé MoTIF, correspond à notre première approche du problème de dictionnaire invariant par translation. Le second, qui est dérivé de l'algorithme K-SVD, permet l'apprentissage de motifs sous le nouveau modèle d'atomes déformés. Là encore, l'algorithme K-SVD est particulièrement adapté à l'utilisation de déformations linéaires

grâce à l'existence d'un adjoint.

Nous montrons expérimentalement l'apport de la structuration en termes de performances de modélisation du dictionnaire appris. Nous étudions également l'influence de la taille du dictionnaire et de l'ensemble des signaux d'apprentissage sur la qualité de l'apprentissage et sur la généralisation du dictionnaire appris à d'autres signaux de la même classe.

Séparation de sources

Dans le chapitre 10, nous expliquons pourquoi le modèle d'atomes structurés est intéressant pour la séparation de sources, et comment l'intégrer à des algorithmes de séparation. Nous présentons deux algorithmes de séparation de sources basés sur l'algorithme Matching Pursuit, dans le cas monocanal et dans le cas multicanal, pour des mélanges sous-déterminés (plus de sources que de capteurs). Les deux algorithmes reposent sur l'existence de données d'apprentissage des différentes sources au préalable. Pour la séparation de sources multicanale, nous supposons également que les paramètres de mélange sont connus, et nous n'étudions que le processus d'estimation des sources en lui-même.

Conclusions et perspectives

Nous dressons dans la troisième partie un bilan des expériences exposées tout au long des chapitres afin de mettre en lumière les principales avancées qu'a permises le modèle d'atome structuré, et d'illustrer ses limites actuelles. Nous présentons également les problèmes de nature théorique ou algorithmique que nous n'avons pu résoudre. Nous terminons ce travail par une conclusion et un exposé des perspectives théoriques, algorithmiques et applicatives.

Première partie

État de l'art

-

Décomposition parcimonieuse sur
un dictionnaire redondant,
apprentissage de dictionnaire et
séparation de sources

Chapitre 1

Représentation parcimonieuse des signaux

Le chapitre qui s'ouvre est consacré à la définition des représentations parcimonieuses de signaux dans un dictionnaire connu et à l'exposition des enjeux et des problèmes qui y sont associés.

Dans un premier temps, nous exposerons, sans évoquer la parcimonie, l'intérêt de représenter les signaux sous une autre forme et nous présenterons la synthèse linéaire à base de dictionnaire, en définissant les notions de signal, d'atome, de dictionnaire et de coefficients. Après deux parties consacrées à la redondance du dictionnaire, et à l'approximation linéaire, nous exposerons les difficultés qui en découlent, concernant le choix d'une représentation parmi toutes les représentations possibles dans un même dictionnaire. Cela nous permettra d'introduire la deuxième section de ce chapitre, consacrée à la parcimonie.

Dans cette deuxième section, après avoir défini la parcimonie, qui rappelons-le, consiste à utiliser seulement une petite partie des atomes du dictionnaire pour décrire un signal, nous expliquerons pourquoi le critère de parcimonie a été retenu pour décider quelle représentation choisir dans un dictionnaire. De façon mathématique, il s'agit de trouver les coefficients \mathbf{c}^{opt} qui optimise un critère $\mathcal{C}(\underline{s}, \mathbf{c}, \mathcal{D})$ favorisant la parcimonie :

$$\mathbf{c}^{\text{opt}} = \arg \min_{\mathbf{c}} \mathcal{C}(\underline{s}, \mathbf{c}, \mathcal{D}) , \quad (1.1)$$

où rappelons-le, \underline{s} désigne le signal à décomposer, \mathbf{c} les coefficients de la décomposition et \mathcal{D} le dictionnaire d'atomes.

Nous définirons les notions de critère, de problème, de solution et d'algorithme. Nous déclinerons différentes mises en œuvre de la parcimonie au sein d'un critère, et donnerons quelques résultats théoriques établis pour les critères exposés.

Enfin, nous rassemblerons les hypothèses de travail, et listerons les difficultés à surmonter, et conclurons sur un sommaire présentant les chapitres suivants.

1.1 Représenter des signaux grâce à un dictionnaire

L'analyse des signaux pour les représenter sous une forme différente est une manière de visualiser les composantes contenues dans ces signaux, afin de comprendre, par exemple, comment ils ont été formés ou de jauger les informations qui pourront en être extraites.

L'analyse d'un signal sert également à obtenir un objet, la représentation, plus facile à manipuler et permettant d'accomplir des tâches de façon plus intuitive et rapide qu'en manipulant les signaux. Mais avant de voir comment effectuer leur analyse, donnons tout d'abord une définition des signaux.

1.1.1 Qu'est-ce qu'un signal ?

Nous avons déjà défini dans l'introduction que les signaux sont notés \underline{s} . La lettre n'est pas importante, en revanche, le fait de souligner la lettre désignera toujours un signal, tout au long de ce document.

Nous utilisons cette notation pour tous types de signaux. Ainsi, nous incluons les signaux temporels, qui, comme par exemple les enregistrements musicaux ou les électrocardiogrammes, comptent une seule dimension temporelle, les images, qui comptent deux dimensions spatiales, ou encore les vidéos, qui comptent deux dimensions spatiales et une dimension temporelle. Notons que par vidéo, nous ne désignons pas l'image et le son, mais uniquement l'image, ou plutôt la suite temporelle d'images.

Pour tous ces signaux, \underline{s} est un vecteur horizontal réel de taille L : $\underline{s} \in \mathbb{R}^{1 \times L}$ ¹, où L est la dimension du signal. Ce vecteur est obtenu directement par la suite des valeurs échantillonnées dans le cas des signaux temporels mono-dimensionnels. Pour les images, le vecteur est obtenu par réordonnement des pixels. Ainsi, une image carrée de 12 pixels par 12 pixels donnera un vecteur \underline{s} de taille 144. Le vecteur est obtenu par le même type de stratégie pour les vidéos.

Les signaux sont supposés réels pour un souci de simplicité et parce que les signaux complexes ne sont pas au cœur de nos préoccupations. De même, ils sont de dimension finie pour plus de simplicité. Ces deux restrictions permettent de couvrir un grand nombre de signaux, qui suffisent à notre étude. Son extension aux signaux complexes et de dimension infinie pourra être envisagée, mais nous ne l'évoquerons plus par la suite.

Dans le cas des enregistrements multicanaux, comme par exemple l'enregistrement stéréo de musique, l'électrocardiogramme (12 pistes), ou l'enregistrement de données sismiques sur plusieurs sites simultanément, le signal obtenu est multicanal. Cela signifie que chaque canal contient un signal unidimensionnel. Pour désigner ces signaux multicanaux, nous utilisons la notation $\underline{\mathbf{S}}$, et nous notons \underline{s}_m le signal présent sur le m^e canal. Là encore, nous supposons que le nombre de canaux, noté M , est fini. Le signal multicanal $\underline{\mathbf{S}}$ peut être vu comme une matrice de taille $M \times L$.

Nous avons choisi de représenter les signaux comme des vecteurs horizontaux afin de suivre la convention adoptée par la communauté de l'ACI

¹Même pour les vecteurs, nous utilisons la notation matricielle pour différencier vecteurs horizontaux et verticaux (la taille $M \times N$ indique qu'une matrice a M lignes et N colonnes).

et de la séparation de sources. Dans la communauté de la parcimonie, la convention est au contraire de noter les signaux verticalement. Nous espérons que les lecteurs de cette communauté ne seront pas trop dérangés par les notations.

Après cette définition des signaux, intéressons-nous à leur représentation sur un dictionnaire d'atomes, et en premier lieu, à l'opérateur de synthèse qui permet de générer les signaux linéairement à partir des atomes.

1.1.2 Synthèse de signaux à partir d'atomes

Nous avons expliqué dans l'introduction l'intérêt que peut avoir la représentation d'un signal sur un dictionnaire d'atomes. Rappelons que c'est un modèle de décomposition du signal en sous-parties élémentaires appelées atomes. Ces atomes peuvent refléter différents niveaux de description. En reprenant l'exemple de la musique, les atomes peuvent être des fréquences pures, des notes, qui sont combinaisons complexes de fréquences contenant une information de hauteur et de timbre, ou encore au niveau supérieur des accords qui sont des combinaisons de notes.

La représentation est définie par un opérateur de synthèse qui transforme un jeu de coefficients c_k en un signal \underline{s} . L'opérateur de synthèse correspondant à la représentation sur un dictionnaire génère le signal comme combinaison linéaire d'atomes :

$$\underline{s} = \sum_{k=1}^K c_k \cdot \underline{d}_k \quad (1.2)$$

Les notations ont déjà été utilisées dans l'introduction. Définissons les objets en présence de façon plus précise.

Les atomes $\underline{d}_k \in \mathbb{R}^{1 \times L}$ sont des signaux de dimension L identique à la dimension du signal \underline{s} . Ils sont regroupés au sein d'un dictionnaire $\mathcal{D} = \{\underline{d}_k, 1 \leq k \leq K\}$. Nous imposons, dans la première partie de la thèse, que les atomes \underline{d}_k soient de norme unité :

$$\|\underline{d}_k\|_2 = 1 . \quad (1.3)$$

et nous imposons également, pour plus de simplicité, que le dictionnaire est de rang plein, ce qui signifie que la famille des atomes est libre si le nombre d'atomes K est inférieur ou égal à la taille des atomes L , et qu'il existe une famille libre de L atomes dans le dictionnaire si le nombre d'atomes est supérieur à L . Rappelons qu'une famille de K vecteurs est libre si et seulement si aucune combinaison linéaire de $K - 1$ vecteurs de la famille n'est égale au vecteur non utilisé.

Dans la première partie de cette thèse, consacrée à l'état de l'art, nous supposons que le nombre d'atomes est fini. Nous relâcherons cette contrainte dans la deuxième partie, et le dictionnaire pourra comprendre une infinité d'atomes. Pour le moment, K est fini et les atomes peuvent être indicés par k .

Puisque le nombre d'atomes est fini, nous pouvons écrire le dictionnaire sous la forme

d'une matrice $\underline{\mathbf{D}}$ de taille $K \times L$:

$$\underline{\mathbf{D}} = \begin{bmatrix} \underline{d}_1 \\ \vdots \\ \underline{d}_K \end{bmatrix}. \quad (1.4)$$

Remarquez que la matrice du dictionnaire est notée de la même manière que les signaux multicanaux, ce qui n'est pas choquant, puisque c'est aussi un "vecteur" de signaux.

Les coefficients $c_k \in \mathbb{R}$ sont réels, et sont la représentation du signal \underline{s} sur le dictionnaire \mathcal{D} . Dans le cas présent où le nombre d'atomes est fini, les coefficients peuvent être regroupés en un vecteur $\mathbf{c} \in \mathbb{R}^{K \times 1}$ défini par :

$$\mathbf{c} = \begin{bmatrix} c_1 \\ \dots \\ c_K \end{bmatrix}. \quad (1.5)$$

Toujours dans le cas d'un nombre d'atomes fini, l'opérateur de synthèse est alors défini uniquement à l'aide de la matrice $\underline{\mathbf{D}}$ et s'écrit comme le produit matriciel :

$$\underline{s} = \mathbf{c}^T \cdot \underline{\mathbf{D}}. \quad (1.6)$$

Nous ne reviendrons pas sur le fait que K est fini dans cette partie, et nous supposons que le lecteur l'a à l'esprit.

1.1.3 Analyse de signaux sur un dictionnaire

Pour pouvoir manipuler un signal, ce n'est pas l'opérateur de synthèse qui nous intéresse, mais l'opérateur d'analyse, afin d'accéder aux coefficients à partir du signal. Voyons comment l'opérateur d'analyse peut être obtenu à partir de l'opérateur de synthèse.

L'opérateur d'analyse est défini à partir de l'opérateur de synthèse par :

$$\begin{array}{ccc} \mathbb{R}^{1 \times L} & \longrightarrow & \mathbb{R}^{K \times 1} \\ \text{Opérateur d'analyse : } \underline{s} & \longrightarrow & \mathbf{c} / \underline{s} = \mathbf{c}^T \cdot \underline{\mathbf{D}} \end{array} \quad (1.7)$$

Cette définition ne caractérise pas l'opérateur d'analyse, car il peut ne pas y en avoir, il peut en exister un unique, ou alors en exister plusieurs, voire une infinité. De même, si un opérateur existe, il n'est pas forcément exprimable aussi simplement que l'opérateur de synthèse et n'est pas forcément linéaire.

1.1.4 Cas particulier où le dictionnaire est une base

Le cas le plus simple est celui où le dictionnaire \mathcal{D} est une base, ce qui signifie que $K = L$ et que les atomes forment une famille libre de vecteurs.

Si le dictionnaire est une base, alors $\underline{\mathbf{D}}$ est une matrice carrée inversible, et l'opérateur d'analyse est donné directement par sa matrice inverse, notée $\underline{\mathbf{D}}^{-1}$:

$$\begin{array}{ccc} \mathbb{R}^{1 \times L} & \longrightarrow & \mathbb{R}^{K \times 1} \\ \text{Opérateur d'analyse : } \underline{\mathbf{s}} & \longrightarrow & \mathbf{c} / \mathbf{c}^T = \underline{\mathbf{s}} \cdot \underline{\mathbf{D}}^{-1} \end{array} \quad (1.8)$$

Dans ce cas, le signal $\underline{\mathbf{s}}$ et sa représentation \mathbf{c} sont équivalents et de même énergie. L'opérateur d'analyse est linéaire, puisqu'il peut être caractérisé par une multiplication matricielle.

Les bases sont très utilisées pour les représentations de signaux. En particulier, toutes les transformées peuvent s'exprimer comme un changement de base, ou, en utilisant les termes de ce travail, comme une représentation sur une base. Ainsi entrent notamment dans ce cadre la transformée de Fourier, la transformée en z et les transformées en ondelettes. A part quelques transformées en ondelettes, ce sont des transformées à coefficients complexes dont nous ne parlerons pas plus ici.

En revanche, présentons deux transformées à coefficients réels, afin d'illustrer nos propos. Nous nous intéressons à la transformée en cosinus, qui est une version de la transformée de Fourier dans le domaine réel, et la transformée de Haar, qui est une transformée en ondelettes à coefficients réels d'expression très simple.

Les atomes \underline{d}_k de la transformée en cosinus s'expriment :

$$\underline{d}_k(l) = \sqrt{\frac{2}{L}} \cdot \cos\left(2\pi \cdot \frac{k \cdot l}{L}\right), \quad \forall l \in \llbracket 1, L \rrbracket, \quad (1.9)$$

où L est la longueur des signaux et des atomes. Les atomes de cette base correspondent à des fréquences pures, et sont très utiles pour analyser la structure fréquentielle d'un signal. L'affichage du spectre, qui est la représentation en échelle logarithmique de la valeur absolue des coefficients \mathbf{c} , offre une nouvelle vue du signal, en mettant en lumière l'articulation des fréquences.

Prenons un exemple musical. Un signal temporel – une note de violon – et son spectre sont représentés sur la figure 1.1. Dans cet exemple, les coefficients prépondérants sont en relation harmonique, ce qui signifie qu'ils correspondent à des atomes de fréquence multiple d'une fréquence fondamentale. En mesurant l'écart de fréquence entre les harmoniques, qui est d'environ 200 Hz, nous pouvons caractériser la hauteur de la note jouée. Ici, la note jouée doit être un sol, puisque la fréquence fondamentale théorique d'un sol est de 196 Hz.

Définissons maintenant la base de Haar qui est la base d'ondelettes la plus simple. Elle est orthogonale, les atomes ont un support fini, et l'ondelette mère est un échelon. Les atomes sont localisés temporellement, et représentent la même ondelette mère dilatée. La dimension L doit être une puissance de deux. Les L atomes sont définis de la façon suivante. Le premier atome est constant : $\underline{d}_1(l) = \frac{1}{\sqrt{L}}$, $\forall l \in \llbracket 1, L \rrbracket$. Les $K - 1$ atomes suivants sont indicés par un indice d'échelle k et un indice de localisation l , et sont définis par :

$$\underline{d}_{k,l}(l) = \begin{cases} +\frac{2^k}{\sqrt{L}} & \text{pour } l \in \llbracket (l-1) \cdot \frac{L}{2^k} + 1, (l-\frac{1}{2}) \cdot \frac{L}{2^k} \rrbracket \\ -\frac{2^k}{\sqrt{L}} & \text{pour } l \in \llbracket (l-\frac{1}{2}) \cdot \frac{L}{2^k} + 1, l \cdot \frac{L}{2^k} \rrbracket, \end{cases} \quad (1.10)$$

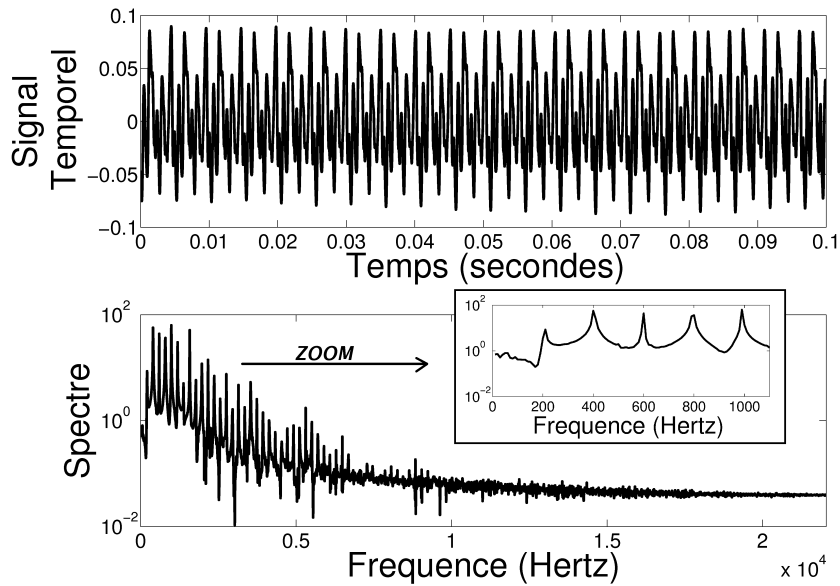


FIG. 1.1 – Note de violon : signal temporel (haut) et spectre (bas).

avec $k \in \llbracket 0, \log_2(\frac{L}{2}) \rrbracket$ et $l \in \llbracket 1, 2^k \rrbracket$.

Nous présentons sur la figure 1.2 quelques atomes de la base de cosinus, et quelques atomes de la base de Haar.

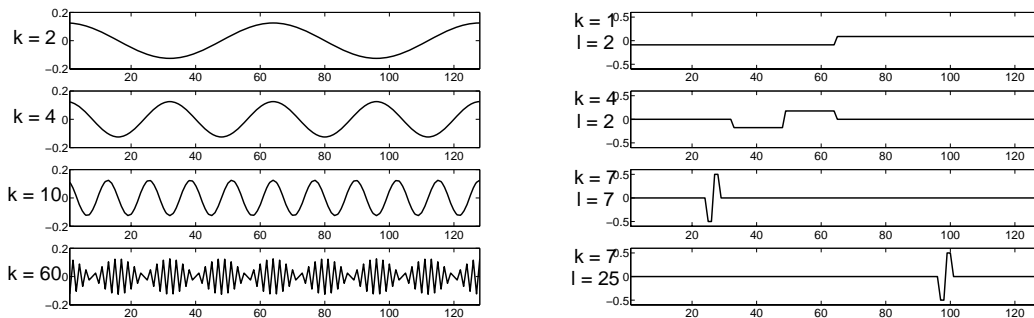


FIG. 1.2 – Quelques atomes de la base de cosinus (gauche) et de la base de Haar (droite), de taille $L = 128$.

Au premier coup d'oeil, la forme des atomes nous apprend que les deux bases sont assez différentes, et ne seront pas utilisées dans le même but, ou pour le même type de signaux. Nous reviendrons par la suite à cet exemple lorsque nous évoquerons la parcimonie des coefficients. En anticipant sur la question de la parcimonie, il paraît probable que des signaux fortement harmoniques seront mieux approchés par quelques atomes de la base de cosinus qu'avec quelques atomes de la base de Haar (la réciproque

est vraie pour des signaux de nature plus impulsive).

1.1.5 Cas d'un dictionnaire redondant

Nous venons de voir que l'opérateur d'analyse se déduit facilement de l'opérateur de synthèse dans le cas où le dictionnaire est une base. En revanche, lorsque le nombre d'atomes K est différent de la taille des signaux L , le problème est beaucoup plus difficile.

Examinons rapidement le cas où $K < L$. Dans ce cas, le dictionnaire n'est pas une famille génératrice de l'espace des signaux, puisque la dimension du dictionnaire est inférieure à la dimension de l'espace à générer. Cela signifie qu'il existe des signaux \underline{s} qui ne peuvent pas s'écrire comme combinaison linéaire des atomes du dictionnaire. L'opérateur de synthèse défini à partir du dictionnaire engendre seulement un sous-espace de l'espace des signaux. Il n'existe donc pas d'opérateur d'analyse permettant d'analyser tous les signaux de $\mathbb{R}^{1 \times L}$. Nous reviendrons dans la section suivante sur ce cas.

Intéressons-nous maintenant au cas où $K > L$. Le dictionnaire contient plus d'atomes que la dimension du signal, et il est dit redondant. La famille des atomes est nécessairement liée, puisque le nombre de vecteurs de la famille est supérieur à la dimension de l'espace à engendrer. Plusieurs jeux de coefficients différents peuvent donc synthétiser le même signal, et par conséquent, il n'existe pas un unique opérateur d'analyse.

Nous pouvons également expliquer que la matrice $\underline{\mathbf{D}}$, de dimension $K \times L$, n'est pas carrée si $K > L$ et elle ne peut donc pas être inversée. Cependant, il existe une infinité de matrices $\underline{\mathbf{D}}_2$ telles que $\underline{\mathbf{D}}_2^T \cdot \underline{\mathbf{D}} = \underline{\mathbf{I}}$ (où $\underline{\mathbf{I}} \in \mathbb{R}^{L \times L}$ est la matrice identité). Ces matrices engendrent autant d'opérateurs d'analyse possibles :

$$\begin{array}{l} \mathbb{R}^{1 \times L} \quad \longrightarrow \quad \mathbb{R}^{K \times 1} \\ \text{Opérateur d'analyse : } \underline{s} \quad \longrightarrow \quad \mathbf{c} / \mathbf{c}^T = \underline{s} \cdot \underline{\mathbf{D}}_2^T \end{array} \quad (1.11)$$

En effet :

$$\underline{s} = \mathbf{c}^T \cdot \underline{\mathbf{D}} = (\underline{s} \cdot \underline{\mathbf{D}}_2^T) \cdot \underline{\mathbf{D}} = \underline{s} . \quad (1.12)$$

Dès lors, comment choisir parmi tous les opérateurs d'analyse ? En choisissant à chaque analyse un opérateur au hasard, il sera difficile d'utiliser les coefficients efficacement, puisque les propriétés des coefficients vont dépendre de l'opérateur utilisé et qu'il n'est pas nécessairement bien connu, puisque choisi au hasard parmi une infinité. Pour que les coefficients arborent toujours les mêmes propriétés, ou en tout cas, pour sélectionner le jeu de coefficients qui présente les propriétés les plus proches de celles voulues, une solution consiste à définir un critère de choix des coefficients, et une fois que le critère de choix des coefficients existe, l'opérateur d'analyse est défini comme l'optimisation de ce critère.

1.1.5.1 Exemple : le dictionnaire de Gabor

Avant de discuter du critère à choisir, qui dans notre cas sera, sans surprise, un critère optimisant la parcimonie des coefficients, présentons un dictionnaire redondant

notamment utilisé pour traiter les signaux sonores, que nous appellerons dictionnaire de Gabor.

Ce dictionnaire est composé d’atomes de Gabor, qui sont localisés en temps et en fréquence. Rappelons-nous, les atomes de la base de cosinus étaient localisés en fréquence, les atomes de la base de Haar étaient localisés en temps. Les atomes de Gabor réunissent donc les deux propriétés et représentent une fréquence pure oscillant sur une courte période. Un atome de Gabor est paramétré par une localisation temporelle l , une fréquence f , une phase φ et une taille de fenêtre L_t :

$$\underline{d}_{t,f,\varphi,L_t}(l) = \begin{cases} C \cdot w\left(\frac{l-t}{L_t}\right) \cdot \cos\left(2\pi \cdot f \cdot \frac{l-t}{L_t} + \varphi\right), & \forall l \in \llbracket t+1, t+L_t \rrbracket, \\ 0 & \text{ailleurs,} \end{cases} \quad (1.13)$$

où C est une constante de normalisation et w est une fonction de fenêtrage, par exemple une fenêtre de Hamming. Notons que ce type d’atome n’entre dans le cadre que nous avons fixé qu’à condition que la phase soit échantillonnée et ne puisse prendre qu’un nombre fini de valeurs, sinon, si la phase est à valeur continue, alors le nombre d’atomes est infini. Ce n’est cependant pas un problème très important.

Sur la figure 1.3 sont représentés quelques atomes de Gabor, qui sont effectivement localisés temporellement, oscillent régulièrement et ont une enveloppe suivant la forme d’une fenêtre de Hamming.

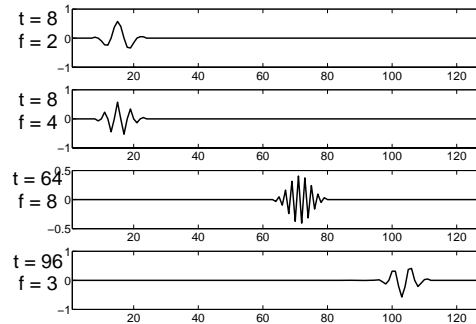


FIG. 1.3 – Quelques atomes de Gabor de taille $L = 128$, utilisant une fenêtre de Hamming de longueur $L_t = 16$.

Comme pour tous les dictionnaires redondants, il existe une infinité d’opérateurs d’analyse associés au dictionnaire de Gabor. Intéressons-nous à un opérateur spécifique, appelé **Transformée de Fourier à Court Terme** (TFCT). La TFCT est très utilisée pour analyser l’évolution du contenu fréquentiel des signaux. L’affichage en échelle logarithmique du module de ses coefficients est appelé spectrogramme, et est une représentation dite “temps-fréquence” très utilisée. Le spectrogramme est à la TFCT ce que le spectre est à la transformée en cosinus. Il faut noter que notre description de la TFCT est abusive car c’est en réalité une transformée à coefficients complexes correspondant à des atomes de Gabor complexes, et la synthèse des coefficients de la TFCT n’est pas exactement la somme des atomes de Gabor pondérés.

Décrivons dans les grandes lignes les quelques étapes de la TFCT. Tout d’abord, des trames de signal de longueur $L_t \ll L$ (petite devant la longueur du signal) extraient une partie du signal à analyser. Ces trames sont généralement décalées de $L_t/2$ échantillons, de façon à ne pas perdre de signal lors du fenêtrage qui va suivre. Un décalage inférieur à la taille d’une trame donne un dictionnaire redondant. Les trames sont ensuite fenêtrées en utilisant la fenêtre w définie précédemment. Enfin, une transformée de Fourier de taille L_t est effectuée sur chaque trame. Pour Q trames, la TFCT donne donc $Q \cdot L_t$ coefficients complexes, qui correspondent chacun à une “case” temps-fréquence, c’est-à-dire qu’ils correspondent à l’amplitude et à la phase d’un atome de Gabor complexe situé au même instant et à la même fréquence.

La TFCT est un moyen très utile d’analyser la structure temps-fréquence d’un signal. Visuellement, la représentation de son module permet de suivre l’évolution des fréquences au cours du temps et de comprendre sa structure harmonique. La figure 1.4 représente la TFCT de la note de violon dont un extrait est représenté dans le domaine temporel sur la figure 1.1. La structure harmonique est très visible sur cette figure, représentée par les traits clairs parallèles correspondant aux coefficients de plus haute énergie. Le bord gauche de la figure représente le début de la note, avec l’apparition des premiers coefficients importants, et le bord droit correspond à l’extinction de la note, avec la disparition des coefficients importants. Nous utiliserons par la suite des spectrogrammes calculés par TFCT pour visualiser la structure temps-fréquence de signaux.

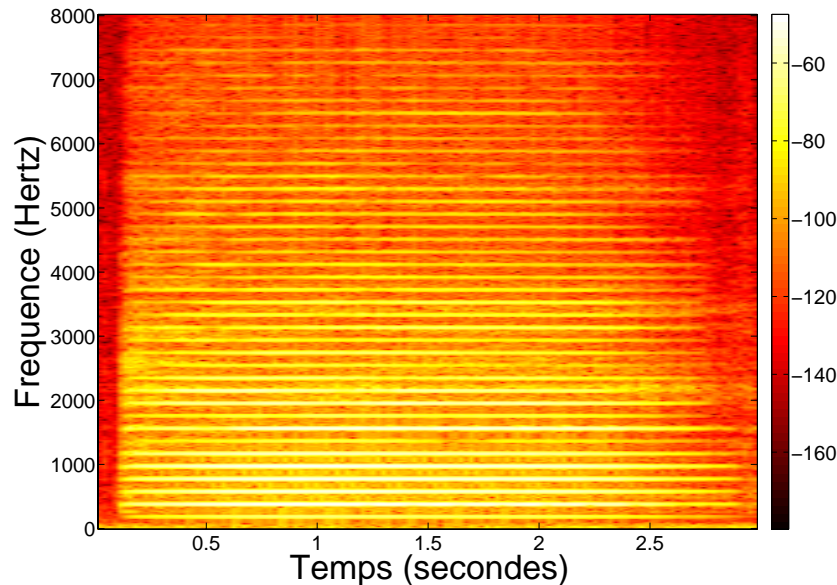


FIG. 1.4 – Transformée de Fourier à Court Terme d’un signal de violon de taille $L = 132\,300$ (3 s de signal échantillonnées à 44 100 Hz), pour des trames de longueur $L_t = 2\,048$ qui se recouvrent de moitié.

1.1.6 Représentation approchée des signaux

Nous avons expliqué dans la section précédente que si le nombre d'atomes est plus petit que L , la longueur du signal, alors il n'existe pas d'opérateur d'analyse permettant une reconstruction parfaite. Il est pourtant intéressant dans de nombreux cas d'utiliser des dictionnaires contenant moins d'atomes que la dimension.

Le meilleur exemple vient de l'Analyse en Composantes Principales (ACP), qui est exposée en détails à la section 3.2. L'un des objectifs de l'ACP est de créer le dictionnaire d'atomes qui permet de réduire la dimension des signaux traités en conservant le maximum d'énergie. En effet, pour certaines applications où les données sont conséquentes, si seuls quelques atomes sont réellement influents et peuvent décrire les données avec une assez bonne précision, il peut être intéressant d'effacer tous les autres atomes du dictionnaire, afin de diminuer le nombre de coefficients à manipuler. Dans ce cas, plus aucun jeu de coefficients ne peut reconstruire exactement les données, ce qui signifie qu'il n'y a pas d'opérateur d'analyse.

Cependant, nous venons d'évoquer la précision avec laquelle les données sont décrites. L'objectif de cette application n'est donc pas de décomposer les données exactement sur un dictionnaire, mais de diminuer significativement la dimension des signaux tout en maîtrisant le niveau d'erreur.

Pour les dictionnaires contenant moins d'atomes que la taille des signaux, une solution pour le calcul des coefficients est donc l'approximation au lieu de la décomposition exacte. Les coefficients optimaux peuvent être choisis comme ceux qui minimisent l'erreur entre le signal d'origine et le signal obtenu après analyse et synthèse, soit, en minimisant l'énergie de l'erreur :

$$\begin{array}{ccc} \mathbb{R}^{1 \times L} & \longrightarrow & \mathbb{R}^{K \times 1} \\ \text{Opérateur d'analyse} : \underline{s} & \longrightarrow & \mathbf{c} = \arg \min_{\mathbf{c}} \|\underline{s} - \mathbf{c}^T \cdot \mathbf{D}\|_2^2 = (\underline{s} \cdot \mathbf{D}^+)^T \end{array} \quad (1.14)$$

où \mathbf{D}^+ est la pseudo-inverse à droite du dictionnaire \mathbf{D} (preuve dans l'annexe A.1).

Pour un dictionnaire ayant plus de colonnes que de lignes, comme ici, la pseudo-inverse à droite est définie par : $\mathbf{D}^+ = \mathbf{D}^T \cdot (\mathbf{D} \cdot \mathbf{D}^T)^{-1}$. Définissons également dès maintenant la pseudo-inverse à gauche quand le nombre de lignes est plus important que le nombre de colonnes : $\mathbf{D}^\dagger = (\mathbf{D}^T \cdot \mathbf{D})^{-1} \cdot \mathbf{D}^T$. Enfin, si la matrice est carrée, la pseudo-inverse est équivalente à l'inverse. La pseudo-inverse n'est définie que pour les matrices de rang plein.

Au delà de cet exemple, l'objectif de la décomposition peut être l'approximation, quel que soit le type de dictionnaire. En effet, même s'il existe des solutions exactes, correspondant à des opérateurs d'analyse bien définis, il peut être intéressant d'accepter un niveau d'erreur en échange de meilleures propriétés sur les coefficients. Évidemment, dans notre cas, l'idée est de calculer des coefficients parcimonieux.

Pour calculer les coefficients, une solution consiste élaborer un critère faisant un compromis entre la qualité de la reconstruction et une mesure des bonnes propriétés des coefficients. L'opérateur d'analyse est alors défini par l'optimisation de ce critère. Rien

ne dit que cet opérateur sera calculable explicitement, ni même qu'il existe une stratégie permettant de trouver une solution. C'est d'ailleurs un des enjeux de la recherche dans le domaine des décompositions parcimonieuses.

Nous avons vu avec les dictionnaires redondants que dans le cadre des décompositions exactes, il faut établir un critère permettant de choisir les coefficients, et il apparaît également que pour tout type de dictionnaire, un critère est nécessaire pour choisir les coefficients lors de l'approximation du signal. Nous avons indiqué dans l'introduction que la parcimonie est une propriété désirée pour les coefficients dans plusieurs applications. Découvrons donc ce qui se cache derrière le terme de parcimonie et comment intégrer cette propriété dans un critère afin de définir un opérateur d'analyse favorisant la parcimonie.

1.2 Principe de parcimonie de la représentation

Nous avons déjà dit plusieurs fois que la parcimonie est une propriété désirée pour les coefficients dans les décompositions de signaux sur un dictionnaire. Dans cette section, nous définissons le terme et rappelons pourquoi c'est une propriété voulue. Après avoir défini ce que nous entendons par problème, critère, algorithme et solution optimale, nous exposerons différents critères de parcimonie et quelques résultats théoriques qui y sont rattachés.

1.2.1 Définition de la parcimonie

La parcimonie est un terme relié en français à la notion d'épargne et même de pingrerie. En informatique, le terme de parcimonie suit la même logique et il est appliqué aux matrices ou aux vecteurs dont quasiment tous les éléments sont égaux, et dont quelques éléments ont une valeur non-standard. Une matrice est alors parcimonieuse s'il est plus économique de la décrire en donnant la valeur standard, ainsi que la valeur et la position de chaque élément non-standard, plutôt qu'en donnant la valeur de tous les éléments.

Dans le cadre de ce travail, un jeu de coefficients est dit parcimonieux au sens strict si une grande majorité des coefficients vaut zéro, et si seuls quelques coefficients ont une valeur non-nulle. De façon plus large, un jeu de coefficient est parcimonieux si la majorité des coefficients a une valeur faible, tandis qu'une minorité prend des grandes valeurs.

1.2.2 Pourquoi vouloir la parcimonie ?

La motivation pour l'utilisation de la parcimonie, au delà de la nécessité de définir un unique opérateur d'analyse, est à la fois conceptuelle et expérimentale.

Tout d'abord, l'idée est de pouvoir exprimer un signal comme l'agrégation de quelques éléments connus facilement manipulables et compréhensibles. Ainsi, une action sur le signal, qui peut paraître ardue et opaque, est décomposée en une action simple et claire sur chacun des éléments. C'est une stratégie à rapprocher du paradigme algorithmique

“divide and conquer” (diviser pour mieux régner, en français) qui consiste à diviser de façon récursive un problème compliqué en sous-problèmes moins compliqués, jusqu’à pouvoir trouver une solution, puis à remonter les solutions jusqu’au niveau original. Notons que dans le cadre des décompositions sur un dictionnaire, la division du signal est linéaire, et donc un tâche de traitement du signal sera simple à diviser et à refusionner si elle est elle-même linéaire.

Revenons à la décomposition du signal en atomes. Intuitivement, il paraît plus simple mentalement de comprendre une représentation si quelques éléments notables jaillissent de la masse des autres coefficients, que si l’énergie est répartie plus ou moins uniformément sur tous les coefficients sans aucune régularité. L’intuition nous dit que cette justification de visualisation s’étend aussi à la manipulation de la représentation. Une représentation parcimonieuse peut être vue comme une façon de bien “ranger” dans des cases distinctes correspondant aux atomes le désordre de la répartition d’énergie au sein du signal. Ainsi, manipuler chacun des atomes porteur de beaucoup d’énergie est plus efficace que manipuler une partie du signal de même taille mais moins énergétique.

La motivation expérimentale de la parcimonie vient de l’observation. Nous avons évoqué dans l’introduction le dictionnaire de Gabor sur lequel les signaux sonores se décomposent de façon parcimonieuse. Nous pouvons le vérifier sur la figure 1.4. Les traits clairs correspondent aux atomes de Gabor les plus actifs. Ils représentent une surface minoritaire de toute la surface, d’autant plus que la figure a été tronquée. En effet, la représentation temps-fréquence monte jusqu’à 22 050 Hz initialement, et seules les basses fréquences, jusqu’à 8 000 Hz sont représentées. Le reste de la figure contient des coefficients de faible poids.

Le même type d’observation a été faite en décomposant des images sur une base d’ondelettes. Nous présentons sur la figure 1.5 la position des coefficients non-nuls de la décomposition approchée d’une image sur un dictionnaire d’ondelettes.

L’image originale est carrée et contient 512^2 pixels. Chaque pixel de l’image est codé avec 8 bits et prend une valeur entière entre 0 et 255. Tous les coefficients inférieurs à un certain seuil ont été mis à zéro. L’approximation est très bonne malgré ce seuillage qui n’a conservé qu’environ 7% des coefficients, puisque le rapport signal à bruit crête (RSBC) est de 35 dB. Avec ce seuillage, le jeu de coefficients peut être codé à 0.5 bit par pixel, au lieu de 8 bits par pixel à l’origine. Le RSBC entre un signal \underline{s} et une approximation de ce signal $\hat{\underline{s}}$ est défini par :

$$\text{RSB}(\underline{s}, \hat{\underline{s}}) = 10 \cdot \log_{10} \frac{C}{\|\underline{s} - \hat{\underline{s}}\|_2^2}, \quad (1.15)$$

où C une constante égale à l’énergie maximale pour un signal \underline{s} . Le RSBC n’est bien sûr utilisable que pour des signaux bornés. Pour l’image précédente, par exemple, $C = 512^2 \cdot 255^2$.

Définissons dès maintenant, pour les signaux non-bornés, le rapport signal à bruit (RSB), que nous utiliserons plus tard :

$$\text{RSB}(\underline{s}, \hat{\underline{s}}) = 10 \cdot \log_{10} \frac{\|\underline{s}\|_2^2}{\|\underline{s} - \hat{\underline{s}}\|_2^2}. \quad (1.16)$$

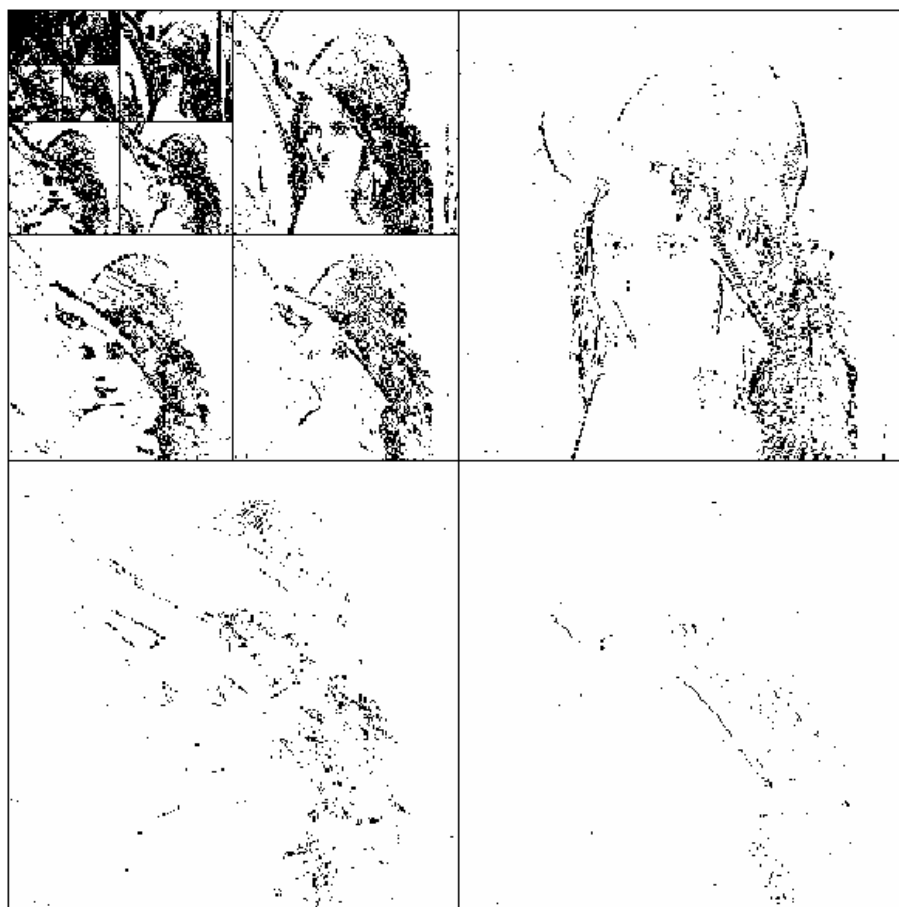


FIG. 1.5 – Carte des coefficients significatifs de la transformée en ondelette de l’image “Léna” (blanc : coefficient nul, noir : coefficient non-nul). Image extraite de [Ma198].

1.2.3 Quelques notions d’optimisation

Pour toutes les raisons exposées précédemment, il est justifié de choisir les coefficients les plus parcimonieux lorsque plusieurs jeux de coefficients sont en concurrence. Rappelons que c’est le cas lorsque le dictionnaire est redondant et lorsque la décomposition est approchée. Pour faire le choix des coefficients, il faut dès lors définir un critère de parcimonie. Nous en proposerons plusieurs dans la section suivante. Dans cette section, nous définissons au préalable quelques notions que nous retrouverons tout au long de ce document.

Tout d’abord, un **critère** est une fonction mathématique qui attribue une valeur scalaire à tous les éléments d’un ensemble. Le critère sert généralement à mesurer la pertinence d’un élément par rapport à une propriété voulue. Il sert à définir un objectif. Dans le cas de la parcimonie, le critère doit mesurer à quel point un jeu de coefficients est parcimonieux, et donner une valeur pour chaque jeu de coefficients. Nous noterons les critères de façon générale par la lettre \mathcal{C} .

Définissons maintenant la notion de problème. Un **problème** est une question posée sous forme mathématique. Toute réponse à ce problème est bien sûr appelée une **solution**. Les problèmes peuvent ne pas avoir de solution, avoir une unique solution, ou avoir plusieurs solutions. Savoir dans quelle catégorie se trouve un problème est d'ailleurs un problème en soi. Nous noterons les problèmes par la lettre \mathcal{P} .

Les critères peuvent être utilisés pour former un type de problèmes particulier : les problèmes d'optimisation de critère. Il s'agit de trouver parmi un ensemble d'éléments celui (ou ceux) qui donne la valeur de critère préférée. L'élément (ou les éléments) gagnant est appelé solution optimale du problème, et si les éléments sont notés t , alors l'élément optimal sera noté t^{opt} .

L'opérateur d'analyse (1.14) est par exemple défini à l'aide d'un problème d'optimisation. Le critère est :

$$\mathcal{C}(\underline{s}, \mathbf{c}, \underline{\mathbf{D}}) = \|\underline{s} - \mathbf{c}^T \cdot \underline{\mathbf{D}}\|_2^2, \quad (1.17)$$

le problème d'optimisation est :

$$\mathcal{P} : \quad \min_{\mathbf{c} \in \mathbb{R}^K} \mathcal{C}(\underline{s}, \mathbf{c}, \underline{\mathbf{D}}), \quad (1.18)$$

et les solutions du problème sont :

$$\mathbf{c}^{\text{opt}} = \arg \mathcal{P} = \arg \min_{\mathbf{c} \in \mathbb{R}^K} \mathcal{C}(\underline{s}, \mathbf{c}, \underline{\mathbf{D}}) \quad (1.19)$$

Enfin, pour résoudre un problème, il faut appliquer une méthode de calcul de la solution. Nous appellerons **algorithme** la description de l'ensemble des calculs, ou instructions, d'une méthode. Dans l'exemple précédent, l'algorithme est très simple, il s'agit d'une seule instruction :

$$\mathbf{c} = (\underline{s} \cdot \underline{\mathbf{D}}^+)^T. \quad (1.20)$$

Souvent, il n'existe pas de manière de calculer la solution de façon exacte. Les algorithmes sont alors conçus pour s'approcher au plus près de la solution optimale. Il existe pour les problèmes d'optimisation des algorithmes classiques (descente de gradient, annulation de gradient, ...) qui peuvent être appliqués pour de nombreux critères. Lorsque ces algorithmes classiques sont trop complexes, ou pas applicables, il peut être intéressant d'utiliser des algorithmes heuristiques, qui reposent généralement sur des règles empiriques simples et rapides, et qui fournissent des résultats raisonnablement proches de la solution optimale. Cette proximité peut être prouvée dans certains cas.

Un algorithme est dit séquentiel si les instructions sont effectuées l'une après l'autre, et itératif s'il contient une boucle, c'est-à-dire si un ensemble d'instructions est appliqué plusieurs fois de suite. Pour les algorithmes itératifs, un test effectué à chaque itération permet de déterminer si la boucle doit être arrêtée ou non. Généralement, le test d'arrêt est effectué en comparant un critère calculé sur les variables à un seuil, et la boucle est arrêtée lorsque le critère franchit le seuil.

La complexité d'un algorithme permet de mesurer son efficacité dans la recherche de solution et dépend de deux paramètres qui varient avec la taille des données : le nombre d'instructions nécessaires pour arriver à la solution, et la quantité de mémoire utilisée.

Pour finir, notons que les solutions optimales sur tout l'ensemble de recherche sont aussi appelées optima globaux, et que les solutions optimales dans un voisinage sont appelées optima locaux.

1.2.4 Critères de parcimonie

Après ces quelques définitions, intéressons-nous aux principaux critères qui ont été élaborés pour mesurer la parcimonie et aux problèmes qui leur sont associés.

Nous avons défini la parcimonie comme le fait qu'une majorité des coefficients vaut zéro. Nous définissons alors la mesure de **diversité** stricte comme le nombre de coefficients non-nuls (le nombre d'atomes utilisés). Nous la notons avec la pseudo-norme ℓ^0 : $\|\mathbf{c}\|_0^0$, et nous expliciterons cette notation par la suite. La diversité est une mesure du poids des coefficients.

Le problème \mathcal{P}_0 est le plus intuitif et consiste à trouver la décomposition exacte qui minimise le nombre de coefficients non-nuls :

$$\mathcal{P}_0 : \quad \min_{\mathbf{c}} \|\mathbf{c}\|_0^0 \text{ tel que } \underline{s} = \mathbf{c}^T \cdot \underline{\mathbf{D}} . \quad (1.21)$$

Il a été prouvé [Nat95] que ce problème est NP-complet, ce qui signifie de façon intuitive qu'il ne peut être résolu que par une recherche exhaustive parmi toutes les combinaisons d'atomes différentes. Lorsque les dimensions sont grandes, il est impossible à résoudre. Il faut donc trouver une autre manière d'imposer la parcimonie.

Pour pallier la limitation introduite par la diversité stricte, Chen, Donoho et Saunders [CDS99] ont assoupli cette mesure de la parcimonie et remplacé la diversité stricte par la norme ℓ^1 définie par :

$$\|\mathbf{c}\|_1 = \sum_{k=1}^K |c_k| . \quad (1.22)$$

La mesure de diversité est minimale si tous les coefficients sont nuls, sauf un, et elle est maximale si le jeu de coefficients est plat, c'est-à-dire si tous les coefficients sont égaux. Pour obtenir un jeu de coefficients parcimonieux, il faut donc minimiser cette diversité. Le problème résultant \mathcal{P}_1 est

$$\mathcal{P}_1 : \quad \min_{\mathbf{c}} \|\mathbf{c}\|_1 \text{ tel que } \underline{s} = \mathbf{c}^T \cdot \underline{\mathbf{D}} . \quad (1.23)$$

Ce problème peut être résolu, notamment par des techniques de programmation linéaire (voir section 2.2). Il est prouvé que si la solution du problème \mathcal{P}_0 a certaines propriétés simples, alors c'est aussi la solution du problème \mathcal{P}_1 (voir la section suivante). Du point de vue probabiliste, la norme ℓ^1 correspond à la log-vraisemblance de coefficients dont la densité de probabilité *a priori* suit une loi de Laplace : $p(\mathbf{c}) \propto e^{-\|\mathbf{c}\|_1}$. Le

problème \mathcal{P}_1 est donc équivalent à une maximisation de la vraisemblance de la densité *a posteriori*.

La diversité peut être également calculée par la norme ℓ^τ :

$$\|\mathbf{c}\|_\tau^\tau = \sum_{k=1}^K |c_k|^\tau, \quad (1.24)$$

avec $\tau \in [0, 1]$. Nous pouvons maintenant expliquer la notation abusive de la diversité stricte par $\|\mathbf{c}\|_0^0 = \lim_{\tau \rightarrow 0} \|\mathbf{c}\|_\tau^\tau$.

Le problème \mathcal{P}_τ associé à cette mesure de diversité est :

$$\mathcal{P}_\tau : \quad \min_{\mathbf{c}} \|\mathbf{c}\|_\tau^\tau \text{ tel que } \underline{\mathbf{s}} = \mathbf{c}^T \cdot \underline{\mathbf{D}}, \quad (1.25)$$

et le théorème d'équivalence entre les problèmes, déjà évoqué et qui sera exposé dans la section suivante, est également valable pour \mathcal{P}_τ . Des algorithmes ont été développés pour résoudre ce problème et sont exposés au chapitre 2. L'interprétation probabiliste de la norme ℓ^τ est que les coefficients suivent une loi gaussienne généralisée. Pour $\tau \in [0, 1]$, cette loi impose des coefficients parcimonieux.

Nous pouvons enfin rappeler que le problème \mathcal{P}_2 , appelé couramment problème des moindres carrés :

$$\mathcal{P}_2 : \quad \min_{\mathbf{c}} \|\mathbf{c}\|_2^2 \text{ tel que } \underline{\mathbf{s}} = \mathbf{c}^T \cdot \underline{\mathbf{D}}, \quad (1.26)$$

a une solution directe : $\mathbf{c}^{\text{opt}} = (\underline{\mathbf{s}} \cdot \underline{\mathbf{D}}^+)^T$. Les coefficients obtenus en minimisant ce critère sont les coefficients de plus petite énergie, et ne sont pas parcimonieux [CDS99]. Du point de vue probabiliste, ce critère correspond à la log-vraisemblance de coefficients avec *a priori* gaussien : $p(\mathbf{c}) \propto e^{-\mathbf{c}^T \cdot \underline{\mathbf{\Sigma}}^{-1} \mathbf{c}}$, où la matrice de covariance $\underline{\mathbf{\Sigma}}$ est l'identité.

Au delà des critères de parcimonie définis par la norme ℓ^τ , Wickerhauser et Coifman [CW92] ont proposé de mesurer la parcimonie en calculant l'entropie de Shannon des coefficients. Nous ne détaillons pas ce critère car nous ne l'utiliserons pas par la suite.

Tous les problèmes que nous venons d'exposer correspondent au cadre des décompositions exactes, et ils n'ont un intérêt que pour des dictionnaires redondants. En effet, les problèmes n'ont pas de solution si le dictionnaire contient moins d'atomes que la dimension, et un seul jeu de coefficients remplit la condition $\underline{\mathbf{s}} = \mathbf{c}^T \cdot \underline{\mathbf{D}}$, lorsque le dictionnaire est une base, ce qui rend inutile la minimisation.

Intéressons-nous maintenant au cas de l'approximation parcimonieuse de signaux. Dans ce cas, le problème est de trouver un jeu de coefficients à la fois parcimonieux et permettant une reconstruction fidèle du signal. De façon classique, la mesure de l'erreur de reconstruction est l'erreur quadratique : $\|\underline{\mathbf{s}} - \mathbf{c}^T \cdot \underline{\mathbf{D}}\|_2^2$, ce qui correspond à supposer, dans le cadre probabiliste, que l'erreur est de distribution gaussienne.

Le problème de l'approximation, noté $\mathcal{P}_{\tau,\lambda}^{DN}$, s'écrit alors, quelle que soit la mesure de parcimonie choisie (avec $0 \leq \tau \leq 2$)

$$\mathcal{P}_{\tau,\lambda}^{DN} : \quad \min_{\mathbf{c}} \|\underline{\mathbf{s}} - \mathbf{c}^T \cdot \underline{\mathbf{D}}\|_2^2 + \lambda \cdot \|\mathbf{c}\|_\tau^\tau. \quad (1.27)$$

La notation DN fait référence au Basis Pursuit De-Noising, proposé dans [CDS99], qui correspond au problème \mathcal{P}_1^{DN} . Le paramètre λ sert à régler la priorité entre les deux termes de la somme. Si λ est très grand, alors la solution optimale sera très parcimonieuse, mais l'erreur peut être grande. De l'autre côté, si λ est très petit, alors l'erreur sera faible, mais les coefficients ne seront pas très parcimonieux. Le réglage de ce paramètre est généralement empirique, mais certaines méthodes [Eng00, KDMR⁺03] permettent de trouver automatiquement une valeur offrant un bon compromis.

De nombreux algorithmes permettent de résoudre les différents problèmes, et les plus marquants sont présentés au chapitre 2. Notons à nouveau que le problème \mathcal{P}_2^{DN} , appelé Method of Frame De-Noising [CDS99], donne des coefficients qui ne sont pas parcimonieux. Ils sont obtenus par $\mathbf{c}^{\text{opt}} = \left(\underline{\mathbf{s}} \cdot \underline{\mathbf{D}}^T (\underline{\mathbf{D}} \cdot \underline{\mathbf{D}}^T + \lambda \cdot \mathbf{I})^{-1} \right)^T$, où $\mathbf{I} \in \mathbb{R}^{K \times K}$ est la matrice identité.

Avant d'exposer quelques résultats théoriques établis sur ces problèmes, faisons un aparté sur l'approximation et le bruit. Nous l'avons présentée jusqu'à maintenant comme une manière d'approcher la décomposition exacte, tout en augmentant la parcimonie des coefficients. L'approximation peut être vue également sous l'angle du modèle de décomposition bruitée :

$$\underline{\mathbf{s}} = \mathbf{c}^T \cdot \underline{\mathbf{D}} + \underline{\mathbf{r}} . \quad (1.28)$$

Ce modèle peut être interprété de façon générative : divers signaux, correspondant aux atomes actifs, sont mesurés par un capteur, et $\underline{\mathbf{r}}$ est le bruit de mesure de ce capteur dû à l'imprécision des instruments de mesure. Le bruit peut également correspondre à un bruit réel capté. Dans les problèmes \mathcal{P}_τ^{DN} , le bruit est supposé gaussien, et la paramètre λ dépend du niveau, qui peut être connu, de ce bruit. Si le niveau du bruit est connu et si l'objectif est d'identifier les atomes actifs dans ce bruit, alors le paramètre λ ne va pas être réglé pour minimiser l'erreur, mais pour que l'erreur soit du même niveau que le bruit.

1.2.5 Résultats théoriques connus

De nombreuses études théoriques ont été menées pour caractériser les solutions des problèmes exposés dans la section précédente. Le théorème le plus important est certainement celui qui prouve sous certaines conditions l'optimalité d'un jeu de coefficients \mathbf{c} pour le problème \mathcal{P}_τ . Donoho et Elad [DE03] ont prouvé le théorème pour $\tau \in \{0, 1\}$ et Gribonval et Nielsen [GN03b] pour $\tau \in [0, 1]$. Le théorème est le suivant :

Théorème 1 *Soit $\underline{\mathbf{D}}$ un dictionnaire arbitraire dans un espace de Hilbert de dimension finie ou infinie. Si $\underline{\mathbf{s}} = \sum_{k=1}^K c_k \cdot \underline{\mathbf{d}}_k$ avec*

$$\|\mathbf{c}\|_0^0 < \frac{1}{2}(1 + 1/M(\underline{\mathbf{D}})) , \quad (1.29)$$

alors \mathbf{c} est l'unique solution du problème \mathcal{P}_τ , pour toutes les valeurs de τ , $0 \leq \tau \leq 1$.

La grandeur $M(\mathbf{D}) = \sup_{k' \neq k} |\langle \underline{d}_{k'}, \underline{d}_k \rangle|$ s'appelle la cohérence du dictionnaire \mathbf{D} et elle mesure la proximité entre les atomes du dictionnaire, dans le pire cas.

Le théorème permet d'assurer qu'un jeu de coefficients \mathbf{c} est l'unique solution optimale du problème \mathcal{P}_τ , mais également qu'il est la solution optimale unique de tous les problèmes \mathcal{P}_τ pour $0 \leq \tau \leq 1$, et notamment de \mathcal{P}_0 , qui, comme nous l'avons déjà expliqué, est un problème NP-complet dans le cas général.

Le théorème prouve qu'un jeu de coefficients est la solution optimale, mais il ne donne pas la méthode permettant de calculer cette solution. Certaines méthodes permettant de tendre vers cette solution sont présentées au chapitre 2.

Notons que le théorème se base sur le pire cas, ce qui implique que les conditions sont très rarement atteintes. Pour un dictionnaire redondant d'atomes normés, la cohérence ne peut pas être inférieure à $1/\sqrt{L}$. Si par exemple les atomes sont de taille $L = 256$, le jeu de coefficients doit donc vérifier $\|\mathbf{c}\|_0^0 < 8$ et $\underline{s} = \mathbf{c}^T \cdot \mathbf{D}$. Le cas de figure où un signal de taille 256 est décomposé exactement comme combinaison linéaire de 8 atomes est très restrictif, et n'arrive jamais en pratique. Gribonval et Nielsen proposent des conditions moins restrictives dans [GN03a].

Un deuxième théorème concerne les seuls problèmes \mathcal{P}_1 et \mathcal{P}_0 . Fuchs [Fuc04] et Tropp[Tro04] ont prouvé que, sous certaines conditions différentes de celles du premier théorème, les deux problèmes sont équivalents :

Théorème 2 *Supposons que le signal \underline{s} se décompose exactement comme $\sum_{k=1}^K c_k \cdot \underline{d}_k$ dans un dictionnaire arbitraire \mathbf{D} , et notons $I = \{k / c_k \neq 0\}$ et T_I l'opérateur de synthèse $\mathbf{c} \rightarrow T_I(\mathbf{c}) = \sum_{k \in I} c_k \cdot \underline{d}_k$. Si*

$$\sup_{k \notin I} \|T_I^+(\underline{d}_k)\|_1 < 1, \quad (1.30)$$

alors \mathbf{c} , la solution qui minimise \mathcal{P}_1 , est aussi la solution qui minimise \mathcal{P}_0 .

Rappelons que la notation T_I^+ désigne la pseudo-inverse de T_I .

Concernant les problèmes d'approximation parcimonieuse, Gribonval, Figueras et Vandergheynst [GFV05] proposent un test déterminant si un jeu de coefficients \mathbf{c} est la solution de \mathcal{P}_0^{DN} , et plus largement mesurant à quel point le jeu de coefficients est proche de la solution optimale.

Après toutes ces définitions et résultats théoriques, illustrons sur quelques exemples simples les notions exposées jusqu'ici.

1.3 Illustration expérimentale

Nous présentons dans cette section quelques résultats expérimentaux permettant d'illustrer trois notions précédemment exposées. Nous montrerons donc qu'un signal peut être parcimonieux dans une base, mais pas dans une autre, et que le choix du dictionnaire doit refléter la nature du signal à décomposer. Dans un deuxième temps, nous illustrerons l'intérêt de la redondance sur un exemple simple. Nous terminerons cette partie expérimentale par une comparaison des différents critères de parcimonie exposés précédemment.

1.3.1 Le dictionnaire doit être adapté à la nature des signaux

Pour illustrer l'intérêt de bien choisir le dictionnaire, nous étudions deux signaux, de taille $L = 128$. Le premier signal est la somme pondérée de dix atomes de la base de cosinus. Le second signal est la somme pondérée de dix atomes de la base de Haar. Ils sont représentés sur la figure 1.6, le premier signal à gauche et le deuxième à droite. Le signal de gauche est oscillé de façon assez régulière tandis que le signal de droite est plus impulsif. Les deux signaux ont des natures très différentes qui s'expriment lors de leur représentation dans les deux bases.

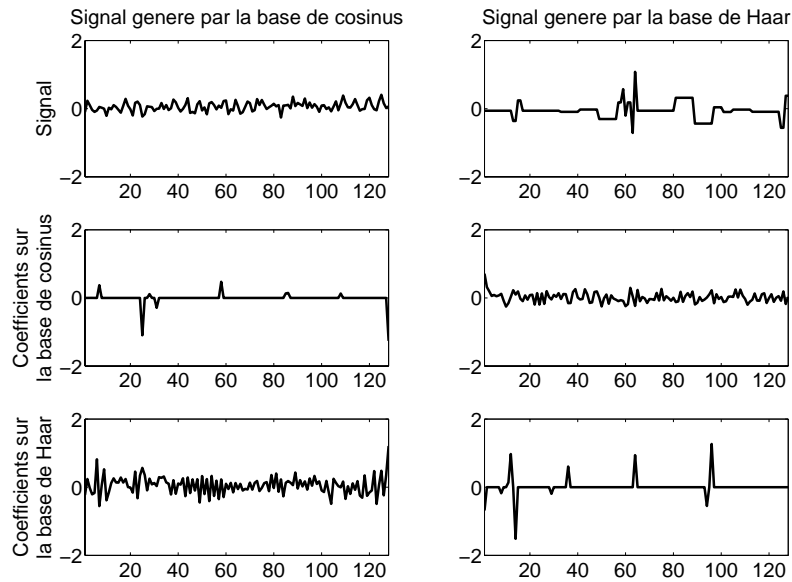


FIG. 1.6 – Gauche : le signal est une somme de sinusoides. Droite : le signal est une somme d'ondelettes de Haar. La ligne du milieu représente les coefficients de leur décomposition sur la base de cosinus, et la ligne du bas les coefficients de leur décomposition sur la base de Haar. Chaque signal est représenté très efficacement dans sa base associée, et pas du tout dans l'autre.

Sur la base de cosinus (au milieu, à gauche), le signal de gauche est représenté avec dix coefficients non-nuls, ce qui correspond à la manière dont il a été synthétisé. La description est très économique et explicite. En revanche, la décomposition du signal sur la base de Haar (en bas, à gauche) donne des coefficients non-nuls répartis sur tous les atomes. Cette description est compliquée à lire, et n'apporte pas une information claire sur la nature du signal.

Le signal de droite suit la même logique. Il se décompose avec seulement dix atomes sur la base de Haar, en revanche tous les coefficients sont nécessaires pour décomposer sur la base de cosinus.

Les mesures de parcimonie de ces exemples sont donnés à la section 1.3.3, pour les

différents critères. De façon visuelle, nous avons montré l'intérêt d'utiliser un dictionnaire approprié pour obtenir une représentation parcimonieuse.

1.3.2 La redondance permet d'augmenter la parcimonie

Nous illustrons maintenant l'intérêt des dictionnaires parcimonieux sur le même exemple. Rappelons tout d'abord qu'en considérant les atomes comme des éléments générateurs du signal à décomposer, il n'y a aucun raison de supposer que seuls L éléments contribuent à générer le signal, mais au contraire que le nombre d'éléments générateurs est plus important que L . Dans le cas redondant, le dictionnaire forme un grand catalogue contenant des formes précises de signal, contrairement à une base qui contient L éléments modélisant grossièrement chaque partie de signal, sans en modéliser une seule précisément. Intuitivement, plus il y a d'atomes générateurs, mieux la création de chaque signal peut s'expliquer comme somme de peu d'atomes.

Utiliser un dictionnaire redondant permet de décomposer un signal sonore en tenant compte de sa structure particulière. Par exemple, le dictionnaire peut être composé à la fois des atomes de la base de cosinus et de ceux de la base de Haar, de façon à bien représenter une superposition de composantes oscillatoires et d'impulsions.

En utilisant justement ce dictionnaire, créons un signal de taille $L = 128$ comme la somme pondérée de 10 atomes de la base de cosinus et de 10 atomes de la base de Haar. Il contient donc des sinusoïdes et des impulsions. Le signal est représenté en haut de la figure 1.7.

Au milieu de la figure 1.7 sont représentés les coefficients qui ont servi à générer le signal à partir du dictionnaire redondant contenant la base de cosinus et la base de Haar. Ce dictionnaire contient $K = 256$ atomes et 256 coefficients sont donc représentés. Seuls 20 d'entre eux sont non-nuls.

En bas de la figure 1.7, les coefficients de la décomposition du signal sur la base de cosinus sont représentés à gauche (à droite pour la base de Haar). Aucune des deux bases ne permet d'expliquer succinctement le signal, car tous les coefficients sont utilisés, et aucun n'est prépondérant. Seul un dictionnaire comprenant les atomes des deux bases permet d'obtenir une décomposition efficace (au milieu). Nous pouvons cependant remarquer que dans la base de cosinus (en bas à gauche), les coefficients les plus énergétiques correspondent aux atomes générateurs du signal (visibles au milieu), les coefficients ayant été rangés dans le même ordre. La même remarque s'applique à la base de Haar.

Notre exemple illustre l'intérêt d'utiliser un dictionnaire redondant pour traiter un signal qui contient des éléments de nature différentes. Aucune bonne décomposition n'aurait pas pu être obtenue en utilisant une base seule. L'intérêt de la redondance grandit avec le nombre de structures différentes dans le signal à traiter.

1.3.3 Comparaison des critères de parcimonie

Nous avons montré dans la section 1.3.1 que le signal généré par la base de cosinus n'est pas parcimonieux dans la base de Haar, et inversement. Vérifions-le de façon

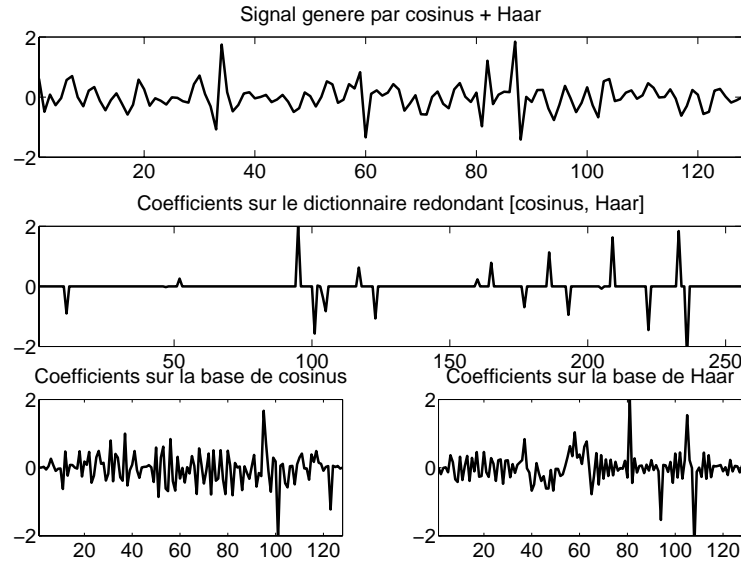


FIG. 1.7 – Le signal composé de cosinus et d’ondelettes de Haar (haut) a une décomposition efficace dans le dictionnaire réunissant les atomes de la base de cosinus et de la base de Haar (milieu). En revanche, aucune des deux bases prise seule ne permet une décomposition efficace.

numérique à l’aide des mesures de diversité présentées précédemment. Nous n’exposons que les résultats pour le signal généré par la base de cosinus, pour des raisons de place et d’intérêt de la lecture. En effet, les résultats sont similaires pour les deux signaux.

	Décomposition sur la base de Haar	Décomposition sur la base de cosinus
$\ \mathbf{c}\ _0$	128	10
$\ \mathbf{c}\ _{0.5}^{0.5}$	40.14	5.51
$\ \mathbf{c}\ _1$	14.69	4.02
$\ \mathbf{c}\ _2^2$	2.75	3.31

La première ligne du tableau traduit le fait que 10 atomes de la base de cosinus ont été sélectionnés pour créer le signal, et que 128 atomes, soit la totalité des atomes, sont nécessaires pour décomposer le signal sur la base de Haar.

A travers les trois premières lignes, l’exemple illustre que les normes ℓ^τ , avec $\tau \in [0, 1]$ permettent de mesurer la diversité. Elles sont toujours plus élevées pour la colonne de gauche, correspondant à la décomposition non parcimonieuse sur la base de Haar, que pour la colonne de droite, qui correspond à la décomposition la plus parcimonieuse.

La ligne 4 montre quant à elle que la norme ℓ^2 n’a pas de rapport avec la parcimonie. En effet, la norme ℓ^2 est plus faible pour la colonne de gauche que pour la colonne de

droite.

Intéressons-nous maintenant au signal généré par 20 atomes issus du dictionnaire redondant formé des deux bases. Ce signal a donc une décomposition parcimonieuse sur le dictionnaire union des deux bases, et la figure 1.7 montre que ni la base de cosinus, ni la base de Haar ne donne à elle seule une décomposition parcimonieuse. Nous confirmons cette analyse de façon chiffrée dans le tableau suivant, à l'aide des différentes mesures de parcimonie.

	Base de cosinus	Base de Haar	Union des bases de cosinus et de Haar
$\ \mathbf{c}\ _0$	128	128	20
$\ \mathbf{c}\ _{0.5}^{0.5}$	62.49	61.80	17.69
$\ \mathbf{c}\ _1$	40.48	38.01	18.85
$\ \mathbf{c}\ _2^2$	26.37	26.37	26.82

Le tableau confirme que quelle que soit la mesure de parcimonie utilisée, pour $\tau \leq 1$, les coefficients de la décomposition sur le dictionnaire redondant sont plus parcimonieux que les coefficients de la décomposition sur une seule des deux bases. Il confirme également, avec la quatrième colonne, que les coefficients d'énergie minimale ne sont pas les plus parcimonieux.

1.4 Conclusion et sommaire

Pour conclure ce chapitre et annoncer les prochains, nous allons rappeler les quelques hypothèses de travail et énoncer les difficultés à surmonter pour utiliser les représentations parcimonieuses. Nous terminerons par un sommaire des chapitres suivants.

1.4.1 Hypothèses de travail

Nous avons déjà annoncé que, pour la première partie de ce document, les signaux sont de dimension finie et sont représentés par un vecteur \underline{s} de taille L .

De même, nous supposons que le nombre d'atomes K dans le dictionnaire est fini, ce qui permet de représenter le dictionnaire par la matrice $\underline{\mathbf{D}}$, dont les K lignes sont les atomes \underline{d}_k de taille L , et les coefficients par le vecteur \mathbf{c} , de taille K .

Rappelons également que nous avons restreint les signaux, les atomes, et les coefficients à être à valeurs réelles.

Enfin, à travers les critères de parcimonie définis par une norme ℓ^τ , avec $\tau \in [0, 1]$, nous supposons que les coefficients de la décomposition sont indépendants entre eux et identiquement distribués, d'un point de vue probabiliste. En effet, sous l'angle des probabilités, le critère $\|\mathbf{c}\|_\tau^\tau$ correspond à la log-vraisemblance des coefficients suivant

de façon indépendante une même loi gaussienne généralisée :

$$\|\mathbf{c}\|_{\tau}^{\tau} \propto \log \prod_{k=1}^K C \cdot e^{-|\alpha \cdot c_k|^{\tau}}, \quad (1.31)$$

où C et α sont des constantes.

Cette hypothèse peut paraître surprenante, car rien ne permet d'affirmer que les coefficients sont réellement indépendants. Par exemple, sur l'exemple de la figure 1.4 qui représente la TFCT d'un signal de violon, les coefficients en relation harmonique ont des comportements très similaires, et certainement pas indépendants. De plus, dans le cas de l'analyse exacte du signal, si le dictionnaire est redondant, alors les coefficients ne peuvent pas être indépendants, puisqu'ils forment une famille liée.

En revanche, l'idée d'indépendance n'est pas complètement éloignée de l'élaboration d'un dictionnaire efficace. En effet, reprenons l'exemple de la TFCT du signal de violon. Si des relations harmoniques fréquentes sont repérées dans le signal, alors pour augmenter la parcimonie, il est judicieux d'ajouter dans le dictionnaire des atomes harmoniques évitant de gaspiller plusieurs coefficients alors qu'un seul suffit. Des coefficients très interdépendants cachent souvent une optimisation possible du dictionnaire. L'indépendance entre les coefficients est l'objectif d'une méthode d'apprentissage de dictionnaire présentée au chapitre 3, et nommée l'analyse en composantes indépendantes (ACI).

Après avoir exposé les quelques hypothèses de travail, listons les difficultés inhérentes aux représentations parcimonieuses.

1.4.2 Difficultés

Nous avons expliqué dans l'introduction et dans ce chapitre que les représentations parcimonieuses permettent d'éclairer la visualisation de signaux et sont un outil précieux pour des applications telles que le codage, la séparation de sources, ou la classification.

Étant donné un dictionnaire d'atomes, le calcul des coefficients n'est en revanche pas direct. Le choix d'un jeu de coefficients particulier, qui fait tout l'intérêt des décompositions parcimonieuses, est également un problème souvent compliqué.

La première difficulté consiste à formuler le problème en fonction des propriétés voulues sur les coefficients. Nous avons exposé dans la section 1.2 les formulations les plus courantes du problème du choix de coefficients parcimonieux et des théorèmes établissant certaines relations d'équivalence entre ces problèmes. Il faut néanmoins choisir quel problème utiliser pour calculer les coefficients, et notamment choisir les paramètres τ et λ , pour les problèmes \mathcal{P}_{τ} et $\mathcal{P}_{\tau, \lambda}^{DN}$.

La deuxième difficulté se trouve dans l'élaboration d'une méthode permettant de résoudre le problème posé. Le problème peut être trivial, comme \mathcal{P}_2 où, en théorie, la méthode de résolution est directe, mais aussi être plus compliqué et nécessiter la mise en œuvre d'algorithmes complexes, dont le résultat n'est pas forcément la solution optimale, et qui peuvent être gourmands en temps de calcul et en mémoire. Notons également qu'une voie parallèle consiste à élaborer un algorithme heuristique, détaché

de toute idée de minimisation de critère, et de prouver ensuite la qualité de son résultat pour approcher la solution optimale d'un problème donné.

Enfin, la troisième grande difficulté des représentations parcimonieuses se trouve dans le choix des atomes, car la parcimonie des coefficients dépend du dictionnaire utilisé, comme nous l'avons montré dans la section 1.3. Ce problème spécifique sera discuté plus en détails dans le chapitre 3, et nous verrons notamment quels critères devront être minimisés. Nous pouvons cependant dévoiler dès maintenant que les méthodes d'apprentissage de dictionnaire consistent en général à optimiser alternativement le dictionnaire et les coefficients de la représentation, et que la mise à jour du dictionnaire est basée sur la minimisation de l'erreur d'approximation de la représentation calculée précédemment. Le calcul des coefficients correspond donc, pour ce genre de méthodes, à un problème d'approximation, où il est nécessaire que l'erreur soit non-nulle pour que le dictionnaire puisse évoluer.

1.4.3 Sommaire

Pour répondre à ces difficultés, nous allons exposer dans les chapitres suivants les réponses apportées par la communauté, principalement depuis quinze ans. Dans le chapitre 2, nous exposerons des méthodes de résolution des différents problèmes de calcul des coefficients. Dans un premier temps, nous exposerons des algorithmes heuristiques, et notamment Matching Pursuit, que nous retrouverons par la suite. Nous trouverons ensuite des méthodes de résolution des problèmes $\mathcal{P}_{\tau,\lambda}^{DN}$, puis des méthodes de seuillage alterné pour le problème $\mathcal{P}_{1,\lambda}^{DN}$.

Le chapitre 3 sera consacré au problème de l'apprentissage de dictionnaire. Nous donnerons les motivations, formaliserons le problème, et exposerons des méthodes de résolution pour l'apprentissage de bases et de dictionnaires redondants, après un préliminaire sur les premières méthodes d'apprentissage que sont l'ACP et l'ACI.

Ensuite, dans le chapitre 4, nous poserons de manière formelle le problème de séparation de sources et nous présenterons les méthodes classiques de résolution utilisant les représentations parcimonieuses.

Enfin, le chapitre 5 fera un bilan critique de cet état de l'art et permettra d'introduire le modèle d'atomes structurés que nous avons évoqué dans l'introduction.

Chapitre 2

Algorithmes de décomposition parcimonieuse

Nous avons expliqué au chapitre 1 que l'approximation d'un signal en utilisant les atomes d'un dictionnaire redondant n'est pas unique, à cause du caractère redondant du dictionnaire, mais aussi de l'erreur de reconstruction non nulle. Il faut donc définir une stratégie pour choisir un jeu de coefficients \mathbf{c} parmi tous les jeux possibles.

Nous présentons dans ce chapitre deux approches. La première est une approche heuristique de sélection successive des atomes actifs. Les algorithmes implémentant cette approche sont appelés algorithmes de poursuite et sont présentés à la section 2.1. Ils sont rapides et conduisent sous certaines conditions à la solution optimale du problème $\mathcal{P}_{1,\lambda}^{DN}$.

La deuxième approche regroupe les stratégies de minimisation d'un critère. Hormis pour le problème $\mathcal{P}_{2,\lambda}^{DN}$, qui a une solution analytique directe, la minimisation des critères exposés dans le chapitre précédent nécessite le recours à des algorithmes complexes, qui peuvent être sous-optimaux selon les cas.

Une première famille d'algorithmes, présentés à la section 2.2, regroupe les algorithmes de programmation linéaire et quadratique qui tendent à résoudre le problème Basis Pursuit $\mathcal{P}_{1,\lambda}^{DN}$ de façon exacte. Si les algorithmes que nous présentons permettent de donner des solutions très proches du minimiseur du critère, ils nécessitent néanmoins un très gros effort de calcul.

La deuxième famille d'algorithmes, présentés à la section 2.3 regroupe les algorithmes de seuillage itératif. Ces algorithmes rapides consistent à calculer les coefficients par minimisation de l'erreur quadratique, puis à les seuiller, et à renouveler ce traitement jusqu'à convergence.

2.1 Algorithmes de poursuite

Les algorithmes de poursuite, ou algorithmes gloutons, cherchent à résoudre le problème $\mathcal{P}_{0,\lambda}^{DN}$ à chaque itération. Cette optimisation locale n'assure cependant pas d'optimisation globale dans le cas général. La logique heuristique des algorithmes est la

suivante : chaque étape ajoute un atome à la décomposition, pour tendre vers une erreur nulle quand le nombre d'atomes est grand. A chaque étape, l'atome qui minimise le plus le critère par son ajout est sélectionné. L'algorithme s'arrête quand un critère d'arrêt défini *a priori* est atteint.

Nous présentons dans un premier temps Matching Pursuit, qui est le premier algorithme glouton, proposé par Mallat et Zhang en 1993 [MZ93]. Dans un deuxième temps, nous exposons l'algorithme Orthogonal Matching Pursuit [PRK93] qui assure une plus grande parcimonie et converge en un nombre fini d'itérations. D'autres algorithmes gloutons [Nat95, CR01, KMPY05] ont été proposés dans la littérature mais nous ne les présentons pas ici. Dans la suite de la thèse, nous expliquerons pourquoi nous avons choisi Matching Pursuit pour nos expériences.

2.1.1 Matching Pursuit

Le principe de Matching Pursuit (MP) est de sélectionner à chaque étape l'atome le plus corrélé avec le signal. La décomposition après Matching Pursuit est de la forme :

$$\underline{s} = \sum_{i=1}^I c_i \cdot \underline{d}_{k_i} + \underline{r}_I . \quad (2.1)$$

Le résidu initial \underline{r}_0 est le signal \underline{s} , et à chaque itération i , l'atome \underline{d}_{k_i} sélectionné est celui dont le produit scalaire avec le résidu \underline{r}_i est le plus grand.

Description de Matching Pursuit :

Initialisation du compteur $i := 0$ et du résidu $\underline{r}_0 := \underline{s}$

Faire

1. Mise à jour du compteur $i := i + 1$
2. Recherche de l'atome le plus corrélé :

$$k_i = \arg \max_k |\langle \underline{r}_{i-1}, \underline{d}_k \rangle|^2 \quad (2.2)$$

3. Calcul du nouveau coefficient : $c_i = \langle \underline{s}, \underline{d}_{k_i} \rangle$
4. Mise à jour du résidu : $\underline{r}_i = \underline{r}_{i-1} - c_i \cdot \underline{d}_{k_i}$

Tant que la convergence n'est pas atteinte

Il faut noter que lors de cette procédure, un même atome \underline{d}_k peut être sélectionné plusieurs fois.

L'algorithme Matching Pursuit présente l'intérêt d'être très rapide car il n'exige que des calculs de produits scalaires, et aucune opération de type inversion de matrice, contrairement à beaucoup des algorithmes que nous présentons plus loin. En

contrepartie, le nombre d'itérations à effectuer avant de converger peut être important, notamment à cause de la possibilité de sélectionner plusieurs fois le même atome.

A chaque itération, l'énergie de l'erreur diminue et la diversité des coefficients augmente. L'erreur tend vers zéro quand le nombre d'itérations tend vers l'infini. Sous certaines conditions assez sévères sur la parcimonie de la solution optimale du problème $\mathcal{P}_{0,\lambda}^{DN}$, il est prouvé que l'algorithme Matching Pursuit tend vers cette solution [GN03b] (voir la section 2.1.3).

2.1.2 Orthogonal Matching Pursuit

L'algorithme Orthogonal Matching Pursuit (OMP) repose sur le même principe que Matching Pursuit et la différence vient de la mise à jour des coefficients. En effet, à chaque itération, au lieu de projeter le résidu sur l'atome sélectionné et de calculer uniquement le dernier coefficient, tous les coefficients sont recalculés, en projetant le signal \underline{s} sur l'espace engendré par tous les atomes sélectionnés.

Cette mise à jour de tous les coefficients est obtenue par une orthogonalisation de Gram-Schmidt sur les atomes déjà sélectionnés. Il faut noter que si tous les atomes sont orthogonaux entre eux, Orthogonal Matching Pursuit et Matching Pursuit sont identiques.

Une propriété de OMP est de converger en un nombre fini d'itérations pour un signal de dimension finie. En effet, dans le pire des cas, si la taille du signal est L , l'erreur sera nulle au bout de L itérations, car les L atomes sélectionnés et orthonormalisés forment une base orthogonale de l'espace \mathbb{R}^L . Il faut signaler aussi que la projection interdit à un atome déjà sélectionné d'être sélectionné de nouveau.

Nous notons $\underline{\mathbf{D}}_i = [\underline{d}_{k_1}^T \cdots \underline{d}_{k_i}^T]^T$ le sous-dictionnaire de $\underline{\mathbf{D}}$ regroupant les i premiers atomes sélectionnés et $\mathbf{c}_i = [c_1 \dots c_i]^T$ le vecteur des coefficients correspondants.

Description d'Orthogonal Matching Pursuit :

Initialisation du compteur $i := 0$ et du résidu $\underline{r}_0 := \underline{s}$

Faire

1. Mise à jour du compteur $i := i + 1$
2. Recherche de l'atome le plus corrélé :

$$k_i = \arg \max_k |\langle \underline{r}_{i-1}, \underline{d}_k \rangle|^2 \quad (2.3)$$

3. Ajout du nouvel atome à la collection : $\underline{\mathbf{D}}_i = \begin{bmatrix} \underline{\mathbf{D}}_{i-1} \\ \underline{d}_{k_i} \end{bmatrix}$
4. Mise à jour des coefficients par projection du résidu sur l'espace engendré par les i premiers atomes : $\mathbf{c} = (\underline{s} \cdot \underline{\mathbf{D}}_i^+)^T$ avec $\underline{\mathbf{D}}_i^+ = (\underline{\mathbf{D}}_i^T \cdot \underline{\mathbf{D}}_i)^{-1} \cdot \underline{\mathbf{D}}_i^T$
5. Mise à jour du résidu : $\underline{r}_i = \underline{r}_{i-1} - \mathbf{c}^T \cdot \underline{\mathbf{D}}_i$

Tant que la convergence n'est pas atteinte

La présentation matricielle de l'algorithme met en lumière l'étape de projection, par application de la pseudo-inverse à droite $\underline{\mathbf{D}}_i^+$ de la collection des i premiers atomes. Il faut noter que cette pseudo-inverse peut être calculée itérativement en fonction de la pseudo-inverse $\underline{\mathbf{D}}_{i-1}^+$. De même, les coefficients peuvent être calculés itérativement à un moindre coût. Pour les détails de ces simplifications, voir [PRK93].

2.1.3 Résultats théoriques des algorithmes de poursuite

L'étude des propriétés de convergence des algorithmes Matching Pursuit et Orthogonal Matching Pursuit a été intense dans les dernières années. Il est prouvé que Matching Pursuit converge – l'énergie du résidu tend vers 0 – à condition que le dictionnaire génère tout l'espace des signaux [Jon87]. Comme expliqué précédemment, Orthonormal Matching Pursuit converge en un nombre fini d'itérations pour des signaux de dimension finie.

Tropp [Tro04] et Gribonval et Vandergheynst [GV06] ont respectivement prouvé que OMP et MP vérifient le théorème suivant, qui montre que sous certaines conditions, les “bons” atomes sont sélectionnés à chaque itération :

Théorème 3 *Supposons que le signal \underline{s} se décompose exactement comme $\sum_{k=1}^K c_k \cdot \underline{d}_k$ dans un dictionnaire arbitraire $\underline{\mathbf{D}}$, et notons $I = \{k \mid c_k \neq 0\}$, et T_I l'opérateur $\mathbf{c} \rightarrow T_I(\mathbf{c}) = \sum_{k \in I} c_k \cdot \underline{d}_k$. Si*

$$\sup_{k \notin I} \|T_I^+(\underline{d}_k)\|_1 < 1, \quad (2.4)$$

alors MP et OMP retrouvent la décomposition, c'est-à-dire qu'à chaque pas, un atome “correct” est choisi : $\forall i, k_i \in I$.

Ce théorème prouve que sous les mêmes conditions que celles du théorème 2, qui prouve que la solution de Basis Pursuit est aussi solution de $\mathcal{P}_{0,\lambda}^{DN}$, les algorithmes MP et OMP donnent la solution de $\mathcal{P}_{0,\lambda}^{DN}$.

2.2 Programmation quadratique

Nous présentons dans cette section la famille des algorithmes de résolution du problème $\mathcal{P}_{\tau,\lambda}^{DN}$, par des techniques de point fixe et de programmation quadratique.

2.2.1 Algorithme de point fixe : FOCUSS

L'algorithme FOCUSS [RKD99, REC⁺03] est de type algorithme du point fixe sur le critère $\mathcal{C}_{\tau,\lambda}^{DN}(\underline{s}, \mathbf{c}, \underline{\mathbf{D}})$:

$$\mathcal{C}_{\tau,\lambda}^{DN}(\underline{s}, \mathbf{c}, \underline{\mathbf{D}}) = \|\underline{s} - \mathbf{c}^T \cdot \underline{\mathbf{D}}\|_2^2 + \lambda \cdot \|\mathbf{c}\|_\tau^\tau. \quad (2.5)$$

Le point clé de l'algorithme FOCUSS est l'hypothèse que le gradient de la diversité, $\nabla_{\mathbf{c}}(\|\mathbf{c}\|_{\tau}^{\tau})$, peut s'écrire comme :

$$\nabla_{\mathbf{c}}(\|\mathbf{c}\|_{\tau}^{\tau}) = \alpha(\mathbf{c}) \cdot \mathbf{\Pi}(\mathbf{c}) \cdot \mathbf{c} , \quad (2.6)$$

où $\alpha(\mathbf{c})$ est strictement positif, et $\mathbf{\Pi}(\mathbf{c})$ est une matrice diagonale définie positive. Si l'hypothèse est vérifiée, alors le gradient du critère $\mathcal{C}_{\tau,\lambda}^{DN}(\underline{\mathbf{s}}, \mathbf{c}, \underline{\mathbf{D}})$ s'écrit :

$$\nabla_{\mathbf{c}}(\mathcal{C}_{\tau,\lambda}^{DN}(\underline{\mathbf{s}}, \mathbf{c}, \underline{\mathbf{D}})) = -2 \cdot \underline{\mathbf{D}} \cdot (\underline{\mathbf{s}} - \mathbf{c}^T \cdot \underline{\mathbf{D}})^T + \lambda \cdot \alpha(\mathbf{c}) \cdot \mathbf{\Pi}(\mathbf{c}) \cdot \mathbf{c} . \quad (2.7)$$

Au point fixe \mathbf{c}^* , le gradient est nul et donc

$$\mathbf{c}^* = \left[\underline{\mathbf{D}} \cdot \underline{\mathbf{D}}^T + \frac{\lambda \cdot \alpha(\mathbf{c}^*)}{2} \cdot \mathbf{\Pi}(\mathbf{c}^*) \right]^{-1} \underline{\mathbf{D}} \cdot \underline{\mathbf{s}}^T . \quad (2.8)$$

A l'aide de la manipulation suivante :

$$\left[\underline{\mathbf{D}} \cdot \underline{\mathbf{D}}^T + \frac{\lambda \cdot \alpha(\mathbf{c}^*)}{2} \cdot \mathbf{\Pi}(\mathbf{c}^*) \right]^{-1} \cdot \underline{\mathbf{D}} = \mathbf{\Pi}^{-1}(\mathbf{c}^*) \cdot \underline{\mathbf{D}} \cdot \left[\underline{\mathbf{D}}^T \cdot \mathbf{\Pi}^{-1}(\mathbf{c}^*) \cdot \underline{\mathbf{D}} + \frac{\lambda \cdot \alpha(\mathbf{c}^*)}{2} \cdot \underline{\mathbf{I}} \right]^{-1} , \quad (2.9)$$

où $\underline{\mathbf{I}}$ est la matrice identité, l'équation (2.8) peut s'écrire sous la forme :

$$\mathbf{c}^* = \mathbf{\Pi}^{-1}(\mathbf{c}^*) \cdot \underline{\mathbf{D}} \cdot \left[\underline{\mathbf{D}}^T \cdot \mathbf{\Pi}^{-1}(\mathbf{c}^*) \cdot \underline{\mathbf{D}} + \frac{\lambda \cdot \alpha(\mathbf{c}^*)}{2} \cdot \underline{\mathbf{I}} \right]^{-1} \cdot \underline{\mathbf{s}}^T . \quad (2.10)$$

Cette nouvelle formulation permet d'utiliser la matrice $\mathbf{\Pi}^{-1}(\mathbf{c})$ qui a une expression plus simple que $\mathbf{\Pi}(\mathbf{c})$ et qui est toujours définie. En effet, dans le cas de la diversité $\|\cdot\|_{\tau}$, la matrice $\mathbf{\Pi}(\mathbf{c})$ est la matrice diagonale de terme diagonal $|c_k|^{\tau-2}$, qui n'est pas définie si un coefficient est nul, pour $\tau \leq 1$. En revanche, le terme diagonal de la matrice $\mathbf{\Pi}^{-1}(\mathbf{c})$ est $|c_k|^{2-\tau}$, ou, pour $\tau = 1$, $|c_k|$, ce qui est toujours défini.

L'équation (2.10) réduit également la complexité calculatoire et les problèmes de conditionnement, car la matrice à inverser est beaucoup plus petite et beaucoup mieux conditionnée. En effet, dans l'équation (2.8), la matrice à inverser est de taille $K \times K$, avec K le nombre d'atomes dans le dictionnaire $\underline{\mathbf{D}}$, tandis que dans l'équation (2.10), la matrice à inverser est de taille $L \times L$, avec L la taille des signaux, qui est petite devant le nombre d'atomes dans le cas des dictionnaires redondants.

L'algorithme qui découle de l'équation du point fixe (2.10) est le suivant :

Description de FOCUSS :

Initialisation du compteur $i := 0$ et des coefficients $\mathbf{c}^{(0)}$

Faire

1. Mise à jour du compteur $i := i + 1$

2. Mise à jour de la matrice $\mathbf{\Pi}^{-1}(\mathbf{c}^{(i-1)})$

$$\mathbf{\Pi}^{-1}(\mathbf{c}^{(i-1)}) = \text{diag}(|c_k^{(i-1)}|^{2-\tau}) \quad (2.11)$$

3. Mise à jour du vecteur de coefficients $\mathbf{c}^{(i)}$

$$\mathbf{c}^{(i)} = \mathbf{\Pi}^{-1}(\mathbf{c}^{(i-1)}) \cdot \underline{\mathbf{D}} \cdot \left[\underline{\mathbf{D}}^T \cdot \mathbf{\Pi}^{-1}(\mathbf{c}^{(i-1)}) \cdot \underline{\mathbf{D}} + \frac{\lambda \cdot \tau}{2} \cdot \underline{\mathbf{I}} \right]^{-1} \cdot \underline{\mathbf{s}}^T, \quad (2.12)$$

Tant que la convergence n'est pas atteinte

Il est prouvé [REC⁺03, Fuc06] que l'algorithme FOCUS converge vers un unique point fixe qui fait partie des minima locaux du critère $\mathcal{C}_{\tau,\lambda}^{DN}(\underline{\mathbf{s}}, \mathbf{c}, \underline{\mathbf{D}})$, pour $\tau \in]0, 1]$ et pour toute valeur de $\lambda \geq 0$. Il faut noter [Fuc06] que lorsque λ est grand, la solution optimale est le vecteur nul.

L'initialisation des coefficients $\mathbf{c}^{(0)}$ a une grande influence sur la vitesse de convergence, ce qui permet d'introduire des connaissances *a priori* lors de l'initialisation [RKD99, REC⁺03]. Il faut noter que l'équation de mise à jour des coefficients (2.12) est telle qu'un coefficient mis à zéro à une itération restera à zéro pour toutes les itérations suivantes. Lors de l'initialisation, il est donc impératif de n'avoir que des coefficients non-nuls dans le vecteur $\mathbf{c}^{(0)}$. Une initialisation standard qui conduit presque sûrement à des coefficients non-nuls est la solution d'énergie minimale :

$$\mathbf{c}^{(0)} = (\underline{\mathbf{s}} \cdot \underline{\mathbf{D}}^+)^T. \quad (2.13)$$

2.2.2 Programmation quadratique

Le critère $\mathcal{C}_{1,\lambda}^{DN}(\underline{\mathbf{s}}, \mathbf{c}, \underline{\mathbf{D}})$ peut être résolu par programmation quadratique. En effet, en posant

$$c_k^+ = \max(c_k, 0) \quad \text{et} \quad (2.14)$$

$$c_k^- = \max(-c_k, 0), \quad (2.15)$$

les coefficients peuvent s'écrire

$$c_k = c_k^+ - c_k^- \quad (2.16)$$

$$|c_k| = c_k^+ + c_k^-. \quad (2.17)$$

Le critère $\mathcal{C}_{1,\lambda}^{DN}(\underline{\mathbf{s}}, \mathbf{c}, \underline{\mathbf{D}})$ s'écrit alors sous la forme du programme quadratique suivant :

$$\begin{aligned} \min_{\tilde{\mathbf{c}}} \quad & \tilde{\mathbf{c}}^T \cdot \tilde{\underline{\mathbf{D}}} \cdot \tilde{\underline{\mathbf{D}}}^T \cdot \tilde{\mathbf{c}} + (\lambda \cdot \underline{\mathbf{1}} - 2 \cdot \underline{\mathbf{s}} \cdot \tilde{\underline{\mathbf{D}}}^T) \cdot \tilde{\mathbf{c}} \\ \text{sous la contrainte} \quad & -c_k^+ \leq 0 \text{ et } c_k^- \leq 0 \quad \forall k \end{aligned} \quad (2.18)$$

avec $\tilde{\mathbf{c}} = [\mathbf{c}^{+T} \ \mathbf{c}^{-T}]^T$ et $\tilde{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ -\mathbf{D} \end{bmatrix}$ et $\mathbf{1} = [1 \dots 1]$.

Ce programme quadratique est convexe et possède au moins un minimum global qui peut être notamment obtenu par des algorithmes de point intérieur [Meh92, CDS99] ou de gradient conjugué [GL96]. Ces méthodes génériques sont largement implémentées dans les logiciels scientifiques mais elles présentent le défaut d'être très coûteuses en temps de calcul et en mémoire.

Des travaux récents ont permis d'accélérer la résolution de ce problème, notamment l'Iterative Reweighted Least-Squares (IRLS) [Fuc06], qui est très proche de l'algorithme FOCUSS, et l'approche par continuation homotopique [MCW05, MF06] qui permet de trouver la solution exacte du problème $\mathcal{P}_{1,\lambda}^{DN}$ en un nombre fini d'itérations.

2.3 Seuillage alterné

Les algorithmes précédents proposent des solutions pour un dictionnaire général. Lorsque le dictionnaire présente une structure particulière, des stratégies peuvent être mises en place pour simplifier et accélérer la résolution. Nous présentons ainsi deux algorithmes de seuillage doux (section 2.3.1) et de seuillage doux alterné (section 2.3.2) pour minimiser $\mathcal{C}_{1,\lambda}^{DN}(\underline{\mathbf{s}}, \mathbf{c}, \mathbf{D})$ lorsque le dictionnaire est respectivement une base orthonormée et une union de bases orthonormées. Nous présentons en section 2.3.3 un algorithme de seuillage itératif, basé sur le même principe, mais applicable à tout type de dictionnaire, sans contrainte de structure.

2.3.1 Seuillage doux

Lorsque le dictionnaire est une base orthonormée, la solution optimale de $\mathcal{P}_{1,\lambda}^{DN}$ est obtenue par seuillage doux [Don95, CDS99].

Rappelons que la matrice \mathbf{D} d'une base orthonormée est carrée, et que son inverse est égale à sa transposée : $\mathbf{D}^T \cdot \mathbf{D} = \mathbf{D} \cdot \mathbf{D}^T = \mathbf{I}$. Les bases d'ondelettes [Mal98] et les cosinus locaux sont exemples de bases orthonormées utilisées pour le traitement de signal ou d'images.

Pour une base orthonormée, le critère $\mathcal{C}_{1,\lambda}^{DN}(\underline{\mathbf{s}}, \mathbf{c}, \mathbf{D})$ peut s'écrire sous la forme :

$$\|\mathbf{e} - \mathbf{c}\|_2^2 + \lambda \cdot \|\mathbf{c}\|_1, \quad (2.19)$$

avec $\mathbf{e}^T = \underline{\mathbf{s}} \cdot \mathbf{D}^T$. Le critère est alors séparable, c'est-à-dire qu'il dépend de façon indépendante de chacun des paramètres c_k , $k \in \llbracket 1, K \rrbracket$ (noter que $K = L$).

Définissons la fonction de seuillage doux η_λ par :

$$\eta_\lambda(c) = \begin{cases} c - \frac{\lambda}{2} & , \text{ si } c \geq \frac{\lambda}{2} \\ 0 & , \text{ si } |c| \leq \frac{\lambda}{2} \\ c + \frac{\lambda}{2} & , \text{ si } c \leq -\frac{\lambda}{2} \end{cases}. \quad (2.20)$$

La fonction de seuillage doux η_λ est représentée sur la figure 2.1.

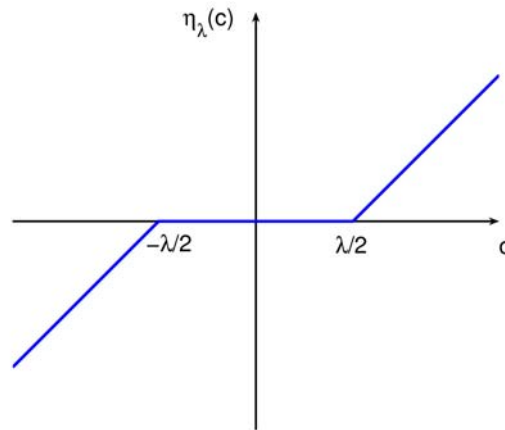


FIG. 2.1 – Graphe de la fonction de seuillage doux.

Donoho a prouvé [Don95] que les coefficients \mathbf{c}^{opt} optimaux sont donnés par :

$$\mathbf{c}^{\text{opt}} = \eta_{\lambda}(\underline{\mathbf{D}} \cdot \underline{\mathbf{s}}^T), \quad (2.21)$$

ce qui correspond à calculer les coefficients de la décomposition exacte, par $\underline{\mathbf{D}} \cdot \underline{\mathbf{s}}^T$, puis à seuiller ces coefficients afin d'augmenter la parcimonie.

2.3.2 Seuillage doux alterné

Sardy, Bruce et Tseng [SBT00] proposent une méthode de seuillage doux alterné, appelée aussi Block Coordinate Relaxation, lorsque le dictionnaire est constitué d'une union de bases orthonormées. Cela correspond à toute union des bases orthonormales précédemment citées dans la section 2.3.1. Par exemple, Daudet et Torrèsani [DT02] proposent d'utiliser l'union de la transformée en cosinus discrète modifiée (TCDM) et d'une base d'ondelettes pour décomposer un signal sonore entre ses composantes tonales (les parties oscillatoires), transitoires (les parties impulsives) et le bruit. Il faut noter qu'ils n'utilisent pas la méthode de seuillage doux alterné pour la décomposition.

Si le dictionnaire est $\underline{\mathbf{D}} = \begin{bmatrix} \underline{\mathbf{D}}_1 \\ \vdots \\ \underline{\mathbf{D}}_P \end{bmatrix}$, l'algorithme consiste, à chaque étape, à choisir une base $\underline{\mathbf{D}}_p$, à fixer les coefficients $\mathbf{c}_{p'}$ associés aux autres bases, et à mettre à jour les coefficients \mathbf{c}_p associés à la base $\underline{\mathbf{D}}_p$ par seuillage doux.

Description du seuillage doux alterné :

Initialisation du compteur $i := 0$ et des coefficients $\mathbf{c}^T = [\mathbf{c}_1^T \dots \mathbf{c}_P^T]$

Faire

1. Sélection d'une base $\underline{\mathbf{D}}_p$
2. Retrait de la contribution des autres bases au signal : $\underline{\mathbf{s}}_p := \underline{\mathbf{s}} - \sum_{p' \neq p} \mathbf{c}_{p'}^T \cdot \underline{\mathbf{D}}_{p'}$
3. Mise à jour des coefficients associés à la base $\underline{\mathbf{D}}_p$ par seuillage doux :

$$\mathbf{c}_p^{(i)} := \eta_\lambda(\underline{\mathbf{D}}_p \cdot \underline{\mathbf{s}}_p^T) \quad (2.22)$$

4. Les autres coefficients sont inchangés :

$$\mathbf{c}_{p'}^{(i)} := \mathbf{c}_{p'}^{(i-1)}, \forall p' \neq p \quad (2.23)$$

Tant que la convergence n'est pas atteinte

La sélection des bases $\underline{\mathbf{D}}_p$ peut s'effectuer dans l'ordre naturel, ou suivre une méthode plus fine qui anticipe sur l'itération suivante pour sélectionner la base qui diminuera le plus le critère $\mathbf{c}_{1,\lambda}^{DN}(\underline{\mathbf{s}}, \mathbf{c}, \underline{\mathbf{D}})$.

A chaque pas, alors que toutes les autres bases sont fixées, la mise à jour de la base $\underline{\mathbf{D}}_p$ est optimale au sens du critère de Basis Pursuit. Il est prouvé que l'algorithme converge vers le minimum du critère $\mathbf{c}_{1,\lambda}^{DN}(\underline{\mathbf{s}}, \mathbf{c}, \underline{\mathbf{D}})$, que les bases $\underline{\mathbf{D}}_p$ soient sélectionnées l'une après l'autre dans un ordre prédéfini, ou qu'elles soient sélectionnées selon une règle de plus grande descente [SBT00]. Ici, la convergence signifie que le poids $\|\mathbf{c}^{(i)}\|_1$ est borné et que tous les points fixes de l'algorithme sont une solution de Basis Pursuit.

Les coefficients doivent être initialisés avant d'appliquer l'algorithme, par exemple par régression linéaire (application de la pseudo-inverse à droite : $\mathbf{c}^{(0)} = (\underline{\mathbf{s}} \cdot \underline{\mathbf{D}}^+)^T$). L'algorithme de seuillage alterné permet d'accélérer la convergence par rapport à des méthodes de programmation quadratique, notamment quand l'approximation est grossière (λ grand). En revanche, il est incapable de résoudre le problème pour $\lambda \rightarrow 0$ car le seuillage doux n'a aucun effet dans ce cas et les coefficients ne sont jamais modifiés.

2.3.3 Seuillage itératif

En se basant sur le principe de mise à jour itérative des coefficients par seuillage, Figueiredo et Nowak [FN03], Daubechies, Defrise et De Mol [DDM04], Combettes et Wajs [CW05] et Elad [Ela] proposent diverses versions d'un algorithme de seuillage itératif applicable à tout type de dictionnaire $\underline{\mathbf{D}}$, et non plus uniquement aux unions de bases orthonormées. L'algorithme est le suivant :

Description du seuillage itératif :

Initialisation du compteur $i := 0$ et des coefficients $\mathbf{c}^{(0)} = \mathbf{0}$

Faire la mise à jour des coefficients :

$$c_k^{(i)} := \eta_\lambda \left(c_k^{(i-1)} + \langle \underline{\mathbf{s}} - \mathbf{c}^{(i-1)T} \cdot \underline{\mathbf{D}}, \underline{\mathbf{d}}_k \rangle \right) \quad (2.24)$$

Tant que la convergence n'est pas atteinte

Le principe de l'algorithme est d'effectuer une descente de gradient avec un pas de 1 le long du gradient de l'erreur de reconstruction par rapport aux coefficients, puis de seuiller les coefficients. Cet algorithme converge vers un minimum local du critère $\mathcal{C}_{1,\lambda}^{DN}(\underline{g}, \mathbf{c}, \mathbf{D})$.

2.4 Conclusion

Nous avons présenté dans ce chapitre trois grandes familles d'algorithmes destinés au calcul de coefficients parcimonieux dans un dictionnaire. Nous avons exposé dans un premier temps les algorithmes heuristiques MP et OMP, qui recherchent les atomes actifs un par un. Il est prouvé qu'ils convergent vers la solution de \mathcal{P}_0 lorsque cette solution est suffisamment parcimonieuse.

Nous avons ensuite présenté l'algorithme FOCUSS qui vise à résoudre le problème $\mathcal{P}_{\tau,\lambda}^{DN}$, puis les algorithmes de programmation quadratique et de seuillage alterné dédiés au problème de Basis Pursuit $\mathcal{P}_{1,\lambda}^{DN}$.

Nous ne disposons pas d'étude comparative entre ces différents algorithmes qui permette de montrer leurs points forts et leurs faiblesses. Nous pouvons cependant rappeler que pour tous ces algorithmes, il existe un théorème de convergence au moins vers un optimum local d'un problème de type \mathcal{P}_τ ou $\mathcal{P}_{\tau,\lambda}^{DN}$.

Notons dès maintenant que, dans la deuxième partie de la thèse, nous étudierons spécifiquement l'algorithme MP, afin de l'adapter à un modèle structuré d'atomes.

Armés de tous ces algorithmes de décomposition parcimonieuse, nous pouvons maintenant nous attaquer au problème d'apprentissage des atomes permettant les décompositions les plus parcimonieuses. De nombreuses méthodes nécessitent en effet le recours à un algorithme de décomposition parcimonieuse tel que ceux que nous venons de présenter.

Chapitre 3

Apprentissage de dictionnaire

Nous avons présenté dans le chapitre précédent divers algorithmes pour le calcul de la décomposition la plus parcimonieuse d'un signal sur un dictionnaire donné. Mais la qualité et la parcimonie d'une décomposition dépendent, au delà du critère et de l'algorithme utilisés, du choix d'un dictionnaire adapté à l'application désirée. Nous discutons dans ce chapitre de la façon de choisir le dictionnaire, soit parmi une famille de dictionnaires connus, soit en calculant, en plus des coefficients, le dictionnaire le plus adapté à un ensemble d'exemples de signaux.

3.1 Motivations

Les deux chapitres précédents sont dédiés au calcul de la décomposition la plus parcimonieuse d'un signal sur un dictionnaire fixé, ce qui correspond au problème :

$$\mathbf{c}^{\text{opt}} = \arg \min_{\mathbf{c}} \mathcal{C}(\underline{\mathbf{s}}, \mathbf{c}, \underline{\mathbf{D}}) . \quad (3.1)$$

Dans ce chapitre, nous traitons de l'adaptation et de l'apprentissage de dictionnaire, ce qui correspond à relâcher un degré de liberté dans (3.1) : le dictionnaire $\underline{\mathbf{D}}$. Il s'agit d'augmenter les chances que les signaux traités soient décomposés de façon parcimonieuse. Pour cela, nous supprimons la contrainte d'un dictionnaire $\underline{\mathbf{D}}$ fixé, et nous choisissons le dictionnaire le plus favorable parmi un ensemble de dictionnaires \mathbb{D} .

Le dictionnaire optimal est le plus adapté aux signaux à traiter, au sens où il reflète le mieux la nature des signaux. Ainsi, pour décomposer efficacement le son d'un instrument, le dictionnaire doit représenter son timbre ainsi que sa tessiture (la palette de notes que l'instrument peut jouer). Pour des images, le dictionnaire doit être adapté à la texture, mais aussi au type de contours. Par exemple, pour des images artificielles, ou des photos de constructions humaines, les contours seront souvent des droites ou des formes géométriques, contrairement aux scènes naturelles où les contours sont plus libres.

D'une façon générale, le dictionnaire doit être en mesure de bien représenter les **motifs** significativement récurrents du signal, par exemple les différentes notes dans un enregistrement de piano, ou le complexe QRS dans un électrocardiogramme (ECG).

Les signaux utilisés pour la sélection ou l'apprentissage du dictionnaire doivent être de la même classe, c'est-à-dire qu'ils doivent présenter des caractéristiques communes qui permettent de les rassembler. Les signaux doivent donc être à la fois assez proches pour former une classe cohérente et assez divers pour inclure toutes les caractéristiques communes de la classe. Si toute la diversité de la classe n'est pas présente dans les signaux d'apprentissage, il est évident que le dictionnaire résultant sera mal adapté aux caractéristiques manquantes.

Il faut noter que la recherche d'un dictionnaire adapté peut servir à réduire la complexité ou la dimension des signaux. L'analyse en composantes principales (ACP), qui est décrite dans la section 3.2, consiste ainsi à sélectionner les composantes les plus énergétiques du signal pour former le dictionnaire.

3.1.1 Espace de recherche du dictionnaire

Le choix du dictionnaire se fait parmi un ensemble \mathbb{D} , qui peut être de plusieurs natures :

- catalogue : c'est un ensemble de dictionnaires connus *a priori* dans lesquels piocher, selon un critère,

$$\text{par exemple, } \mathbb{D} = \{\underline{\mathbf{D}}^{Fourier}, \underline{\mathbf{D}}^{Haar}\}, \quad (3.2)$$

- paramétrique : tous les dictionnaires sont analytiques et dépendent d'un ensemble de paramètres. Le choix du dictionnaire consiste à optimiser ce jeu de paramètres,

$$\text{par exemple, } \mathbb{D} = \{\underline{\mathbf{D}}_{L_t}^{TFCT} \text{ avec } L_t \text{ la taille des fenêtres}\}, \quad (3.3)$$

- sous-ensemble de $\mathbb{R}^{L \times K}$: les atomes sont décrits uniquement par leur forme d'onde (ils n'ont pas de description analytique),

$$\text{par exemple, } \mathbb{D} = \{\underline{\mathbf{D}} \in \mathbb{R}^{L \times K} / \|\underline{d}_k\| = 1, \forall k \in \llbracket 1, K \rrbracket\}. \quad (3.4)$$

Nous ne traitons que le troisième point dans la suite de ce travail, bien que la section suivante puisse s'appliquer aux trois points.

3.1.2 Méthode d'optimisation du dictionnaire

Le choix du dictionnaire peut être fait soit par connaissance experte des signaux traités et des dictionnaires habituellement associés, soit par optimisation d'un critère.

Le premier cas, le choix par connaissance experte *a priori*, correspond surtout à la sélection de dictionnaires dans un catalogue, ou à un dictionnaire dépendant de peu de paramètres. Par exemple, comme mentionné dans la section 2.3.2, un signal musical composé majoritairement de tonales (notes tenues, par exemple) se décompose de façon parcimonieuse sur un dictionnaire de type TCDM, et un signal composé de transitoires (attaques de notes, par exemple) se décompose de façon parcimonieuse sur un dictionnaire d'ondelettes [DT02]. D'une façon analogue, il a été montré que les images naturelles aux textures complexes se décomposent parcimonieusement sur une TDCM

à deux dimensions, tandis que des dessins de bande dessinée sont mieux décomposés par une transformée en “curvelettes” [SED04].

Le deuxième cas, l’optimisation d’un critère, peut correspondre à la recherche paramétrique du dictionnaire. Par exemple, Coifman et Wickerhauser [CW92] ont proposé la méthode Best Basis Selection qui choisit sur quelle base d’ondelettes $\underline{\mathbf{D}}_i$ est décomposée chaque trame du signal $\underline{\mathbf{s}}_q$, en fonction de l’entropie de sa décomposition $\underline{\mathbf{s}}_q \cdot \underline{\mathbf{D}}_i^{-1}$.

De façon générale, le problème d’optimisation d’un critère de choix de dictionnaire est le suivant :

$$\underline{\mathbf{D}}^{\text{opt}} = \arg \min_{\underline{\mathbf{D}} \in \mathbb{D}} \left[\sum_{q=1}^Q \min_{\mathbf{c}} \mathcal{C}(\underline{\mathbf{s}}_q, \mathbf{c}, \underline{\mathbf{D}}) \right] . \quad (3.5)$$

Le problème (3.5) revient à chercher le dictionnaire qui permet la décomposition la plus parcimonieuse, en moyenne, de Q signaux $\underline{\mathbf{s}}_q$. Il est une double généralisation du problème (3.1), car le signal $\underline{\mathbf{s}}$ est remplacé par une classe de signaux $\underline{\mathbf{s}}_q$ et le dictionnaire est un paramètre libre à optimiser.

Le problème (3.5) peut se noter également

$$\underline{\mathbf{D}}^{\text{opt}} = \arg \min_{\underline{\mathbf{D}} \in \mathbb{D}} \left[\min_{\mathbf{C}} \mathcal{C}(\underline{\mathbf{S}}, \mathbf{C}, \underline{\mathbf{D}}) \right] . \quad (3.6)$$

où $\underline{\mathbf{S}} = \begin{bmatrix} \underline{\mathbf{s}}_1 \\ \vdots \\ \underline{\mathbf{s}}_Q \end{bmatrix}$ et $\mathbf{C} = [\mathbf{c}_1 \dots \mathbf{c}_Q]$ sont respectivement les Q trames de signal et les coefficients de leur décomposition sur le dictionnaire $\underline{\mathbf{D}}$. La notation des critères est étendue des vecteurs aux matrices par :

$$\mathcal{C}(\underline{\mathbf{S}}, \mathbf{C}, \underline{\mathbf{D}}) = \sum_{q=1}^Q \mathcal{C}(\underline{\mathbf{s}}_q, \mathbf{c}_q, \underline{\mathbf{D}}) . \quad (3.7)$$

Notons que le problème (3.5) repose sur l’hypothèse probabiliste de signaux indépendants entre eux. Étant donné qu’ils participent à la même classe, les signaux ont pourtant des caractéristiques communes qui doivent transparaître dans leur distribution conjointe. L’indépendance supposée entre les signaux n’est due qu’à des raisons de simplicité de calcul, de la même façon que les coefficients de la décomposition ont été supposés indépendants. Nous ne chercherons pas par la suite à détailler les conséquences de cette approximation.

3.1.3 Sommaire

Les trois dernières sections de ce chapitre décrivent les trois grandes familles d’apprentissage d’un dictionnaire non-analytique, qui tentent de répondre au problème (3.5). Elles correspondent au troisième point de la section 3.1.1.

Dans la section 3.2, nous exposons l’Analyse en Composantes Principales (ACP), qui est un outil classique d’analyse de données et de réduction d’information, mais qui ne

favorise pas la parcimonie. Dans la section 3.3, nous décrivons l'Analyse en Composantes Indépendantes, qui vise à dépasser la décorrélation apportée par l'ACP, et pose les principes de l'apprentissage de dictionnaire pour les décompositions parcimonieuses, ou "sparse coding", qui est traité dans les sections 3.4 et 3.5.

3.2 Analyse en Composantes Principales

L'Analyse en Composantes Principales (ACP) est un outil classique de traitement de signal qui permet de réduire la dimension des données à traiter. Dans le cadre de l'apprentissage de dictionnaire, l'ACP revient à calculer une base orthonormale, aux propriétés énergétiques particulières, mais sans lien avec la parcimonie. Les coefficients obtenus par transformation du signal sur ce dictionnaire sont notamment décorrés entre eux.

Les travaux de Pearson [Pea01] sont les premiers pas vers l'ACP et permettent de trouver, pour un nuage de points, les axes principaux qui génèrent l'hyperplan tel que le carré de la distance de l'hyperplan aux points est minimal en moyenne. Hotelling [Hot33], sans lien avec les travaux de Pearson, a établi le calcul des composantes principales, afin d'étudier les relations de corrélation entre les différentes composantes d'un vecteur d'observation.

L'ACP, aussi appelée transformée de Karhunen-Loève [Kar46, Loè55] ou transformée de Hotelling [Hot33] consiste, à partir d'un nuage de vecteurs \underline{s}_q de dimension L , et pour une dimension K donnée inférieure à L , à trouver une des bases orthogonales du sous-espace de dimension K qui conservent le plus d'énergie des vecteurs.

Définissons $\mathbb{D}_K^{BON} = \{\underline{\mathbf{D}} \in \mathbb{R}^{K \times L} / \underline{\mathbf{D}} \cdot \underline{\mathbf{D}}^T = \mathbf{I}_K\}$ l'ensemble des bases orthonormales du sous-espace de dimension K . L'objectif de l'ACP est de trouver la base orthonormale $\underline{\mathbf{D}}$ de l'espace de dimension L telle que chacune de ses sous-matrices $\underline{\mathbf{D}}_K$ (les K premières lignes de $\underline{\mathbf{D}}$) soit l'élément de \mathbb{D}_K^{BON} donnant les coefficients les plus énergétiques. En termes mathématiques, chaque sous-matrice $\underline{\mathbf{D}}_K$, pour $K \in \llbracket 1, L \rrbracket$, minimise :

$$\sum_{q=1}^Q \|\underline{s}_q \cdot \underline{\mathbf{D}}_K^T\|_2^2. \quad (3.8)$$

Une première manière pour trouver les atomes \underline{d}_k du dictionnaire consiste à minimiser successivement le critère (3.8) pour K allant de 1 à L , en forçant les $K - 1$ lignes de $\underline{\mathbf{D}}_K$ à être la matrice $\underline{\mathbf{D}}_{K-1}$ obtenue au rang précédent.

Une deuxième méthode consiste à effectuer une décomposition en valeurs singulières de la matrice $\underline{\mathbf{S}} = [\underline{s}_1^T \dots \underline{s}_Q^T]$:

$$\underline{\mathbf{S}} = \underline{\mathbf{U}}^T \cdot \underline{\mathbf{\Delta}} \cdot \underline{\mathbf{V}}, \quad (3.9)$$

où $\underline{\mathbf{U}}$ et $\underline{\mathbf{V}}$ sont des matrices orthonormales, et où les L premières lignes de $\underline{\mathbf{\Delta}}$ forment une matrice diagonale contenant les valeurs singulières, en ordre décroissant d'énergie, alors que les autres lignes sont nulles. Le dictionnaire $\underline{\mathbf{D}}$ optimal est alors donné par la matrice $\underline{\mathbf{V}}$ (notée sans gras pour indiquer qu'elle est une matrice carrée de taille $L \times L$).

L'ACP est un outil puissant d'analyse statistique et de réduction de dimension, et dans le cadre de l'analyse de signaux, elle permet d'extraire notamment la forme d'onde la plus corrélée avec toutes les trames \underline{s}_q . Nous verrons dans la deuxième partie de la thèse que cette propriété peut être exploitée pour élaborer des dictionnaires aux propriétés parcimonieuses.

En revanche, en tant que telle, la matrice orthogonale obtenue par ACP est un dictionnaire qui n'a aucune bonne propriété de parcimonie. La contrainte d'orthogonalité entre les atomes est notamment très restrictive. La deuxième restriction concerne la non-redondance de ce dictionnaire. En effet, nous avons déjà évoqué le fait que la redondance permet d'augmenter le choix d'atomes lors de la décomposition et ainsi augmente la capacité à produire des décompositions parcimonieuses.

En rappelant que L est la taille des signaux, l'ACP a $\frac{L(L-1)}{2}$ degrés de libertés, et non $L \times L$, de par la contrainte d'orthogonalité.

La section suivante propose des méthodes statistiques basées sur l'indépendance entre les coefficients permettant de relâcher la contrainte d'orthogonalité entre les atomes.

3.3 Analyse en Composantes Indépendantes

L'Analyse en Composantes Indépendantes (ACI) est une technique d'analyse statistique de données multivariées qui cherche à extraire les processus indépendants qui génèrent ces données. Du point de vue de l'apprentissage de dictionnaire, il s'agit d'apprendre une matrice carrée $\underline{\mathbf{D}}$ inversible. Comme pour l'ACP, le dictionnaire est une base, et non pas un dictionnaire redondant.

Le modèle de l'ACI est :

$$\underline{\mathbf{S}} = \mathbf{C}^T \cdot \underline{\mathbf{D}}, \quad (3.10)$$

où $\underline{\mathbf{S}}$ représente les données multivariées, et \mathbf{C} représente les composantes indépendantes. L'objectif de l'ACI est de trouver, à partir des données $\underline{\mathbf{S}}$, une matrice de séparation $\overline{\mathbf{B}} = \underline{\mathbf{D}}^{-1}$, telle que les composantes \mathbf{Y} :

$$\mathbf{Y} = (\underline{\mathbf{S}} \cdot \overline{\mathbf{B}})^T, \quad (3.11)$$

soient les plus indépendantes possible entre elles, au sens statistique. La notation $\overline{\mathbf{B}}$, avec la barre au dessus plutôt qu'en dessous, signifie que la taille de la matrice est $L \times K$. Ici, la différence de notation est surtout sémantique puisque $\underline{\mathbf{D}}$ et $\overline{\mathbf{B}}$ sont carrées.

Rappelons que l'indépendance statistique entre deux variables signifie que la connaissance de la valeur de l'une n'apporte aucune connaissance sur la valeur de l'autre. Mathématiquement, l'indépendance entre les composantes \mathbf{y}_k est définie à partir des densités de probabilité marginales et de la densité de probabilité jointe par :

$$p(\mathbf{y}_1, \dots, \mathbf{y}_K) = \prod_{k=1}^K p(\mathbf{y}_k) \quad (3.12)$$

Par rapport à l'ACP, l'ACI va plus loin en passant de la décorrélation entre coefficients à l'indépendance statistique. En effet, les composantes principales sont orthogonales entre elles, ce qui correspond à une intercorrélacion nulle entre deux composantes du point de vue statistique. En revanche, l'orthogonalité n'impose rien pour les moments d'ordre supérieurs à deux, et n'impose donc pas l'indépendance.

Pour une variable gaussienne :

$$p_x(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad (3.13)$$

où μ est la moyenne, et σ^2 est la variance, tous les moments d'ordre supérieur à deux sont nuls et les notions de décorrélation et d'indépendance sont équivalentes. En un sens, l'ACP est donc une ACI où les coefficients sont supposés de distribution gaussienne.

Tout l'intérêt de l'ACI est justement de s'éloigner le plus possible de l'ACP et de chercher les distributions les "moins gaussiennes" pour tendre vers l'indépendance. Une première technique, exposée à la section 3.3.1, consiste à maximiser un critère de "non-gaussianité" de la distribution. Sachant que la distribution des composantes issues de l'ACP est gaussienne et donc très peu parcimonieuse, une autre technique, exposée à la section 3.3.2, consiste à imposer aux composantes des distributions connues pour leurs propriétés de parcimonie, tout en supposant que les composantes sont indépendantes. Cette deuxième technique est le point de départ historique des méthodes d'apprentissage de dictionnaire pour les décompositions parcimonieuses, qui est le sujet principal de cette thèse.

Historiquement, l'ACI est apparue au milieu des années 80 pour des applications de neurophysiologie, grâce à Hérault, Jutten et Ans [HJA85]. Cardoso [Car89] et Comon [Com89] ont introduit en 1989 les méthodes à base de tenseurs de cumulants d'ordre supérieur – implicitement, d'ordre supérieur à 2 – et l'algorithme JADE, basé sur ces méthodes a été proposé en 1993 par Cardoso [CS93]. Le principe d'information-maximisation (infomax) a été utilisé par Bell et Sejnowski en 1995 pour l'ACI [BS95]. Enfin, Hyvärinen et Oja ont présenté en 1997 [HO97] l'algorithme de point fixe FastICA, qui permet de traiter des données importantes grâce à sa rapidité calculatoire. Pour plus de détails, se référer à l'historique de l'ACI exposé dans [HKO01].

Le problème de la séparation de sources, qui est exposé au chapitre 4, est l'application majeure de l'ACI. Nous discutons dans la section 5.1 du lien très fort entre les trois notions d'ACI, de séparation de sources, et d'apprentissage de dictionnaire.

Comme mentionné plus tôt, une description détaillée des critères et algorithmes d'ACI est disponible dans [HKO01]. Nous ne les détaillons pas ici, mais listons les quelques critères permettant de tendre vers l'indépendance.

3.3.1 ACI par non-gaussianité

Une première famille de méthodes [Car89, Com89] consiste à assimiler indépendance à "non-gaussianité". En effet, le théorème de la limite centrale établit que sous certaines conditions, la distribution de la somme de k variables aléatoires tend vers une distribution gaussienne lorsque k tend vers l'infini. Cela signifie qu'en général la distribution

d'une somme de deux variables aléatoires est plus proche d'une loi normale que chacune des deux distributions initiales. Ainsi, pour retrouver les distributions initiales à partir d'un mélange, le principe est de chercher les distributions les moins proches d'une loi normale. Cette méthode ne peut marcher que si au plus une distribution initiale est normale (cela correspond d'ailleurs souvent, lorsque l'ACI est utilisée pour la séparation de sources, à la modélisation du bruit).

La non-gaussianité est généralement mesurée à l'aide du cumulatif d'ordre 4, nommé kurtosis, qui doit être maximisé, à variance fixée. Une autre mesure de non-gaussianité est la "negentropie", définie comme la différence entre l'entropie d'une variable gaussienne et l'entropie de la composante \mathbf{y} . Elle est basée sur le fait que l'entropie est maximale quand la variable est gaussienne. L'objectif est alors de maximiser, à variance fixée, la negentropie, qui est toujours positive ou nulle, et vaut zéro lorsque les composantes sont gaussiennes.

3.3.2 ACI par maximum de vraisemblance

La deuxième grande famille de méthodes [BS95, HO97] est basée sur le principe de maximum de vraisemblance, qui choisit la matrice de séparation $\overline{\mathbf{B}}$ maximisant la vraisemblance des données $\underline{\mathbf{S}}$. Les coefficients, de distribution supposée connue *a priori*, sont des données cachées car ils peuvent être exprimés avec $\underline{\mathbf{S}}$ et $\overline{\mathbf{B}}$. Toute la difficulté réside dans le choix de la distribution *a priori* sur les composantes. Souvent, la distribution choisie est parcimonieuse, comme par exemple la loi laplacienne. C'est avec les méthodes d'ACI par maximum de vraisemblance qu'est né le champ spécifique de l'apprentissage de dictionnaire pour des décompositions parcimonieuses, détaillé dans les sections 3.4 et 3.5.

Dans de nombreuses méthodes d'ACI, la moitié du travail est effectuée lors d'une étape de blanchiment. Cette étape transforme les observations $\underline{\mathbf{S}}$ en observations blanches, qui ont une matrice de covariance identité, ce qui signifie que ces observations sont orthogonales, et de variance unité. Grâce à cette étape préalable, la deuxième moitié du travail, qui consiste à tendre vers l'indépendance, est limitée à la recherche d'une matrice orthogonale. Le nombre de paramètres à optimiser passe donc de L^2 à $\frac{L(L-1)}{2}$.

L'ACI a un lien particulier avec la recherche de la parcimonie, car les méthodes par maximum de vraisemblance imposent aux coefficients d'avoir une distribution parcimonieuse. Dans le cadre bayésien, imposer une loi laplacienne sur les coefficients :

$$p_y(y) = \frac{\lambda}{2} \exp(-\lambda|y|) , \quad (3.14)$$

conduit à la minimisation suivante :

$$\overline{\mathbf{B}}^{\text{opt}} = \arg \min_{\overline{\mathbf{B}}} \|\overline{\mathbf{B}} \cdot \underline{\mathbf{S}}\|_1 . \quad (3.15)$$

Il s'agit de minimiser la diversité en norme l^1 des coefficients, par rapport aux coefficients et au dictionnaire. Les coefficients étant obtenus par inversion du dictionnaire (application de $\overline{\mathbf{B}}$), le critère à minimiser ne dépend que du dictionnaire.

L'ACI est limitée au cas d'une matrice de séparation carrée, ce qui empêche toute redondance. Quelques pistes ont été proposées pour relâcher cette contrainte en proposant de chercher des composantes quasi-orthogonales, après blanchiment, au lieu de la stricte orthogonalité [HCO99, Car92]. Le problème est cependant plus compliqué dans le cas redondant : il ne suffit pas de trouver le dictionnaire, il faut également estimer les composantes. Afin de tendre vers la parcimonie, notamment pour des dictionnaires redondants, il convient donc d'adopter des critères spécifiques.

Nous présentons dans la section suivante le domaine de l'apprentissage de dictionnaire pour les décompositions parcimonieuses, qui a émergé du monde de l'ACI pour devenir une discipline à part entière vers 1995.

3.4 Apprentissage de base pour la décomposition parcimonieuse exacte

Le problème de l'apprentissage de dictionnaire est de trouver le dictionnaire $\underline{\mathbf{D}}$ qui permet la décomposition la plus parcimonieuse d'un ensemble de signaux d'apprentissage $\underline{\mathbf{s}}_q$.

Historiquement, le problème a été abordé du point de vue probabiliste, sous le formalisme bayésien. Dans cette thèse, nous établissons systématiquement le lien entre ce formalisme et la minimisation d'un critère dont nous expliquons la signification de chaque terme.

Selon les *a priori* et les approximations du modèle probabiliste, trois contraintes contradictoires pourront être présentes sous différentes formes dans le critère à minimiser, à savoir :

1. le poids de l'erreur de reconstruction,
2. la diversité des coefficients,
3. le poids du dictionnaire,

soit

$$\sum_{q=1}^Q \|\underline{\mathbf{s}}_q - \mathbf{c}_q^T \cdot \underline{\mathbf{D}}\|_2^2 + \sum_{q=1}^Q f(\mathbf{c}_q) + g(\underline{\mathbf{D}}) \quad (3.16)$$

Dans le cas où la décomposition n'est pas exacte, le premier terme sert à maîtriser le poids de l'erreur, pour coller au mieux aux signaux d'apprentissage.

Le deuxième terme impose quant à lui une certaine parcimonie aux coefficients.

Enfin, le troisième terme est nécessaire pour éviter, en vertu de l'égalité

$$\mathbf{c}^T \cdot \underline{\mathbf{D}} = (\alpha \cdot \mathbf{c}^T) \cdot \left(\frac{1}{\alpha} \cdot \underline{\mathbf{D}}\right), \quad (3.17)$$

pour tout $\alpha \neq 0$, la situation inintéressante où α tend vers zéro, ce qui fait artificiellement diminuer le poids des coefficients et augmenter celui du dictionnaire, sans résoudre le problème. Le troisième terme sert à normaliser le dictionnaire, et prendra différentes formes selon les cas.

Nous traitons dans cette section le problème de l'apprentissage d'un dictionnaire non-redondant, ou base, permettant la décomposition parcimonieuse exacte des signaux d'apprentissage. Dans la section 3.5, nous nous intéresserons à l'apprentissage d'un dictionnaire redondant, permettant une approximation parcimonieuse des signaux d'apprentissage.

Dans cette section, l'approche [BS95, BS97, HO97] est très liée à l'ACI, car le dictionnaire est une matrice carrée, et l'erreur de reconstruction est nulle. Cette approche a déjà été abordée à la section 3.3.2 sous l'angle du maximum de vraisemblance. Le modèle est :

$$\underline{s} = \mathbf{c}^T \cdot \underline{\mathbf{D}} . \quad (3.18)$$

3.4.1 Principe probabiliste : maximum de vraisemblance

Si la distribution $p_{\mathbf{c}}(\mathbf{c})$ des coefficients \mathbf{c} est connue *a priori*, et si les coefficients (inconnus) sont remplacés par $(\underline{s} \cdot \underline{\mathbf{D}}^{-1})^T$, la log-vraisemblance des signaux observés \underline{s}_q connaissant le dictionnaire $\underline{\mathbf{D}}$ [BS95, HKO01] s'exprime comme :

$$\mathcal{L}(\underline{\mathbf{D}}) = \left(\frac{1}{Q} \sum_{q=1}^Q \log p_{\mathbf{c}} \left((\underline{s}_q \cdot \underline{\mathbf{D}}^{-1})^T \right) \right) + \log |\det \underline{\mathbf{D}}^{-1}| , \quad (3.19)$$

avec $\det \underline{\mathbf{D}}^{-1}$ le déterminant de la matrice $\underline{\mathbf{D}}^{-1}$.

En notant la moyenne $E\{t\} = \frac{1}{Q} \sum_{q=1}^Q t_q$, maximiser la vraisemblance correspond donc au problème de minimisation suivant :

$$\underline{\mathbf{D}}^{\text{opt}} = \arg \min_{\underline{\mathbf{D}}} \lambda \cdot E\{f(\underline{s} \cdot \underline{\mathbf{D}}^{-1})\} + \log |\det \underline{\mathbf{D}}| , \quad (3.20)$$

où la densité de probabilité s'écrit $p_{\mathbf{c}}(\mathbf{c}) \propto \exp(-\lambda f(\mathbf{c}^T))$. Si les coefficients sont supposés laplaciens, par exemple, $f(\mathbf{c}) = \|\mathbf{c}\|_1$.

3.4.2 Interprétation en tant que critère

Le critère à minimiser dans le problème (3.20) se compose de deux termes seulement. En effet, le terme de minimisation de l'erreur n'a pas lieu d'être, puisque l'erreur est nulle. Le premier terme du critère est une mesure de diversité, qui peut prendre des formes différentes selon la loi *a priori* des coefficients. Minimiser ce terme correspond à augmenter la parcimonie des coefficients. Le deuxième terme contrôle le poids du dictionnaire pour répondre au besoin de normalisation du dictionnaire.

Il est intéressant de noter, dans le cas d'un dictionnaire carré et en l'absence de bruit, que les coefficients \mathbf{c} disparaissent du critère, et que la minimisation se fait directement sur le dictionnaire.

3.4.3 Dérivation du critère

Nous introduisons la matrice inverse $\overline{\mathbf{B}} = \underline{\mathbf{D}}^{-1}$, car la minimisation du critère (3.20) par rapport à $\overline{\mathbf{B}}$ est équivalente à la minimisation par rapport à $\underline{\mathbf{D}}$, et la solution est plus facile à obtenir. Ainsi, le problème devient :

$$\overline{\mathbf{B}}^{\text{opt}} = \arg \min_{\overline{\mathbf{B}}} \lambda \cdot E\{f(\underline{\mathbf{s}} \cdot \overline{\mathbf{B}})\} - \log |\det \overline{\mathbf{B}}|, \quad (3.21)$$

La dérivation du critère du problème (3.21) par rapport à l'inverse du dictionnaire est la suivante :

$$\nabla_{\overline{\mathbf{B}}} \mathcal{L}(\overline{\mathbf{B}}) = \lambda \cdot E\{\nabla_{\overline{\mathbf{B}}} f(\underline{\mathbf{s}} \cdot \overline{\mathbf{B}})\} - \left(\overline{\mathbf{B}}^T\right)^{-1}. \quad (3.22)$$

Si nous supposons que les coefficients sont mutuellement indépendants, alors $f(\mathbf{c}) = \sum_{k=1}^K f_k(c_k)$, et :

$$\nabla_{\overline{\mathbf{B}}} f(\underline{\mathbf{s}} \cdot \overline{\mathbf{B}}) = \underline{\mathbf{s}}^T \cdot \mathbf{g}^T(\underline{\mathbf{s}} \cdot \overline{\mathbf{B}}), \quad (3.23)$$

où $\mathbf{g}^T(\mathbf{c}) = [g_1(c_1) \cdots g_K(c_K)]$, avec $g_k = f'_k$ qui est la dérivée de f_k . Il faut noter qu'en général, les coefficients sont également supposés identiquement distribués, ce qui signifie que toutes les fonctions f_k sont identiques, ainsi que les fonctions g_k .

L'hypothèse d'indépendance entre les coefficients fait un lien direct avec l'ACI. Il faut noter que dans l'immense majorité des méthodes d'apprentissage de dictionnaire, ou de décomposition parcimonieuse, les coefficients sont supposés indépendants. Par exemple, l'hypothèse de coefficients laplacien et indépendants explique de façon probabiliste le choix de la mesure de diversité $\|\mathbf{c}\|_1$.

3.4.4 Algorithme de Bell-Sejnowski

Plusieurs algorithmes découlent de ce formalisme. L'algorithme proposé par Bell et Sejnowski [BS95] est un algorithme du gradient, où la mise à jour de l'inverse du dictionnaire $\overline{\mathbf{B}}_i$ est :

$$\overline{\mathbf{B}}_{i+1} = \overline{\mathbf{B}}_i + \mu \cdot (\lambda \cdot E\{\underline{\mathbf{s}}^T \cdot \mathbf{g}^T(\underline{\mathbf{s}} \cdot \overline{\mathbf{B}}_i)\} - \left(\overline{\mathbf{B}}_i^T\right)^{-1}), \quad (3.24)$$

avec μ le pas de l'algorithme.

La genèse de cet algorithme est expliquée dans [BS95] non par maximum de vraisemblance, mais par le principe d'infomax.

3.4.5 Algorithme de Hyvärinen

L'utilisation du gradient naturel permet d'améliorer la vitesse de convergence et le conditionnement de la vraisemblance. Le gradient naturel, défini par Amari [Ama98], s'écrit dans le cas d'un gradient par rapport à une matrice comme :

$$\nabla_{\overline{\mathbf{B}}}^{\text{nat}} \mathcal{L}(\overline{\mathbf{B}}) = \overline{\mathbf{B}} \cdot \overline{\mathbf{B}}^T \cdot \nabla_{\overline{\mathbf{B}}} \mathcal{L}(\overline{\mathbf{B}}). \quad (3.25)$$

Il permet de conserver le point d'équilibre d'un critère, tout en améliorant la stabilité autour de ce point. Ainsi, l'algorithme du gradient naturel, dérivé de l'algorithme décrit en (3.24), s'écrit :

$$\bar{\mathbf{B}}_{i+1} = \bar{\mathbf{B}}_i + \mu \cdot \bar{\mathbf{B}}_i \cdot (\lambda \cdot E\{\mathbf{c}_i \cdot \mathbf{g}^T(\mathbf{c}_i)\} - \mathbf{I}) , \quad (3.26)$$

où $\mathbf{c}_i^T = \underline{s} \cdot \bar{\mathbf{B}}_i$. La phase d'inversion de $\bar{\mathbf{B}}_i^T$ présente dans (3.24) a ainsi disparu.

En blanchissant au préalable les signaux d'apprentissage, cet algorithme devient FastICA [HO97], un algorithme classique d'ICA. Hyvärinen conseille d'effectuer le blanchiment pour améliorer la rapidité et le conditionnement de la résolution.

3.4.6 Choix de la distribution des coefficients

Le choix de la fonction f peut être fait par rapport à la mesure de diversité, comme expliqué précédemment ($f_1(t) = |t|$), mais aussi, comme dans le domaine des réseaux de neurones, parmi un ensemble de fonctions de non-linéarité classiques ($f_2(t) = -\arctan(t)$, $f_3(t) = -\tanh(t)/2$ et $f_4(t) = -\frac{1}{1+e^{-t}}$). Le choix de la fonction, correspondant au choix de l'*a priori* sur les coefficients, est l'objet de recherches dans les domaines de l'apprentissage de dictionnaires, de l'ACI et des réseaux de neurones, et repose essentiellement sur l'expertise.

3.5 Apprentissage de dictionnaire redondant pour l'approximation parcimonieuse

Comme expliqué en section 1.3.2, l'intérêt des décompositions parcimonieuses réside particulièrement dans l'utilisation de dictionnaires redondants, qui par leur plus grande diversité d'atomes permettent de mieux correspondre à l'esprit de décomposition d'un signal comme somme de quelques atomes.

Les études dédiées à l'apprentissage de dictionnaire redondant considèrent une décomposition approchée, et non pas exacte, du signal sur le dictionnaire. Le modèle est le suivant :

$$\underline{s} = \mathbf{c}^T \cdot \underline{\mathbf{D}} + \underline{r} . \quad (3.27)$$

Il faut noter que les algorithmes proposés dans cette section permettent également d'apprendre une base non-redondante, dans le cas bruité.

Contrairement au cas précédent d'une base, et d'une erreur de reconstruction nulle, les coefficients \mathbf{c} ne peuvent pas être remplacés dans les critères par $(\underline{s} \cdot \underline{\mathbf{D}}^{-1})^T$, ou toute autre expression analytique. En effet, comme nous l'avons expliqué au chapitre 1, il faut recourir à une méthode spécifique pour déterminer les coefficients en raison de la redondance et de l'erreur de reconstruction.

3.5.1 Interprétation probabiliste : maximum de vraisemblance

Le principe de toutes les méthodes d'apprentissage de dictionnaire [OF96b, OF97, OF96a, LO99, LS00, LLGS99, KDMR⁺03, EAH, ERKD99, Gir01, OM00] est de trouver le dictionnaire qui maximise la log-vraisemblance \mathcal{L} des données \underline{s} connaissant le dictionnaire $\underline{\mathbf{D}}$:

$$\mathcal{L} = E\{\log p(\underline{s}|\underline{\mathbf{D}})\} , \quad (3.28)$$

où $E\{\cdot\}$ correspond à l'espérance sur les données, et par abus de notation, à la moyenne sur les Q signaux \underline{s}_q . Par principe d'ergodicité, en effet, l'espérance peut être d'autant mieux approchée par la moyenne sur Q trames que Q est élevé et que les trames couvrent une grande partie de l'espace des signaux.

Deux types de méthodes sont communément utilisées pour maximiser la vraisemblance : les méthodes utilisant le gradient par rapport au dictionnaire :

$$\nabla_{\underline{\mathbf{D}}} \mathcal{L} = \frac{\partial E\{\log p(\underline{s}|\underline{\mathbf{D}})\}}{\partial \underline{\mathbf{D}}} , \quad (3.29)$$

qui sont exposées dans les sections suivantes, et l'algorithme "Expectation-Maximization" (EM [DLD77]) exposé dans la section 3.5.5.

Pour les méthodes utilisant le gradient, le dictionnaire est soit obtenu directement par annulation du gradient :

$$\nabla_{\underline{\mathbf{D}}_{\text{opt}}} \mathcal{L} = 0 \quad (3.30)$$

soit par descente de gradient à partir d'un dictionnaire initial, avec μ le pas d'apprentissage :

$$\underline{\mathbf{D}} := \underline{\mathbf{D}} + \mu \cdot \nabla_{\underline{\mathbf{D}}} \mathcal{L} . \quad (3.31)$$

Toute la difficulté réside dans le calcul du gradient $\nabla_{\underline{\mathbf{D}}} \mathcal{L}$, qui peut s'écrire (calculs en annexe A.2) comme :

$$\nabla_{\underline{\mathbf{D}}} \mathcal{L} = E\left\{E\left\{\frac{\partial \log p(\underline{s}|\underline{\mathbf{D}}, \mathbf{c})}{\partial \underline{\mathbf{D}}}\right\}_{p(\mathbf{c}|\underline{s}, \underline{\mathbf{D}})}\right\} . \quad (3.32)$$

Il faut en effet être en mesure de calculer l'espérance de $\frac{\partial \log p(\underline{s}|\underline{\mathbf{D}}, \mathbf{c})}{\partial \underline{\mathbf{D}}}$ sous la distribution *a posteriori* des coefficients $p(\mathbf{c}|\underline{s}, \underline{\mathbf{D}})$, qui ne permet pas de calculs analytiques pour la distribution *a priori* des coefficients $p(\mathbf{c})$ classique (loi laplacienne). Il faut donc approcher la loi *a posteriori* d'une certaine manière pour pouvoir calculer le gradient.

La loi $p(\underline{s}|\underline{\mathbf{D}}, \mathbf{c})$ correspond quant à elle à la distribution du bruit \underline{r} , qui est de manière classique supposée gaussienne i.i.d. de moyenne nulle et variance $\sigma_{\underline{r}}^2$ connue. Par conséquent, la dérivée de $\log p(\underline{s}|\underline{\mathbf{D}}, \mathbf{c})$ peut se calculer et vaut :

$$\frac{\partial \log p(\underline{s}|\underline{\mathbf{D}}, \mathbf{c})}{\partial \underline{\mathbf{D}}} = \frac{1}{2 \cdot \sigma_{\underline{r}}^2} (\underline{s} - \mathbf{c}^T \cdot \underline{\mathbf{D}})^T \cdot \mathbf{c}^T \quad (3.33)$$

Nous exposons dans les sections suivantes les méthodes correspondant à différentes modélisations de la loi *a posteriori* :

1. section 3.5.2 : la loi $p(\mathbf{c}|\underline{s}, \mathbf{D})$ est $\delta_{\mathbf{c}=\hat{\mathbf{c}}}$, ce qui correspond à remplacer les coefficients \mathbf{c} , dans l'intégrale, par les coefficients $\hat{\mathbf{c}}$ calculés explicitement. Il n'y a alors plus de degré de liberté sur les coefficients.
2. section 3.5.3 : la loi $p(\mathbf{c}|\underline{s}, \mathbf{D})$ est une loi gaussienne centrée autour des coefficients estimés $\hat{\mathbf{c}}$. Les coefficients sont donc fortement influencés par $\hat{\mathbf{c}}$ mais conservent une certaine liberté autour de ce point, dans la limite de la variance autorisée.
3. section 3.5.4 : la loi $p(\mathbf{c}|\underline{s}, \mathbf{D})$ peut être calculée analytiquement à partir de la loi *a priori* $p(\mathbf{c})$. L'intégrale est estimée par une méthode de Monte-Carlo, qui consiste à tirer un grand nombre de jeux de coefficients \mathbf{c}_n suivant la loi $p(\mathbf{c}|\underline{s}, \mathbf{D})$ et à approcher l'intégrale par la moyenne sur tous ces tirages.

3.5.2 Mise à jour alternée des coefficients et du dictionnaire - calcul explicite des coefficients

Dans les premiers travaux de Olshausen et Field [OF96b, OF97, OF96a], et pour les algorithmes du même type [KDMR⁺03, EAH, ERKD99, AEB05, AEB06], la distribution $p(\mathbf{c})$ est approchée par $\delta_{\mathbf{c}=\hat{\mathbf{c}}}$, où $\hat{\mathbf{c}}$ sont les coefficients estimés avant la mise à jour de \mathbf{D} , par un algorithme de décomposition parcimonieuse. Cela signifie que les coefficients sont fixés lors du calcul du dictionnaire. Tous les algorithmes de cette section sont basés sur le principe suivant :

Principe des algorithmes d'apprentissage :

Initialisation du dictionnaire \mathbf{D}_0

Faire

1. Pour chaque signal d'apprentissage \underline{s}_q , calcul des coefficients $\hat{\mathbf{c}}_q$ sur le dictionnaire \mathbf{D}_i par un algorithme de décomposition parcimonieuse
2. Mise à jour du dictionnaire \mathbf{D}_{i+1} à partir des signaux \underline{s}_q et des coefficients $\hat{\mathbf{c}}_q$ fixés :

$$\mathbf{D}_{i+1} = \arg \min_{\mathbf{D}} E\{\|\underline{s} - \hat{\mathbf{c}}^T \cdot \mathbf{D}\|_2^2\} . \quad (3.34)$$

Tant que la convergence n'est pas atteinte

Pour l'étape 1 de calcul des coefficients, Olshausen et Field [OF96b, OF97, OF96a] utilisent un algorithme de descente de gradient. Kreutz-Delgado *et coll.* [KDMR⁺03], ainsi que Engan *et coll.* [EAH, ERKD99] utilisent l'algorithme FOCUSS. Enfin, Aharon et Elad [AEB05, AEB06] laissent le choix de l'algorithme à leur lecteur.

Pour l'étape 2 de mise à jour du dictionnaire, remarquons qu'il s'agit uniquement de minimiser l'erreur quadratique, et qu'il n'existe pas de terme de parcimonie. La parcimonie est présente dans le critère par le biais des coefficients précédemment calculés

$\hat{\mathbf{c}}$. De même, la normalisation du dictionnaire n'est pas incluse dans le critère, et elle est donc effectuée après mise à jour.

Le gradient de \mathcal{L} s'écrit :

$$\nabla_{\underline{\mathbf{D}}} \mathcal{L} = E\left\{\frac{1}{2 \cdot \sigma^2} \hat{\mathbf{c}} \cdot (\underline{\mathbf{s}} - \hat{\mathbf{c}}^T \cdot \underline{\mathbf{D}})\right\} . \quad (3.35)$$

La mise à jour du dictionnaire, dans les algorithmes proposés par Olshausen et Field [OF96b, OF97, OF96a], repose sur une descente de gradient :

$$\underline{\mathbf{D}}_{i+1} = \underline{\mathbf{D}}_i + \mu \cdot E\{\hat{\mathbf{c}} \cdot (\underline{\mathbf{s}} - \hat{\mathbf{c}}^T \cdot \underline{\mathbf{D}})\} , \quad (3.36)$$

et chaque atome du dictionnaire est ensuite multiplié par un facteur permettant à tous les coefficients c_k d'être de variance unité. Kreutz-Delgado *et coll.* [KDMR⁺03] proposent la même mise à jour du dictionnaire, mais la normalisation s'effectue soit en multipliant $\underline{\mathbf{D}}$ pour que sa plus grande valeur propre soit égale à 1 – l'algorithme s'appelle FOCUSS-FDL pour Fröbenius-norm Dictionary Learning –, soit en normalisant chaque atome – l'algorithme s'appelle FOCUSS-CNDL pour Column Normalized Dictionary Learning.

Les algorithmes proposés par Engan [EAH, ERKD99] reposent quant à eux sur l'annulation du gradient :

$$\underline{\mathbf{D}}_{i+1} = E\{\mathbf{c} \cdot \mathbf{c}^T\}^{-1} \cdot E\{\mathbf{c} \cdot \underline{\mathbf{s}}\} , \quad (3.37)$$

et les atomes sont normalisés un à un.

L'algorithme K-SVD [AEB05, AEB06], élaboré par Aharon et Elad, suit une logique un peu différente des algorithmes précédents pour la mise à jour du dictionnaire. En effet, le dictionnaire n'est pas mis à jour de façon globale, mais au contraire atome par atome.

L'algorithme K-SVD est appelé K-SVD en référence à l'algorithme de classification K-means et à la SVD (Singular Value Decomposition ou décomposition en valeurs singulières). Il reprend le principe de mise à jour alternée de parties du dictionnaire que nous avons proposé dans [LGBB05], et que nous exposons au chapitre 6. Il nous a en retour servi comme principe de départ pour l'algorithme que nous présentons au chapitre 9.

La mise à jour du dictionnaire est la suivante :

Mise à jour des atomes par K-SVD :

Pour tout atome \underline{d}_k , $k \in \llbracket 1, K \rrbracket$, du dictionnaire $\underline{\mathbf{D}}$

Faire

- a) Retrait de la contribution de tous les autres atomes dans les signaux :

$$\check{\underline{s}}_q \leftarrow \underline{s}_q - \sum_{k' \neq k} \hat{c}_{q,k'} \cdot \underline{d}_{k'} = \hat{c}_{q,k} \cdot \underline{d}_k + \underline{r}_q, \forall q \in \llbracket 1, Q \rrbracket \quad (3.38)$$

- b) Mise à jour de l'atome \underline{d}_k et des coefficients $\hat{c}_{q,k}$, $q \in \llbracket 1, Q \rrbracket$ en minimisant l'énergie de l'erreur résiduelle

$$\left(\underline{d}_k, \{\hat{c}_{q,k}\}_{q \in \llbracket 1, Q \rrbracket} \right) \leftarrow \arg \min_{\|\underline{d}\|_2=1, \{\hat{c}_{q,k}\}_{q \in \llbracket 1, Q \rrbracket}} \sum_{q=1}^Q (\|\underline{\check{s}}_q - \hat{c}_{q,k} \cdot \underline{d}_k\|_2^2) , \quad (3.39)$$

ce qui correspond à :

$$\underline{d}_k = \arg \max_{\|\underline{d}\|_2=1} \sum_{q=1}^Q |\langle \underline{s}_q, \underline{d} \rangle|^2 \quad (3.40)$$

$$= \arg \max_{\|\underline{d}\|_2=1} \underline{d} \cdot \left(\sum_{q=1}^Q \underline{s}_q^T \cdot \underline{s}_q \right) \cdot \underline{d}^T , \quad (3.41)$$

dont la solution \underline{d}_k est le vecteur propre de la matrice $\sum_{q=1}^Q \underline{s}_q^T \cdot \underline{s}_q$ associé à la plus grande valeur propre. Le coefficient $\hat{c}_{q,k}$ est alors mis à jour par :

$$\hat{c}_{q,k} = \langle \underline{s}_q, \underline{d}_k \rangle . \quad (3.42)$$

- c) Mise à jour du résidu :

$$\underline{r}_q \leftarrow \underline{\check{s}}_q - \hat{c}_{q,k} \cdot \underline{d}_k \quad (3.43)$$

La preuve de la solution optimale de (3.41) est disponible dans l'annexe A.3.

Les atomes sont contraints à être de norme unité. Cette normalisation est obtenue lors de la mise à jour des atomes, en choisissant le vecteur propre de norme unité.

Notons que dans l'algorithme K-SVD, les coefficients $\hat{\mathbf{c}}$ sont mis à jour en même temps que leurs atomes associés, afin de minimiser l'erreur. Cela peut diminuer leur parcimonie, car ils ne sont pas contraints, mais le nombre d'atomes actifs, *i.e.* la parcimonie stricte, n'est pas modifié.

Par rapport à une mise à jour globale de la matrice dictionnaire $\underline{\mathbf{D}}$, la mise à jour individuelle des atomes présente l'avantage d'une évolution douce du dictionnaire au cours des itérations d'apprentissage, car la mise à jour individuelle des atomes empêche toute opération matricielle modifiant l'ensemble des atomes. Ainsi, d'une itération à l'autre, seuls les atomes qui n'ont pas encore convergé sont mis à jour, sans risquer de modifier tout le dictionnaire.

La gestion individuelle des atomes présente un autre avantage qui sera exposé longuement dans la deuxième partie de la thèse. En contraignant le modèle des atomes, cette gestion individuelle autorise en effet l'utilisation d'atomes de longueur différente, et même de nature différente.

3.5.3 Algorithme de Lewicki

Lewicki *et coll.* [LO99, LS00, LLGS99] proposent de calculer les coefficients avant la mise à jour du dictionnaire, de la même manière que dans la section précédente. La distribution *a posteriori* des coefficients est en revanche différente. Au lieu d'être modélisée par un Dirac situé à la valeur $\hat{\mathbf{c}}$, elle est modélisée par une loi gaussienne centrée en $\hat{\mathbf{c}}$ dont la variance est calculée *a posteriori*.

Le problème peut s'exprimer comme la minimisation d'un critère :

$$(\underline{\mathbf{D}}^{\text{opt}}, \mathbf{c}^{\text{opt}}) = \arg \min_{\underline{\mathbf{D}}, \mathbf{c}} E\{\|\underline{\mathbf{s}} - \mathbf{c}^T \cdot \underline{\mathbf{D}}\|_2^2 + \lambda \cdot f(\mathbf{c}) + \log |\det \mathbf{H}|\} , \quad (3.44)$$

où \mathbf{H} est un terme comprenant le Hessien de $p(\mathbf{c})$, pris au point $\hat{\mathbf{c}}$, la log-distribution des coefficients \mathbf{c} , prise au point $\hat{\mathbf{c}}$, et un terme de poids du dictionnaire proportionnel à $\underline{\mathbf{D}} \cdot \underline{\mathbf{D}}^T$.

Le troisième terme du critère contrôle donc la norme du dictionnaire et l'empêche de croître indéfiniment, tandis que comme précédemment, le premier terme du critère mesure l'erreur de reconstruction et le deuxième terme du critère mesure la diversité des coefficients. Sans exposer les détails, l'algorithme repose sur une descente de gradient naturel du critère (3.44). Le gradient naturel est utilisé afin de promouvoir la stabilité autour du point d'équilibre.

3.5.4 Méthodes de Monte-Carlo

Pour certains choix de la distribution *a priori* $p(\mathbf{c})$, il est possible de générer aléatoirement des coefficients suivant approximativement la distribution *a posteriori* $p(\mathbf{c}|\underline{\mathbf{s}}, \underline{\mathbf{D}})$. En générant suffisamment de coefficients \mathbf{c}^n et en s'assurant que ces coefficients explorent tout l'espace, l'intégrale sur les coefficients peut être approchée par une somme sur tous les tirages :

$$E\left\{\frac{\partial \log p(\underline{\mathbf{s}}|\underline{\mathbf{D}}, \mathbf{c})}{\partial \underline{\mathbf{D}}}\right\}_{p(\mathbf{c}|\underline{\mathbf{s}}, \underline{\mathbf{D}})} = \int \frac{\partial \log p(\underline{\mathbf{s}}|\underline{\mathbf{D}}, \mathbf{c})}{\partial \underline{\mathbf{D}}} \cdot p(\mathbf{c}|\underline{\mathbf{s}}, \underline{\mathbf{D}}) d\mathbf{c} \quad (3.45)$$

$$\approx \frac{1}{N} \cdot \sum_{n=1}^N \frac{\partial \log p(\underline{\mathbf{s}}|\underline{\mathbf{D}}, \mathbf{c}^n)}{\partial \underline{\mathbf{D}}} \quad (3.46)$$

Cette technique est une méthode de Monte-Carlo [CR04]. Elle peut être implémentée par l'algorithme Metropolis-Hasting ou un cas particulier, l'échantillonnage de Gibbs.

Olshausen et Millman proposent dans [OM00] de modéliser les coefficients *a priori* par un mélange de gaussiennes :

$$p(\mathbf{c}|\mathbf{q}) = \prod_{k=1}^K (q_k \cdot \mathcal{N}_{c_k}(0, \sigma_{1,k}^2) + (1 - q_k) \cdot \mathcal{N}_{c_k}(0, \sigma_{2,k}^2)) , \quad (3.47)$$

où q_k est un indicateur valant 0 ou 1 qui permet de choisir entre les deux distributions. Il donne l'état du modèle de mélange de gaussiennes pour le coefficient c_k . La probabilité

$p(q_k = 1)$ est la probabilité que le coefficient c_k soit généré par la gaussienne $\mathcal{N}_{c_k}(0, \sigma_{1,k}^2)$, et $p(q_k = 0)$ est la probabilité qu'il soit généré par $\mathcal{N}_{c_k}(0, \sigma_{2,k}^2)$.

Ce mélange de gaussiennes reflète la nature des coefficients parcimonieux, qui sont constitués de beaucoup de coefficients proches de zéro, et de quelques coefficients plus grands. Ainsi, la première gaussienne a une grande variance pour modéliser les coefficients de poids fort, et l'autre a une petite variance pour modéliser les coefficients de poids faible.

Blumensath et Davies [BD06], dans le cadre d'atomes invariants par translation que nous présenterons dans la deuxième partie de la thèse, proposent le modèle suivant, légèrement différent :

$$p(\mathbf{c}|\mathbf{q}) = \prod_{k=1}^K (q_k \cdot p_{nz}(c_k) + (1 - q_k) \cdot \delta_0(c_k)) , \quad (3.48)$$

où $p_{nz}(c_k)$, qui peut être une gaussienne, modélise la probabilité des coefficients non-nuls par exemple. Un coefficient c_k a donc la probabilité $p(q_k = 1)$ d'être généré par $p_{nz}(c_k)$, et la probabilité $p(q_k = 0)$ d'être nul.

Blumensath et Davies proposent de modéliser la distribution de l'indicateur q_k par la loi discrète :

$$p(q_k) = \mu_k^{q_k} + (1 - \mu_k)^{1-q_k} , \quad (3.49)$$

où μ_k est la moyenne de l'état q_k et elle est couramment appelée poids de la gaussienne $\mathcal{N}_{c_k}(0, \sigma_{1,k}^2)$. De façon symétrique, $(1 - \mu_k)$ est le poids de la deuxième gaussienne.

Le modèle contient un grand nombre de paramètres que nous notons $\boldsymbol{\theta} = [\theta_1 \dots \theta_P]^T$:

- le dictionnaire $\underline{\mathbf{D}}$,
- la variance du bruit $\sigma_{\underline{\mathbf{r}}}$ (nous avons supposé que la loi $p(\underline{\mathbf{s}}|\underline{\mathbf{D}}, \mathbf{c})$ est une loi gaussienne centrée),
- les $2 \cdot K$ variances $\sigma_{1,k}^2$ et $\sigma_{2,k}^2$, dans le cas du mélange de deux gaussiennes, de Olshausen et Millman (ce seraient les paramètres de la loi $p_{nz}(c_k)$, pour Blumensath et Davies) et
- les K paramètres μ_k des lois $p(q_k)$ des indicateurs.

Afin d'estimer les paramètres optimaux, et notamment le dictionnaire optimal, tous les paramètres θ_p sont initialisés, puis mis à jour successivement par descente de gradient jusqu'à convergence. L'étape de descente de gradient du paramètre θ_p est :

$$\theta_p := \theta_p + \mu \cdot \nabla_{\theta_p} \mathcal{L} . \quad (3.50)$$

Comme le calcul de chaque gradient $\nabla_{\theta_p} \mathcal{L}$ nécessite une intégration sur tous les coefficients, un échantillonnage de Gibbs est utilisé à chaque étape pour remplacer l'intégrale par une somme sur un ensemble de réalisations \mathbf{c}^n . Chaque réalisation \mathbf{c}^n des coefficients est obtenue de la manière suivante :

1. Pour chaque numéro de coefficient k , tirage d'une réalisation q_k^n de l'état q_k selon la loi $p(q_k)$,

2. Le jeu de coefficients \mathbf{c}^n est tiré selon la loi $p(\mathbf{c}|\mathbf{q}_k^n)$, qui est réduite, pour chaque coefficient c_k à la gaussienne correspondant à l'état q_k^n .

L'apprentissage de dictionnaire exposé ici, qui est obtenu en modélisant les coefficients par un mélange de distributions, s'avère plus efficace que les méthodes exposées dans les sections précédentes où les coefficients et le dictionnaire sont mis à jour alternativement [BD06]. En revanche, le temps de calcul est beaucoup plus élevé à cause de l'échantillonnage de Gibbs.

3.5.5 Algorithme EM

Au lieu d'utiliser une technique de calcul des paramètres par descente de gradient, il est possible théoriquement d'utiliser un algorithme EM pour estimer les paramètres, dont le dictionnaire, avec la modélisation en mélange de gaussiennes. La complexité algorithmique rend malheureusement son application impossible. Nous exposons dans la suite plusieurs techniques pour contourner cette difficulté.

L'algorithme EM alterne deux étapes. L'étape E ("expectation") consiste à calculer l'espérance de la vraisemblance \mathcal{L} sur les paramètres cachés, ici les coefficients \mathbf{c} et les états des gaussiennes \mathbf{q} , connaissant les données observées \underline{s}_q et les paramètres $\boldsymbol{\theta}^{i-1}$ à l'étape précédente. L'étape M ("maximization") consiste à maximiser la vraisemblance par rapport aux paramètres $\boldsymbol{\theta}$ pour trouver les nouveaux paramètres $\boldsymbol{\theta}^i$.

Théoriquement, l'algorithme EM peut être appliqué dans le cadre du mélange de gaussiennes [DM04], car les espérances et la maximisation des paramètres se calculent analytiquement. En revanche, l'étape E comprend le calcul de K espérances sur les coefficients c_k , et surtout de 2^K espérances sur les états. En effet, il existe 2^K états correspondant à toutes les combinaisons de K états binaires.

Ce nombre d'états est acceptable pour de petites dimensions (1024 états pour 10 atomes), mais impossible à traiter dès que les dimensions sont un peu plus grandes ($1.3 \cdot 10^{30}$ états pour 100 atomes). Il faut donc recourir à des méthodes permettant de réduire cette complexité.

Davies et Mitianoudis [DM04] proposent de tronquer le mélange de gaussiennes pour ne garder que les plus énergétiques. Attias [Att99], et Girolami [Gir01] proposent des méthodes variationnelles en approchant $p(\mathbf{c})$, dans le cas laplacien $\exp(-\|\mathbf{c}\|_1)$, non plus par un mélange de gaussiennes mais par :

$$\sup_{\boldsymbol{\xi}} \{C(\boldsymbol{\xi}) \cdot \mathcal{N}_{\mathbf{c}}(\mathbf{0}, \boldsymbol{\Delta}(\boldsymbol{\xi}))\} , \quad (3.51)$$

où $C(\boldsymbol{\xi})$ est un facteur de normalisation, $\boldsymbol{\xi}$ est un vecteur de K paramètres correspondant aux K coefficients de \mathbf{c} , et $\boldsymbol{\Delta}(\boldsymbol{\xi}) = \text{diag}(|\xi_1|, \dots, |\xi_K|)$ est la matrice de covariance de la distribution gaussienne.

En utilisant cette approximation, il est possible de calculer une borne inférieure de la vraisemblance grâce à la distribution gaussienne introduite pour modéliser les coefficients. En utilisant un algorithme Expectation-Maximization (EM), cette borne inférieure augmente au fur et à mesure des itérations, et par suite la vraisemblance également. L'étape E ne comprend dans ce cas que K espérances sur les coefficients c_k

qui sont les paramètres cachés. Les paramètres sont le dictionnaire $\underline{\mathbf{D}}$, la variance σ_r du bruit, et les paramètres $\underline{\boldsymbol{\xi}}_q$ pour chaque trame de signal $\underline{\mathbf{s}}_q$. Grâce à l'algorithme EM, le dictionnaire est mis à jour de manière itérative sans calcul explicite des coefficients \mathbf{c} . Comme pour les méthodes des sections précédentes, le dictionnaire est normé après chaque mise à jour.

3.6 Conclusion

Au cours de ce chapitre, nous avons expliqué l'intérêt d'apprendre un dictionnaire adapté aux signaux à décomposer. Nous avons ensuite présenté l'ACP et l'ACI comme des principes ayant fait émerger historiquement l'apprentissage de dictionnaire. Nous avons ensuite présenté divers algorithmes pour l'apprentissage de bases, puis de dictionnaires redondants.

Blumensath et Davies [BD06] ont présenté une étude comparative de quelques algorithmes, montrant que les algorithmes utilisant une méthode de Monte-Carlo offrent un meilleur apprentissage, au détriment de la rapidité de calcul, que les méthodes d'optimisation alternée sur les coefficients et le dictionnaire. Nous ne disposons pas d'autre étude comparative sur ces méthodes, et les travaux théoriques sur la convergence des algorithmes, ou sur l'unicité des solutions optimales des problèmes sont à leurs débuts.

Dans la deuxième partie, nous étudierons particulièrement l'algorithme K-SVD, et nous l'adapterons au modèle d'atomes structurés que nous proposons dans cette thèse. Nous avons notamment choisi cet algorithme parce qu'il est dans la lignée de travaux que nous avons mené précédemment, et parce qu'il peut s'adapter au modèle d'atomes structurés de façon simple.

Nous avons présenté dans les deux derniers chapitres comment calculer des décompositions parcimonieuses, et comment apprendre le dictionnaire, intéressons-nous maintenant aux applications mettant en œuvre les représentations parcimonieuses, et en particulier la séparation de sources. Nous présenterons le problème, délimiterons le cadre de notre étude, et présenterons les méthodes classiques permettant de séparer des sources à l'aide de la parcimonie.

Chapitre 4

Décompositions parcimonieuses pour la séparation de sources sous-déterminée

Le problème de la séparation de sources repose sur l'hypothèse qu'un signal, appelé mélange, a été généré par la somme de plusieurs signaux, appelés sources, et l'objectif de la séparation de sources est de retrouver ces sources à partir du mélange.

Ce problème a un lien très fort avec le principe de décomposition parcimonieuse. En effet, considérons l'exemple suivant. Un mélange constitué de la somme de deux sources distinctes peut être décomposé de façon très parcimonieuse : si les deux sources sont deux atomes du dictionnaire, alors la décomposition du mélange comme somme de ces deux atomes est la plus parcimonieuse possible. Dans ce cas, la séparation peut donc être effectuée simplement par une décomposition parcimonieuse du mélange.

Bien sûr, cet exemple n'est pas réaliste, car la séparation n'a de sens que si les sources sont inconnues et donc les sources ne peuvent pas se trouver dans le dictionnaire. En revanche, en supposant que chaque source a une décomposition parcimonieuse sur un dictionnaire distinct, alors la séparation de sources peut reposer sur une décomposition du mélange sur un dictionnaire, suivie d'un regroupement des atomes de même type en vue d'estimer les sources. Ce regroupement nécessite des connaissances *a priori* sur les sources attendues.

Nous détaillons les tenants et aboutissants de la séparation de sources dans la section 4.1, et dans la section 4.2, nous exposons les diverses méthodes de séparation.

4.1 Le problème de la séparation de sources

La séparation de sources est un problème classique en traitement de signal. Il vise à extraire d'un signal les différentes composantes qui l'ont généré. Prenons l'exemple d'un enregistrement musical. Le signal que nous appelons **mélange** est un enregistrement stéréophonique et les sources sont les différents instruments. La séparation de sources consiste à extraire les instruments à partir du mélange et de certaines connaissances sur

les sources.

4.1.1 Configuration du mélange

La séparation de sources regroupe des réalités très diverses et la difficulté est très variable selon les configurations.

4.1.1.1 Conditions favorables

Décrivons un premier cas facile à séparer. Supposons que le mélange est créé artificiellement par le mixage d'une voix et d'un instrument, de telle façon que le chanteur est placé au milieu (même contribution sur les canaux de gauche et de droite) et que l'instrument est placé plutôt à gauche (contribution supérieure sur le canal de gauche que sur le canal de droite). Dans ce cas, nous pouvons obtenir l'instrument, débarrassé de la voix, uniquement en faisant la différence entre les deux canaux. Cette technique est utilisée dans la plupart des logiciels de karaoké, car en général, la voix du chanteur est mixée au milieu alors que l'accompagnement se trouve autour.

Pour n'obtenir que la voix, il suffirait de faire la différence entre les deux canaux, après avoir appliqué un gain différent sur chacun des canaux, correspondant à la direction de l'instrument. Nous expliquons plus loin que cette méthode revient juste à une inversion de matrice. Elle repose sur une propriété importante : il y a autant de capteurs (microphones) que de sources.

4.1.1.2 Conditions difficiles

Décrivons maintenant un cas très difficile à séparer. Un orchestre symphonique joue dans une église et il est enregistré avec deux microphones. L'objectif est d'obtenir l'enregistrement de chaque instrument séparé. Plusieurs facteurs font que c'est impossible à réaliser.

Tout d'abord, le nombre d'instruments est bien supérieur au nombre de capteurs, ce qui rend le système fortement sous-déterminé.

Ensuite, les sons enregistrés par les microphones correspondent aux sons émis par les instruments puis modifiés selon l'acoustique de la salle. Dans le cas d'une église, par exemple, la résonance est importante, et une note reste audible près d'une dizaine de secondes après que l'instrumentiste l'ait jouée. Ainsi, le mélange ne peut pas être considéré comme une simple addition de sources et nécessite une modélisation convolutive, dans laquelle un filtre décrit le trajet entre chaque instrument et chaque microphone.

Une autre difficulté provient de la proximité spatiale entre certains instruments : deux instruments placés l'un à côté de l'autre pourront à peine être distingués, et il faut faire appel à d'autres informations que les informations de localisation pour les séparer.

Enfin, la dernière difficulté est due aux timbres proches ou identiques de différents instruments. En effet, il est utopique de penser séparer deux violons jouant la même mélodie, car leur différence de timbre ne peut pas être distinguée. Cette difficulté est augmentée dans le cas d'un orchestre par la volonté de former un son global harmonieux et non pas une somme de sonorités différentes.

4.1.1.3 Autres difficultés

Trois autres paramètres peuvent augmenter la difficulté de modélisation. Tout d'abord, le modèle peut être variable avec le temps, ce qui arrive par exemple quand les musiciens se déplacent sur scène alors que les microphones sont fixes.

Ensuite, si un seul microphone enregistre la scène, alors aucune notion de direction spatiale ne peut être utilisée, et seule la différence de timbre peut servir à séparer les instruments.

Enfin, l'ajout d'effets par post-production de l'enregistrement est très néfaste à la séparation, car les distorsions introduites sont en général non-linéaires et de nature inconnue. De même, la compression des fichiers (MP3, OggVorbis, ...) déforme le signal de façon inaudible mais dommageable pour la séparation.

4.1.1.4 Configuration de notre étude

Dans notre étude, la configuration se situe entre les deux exemples précédents. Les hypothèses de travail sont :

1. le nombre de sources est strictement supérieur au nombre de capteurs, mais faible (entre 2 et 5)
2. le mélange est linéaire instantané (pas de filtrage, pas de délais)
3. les paramètres de mixage sont connus
4. deux cas seront étudiés : un seul capteur (monocanal) ou au moins deux capteurs (multicanal)
5. le mélange est artificiel
6. le modèle est invariant dans le temps
7. pas d'effets, ni de compression

Détaillons ces différents points. Le point 1 place notre étude dans le domaine des problèmes sous-déterminés où les variables inconnues sont plus nombreuses que les variables connues. C'est un cas difficile, et la plupart des travaux initiaux de séparation de sources ont été fait dans le cas déterminé, en utilisant les algorithmes d'ACI notamment. Les premiers travaux de séparation dans le cas sous-déterminé datent de 1991, avec l'extension au cas sous-déterminé de l'ACI [Car91]. Nous supposons que le nombre de sources est assez faible, de l'ordre de 2 à 5 sources, parce qu'il est réaliste de penser que seuls quelques instruments jouent ensemble, mais aussi pour satisfaire des hypothèses qui sont données plus loin.

Le point 2 correspond au cas le plus simple, où aucun délai, ni aucun filtrage n'est modélisé entre les sources et les capteurs, et où seule une amplitude modélise l'ampleur de la contribution de chaque source sur chaque capteur. Les modèles de mélange avec délai ou filtrage nécessitent une mise en œuvre plus complexe. Nous proposons au chapitre 10 un algorithme pour traiter ces cas, mais dans l'état de l'art de ce chapitre et dans les expériences que nous exposons, nous ne traitons que le cas linéaire.

Le point 3 est une restriction importante du problème. Nous ne pouvons plus dès lors appeler le problème "séparation de sources", mais uniquement "estimation des sources",

puisque l'estimation des paramètres du mélange constitue la moitié du travail. Dans le cas déterminé, l'estimation des sources est directe lorsque les paramètres de mixage sont connus, comme nous le verrons par la suite. En revanche, dans le cas sous-déterminé, estimer les sources alors que les paramètres de mixage sont connus est un vrai challenge.

Dans le point 4 nous découpons le problème en deux sous-problèmes qui ne peuvent pas être attaqués avec les mêmes outils. En effet, dans le cas monocanal, la connaissance des paramètres de mixage est inutile, tandis que dans le cas multicanal, elle permet de discriminer les sources en fonction de leur position spatiale. Nous exposons un état de l'art plus fourni pour le cas multicanal, mais nous proposons et expérimentons une méthode de séparation pour les deux cas, monocanal et multicanal.

Les points 5, 6 et 7 permettent de manipuler des données qui suivent exactement le modèle et dont nous maîtrisons les paramètres. Pour des données venant d'un enregistrement réel, le modèle linéaire est trop restrictif à cause des résonances, de plus les paramètres de mixage seront modifiés au cours du temps par le mouvement des sources et même par le brassage de l'air dans la pièce. De plus, pour rendre compte de sources réelles, il faudrait les modéliser comme des sources étendues occupant une partie de l'espace, et non pas juste un point. Enfin, nous ne nous intéressons pas ici au problème spécifique de l'annulation des effets de post-production, qui peut s'avérer très difficile.

4.1.2 Modèle linéaire instantané

Sous ces conditions, le mélange mesuré par un capteur suit le modèle suivant :

$$\underline{x} = \sum_{n=1}^N a_n \cdot \underline{s}_n + \underline{r} , \quad (4.1)$$

où \underline{s}_n est la source n , a_n est le facteur correspondant à la contribution de la source \underline{s}_n au mélange \underline{x} , \underline{r} est le bruit qui inclut l'erreur de modélisation et un bruit additif éventuel (modélisation du bruit ambiant et de la mauvaise fidélité du capteur, par exemple). Il faut noter que dans le cas monocanal, le bruit peut être considéré comme une source supplémentaire.

Lorsque plusieurs capteurs enregistrent simultanément, le modèle linéaire multicanal s'écrit :

$$\underline{\mathbf{X}} = \mathbf{A} \cdot \underline{\mathbf{S}} + \underline{\mathbf{R}} , \quad (4.2)$$

où $\underline{\mathbf{S}}$ est le vecteur des N sources \underline{s}_n , $\underline{\mathbf{X}}$ est le vecteur des M capteurs \underline{x}_m et \mathbf{A} , appelée matrice de mélange, est la matrice de taille $M \times N$ composée des facteurs $a_{m,n}$. Enfin, $\underline{\mathbf{R}}$ est le vecteur de bruit, de taille M .

La formule (4.2) peut s'écrire comme :

$$\underline{\mathbf{X}} = \sum_{n=1}^N \mathbf{a}_n \cdot \underline{s}_n + \underline{\mathbf{R}} , \quad (4.3)$$

où \mathbf{a}_n est la n^e colonne de la matrice \mathbf{A} . Cette colonne donne la répartition de la source \underline{s}_n sur chacun des capteurs \underline{x}_m . Si les deux capteurs correspondent aux voies droite

et gauche, la colonne est assimilée à la **direction spatiale** de la source associée. Par exemple, $\mathbf{a}_n = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ représente une source située à gauche et $\mathbf{a}_n = \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}$ une source située au milieu.

4.1.3 Connaissances sur les sources

La difficulté de la séparation dépend beaucoup des connaissances disponibles *a priori* sur les sources. Il faut savoir ce qui permet de différencier les sources. Dans le cas multicanal (plusieurs microphones), savoir qu’il existe quatre instruments qui sont dans des directions spatiales différentes peut permettre de les retrouver. En revanche, sans plus d’information, il est impossible de savoir à quel instrument correspond chaque source. Pour pouvoir les nommer après séparation, il est nécessaire d’avoir un modèle *a priori* des sources.

Lorsqu’aucun modèle discriminant des sources n’est connu *a priori*, la séparation de sources est dite **aveugle** [VFG⁺03b], ce qui signifie que les sources ne peuvent être connues qu’à une permutation près. Notons que si les paramètres de mixage sont inconnus, alors les sources ne peuvent être connues qu’à un facteur multiplicatif près, dans le cas linéaire ($\mathbf{a}_n \cdot \underline{s}_n = (\lambda \cdot \mathbf{a}_n) \cdot (\frac{1}{\lambda} \cdot \underline{s}_n)$), ou à un filtre près, dans le cas convolutif. En revanche, si les paramètres de mixage sont connus, cette incertitude est levée.

La définition du mot “aveugle” a été donnée auparavant par Cardoso [Car98], pour désigner le fait que les sources ne sont pas observées et qu’aucune information n’est disponible sur les paramètres de mélange.

Dans le cadre de notre étude, les sources peuvent être discriminées selon les méthodes suivantes. Dans le cas multicanal, si aucune information n’est disponible sur le mélange, l’ACI, qui a déjà été présentée en section 3.3 permet de discriminer les sources par leur indépendance mutuelle et permet notamment d’estimer les paramètres de mélange.

Les contributions des différents instruments peuvent également être discriminées grâce à la différence de position spatiale. Enfin, la différence de timbre, qui est modélisée par des distributions statistiques différentes selon les sources, permet également de séparer les sources, que ce soit dans le cas monocal ou multicanal.

4.2 Estimation des sources lorsque les modèles des sources sont connus

La séparation de sources suit généralement deux étapes. La première étape consiste à estimer les paramètres du modèle de mélange (nombre de sources, matrice de mélange). La deuxième étape consiste à inverser le système afin d’estimer les sources, à partir des signaux captés (mélange) et des paramètres du modèle précédemment estimés.

Dans le cas où le nombre de sources est égal au nombre de capteurs, la matrice de mélange \mathbf{A} est carrée et le système est inversible. Si l’erreur \mathbf{R} est nulle, alors les sources sont directement obtenues par $\underline{\mathbf{S}} = \mathbf{A}^{-1} \cdot \underline{\mathbf{X}}$. Tout l’effort de la séparation réside alors dans l’estimation de la matrice de mélange.

Dans le cas que nous allons traiter, à savoir plus de sources que de capteurs, le système n'est pas inversible, et une infinité de solutions sont possibles, même si la matrice de mélange est connue parfaitement. Pour choisir entre tous les jeux de sources possibles, nous devons faire des hypothèses sur la modélisation de ces sources.

La modélisation des sources et leur estimation, lorsque les paramètres du mélange sont connus, est le seul versant de la séparation de sources que nous traitons dans cette thèse. Nous citons cependant dans la section suivante les quelques méthodes d'estimation de la matrice de mélange.

4.2.1 Estimation des paramètres de mélange

Les méthodes d'estimation des paramètres de mélange, plus particulièrement de la matrice de mélange, sont réparties en deux grandes familles.

La première famille de méthodes, de type ACI, suppose les sources indépendantes statistiquement, et estime la matrice de mélange grâce aux statistiques d'ordre supérieur [Com98, JH91].

La deuxième famille de méthodes est basée sur l'hypothèse que les sources sont parcimonieuses (ou qu'une transformée ou une décomposition sur un dictionnaire rend ses coefficients parcimonieux) et estime les directions de la matrice de mélange par des algorithmes de clustering de type carte de Kohonen [TLP04] (ne retrouve pas les sources sous-représentées en nombre de points car l'algorithme ne tient pas compte de l'énergie), réseau bayésien [ZSC06], K-means [YR04] (lent et convergence très dépendante de l'initialisation) et classification utilisant les propriétés locales [AD05, AGB06]. La classification peut être précédée d'une réduction de dimensions [VSI⁺04].

Nous présentons maintenant quelques méthodes reposant sur la parcimonie des sources dans le plan temps-fréquence (section 4.2.2) ou sur un dictionnaire (section 4.2.3). Notons que d'autres méthodes, basées sur des modélisations statistiques des sources, ont été proposées pour l'estimation des sources [VR04, FG06].

4.2.2 Parcimonie dans le plan temps-fréquence

La représentation dans le plan temps-fréquence des signaux sonores est un outil fondamental pour l'analyse et le traitement de son. Une des propriétés de ce type de représentation, fondamentale pour la séparation de sources, est la parcimonie des sons dans le plan temps-fréquence.

4.2.2.1 Principe

La représentation temps-fréquence d'un signal sonore expose l'évolution des fréquences présentes dans le signal au cours du temps. Elle peut être calculée par TFCT, qui effectue une transformée de Fourier des trames successives du signal. Dans ce cas, la figure est constituée d'un ensemble de "cases temps-fréquence" paramétrées par un instant t et une fréquence f . Chaque case temps-fréquence est remplie d'une couleur d'autant plus claire que l'amplitude de la composante fréquentielle f de la trame à l'instant t est grande.

Pour un signal sonore, la plupart des cases temps-fréquence sont d'amplitude faible, car l'activité se situe principalement sur les basses fréquences et sur les motifs harmoniques qui évoluent selon la mélodie, pour la musique, ou la prosodie, pour la parole.

Cette parcimonie est illustrée sur la figure 4.1. Une gamme chromatique – une succession de notes décalées d'un demi-ton – jouée avec une contrebasse est représentée dans le plan temps-fréquence. La représentation temps-fréquence est calculée par TFCT et seules les basses fréquences sont représentées, pour plus de lisibilité.

Les points clairs, correspondant aux zones d'activité principales, occupent seulement une petite partie du plan, tandis que la plupart des cases temps-fréquence sont de couleur claire, correspondant à une faible amplitude. Cette lecture de la représentation temps-fréquence fait donc ressortir le caractère parcimonieux des coefficients obtenus par TFCT, avec beaucoup de coefficients faibles, et peu de coefficients élevés.

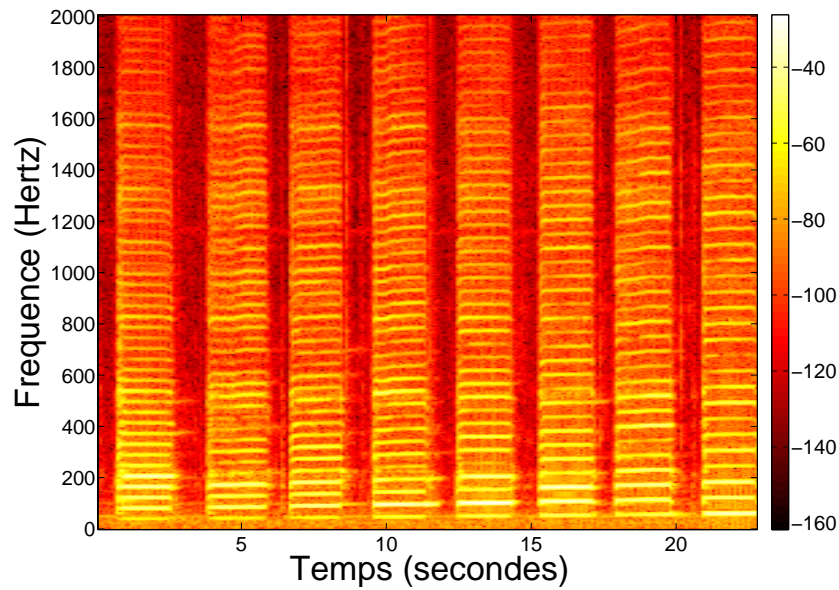


FIG. 4.1 – Représentation temps-fréquence (TFCT) d'une gamme chromatique de contrebasse.

L'intérêt de la parcimonie pour la séparation de sources est expliqué dans [YR04] et provient de la linéarité de la TFCT :

$$\tilde{\mathbf{X}} = \mathbf{A} \cdot \tilde{\mathbf{S}} + \tilde{\mathbf{R}}, \quad (4.4)$$

où $\tilde{\mathbf{X}}$, $\tilde{\mathbf{S}}$ et $\tilde{\mathbf{R}}$ sont respectivement le vecteur des TFCT des M capteurs, des N sources et des M résidus.

Étant donné que la plupart des cases temps-fréquence sont vides, et que le nombre de sources est faible, en supposant que les sources n'ont aucune raison de jouer les mêmes notes en même temps, alors la probabilité est grande pour que chaque case

temps-fréquence active soit due à une seule source. Lorsque deux sources ont la même fréquence active au même moment, nous pouvons supposer que c'est un cas isolé.

La parcimonie des sources dans le plan temps-fréquence implique donc que, **pour la plupart des cases temps-fréquence, seule une source est active**, *i.e.* seule une source a un coefficient non-nul. En conséquence, si cette hypothèse est stricte, en notant $\tilde{\mathbf{S}}^n = [\dots \mathbf{0}^T \tilde{\mathbf{g}}_n^T \mathbf{0}^T \dots]^T$ la restriction de $\tilde{\mathbf{S}}$ aux cases temps-fréquence où seule la source n est active, le mélange $\tilde{\mathbf{X}}$ peut être scindé en N sous-parties $\tilde{\mathbf{X}}^n = \mathbf{a}_n \cdot \tilde{\mathbf{g}}_n$ plus une contenant le bruit $\tilde{\mathbf{R}}$. Réussir à effectuer ce découpage de $\tilde{\mathbf{X}}$, avec attribution des cases temps-fréquence à l'une ou l'autre source, revient à effectuer la séparation de sources.

La figure 4.2 illustre graphiquement l'intérêt de la parcimonie pour la séparation de sources. Un mélange stéréo artificiel de quatre sources est créé, de façon à maximiser la distance angulaire entre les sources. Ainsi, en attribuant l'angle 0 au capteur droit, et l'angle $\pi/2$ au capteur gauche, les quatre sources sont situées aux angles 0, $\pi/6$, $\pi/3$ et $\pi/2$. Les sources sont respectivement une basse, une batterie, une guitare solo et une guitare rythmique, et elles jouent le même morceau de façon synchrone.

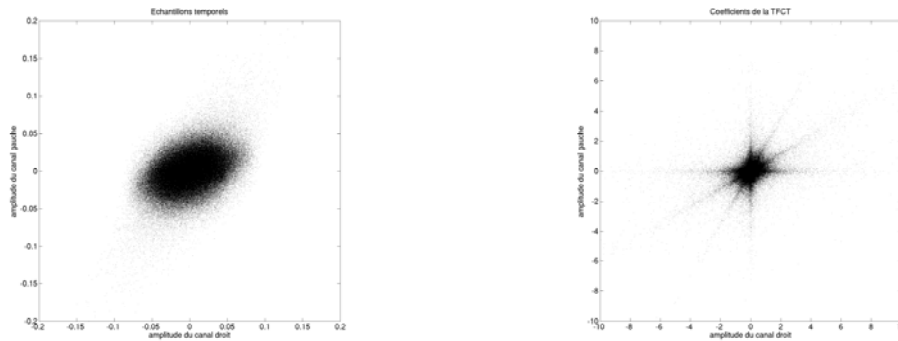


FIG. 4.2 – Amplitude du canal droit contre amplitude du canal gauche pour un mélange stéréo de quatre sources. Gauche : échantillons du signal temporel. Droite : coefficients de la TFCT.

La figure 4.2 représente, à gauche, l'amplitude du canal droit du mélange contre l'amplitude du canal gauche. Chaque point correspond à un instant l : $\mathbf{X}(l)$. La figure de gauche n'est d'aucune information pour déterminer les directions. En effet, à chaque instant, toutes les sources sont actives et leurs contributions se superposent.

Chaque canal est transformé par TFCT, afin d'obtenir les coefficients des cases temps-fréquence. La figure 4.2 représente, à droite, les coefficients temps-fréquence du canal de droite contre les coefficients temps-fréquence du canal de gauche. Chaque point correspond à une case temps-fréquence (t, f) : $\tilde{\mathbf{X}}(t, f)$. Il apparaît que pour les cases temps-fréquence les plus énergétiques, le point est aligné avec une des directions \mathbf{a}_n de la matrice \mathbf{A} . Cela signifie que dans cette case, seule la source \mathbf{g}_n est active.

Dans la réalité, l'hypothèse d'une seule source active n'est pas vérifiée strictement, car sinon tous les points sur la figure de droite seraient placés sur une des N droites

correspondant aux directions des sources \mathbf{a}_n .

4.2.2.2 Algorithme DUET

La méthode proposée par Yilmaz et Rickard [YR04], appelée DUET, est la suivante. Rappelons que, dans cette thèse, nous supposons que l'étape d'estimation de la matrice de mélange a déjà été effectuée. Les directions \mathbf{a}_n sont donc connues. Après avoir appliqué la TFCT au mélange \mathbf{X} , chaque point temps-fréquence $\tilde{\mathbf{X}}(t, f)$ est comparé aux N directions \mathbf{a}_n , est attribué à la source la plus proche, et le coefficient attribué à cette source correspond à la projection du point sur la direction. Ensuite, chaque source est reconstruite à partir de tous les points temps-fréquence qui lui ont été attribués, le reste du plan temps-fréquence étant à zéro. L'algorithme est résumé ci-dessous :

Description de l'algorithme DUET :

1. Transformée de Fourier à court-terme du mélange :

$$\tilde{\mathbf{X}} = \text{TFCT}(\mathbf{X}) \quad (4.5)$$

2. Initialisation des TFCT des sources estimées $\tilde{\hat{\mathbf{x}}}_n$ à zéro
3. **Pour** toutes les cases temps-fréquence (t, f)

Faire

- (a) Recherche de la direction la plus proche

$$n^{\text{opt}} = \arg \max_n |\langle \mathbf{a}_n, \tilde{\mathbf{X}}(t, f) \rangle| \quad (4.6)$$

- (b) Projection sur la source la plus proche

$$\tilde{\hat{\mathbf{x}}}_{n^{\text{opt}}}(t, f) = \langle \mathbf{a}_{n^{\text{opt}}}, \tilde{\mathbf{X}}(t, f) \rangle \quad (4.7)$$

4. Reconstruction des sources estimées dans le domaine temporel

$$\hat{\mathbf{x}}_n = \text{TFCT}^{-1}(\tilde{\hat{\mathbf{x}}}_n) \text{ pour tout } n \in \llbracket 1, N \rrbracket \quad (4.8)$$

4.2.2.3 Extensions de l'algorithme DUET

L'algorithme DUET sous-tend l'hypothèse d'une seule source active par case temps-fréquence. Les méthodes proposées par Aïssa El Bey *et coll.* [AEBLTAM⁺06] et Balan et Rosca [BR05] supposent que plusieurs sources peuvent être actives dans chaque case temps-fréquence. Ainsi, si nous supposons que $N_{act} < M$ sources sont actives, l'étape 3a

de l'algorithme DUET devient la recherche des N_{act} directions permettant la meilleure projection :

$$\beta^{\text{opt}} = \arg \min_{\beta} \|(\mathbf{I} - \mathbf{A}_{\beta} \cdot \mathbf{A}_{\beta}^{\dagger}) \cdot \tilde{\mathbf{X}}(t, f)\|_2^2, \quad (4.9)$$

où $\beta = (n_1, \dots, n_{N_{act}})$ est l'ensemble des indices des N_{act} sources actives, \mathbf{A}_{β} est la sous-matrice \mathbf{A} constituée des colonnes correspondant aux sources de β , et $\mathbf{A}_{\beta}^{\dagger} = (\mathbf{A}_{\beta}^T \cdot \mathbf{A}_{\beta})^{-1} \cdot \mathbf{A}_{\beta}^T$ sa pseudo-inverse. L'étape 3b de l'algorithme DUET devient alors la projection sur l'espace généré par les colonnes d'indices β^{opt} :

$$\text{pour tout } n \in \beta^{\text{opt}}, \tilde{\underline{\mathbf{x}}}_n(t, f) = (\mathbf{A}_{\beta}^{\dagger} \cdot \tilde{\mathbf{X}}(t, f))_n, \quad (4.10)$$

où $(\mathbf{A}_{\beta}^{\dagger} \cdot \tilde{\mathbf{X}}(t, f))_n$ désigne le n^e coefficient du vecteur $\mathbf{A}_{\beta}^{\dagger} \cdot \tilde{\mathbf{X}}(t, f)$.

L'hypothèse extrême consiste à supposer que M sources sont actives dans chaque case temps-fréquence [BZ01, TL02], et donc que chacune de ces sources récupère une partie du coefficient du mélange, en fonction de la proximité de sa direction au point temps-fréquence. Comme le nombre de sources actives est égal au nombre de capteurs, la projection n'induit aucune erreur. Le plus court chemin est utilisé, afin de choisir parmi tous les ensembles d'indices β possibles. Il correspond à la projection qui utilise les coefficients les moins lourds au sens de la norme l^1 . L'étape 3a de l'algorithme DUET est :

$$\beta^{\text{opt}} = \arg \min_{\beta} \|\mathbf{A}_{\beta}^{-1} \cdot \tilde{\mathbf{X}}(t, f)\|_1, \quad (4.11)$$

où $\beta = (n_1, \dots, n_{N_{act}})$, avec $N_{act} = M$, et l'étape 3b de l'algorithme DUET est :

$$\text{pour tout } n \in \beta^{\text{opt}}, \tilde{\underline{\mathbf{x}}}_n(t, f) = (\mathbf{A}_{\beta}^{-1} \cdot \tilde{\mathbf{X}}(t, f))_n. \quad (4.12)$$

4.2.3 Séparation de sources utilisant la parcimonie dans un dictionnaire

Le principe de parcimonie utilisé dans l'algorithme DUET a été étendu de la TFCT aux décompositions parcimonieuse sur un dictionnaire. Ainsi, dans un premier temps, nous présentons le principe de la séparation dans le cas multicanal puis un algorithme restreint au cas d'un mélange stéréo effectuant la séparation grâce à la décomposition parcimonieuse de chacun des capteurs sur un même dictionnaire.

Ensuite, nous présentons la méthode d'analyse en composantes morphologiques (ACM) qui, dans le cas monocal, permet d'effectuer la séparation en utilisant la seule différence de **morphologie** entre les sources.

4.2.3.1 Mélange multicanal

Dans le cas multicanal, si un dictionnaire $\underline{\mathbf{D}}$ permet de décomposer chacune des sources $\underline{\mathbf{s}}_n$ de façon parcimonieuse, alors par définition, chaque atome $\underline{\mathbf{d}}_k$ de $\underline{\mathbf{D}}$ est utilisé peu de fois :

$$\underline{\mathbf{s}}_n = \mathbf{c}_n^T \cdot \underline{\mathbf{D}} \text{ avec } \|\mathbf{c}_n\|_0 \text{ petit.} \quad (4.13)$$

Le mélange $\underline{\mathbf{X}} = \mathbf{A} \cdot \underline{\mathbf{S}}$ peut alors s'écrire comme :

$$\underline{\mathbf{X}} = \mathbf{A} \cdot \mathbf{C} \cdot \underline{\mathbf{D}}, \quad (4.14)$$

où $\mathbf{C} = [\mathbf{c}_1^T \dots \mathbf{c}_N^T]^T$ comprend les coefficients parcimonieux de toutes les sources. Cette description, proposée en 2001 par Zibulevsky et Pearlmutter [ZP01], conduit au problème de séparation suivant : *Connaissant le mélange $\underline{\mathbf{X}}$ et le dictionnaire $\underline{\mathbf{D}}$, trouver une matrice de mélange \mathbf{A} et des coefficients \mathbf{C} tels que $\underline{\mathbf{X}} \approx \mathbf{A} \cdot \mathbf{C} \cdot \underline{\mathbf{D}}$ et \mathbf{C} soit le plus parcimonieux possible.*

En effet, la parcimonie de \mathbf{C} permet de faire l'hypothèse que chaque atome de $\underline{\mathbf{D}}$ est utilisé par une seule source, *i.e.* chaque colonne de \mathbf{C} comprend un seul coefficient non nul. Sous cette hypothèse, la décomposition de $\underline{\mathbf{X}}$ sur le dictionnaire $\underline{\mathbf{D}}$ s'écrit $\underline{\mathbf{X}} = \mathbf{Z} \cdot \underline{\mathbf{D}}$, où \mathbf{Z} , vecteur de taille M , contient les coefficients parcimonieux, et la séparation consiste à décomposer \mathbf{Z} comme $\mathbf{A} \cdot \hat{\mathbf{C}}$. Nous montrerons dans la partie 2 que pour atteindre cet objectif, il suffit d'appliquer les mêmes principes de classification des directions que dans la section 4.2.2.

Gribonval a proposé en 2002 [Gri02] un algorithme basé sur une adaptation stéréo de l'algorithme Matching Pursuit permettant d'effectuer la décomposition d'un mélange stéréo d'au moins trois sources comme : $\underline{\mathbf{X}} = \mathbf{Z} \cdot \underline{\mathbf{D}}$. Nous réduisons ici l'algorithme au cas instantané, mais il a été initialement proposé pour les mélanges avec délai :

Algorithme Matching Pursuit Stéréo :

Initialisation du compteur $i := 0$ et du résidu $\underline{\mathbf{R}}^0 := \underline{\mathbf{X}}$

Faire

1. Mise à jour du compteur $i := i + 1$
2. Calcul de $\langle \underline{\mathbf{r}}_m^{i-1}, \underline{\mathbf{d}}_k \rangle$, pour les deux canaux $m = 1, 2$ et pour tous les atomes $k \in \llbracket 1, K \rrbracket$
3. Recherche de l'atome le plus corrélé conjointement sur les deux canaux :

$$k_i := \arg \max_k |\langle \underline{\mathbf{r}}_1^{i-1}, \underline{\mathbf{d}}_k \rangle|^2 + |\langle \underline{\mathbf{r}}_2^{i-1}, \underline{\mathbf{d}}_k \rangle|^2 \quad (4.15)$$

4. Calcul des nouveaux coefficients :

$$z_{i,1} = \alpha_i \cdot \cos \theta_i \quad (4.16)$$

$$z_{i,2} = \alpha_i \cdot \sin \theta_i, \quad (4.17)$$

avec

$$\alpha_i = \sqrt{|\langle \underline{\mathbf{r}}_1^{i-1}, \underline{\mathbf{d}}_{k_i} \rangle|^2 + |\langle \underline{\mathbf{r}}_2^{i-1}, \underline{\mathbf{d}}_{k_i} \rangle|^2} \quad (4.18)$$

$$\theta_i = \arctan \left(\frac{\langle \underline{\mathbf{r}}_2^{i-1}, \underline{\mathbf{d}}_{k_i} \rangle}{\langle \underline{\mathbf{r}}_1^{i-1}, \underline{\mathbf{d}}_{k_i} \rangle} \right) \quad (4.19)$$

5. Mise à jour du résidu :

$$\underline{\mathbf{R}}^i = \underline{\mathbf{R}}^{i-1} - \begin{bmatrix} z_{i,1} \cdot \underline{\mathbf{d}}_{k_i} \\ z_{i,2} \cdot \underline{\mathbf{d}}_{k_i} \end{bmatrix} \quad (4.20)$$

Tant que la convergence n'est pas atteinte

Étant donné que les directions se résument à un angle dans le cas stéréo, Gribonval propose ensuite de séparer les sources par une partition manuelle des angles θ , ce qui revient à attribuer tous les atomes stéréo situés dans une même portion à une seule source.

4.2.3.2 Mélange monocanal

Dans le cas d'un mélange monocanal (4.1), un raisonnement un peu différent, basé également sur les décompositions parcimonieuses, permet de répondre au problème de séparation de sources. La méthode, appelée Analyse en Composantes Morphologiques (ACM) [SED04], impose de connaître, pour chacune des sources $\underline{\mathbf{s}}_n$, un dictionnaire différent $\underline{\mathbf{D}}_n$ permettant de décomposer parcimonieusement la source $\underline{\mathbf{s}}_n$. Le mélange est alors théoriquement le suivant :

$$\underline{\mathbf{x}} = \sum_{n=1}^N \mathbf{z}_n^T \cdot \underline{\mathbf{D}}_n + \underline{\mathbf{r}}, \quad (4.21)$$

où $\underline{\mathbf{r}} = \sum_{n=1}^N \underline{\mathbf{r}}_n$. La séparation consiste à retrouver les coefficients \mathbf{c}_n à partir de la seule connaissance du mélange et des dictionnaires. Il est prouvé [SED04] que si les sources se décomposent de façon suffisamment parcimonieuse sur leur dictionnaire associé, et si ces dictionnaires sont suffisamment différents les uns des autres, alors des algorithmes classiques de décomposition parcimonieuse (Matching Pursuit, et algorithmes de type Basis Pursuit) permettent de résoudre le problème de séparation de source. L'algorithme est le suivant :

Description de l'analyse en composantes morphologiques :

1. Décomposition parcimonieuse du mélange sur l'union des dictionnaires $[\underline{\mathbf{D}}_1 \cdots \underline{\mathbf{D}}_N]$:

$$\underline{\mathbf{x}} = \sum_{n=1}^N \mathbf{z}_n^T \cdot \underline{\mathbf{D}}_n + \underline{\mathbf{r}} \quad (4.22)$$

Pour l'algorithme de décomposition parcimonieuse, voir le chapitre 2.

2. Reconstruction des sources estimées

$$\hat{\underline{\mathbf{z}}}_n = \mathbf{z}_n^T \cdot \underline{\mathbf{D}}_n \text{ pour tout } n \in \llbracket 1, N \rrbracket \quad (4.23)$$

Les deux approches présentées dans le cas multicanal et monocanal peuvent être réunies pour une séparation multicanale efficace, utilisant à la fois les informations de direction spatiale et de différence morphologique. Cette fusion des approches fait partie du travail de thèse et est présentée au chapitre 10.

4.3 Conclusion

Dans ce chapitre consacré au problème de la séparation de sources, nous avons présenté les facteurs de difficulté, et nous avons choisi de nous intéresser particulièrement au cas linéaire sous-déterminé où le nombre de sources est supérieur au nombre de capteurs. Nous avons exposé l'algorithme classique DUET qui se base sur la parcimonie des coefficients de la TFCT pour la séparation de sources audio, et qui utilise la différence de direction spatiale des sources. Nous avons ensuite présenté l'extension qui a été faite de ce principe à un dictionnaire quelconque. Nous avons fini en présentant l'ACM qui permet d'effectuer une séparation dans le cas monocanal, différenciant les sources par leur morphologie, à partir d'un dictionnaire adapté à chaque source.

Rappelons que nous nous sommes placés dans le cadre où les paramètres de mixage sont connus (ou ont déjà été estimés), et nous nous sommes intéressés uniquement à l'estimation des sources, connaissant le mélange et les paramètres de mixage.

Dans la suite de ce document, nous proposerons de réunir l'ACM et la séparation de type DUET sur un dictionnaire pour utiliser conjointement les informations spatiales et morphologiques sur les sources. Nous proposerons notamment d'utiliser des dictionnaires appris sur des exemples des sources pour l'ACM.

Ce chapitre conclut l'état de l'art concernant les représentations parcimonieuses. Nous établissons dans le chapitre suivant une analyse critique du modèle de dictionnaire utilisé dans les chapitres 1, 2, 4 et 3, et expliquons l'intérêt de structurer le modèle d'atome pour apprendre des dictionnaires mieux adaptés aux signaux, et de description plus économique.

Chapitre 5

Analyse critique du modèle de dictionnaire non-structuré et proposition

Le modèle $\underline{s} = \mathbf{c}^T \cdot \underline{\mathbf{D}}$ utilisé pour la séparation de sources et les décompositions parcimonieuses, a évolué au cours des années. Dans les premiers travaux, l'objectif était d'analyser un signal de taille L comme combinaison de $K = L$ causes indépendantes. L'ACP, puis l'ACI permettent d'apprendre une base, *i.e.* un dictionnaire $\underline{\mathbf{D}}$ carré, mais aussi une matrice de mélange. La redondance du dictionnaire a apporté de nouvelles perspectives et de nouvelles difficultés.

Nous discutons dans la section 5.1 de l'analogie historique de modélisation et de résolution entre la séparation de sources et l'apprentissage de dictionnaire. Nous mettons en lumière les différences fondamentales qui justifient de traiter ces deux problèmes séparément. Dans la section 5.2, nous pointons les insuffisances du modèle matriciel pour les décompositions parcimonieuses et introduisons le concept de motifs déformables.

5.1 Similarités et différences entre la séparation de sources et l'apprentissage de dictionnaire

Le modèle de séparation de sources et le modèle d'apprentissage de dictionnaire sont proches au niveau de la forme, mais diffèrent par leur dimension et leur nature. Malgré leurs différences, ils sont souvent confondus et traités de la même manière. Nous proposons quelques éclaircissements dans cette section.

5.1.1 Similarités

Les deux modèles sont de même forme matricielle, comme l'indique la figure 5.1. Nous pouvons ainsi faire une analogie respectivement entre :

- le mélange $\underline{\mathbf{X}}$ et les trames de signal $\underline{\mathbf{S}}^T$,
- la matrice de mélange \mathbf{A} et le dictionnaire $\underline{\mathbf{D}}^T$,

- les sources $\underline{\mathbf{S}}$ et les coefficients \mathbf{C}^T , et
- le bruit $\underline{\mathbf{R}}$ et le résidu $\underline{\mathbf{R}}^T$.

$$\begin{array}{c}
 \begin{array}{c} \boxed{\underline{\mathbf{X}}} \\ \leftarrow L \rightarrow \\ \updownarrow M \end{array} = \begin{array}{c} \boxed{\mathbf{A}} \\ \leftarrow N \rightarrow \\ \updownarrow M \end{array} \cdot \begin{array}{c} \boxed{\underline{\mathbf{S}}} \\ \leftarrow L \rightarrow \\ \updownarrow N \end{array} + \begin{array}{c} \boxed{\underline{\mathbf{R}}} \\ \leftarrow L \rightarrow \\ \updownarrow M \end{array} \\
 \\
 \begin{array}{c} \boxed{\underline{\mathbf{S}}^T} \\ \leftarrow Q \rightarrow \\ \updownarrow L \end{array} = \begin{array}{c} \boxed{\mathbf{D}^T} \\ \leftarrow K \rightarrow \\ \updownarrow L \end{array} \cdot \begin{array}{c} \boxed{\mathbf{C}^T} \\ \leftarrow Q \rightarrow \\ \updownarrow K \end{array} + \begin{array}{c} \boxed{\underline{\mathbf{R}}^T} \\ \leftarrow Q \rightarrow \\ \updownarrow L \end{array}
 \end{array}$$

FIG. 5.1 – Similarité morphologique entre le modèle de séparation de sources (haut) et le modèle d’apprentissage de dictionnaire (bas).

Dans le cas non-redondant, le nombre de capteurs M et de sources N sont égaux, et le nombre d’atomes K est égal à la taille des signaux L . Dans ces deux situations, la matrice carrée peut être calculée par ACI. Tous les premiers algorithmes classiques d’ACI [CS93, BS95, HO97] ont été développés à des fins de séparation de sources dans le cas déterminé.

La prédominance de l’ACI pour les deux tâches d’apprentissage de dictionnaire et de séparation de sources a certainement contribué à entretenir un niveau de confusion entre les deux problèmes.

La similarité vient également, dans le cas d’une approche parcimonieuse de la séparation de sources, de l’interprétation qu’on peut donner aux sources. Dans le cadre parcimonieux, le mélange est supposé avoir été transformé, par exemple par TFCT, afin de satisfaire l’hypothèse d’une seule source active par atome, ou par case temps-fréquence.

Le résultat est que chaque colonne de $\underline{\mathbf{X}}$ est théoriquement proportionnelle à une colonne de \mathbf{A} , et chaque colonne de $\underline{\mathbf{S}}$ ne contient qu’un coefficient non-nul. Nous pouvons alors interpréter la séparation comme la décomposition ultra-parcimonieuse, à un coefficient, du mélange $\underline{\mathbf{X}}$ sur le “dictionnaire” \mathbf{A} dont les atomes sont les directions $\underline{\mathbf{a}}_n$. Cette décomposition peut être assimilée à une quantification vectorielle [LBG80]. Le dictionnaire est alors un dictionnaire spatial, et le mélange transformé est parcimonieux sur les directions spatiales.

5.1.2 Différences

La différence entre les deux modèles de séparation de sources et d’apprentissage de dictionnaire vient surtout des dimensions et de la nature des objets manipulés.

En ordre de grandeur, le nombre de capteurs M et de sources N est généralement de 1 à 10, et la taille des signaux L peut varier de 100 à 10000 échantillons ou plus. Le nombre d’atomes K est en général de l’ordre de 1 à 10 fois la taille du signal L , et

pour un apprentissage correct, le nombre de trames Q doit être d'au moins 20 fois le nombre d'atomes K , comme nous l'avons montré dans l'article [LGBB05] concernant l'algorithme d'apprentissage de dictionnaire que nous présentons au chapitre 6. Le dictionnaire contient donc, selon les cas, entre 10^2 et 10^9 fois plus d'éléments que la matrice de mélange.

L'émergence de la redondance, pour la matrice de mélange ou pour le dictionnaire, et de l'erreur de reconstruction dans le modèle ont accentué la différence entre la résolution des deux problèmes. En effet, le traitement doit alors être non-linéaire pour être efficace, et contient une étape d'estimation des sources, ou d'estimation des coefficients, qui est lourde et constitue un nouveau problème sans solution analytique.

L'ACI classique n'est plus applicable, et les approches de résolution des deux problèmes sont différentes et s'appuient sur la nature et la dimension des objets.

Pour la séparation de sources, le modèle est de petite taille, et la matrice de mélange comporte des informations de direction spatiale qui dépendent du dispositif d'enregistrement (position des sources et des capteurs, et réverbérations dans le cas convolutif). La plupart des méthodes d'estimation de la matrice de mélange sont géométriques, et se basent sur la différence entre les angles d'arrivée des sources. Les algorithmes de clustering sont utilisés pour estimer les directions.

Pour l'apprentissage de dictionnaire, le modèle est de grande taille, et les atomes sont des constituants élémentaires du signal. Les méthodes d'apprentissage de dictionnaire visent généralement à maximiser la vraisemblance des signaux d'apprentissage connaissant les atomes qui les génèrent.

5.1.3 Complémentarité

Avec l'apparition de la redondance dans le modèle, la matrice de mélange \mathbf{A} et le dictionnaire \mathbf{D} sont devenus deux objets de nature et de taille différente, obtenus par des moyens différents. Ils sont cependant complémentaires, comme nous l'avons montré dans la section 4.2.3.1, où \mathbf{A} et \mathbf{D} apparaissent dans la modélisation du mélange multicanal :

$$\mathbf{X} = \mathbf{A} \cdot \mathbf{C} \cdot \mathbf{D} + \mathbf{R} . \quad (5.1)$$

Cette modélisation peut alors être vue comme une décomposition du mélange \mathbf{X} sur un **dictionnaire multicanal** qui inclut des informations spatiales, avec la matrice de mélange \mathbf{A} , et des informations de forme d'onde, avec le dictionnaire \mathbf{D} . Des méthodes d'estimation conjointe de la matrice de mélange et du dictionnaire peuvent être envisagées, et nous y reviendrons dans les perspectives, au chapitre 11. Nous rappelons que dans cette thèse, nous ne traitons pas le problème de l'estimation de la matrice de mélange.

5.2 Insuffisance de la modélisation

La première partie de cette thèse a montré comment des algorithmes de décomposition parcimonieuse et d'apprentissage de dictionnaire \mathbf{D} ont été dérivés du modèle

$\underline{s} = \mathbf{c}^T \cdot \mathbf{D}$. Ils permettent d'extraire les atomes qui sont les éléments générateurs des signaux étudiés.

5.2.1 Les atomes appris sont significatifs...

Pour les images, comme l'ont prouvé Olshausen et Field [OF96b, OF96a], les atomes appris sont analogues aux réponses des cellules du premier cortex visuel, qui traitent les images perçues par l'œil afin d'en réduire la complexité et d'en faciliter le traitement postérieur. Cette expérience a donc confirmé l'intérêt de la recherche de parcimonie pour représenter les signaux de façon efficace, en imitant la stratégie du cerveau. La figure 5.2 présente les atomes appris avec l'algorithme proposé par Olshausen [OF96b] sur des patches d'images naturelles, de taille 12x12. Chacun de ces atomes est localisé dans l'espace, orienté selon une direction principale, et a une bande de fréquences propre.

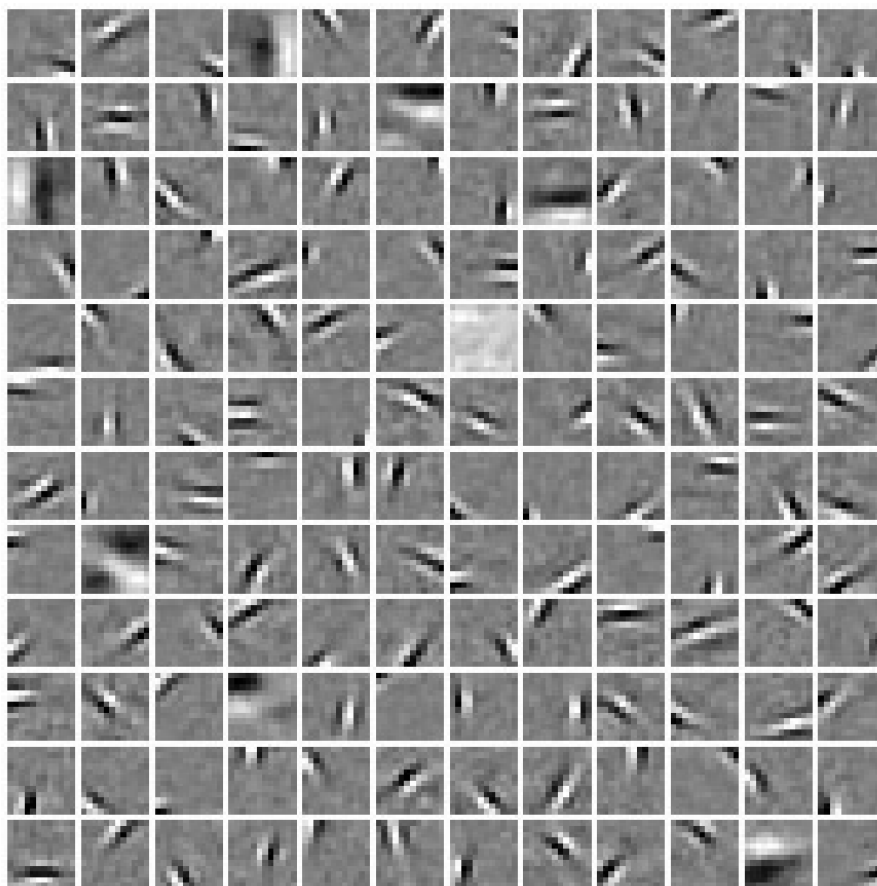


FIG. 5.2 – 144 atomes de taille 12x12, appris à partir d'images naturelles (figure tirée de [OF96b])

5.2.2 ... mais peuvent être factorisés

Si les atomes appris par Olshausen, représentés sur la figure 5.2, ont des propriétés spatiales intéressantes, on remarque cependant que plusieurs de ces atomes représentent le même objet, uniquement translaté dans l'espace. De même, le même objet est présent dans plusieurs atomes, avec des orientations différentes. Ces 144 atomes pourraient donc être résumés par :

- 5 objets de type atome de Gabor qui diffèrent par leur fréquence et leur taille,
- deux transformations – translation et rotation – qui changent ces objets en atomes.

Cette factorisation est également valable pour les signaux temporels. En effet, Abdallah et Plumbley [AP01] ont appris une base d'atomes par ACI sur un enregistrement de radio de plusieurs jours. La radio est BBC Radio 4, qui diffuse majoritairement des émissions de débat, et donc des signaux de parole. Ils ont montré la similitude entre cette base et une base d'ondelettes. En effet, plus la fréquence centrale d'un atome grandit, plus la bande de fréquence s'élargit et plus le support temporel devient petit.

Les atomes peuvent donc être factorisés. Au lieu de laisser tous les paramètres des atomes libres, la structure de type ondelettes obtenue expérimentalement peut être imposé dès le départ aux atomes, en apprenant des objets de base qui sont ensuite dilatés et translatés pour générer le dictionnaire.

5.3 Proposition : la structuration du dictionnaire

Pour les décompositions parcimonieuses, les deux principales stratégies sont l'utilisation :

- d'un dictionnaire standard, comme la TFCT ou les ondelettes,

ou

- d'un dictionnaire appris de façon non-supervisée.

Nous proposons d'étudier une stratégie médiane consistant à intégrer les connaissances disponibles *a priori* sur les signaux d'apprentissage en imposant une structure aux atomes appris.

Le premier exemple simple concerne les signaux temporels. Au lieu d'effectuer un tramage régulier des signaux à l'aide d'une fenêtre de taille fixe, nous travaillerons sur le signal entier, en plaçant les atomes à l'instant approprié, sans la contrainte du tramage. Cette approche a déjà été proposée ces dernières années par Olshausen, Smith et Lewicki, et Blumensath et Davies [Ols00, SL05, BD06].

Au delà de cet exemple simple, nous proposons un formalisme de **motifs** qui subissent une **déformation** linéaire pour générer les atomes du dictionnaire. Cette déformation peut être la translation, le déphasage, la dilatation, l'expansion multicanale, la rotation d'images, ou toute combinaison de ces exemples.

La deuxième partie du document présente les contributions théoriques et expérimentales de cette thèse. Nous exposons le principe de structuration de dictionnaire, puis le modèle de déformation. Nous expliquons ensuite comment l'algorithme Matching Pursuit peut être adapté au modèle pour permettre des décompositions parcimonieuses.

Nous présentons ensuite deux algorithmes d'apprentissage de motifs, puis nous montrons l'intérêt de ce modèle pour la séparation de sources.

Deuxième partie

Contributions

-

Modèle de dictionnaire structuré,
décomposition parcimonieuse,
apprentissage de dictionnaire
structuré, séparation de sources

Chapitre 6

Exemple de structure : union de bases orthonormées

Nous avons conclu la première partie par l'intérêt potentiel de structurer le dictionnaire, en vue d'intégrer des connaissances *a priori* sur les signaux à décomposer. Dans les chapitres suivants, nous exposons le concept d'atomes formés par des motifs déformables, qui est la principale contribution de cette thèse.

Dans ce chapitre, nous proposons un type de structuration plus élémentaire, qui conserve le modèle matriciel (3.10) de la partie précédente, et qui consiste uniquement à imposer au dictionnaire d'être une union de bases orthonormées. Nous expliquons les raisons de ce choix juste après la description du modèle.

Le modèle de décomposition s'écrit :

$$\underline{s} = \mathbf{c}^T \cdot \underline{\mathbf{D}} + \underline{r} = [\mathbf{c}_1^T \cdots \mathbf{c}_P^T] \cdot \begin{bmatrix} \underline{\mathbf{D}}_1 \\ \vdots \\ \underline{\mathbf{D}}_P \end{bmatrix} + \underline{r} \quad (6.1)$$

où chaque base $\underline{\mathbf{D}}_p$ est une base orthonormée : $\underline{\mathbf{D}}_p \cdot \underline{\mathbf{D}}_p^T = \underline{\mathbf{D}}_p^T \cdot \underline{\mathbf{D}}_p = \underline{I}_L$. Le dictionnaire est donc redondant, de taille $PL \times L$.

Le choix de ce modèle a été initialement fait dans l'optique de combiner structuration de dictionnaire et apprentissage de dictionnaire. L'apprentissage de dictionnaire est le centre de notre travail, et nous avons expliqué au chapitre précédent les raisons qui nous poussent à vouloir structurer le dictionnaire.

Dans ce travail préliminaire, nous avons choisi la structure d'union de bases orthonormées pour sa simplicité et ses propriétés spécifiques intéressantes pour l'apprentissage. Ce choix se justifie également par l'utilisation fréquente d'union de bases orthonormées dans la littérature, pour décrire des signaux exhibant des caractéristiques de nature différente.

Ainsi, nous avons mentionné dans la section 3.1.2 que ce modèle a été utilisé par Daudet et Torrèsani [DT02] pour la décomposition de sons musicaux, avec $P = 2$ bases. La base $\underline{\mathbf{D}}_1$ est une base de cosinus (TCDM) qui code les parties tonales du son et $\underline{\mathbf{D}}_2$ est une base d'ondelettes qui code les parties transitoires. Nous avons également évoqué

le modèle proposé par Starck, Elad et Donoho [SED04] pour décomposer des images. Ils utilisent une base $\underline{\mathbf{D}}_1$ de cosinus (TDCM à deux dimensions) pour coder les textures d'images naturelles et une frame $\underline{\mathbf{D}}_2$ de "curvelettes" pour les contours.

La structure d'union de bases orthonormées permet également d'appliquer des stratégies spécifiques pour améliorer la décomposition. Ainsi, pour décomposer le signal $\underline{\mathbf{g}}$ sur le dictionnaire $\underline{\mathbf{D}}$, nous proposons d'utiliser le seuillage doux alterné présenté en section 2.3.2. Rappelons que cet algorithme démarre avec un vecteur de coefficients \mathbf{c} initial, et que, successivement, les sous-vecteurs de coefficients \mathbf{c}_p sont mis à jour individuellement alors que tous les autres $\mathbf{c}_{p'}$, avec $p' \neq p$, sont fixés.

Dans la section 6.1, nous proposons un algorithme d'apprentissage de dictionnaire intégrant la contrainte d'union de bases orthonormées. Cet algorithme met à jour les bases de façon successive. La section 6.2 évalue la capacité de l'algorithme proposé à extraire des bases représentatives des composantes du signal, et la section 6.3 montre que la contrainte structurelle est généralement trop forte quand on traite des signaux naturels musicaux. Nous rappelons également que ce modèle suit le modèle matriciel (3.10), dont les défauts ont été présentés au chapitre 5.

6.1 Mise à jour des bases par DVS

L'apprentissage d'une union de bases orthonormées suit le principe exposé dans la section 3.5, à savoir une alternance entre le calcul des meilleurs coefficients pour le dictionnaire courant, et la mise à jour du dictionnaire en gardant les coefficients fixes. Nous avons expliqué que le calcul des coefficients se fait par seuillage doux alterné. Nous exposons ici l'étape de mise à jour du dictionnaire, les coefficients étant fixés.

La méthode d'apprentissage d'une union de bases orthonormées proposée dans cette section a été présentée dans la conférence ICASSP 05 [LGBB05].

Nous exposons dans la section 6.1.1 le critère et l'algorithme d'apprentissage sur l'exemple simple d'un dictionnaire formé d'une seule base orthonormée. Nous le généralisons ensuite, dans la section 6.1.2, à un dictionnaire union de bases orthonormées.

6.1.1 Apprentissage d'une seule base orthonormée

Nous supposons dans cette section que le dictionnaire $\underline{\mathbf{D}}$ est une base orthonormée (taille $L \times L$), et donc le dictionnaire est non redondant. Nous modélisons la décomposition de l'ensemble des signaux $\underline{\mathbf{s}}_q$, regroupés dans la matrice $\underline{\mathbf{S}}$, comme :

$$\underline{\mathbf{S}} = \mathbf{C} \cdot \underline{\mathbf{D}} + \underline{\mathbf{R}}. \quad (6.2)$$

Le critère à optimiser est similaire à (3.34), auquel il faut ajouter la contrainte d'orthonormalité sur le dictionnaire. Le deuxième terme de (3.34), qui impose la parcimonie des coefficients \mathbf{C} , est inutile puisque les coefficients sont fixés lors de la mise à jour de la base. Notons que dans les perspectives, à la section 11.2, nous évoquons l'optimisation conjointe de la base et des coefficients, mais pour le moment, nous supposons que les coefficients ont été calculés et nous nous intéressons uniquement à la mise à jour de la base.

L'objectif est donc le suivant :

$$\underline{\mathbf{D}}^{\text{opt}} = \arg \min_{\underline{\mathbf{D}}} \|\underline{\mathbf{S}} - \mathbf{C} \cdot \underline{\mathbf{D}}\|_2^2 \text{ tel que } \underline{\mathbf{D}} \text{ soit une base orthonormée} \quad (6.3)$$

Ce problème a une solution analytique, qui est obtenue par le biais de la décomposition en valeurs singulières (DVS) de la matrice $\mathbf{C}^T \cdot \underline{\mathbf{S}}$.

Nous notons $\mathbf{U}^T \cdot \underline{\mathbf{\Delta}} \cdot \underline{\mathbf{V}}$ la DVS de $\mathbf{C}^T \cdot \underline{\mathbf{S}}$ (ou une DVS parmi toutes les DVS possibles). Les matrices \mathbf{U} et $\underline{\mathbf{V}}$ sont orthonormales et de taille $L \times L$, avec L la taille des signaux. La matrice $\underline{\mathbf{\Delta}}$, également de taille $L \times L$, est diagonale et la diagonale porte les valeurs singulières en ordre décroissant.

Les valeurs singulières sont toutes strictement positives si et seulement si la matrice $\mathbf{C}^T \cdot \underline{\mathbf{S}}$ est inversible. La DVS est unique si et seulement si toutes les valeurs singulières sont strictement positives, et toutes différentes.

Après ce rappel théorique sur la décomposition en valeurs singulières, présentons le théorème donnant la solution du problème (6.3).

Théorème 4 *La base $\underline{\mathbf{D}}^{\text{opt}}$, solution du problème (6.3), est donnée par*

$$\underline{\mathbf{D}}^{\text{opt}} = \mathbf{U}^T \cdot \underline{\mathbf{V}}, \quad (6.4)$$

où les matrices \mathbf{U} et $\underline{\mathbf{V}}$ sont obtenues par la décomposition en valeurs singulières de la matrice $\mathbf{C}^T \cdot \underline{\mathbf{S}}$:

$$\mathbf{U}^T \cdot \underline{\mathbf{\Delta}} \cdot \underline{\mathbf{V}} = \mathbf{C}^T \cdot \underline{\mathbf{S}}. \quad (6.5)$$

Si la matrice $\mathbf{C}^T \cdot \underline{\mathbf{S}}$ est inversible, alors $\mathbf{U}^T \cdot \underline{\mathbf{V}}$ est l'unique solution du problème (6.3). Si elle n'est pas inversible, alors $\mathbf{U}^T \cdot \underline{\mathbf{V}}$ est une solution parmi l'ensemble des solutions du problème (6.3).

La preuve complète de ce théorème est donnée dans l'annexe A.4. Expliquons en quelques lignes son principe.

En premier lieu, la contrainte d'orthonormalité peut être intégrée au critère en utilisant les multiplicateurs de Lagrange :

$$(\underline{\mathbf{D}}^{\text{opt}}, \underline{\mathbf{\Lambda}}^{\text{opt}}) = \arg \min_{\underline{\mathbf{D}}, \underline{\mathbf{\Lambda}}} \|\underline{\mathbf{S}} - \mathbf{C} \cdot \underline{\mathbf{D}}\|_2^2 + \text{Tr} [(\underline{\mathbf{D}} \cdot \underline{\mathbf{D}}^T - \mathbf{I}_L) \cdot \underline{\mathbf{\Lambda}}] , \quad (6.6)$$

où $\text{Tr}[\cdot]$ est l'opérateur de trace, et $\underline{\mathbf{\Lambda}}$ est une matrice symétrique de taille $L \times L$ de multiplicateurs de Lagrange. Le problème devient alors convexe, peut être résolu par une annulation de gradient, et la solution $\underline{\mathbf{D}}^{\text{opt}}$ est la même que celle du problème (6.3).

L'annulation des gradients de (6.6) par rapport à la base $\underline{\mathbf{D}}$ et par rapport à la matrice $\underline{\mathbf{\Lambda}}$ permet de montrer que $\underline{\mathbf{D}}^{\text{opt}}$ doit vérifier :

$$\underline{\mathbf{Z}}^T \cdot \underline{\mathbf{Z}} = \underline{\mathbf{D}}^{\text{opt}T} \cdot \underline{\mathbf{Z}} \cdot \underline{\mathbf{Z}}^T \cdot \underline{\mathbf{D}}^{\text{opt}}, \quad (6.7)$$

avec $\underline{\mathbf{Z}} = \mathbf{C}^T \cdot \underline{\mathbf{S}}$. En notant $\underline{\mathbf{Z}} = \mathbf{U}^T \cdot \underline{\mathbf{\Delta}} \cdot \underline{\mathbf{V}}$ la DVS de la matrice $\underline{\mathbf{Z}}$, nous prouvons alors que $\underline{\mathbf{D}}^{\text{opt}}$ vérifie :

$$\underline{\mathbf{D}}^{\text{opt}} = \mathbf{U}^T \cdot \underline{\mathbf{R}}^{\text{opt}^T} \cdot \underline{\mathbf{V}}, \quad (6.8)$$

où $\underline{\mathbf{R}}^{\text{opt}}$ est une matrice vérifiant $\underline{\mathbf{\Delta}}^2 \cdot \underline{\mathbf{R}}^{\text{opt}} = \underline{\mathbf{R}}^{\text{opt}} \cdot \underline{\mathbf{\Delta}}^2$. Nous prouvons enfin que $\underline{\mathbf{R}}^{\text{opt}} = \underline{\mathbf{L}}_L$ et $\underline{\mathbf{D}}^{\text{opt}} = \mathbf{U}^T \cdot \underline{\mathbf{V}}$ sont les solutions optimales, et nous montrons les conditions d'unicité de la solution.

6.1.2 Apprentissage d'une union de bases orthonormées

Lorsque le dictionnaire est formé de P bases orthonormées $\underline{\mathbf{D}}_p$, le modèle de décomposition de la matrice $\underline{\mathbf{S}}$ est :

$$\underline{\mathbf{S}} = \sum_{p=1}^P \mathbf{C}_p \cdot \underline{\mathbf{D}}_p + \underline{\mathbf{R}}. \quad (6.9)$$

Connaissant le signal $\underline{\mathbf{S}}$ et les coefficients précédemment calculés \mathbf{C} , le dictionnaire optimal est celui qui minimise l'erreur de reconstruction sous la contrainte d'orthonormalité :

$$\underline{\mathbf{D}}^{\text{opt}} = \arg \min_{\underline{\mathbf{D}}} \|\underline{\mathbf{S}} - \sum_{p=1}^P \mathbf{C}_p \cdot \underline{\mathbf{D}}_p\|_2^2 \text{ tel que les bases } \underline{\mathbf{D}}_p \text{ soient orthonormées.} \quad (6.10)$$

Le problème (6.10) n'a pas de solution explicite, contrairement au cas précédent d'un dictionnaire contenant une seule base orthonormée. Pour le résoudre, nous proposons une technique sous-optimale de mise à jour successive des bases $\underline{\mathbf{D}}_p$. Le principe de l'algorithme est le suivant :

1. Choix de la base $\underline{\mathbf{D}}_p$ à mettre à jour,
2. Retrait de la contribution des autres bases au signal

$$\underline{\mathbf{S}}_p = \underline{\mathbf{S}} - \sum_{p' \neq p} \mathbf{C}_{p'} \cdot \underline{\mathbf{D}}_{p'}, \quad (6.11)$$

3. Mise à jour de la base $\underline{\mathbf{D}}_p$:

$$\underline{\mathbf{D}}_p^{\text{opt}} = \arg \min_{\underline{\mathbf{D}}_p} \|\underline{\mathbf{S}}_p - \mathbf{C}_p \cdot \underline{\mathbf{D}}_p\|_2^2 \text{ tel que } \underline{\mathbf{D}}_p \text{ soit une base orthonormée} \quad (6.12)$$

4. Retour au pas 1

Le pas 3 correspond au problème traité dans la section précédente, qui se résout analytiquement. Les bases $\underline{\mathbf{D}}_p$ peuvent être choisies, au pas 1, dans l'ordre de leur numérotation, ou par une stratégie plus élaborée. Nous ne traitons pas cette question de la stratégie de choix des bases et dans les expériences, nous mettons à jour les bases dans l'ordre de leur numérotation.

L'algorithme d'apprentissage d'une union de bases orthonormées, dans son ensemble, est le suivant :

Apprentissage d'une union de bases orthonormées :

1. Initialiser le dictionnaire $\underline{\mathbf{D}} = [\underline{\mathbf{D}}_1 \cdots \underline{\mathbf{D}}_P]$
2. **Faire**
 - (a) Calculer un jeu de coefficients $\mathbf{C} = [\mathbf{C}_1 \cdots \mathbf{C}_P]$ parcimonieux pour la décomposition des signaux $\underline{\mathbf{S}}$ sur le dictionnaire $\underline{\mathbf{D}}$. L'algorithme de seuillage doux alterné est adapté à ce modèle de dictionnaire.
 - (b) **Pour** p allant de 1 à P
Faire la mise à jour de la base $\underline{\mathbf{D}}_p$:
 - i. Retrait de la contribution des autres bases au signal

$$\underline{\mathbf{S}}_p = \underline{\mathbf{S}} - \sum_{p' \neq p} \mathbf{C}_{p'} \cdot \underline{\mathbf{D}}_{p'} , \quad (6.13)$$

- ii. Décomposition en valeurs singulières de $\mathbf{C}_p^T \cdot \underline{\mathbf{S}}_p$:

$$\mathbf{C}_p^T \cdot \underline{\mathbf{S}}_p = \mathbf{U}^T \cdot \underline{\mathbf{\Delta}} \cdot \underline{\mathbf{V}} , \quad (6.14)$$

- iii. Mise à jour de la base $\underline{\mathbf{D}}_p$:

$$\underline{\mathbf{D}}_p = \mathbf{U}^T \cdot \underline{\mathbf{V}} . \quad (6.15)$$

Tant que l'algorithme n'a pas convergé.

6.1.3 Rôle du paramètre λ

Nous avons mentionné dans l'introduction que le calcul des coefficients est effectué avant la mise à jour du dictionnaire, et comme nous nous intéressons uniquement à cette mise à jour alors que les coefficients sont fixés, nous avons enlevé le terme de mesure de la parcimonie des coefficients dans le critère à minimiser.

Il faut cependant évoquer le critère dans son ensemble, et notamment le paramètre de compromis λ entre poids de l'erreur et poids des coefficients :

$$\|\underline{\mathbf{S}} - \mathbf{C} \cdot \underline{\mathbf{D}}\|_2^2 + \lambda \cdot \|\mathbf{C}\|_1 . \quad (6.16)$$

En effet, si ce paramètre est nul, ou s'il est trop élevé, l'algorithme de mise à jour des bases n'a pas d'utilité.

Montrons-le dans le cas de plusieurs bases. Si $\lambda = 0$, alors les coefficients optimaux sont obtenus par application de la pseudo-inverse à droite de $\underline{\mathbf{D}}$, qui vaut pour une union de P bases orthonormées :

$$\underline{\mathbf{D}}^+ = \frac{1}{P} \cdot \underline{\mathbf{D}}^T , \quad (6.17)$$

et donc les coefficients optimaux sont :

$$\mathbf{C}_p = \frac{1}{P} \cdot \underline{\mathbf{S}} \cdot \underline{\mathbf{D}}_p^T . \quad (6.18)$$

La matrice $\mathbf{C}_p^T \cdot \underline{\mathbf{S}}_p$ vaut alors :

$$\mathbf{C}_p^T \cdot \underline{\mathbf{S}}_p = \frac{1}{P} \cdot \underline{\mathbf{D}}_p \cdot \underline{\mathbf{S}}^T \cdot \left(\underline{\mathbf{S}} - \sum_{p' \neq p} \frac{1}{P} \cdot \underline{\mathbf{S}} \cdot \underline{\mathbf{D}}_{p'}^T \cdot \underline{\mathbf{D}}_{p'} \right) \quad (6.19)$$

$$= \frac{1}{P^2} \cdot \underline{\mathbf{D}}_p \cdot \underline{\mathbf{S}}^T \cdot \underline{\mathbf{S}} , \quad (6.20)$$

en remplaçant les coefficients par leur valeur, et en utilisant le fait que les bases $\underline{\mathbf{D}}_{p'}$ sont orthonormées. En notant $\underline{\mathbf{S}}^T \cdot \underline{\mathbf{S}} = \underline{\mathbf{V}}^T \cdot \underline{\mathbf{\Delta}} \cdot \underline{\mathbf{V}}$ la DVS de $\underline{\mathbf{S}}^T \cdot \underline{\mathbf{S}}$, alors la DVS de la matrice $\mathbf{C}_p^T \cdot \underline{\mathbf{S}}_p$, correspondant à l'étape (6.14) s'écrit :

$$\mathbf{C}_p^T \cdot \underline{\mathbf{S}}_p = (\underline{\mathbf{D}}_p \cdot \underline{\mathbf{V}}^T) \cdot \frac{\underline{\mathbf{\Delta}}}{P^2} \cdot \underline{\mathbf{V}} , \quad (6.21)$$

et la mise à jour (6.15) s'écrit :

$$\underline{\mathbf{D}}_p^i = \underline{\mathbf{D}}_p^{i-1} \cdot \underline{\mathbf{V}}^T \cdot \underline{\mathbf{V}} = \underline{\mathbf{D}}_p^{i-1} , \quad (6.22)$$

ce qui signifie que le dictionnaire reste inchangé.

L'algorithme présenté n'a donc une action que si le paramètre de compromis λ est supérieur à zéro.

A l'opposé, si ce paramètre est trop élevé, la parcimonie est maximale, et les coefficients valent zéro. Dans ce cas, optimiser le dictionnaire à coefficients fixés n'a pas de sens, car toutes les unions de bases orthonormées possibles donnent la même erreur.

Cela se vérifie en appliquant l'algorithme, car le calcul de la DVS de $\mathbf{C}_p^T \cdot \underline{\mathbf{S}}_p$ donne une matrice $\underline{\mathbf{\Delta}}$ nulle et n'importe quel couple de matrices orthonormales ($\underline{\mathbf{U}}$, $\underline{\mathbf{V}}$). La base mise à jour $\underline{\mathbf{D}}_p = \underline{\mathbf{U}}^T \cdot \underline{\mathbf{V}}$ peut donc être n'importe quelle base orthonormée et est obtenue par les matrices $\underline{\mathbf{U}}$ et $\underline{\mathbf{V}}$ dont le choix dépend du hasard de l'algorithme de DVS utilisé.

Le paramètre λ doit donc être choisi de façon à ce que les coefficients ne soient pas nuls (diversité nulle et parcimonie maximale), mais également de façon à ce que les coefficients ne reconstruisent pas exactement le signal (erreur nulle), car dans ces deux cas extrêmes, l'algorithme de mise à jour des bases est inutile.

6.2 Evaluation de la qualité de l'apprentissage du dictionnaire

Nous allons montrer dans cette partie expérimentale sous quelles conditions l'algorithme d'apprentissage d'union de bases orthonormées permet de retrouver un dictionnaire ayant généré des données parcimonieuses artificielles. Nous montrerons également que cet algorithme est limité par la taille des signaux qu'il peut traiter.

Dans un premier temps, nous allons définir les outils de mesure de la performance que nous utiliserons dans les expériences. Nous expliquerons ensuite les intentions expérimentales, et détaillerons trois expériences concernant l'influence de la quantité de données d'apprentissage, d'un mauvais dimensionnement du dictionnaire, et de la dimension des signaux traités.

6.2.1 Métriques

Nous présentons dans cette section des outils servant à mesurer la ressemblance entre deux dictionnaires. Nous définissons tout d'abord le corrélogramme qui représente les corrélations entre les atomes des deux dictionnaires, puis nous en déduisons deux mesures, le taux de rappel, et le taux de précision, qui permettent de compter et d'analyser le nombre d'atomes en commun dans les deux dictionnaires.

Nous donnons une définition du corrélogramme qui sera étendue dans la suite de la thèse. Pour le moment, la définition est la suivante. Soient deux dictionnaires $\underline{\mathbf{D}}_{ref}$ et $\underline{\mathbf{D}}_{app}$ contenant respectivement K_{ref} et K_{app} atomes de taille L . Le corrélogramme, noté $\mathbf{C}(\underline{\mathbf{D}}_{ref}, \underline{\mathbf{D}}_{app})$, est la matrice contenant tous les produits scalaires entre les atomes des deux dictionnaires, définie par :

$$\mathbf{C}(\underline{\mathbf{D}}_{ref}, \underline{\mathbf{D}}_{app}) = \underline{\mathbf{D}}_{ref} \cdot \underline{\mathbf{D}}_{app}^T = \begin{bmatrix} \vdots & & \\ \langle \underline{d}_i^{ref}, \underline{d}_j^{app} \rangle & \dots & \\ \vdots & & \end{bmatrix} \leftarrow i, \quad \begin{matrix} j \\ \downarrow \\ \vdots \end{matrix} \quad (6.23)$$

où \underline{d}_i^{ref} est le i^e atome du dictionnaire $\underline{\mathbf{D}}_{ref}$.

Lorsque les atomes sont normés, la matrice comprend des valeurs entre -1 et 1 . Le corrélogramme peut être affiché comme une image de même taille, en attribuant à chaque pixel une couleur correspondant à la valeur de la case associée dans la matrice. Il peut aussi être résumé par les corrélations maximales sur les lignes et les colonnes, et de façon encore plus concise, avec des mesures de rappel et de précision.

La première façon de résumer le corrélogramme consiste à calculer, pour chaque ligne du corrélogramme, la corrélation maximale du motif de référence avec tous les motifs appris ; et pour chaque colonne, la corrélation maximale du motif appris avec tous les motifs de référence. L'affichage du premier nuage de points indique à quel point les motifs de référence ont été retrouvés, et l'affichage du deuxième nuage de points indique à quel point les motifs appris sont corrects, c'est-à-dire correspondent à un motif de référence. Nous utiliserons une figure de ce style à la section 9.4.

La deuxième façon de résumer le corrélogramme est d'utiliser les taux de rappel et de précision, qui découlent de ces nuages de points. C'est celle que nous utiliserons dans cette section. Les mesures de rappel et de précision sont classiques dans le domaine de la recherche d'information. Elles servent notamment à mesurer la qualité de la réponse d'un système à une requête de recherche de documents dans une base de données. Ainsi, la mesure de rappel est définie par la proportion de documents pertinents trouvés par le

système, parmi l'ensemble de documents pertinents dans la base. La mesure de précision est définie par la proportion de documents pertinents parmi l'ensemble des réponses du système.

Dans le cadre de la comparaison d'atomes, le dictionnaire de référence $\underline{\mathbf{D}}_{ref}$ correspond à l'ensemble des documents pertinents dans la base de l'exemple, tandis que le dictionnaire appris $\underline{\mathbf{D}}_{app}$ correspond à l'ensemble des documents donnés en réponse par le système. La vision est tout de même un peu différente, puisque les atomes ne peuvent pas forcément être appariés. En effet, il se peut que deux atomes appris correspondent au même atome de référence, par exemple.

Nous définissons les mesures de rappel $\mathcal{R}(\underline{\mathbf{D}}_{ref}, \underline{\mathbf{D}}_{app})$ et de précision $\mathcal{P}(\underline{\mathbf{D}}_{ref}, \underline{\mathbf{D}}_{app})$ par :

$$\mathcal{R}(\underline{\mathbf{D}}_{ref}, \underline{\mathbf{D}}_{app}) = \frac{\text{nombre d'atomes de } \underline{\mathbf{D}}_{ref} \text{ présents dans } \underline{\mathbf{D}}_{app}}{\text{nombre d'atomes de } \underline{\mathbf{D}}_{ref}} \quad (6.24)$$

$$\mathcal{P}(\underline{\mathbf{D}}_{ref}, \underline{\mathbf{D}}_{app}) = \frac{\text{nombre d'atomes de } \underline{\mathbf{D}}_{app} \text{ présents dans } \underline{\mathbf{D}}_{ref}}{\text{nombre d'atomes de } \underline{\mathbf{D}}_{app}} \quad (6.25)$$

Le nombre d'atomes d'un dictionnaire présents dans l'autre dictionnaire est défini à l'aide du corrélogramme. Nous définissons qu'un atome est présent dans un dictionnaire si sa corrélation avec un au moins des atomes du dictionnaire est supérieure à un seuil donné. Le test d'appartenance d'un atome \underline{d} à un dictionnaire $\underline{\mathbf{D}}$, conditionnellement à un seuil ξ , est défini par :

$$\mathcal{A}(\underline{d}, \underline{\mathbf{D}}, \xi) = \begin{cases} 1 & \text{si } \exists \underline{d}_k \in \underline{\mathbf{D}} / \frac{|\langle \underline{d}, \underline{d}_k \rangle|^2}{\|\underline{d}\|_2 \cdot \|\underline{d}_k\|_2} \geq \xi \\ 0 & \text{sinon} \end{cases} \quad (6.26)$$

A l'aide de cette mesure, le rappel et la précision sont définis par :

$$\mathcal{R}(\underline{\mathbf{D}}_{ref}, \underline{\mathbf{D}}_{app}, \xi) = \frac{1}{K_{ref}} \cdot \text{card} \{ \underline{d}_k \in \underline{\mathbf{D}}_{ref} / \mathcal{A}(\underline{d}_k, \underline{\mathbf{D}}_{app}, \xi) = 1 \} \quad (6.27)$$

$$\mathcal{P}(\underline{\mathbf{D}}_{ref}, \underline{\mathbf{D}}_{app}, \xi) = \frac{1}{K_{app}} \cdot \text{card} \{ \underline{d}_k \in \underline{\mathbf{D}}_{app} / \mathcal{A}(\underline{d}_k, \underline{\mathbf{D}}_{ref}, \xi) = 1 \} \quad (6.28)$$

Ces mesures peuvent être calculées en calculant les valeurs maximales du carré de chaque ligne et de chaque colonne du corrélogramme. Le nombre de maxima des lignes du corrélogramme supérieurs au seuil, rapporté au nombre de lignes, donne le rappel, tandis que le nombre de maxima des colonnes du corrélogramme supérieurs au seuil, rapporté au nombre de colonnes, donne la précision.

Le seuil doit être choisi entre 0 et 1 lorsque tous les atomes sont normés. Si le seuil vaut 0, alors quels que soient les dictionnaires, le rappel et la précision valent 1, ce qui signifie que tous les atomes du dictionnaire de référence ont été retrouvés et que tous les atomes appris appartiennent au dictionnaire de référence. Si le seuil vaut 1, alors le rappel donne le ratio d'atomes du dictionnaire de référence qui se retrouvent exactement, et la précision donne le ratio d'atomes appris qui sont exactement des atomes de référence.

Après la définition de ces mesures, exposons nos intentions expérimentales concernant l'apprentissage de dictionnaire structuré comme l'union de bases orthonormées.

6.2.2 Intentions expérimentales

Nous désirons évaluer les capacités de l'algorithme présenté en section 6.1.2 à retrouver un dictionnaire de référence, à partir de données artificielles générées de façon parcimonieuses par ce dictionnaire de référence, c'est-à-dire à retrouver les atomes sur lesquels les données d'apprentissage sont intrinsèquement parcimonieuses.

Nous chercherons quel est l'ordre de grandeur de la taille des données d'apprentissage nécessaire à un bon apprentissage. Nous examinerons ensuite l'influence d'une erreur de modélisation concernant le nombre d'atomes recherchés, par rapport au nombre d'atomes réel.

Enfin, nous étudierons l'influence de la taille des signaux traités sur les propriétés de convergence de l'algorithme, afin de comprendre pour quels types de signaux l'algorithme est utilisable.

6.2.3 Influence de la taille de l'ensemble d'entraînement

L'algorithme nécessite un ensemble de signaux d'entraînement afin d'apprendre les atomes permettant de les décomposer de façon parcimonieuse. Il est intuitif que si l'ensemble est trop petit, alors tous les atomes ne seront pas représentés et il sera impossible d'apprendre toute la diversité du dictionnaire. Nous essaierons dans cette expérience de déterminer à partir de quelle taille l'ensemble de données est suffisamment grand pour permettre d'apprendre correctement le dictionnaire.

6.2.3.1 Principe de l'expérience

Le principe de l'expérience est le suivant. Nous créons un dictionnaire de référence, structuré comme une union de bases orthonormées, puis des signaux artificiels, à la parcimonie contrôlée, à partir de ce dictionnaire de référence. Le nombre de signaux créés est le paramètre à étudier et nous le faisons varier. Nous apprenons ensuite, sur chacun de ces signaux, un dictionnaire de même taille que le dictionnaire de référence.

Nous mesurons ensuite la qualité des atomes appris à l'aide des mesures de rappel et de précision, et nous analysons ces mesures par rapport au nombre de signaux d'apprentissage utilisés.

Étant donné que le nombre de paramètres à apprendre dans le dictionnaire est $L \cdot (L - 1)$ (voir la section 3.2), le nombre de signaux Q sera du type $n \cdot L$ de sorte que le nombre d'échantillons dans les signaux d'entraînement est (presque) proportionnel au nombre de paramètres libres du modèle : $n \cdot L^2$.

L'expérience est réalisée un grand nombre de fois et nous présentons la moyenne des résultats de toutes ces expériences. De plus, l'expérience sera réalisée pour plusieurs tailles de signaux, afin d'éviter de tirer des conclusions à partir d'un cas particulier.

6.2.3.2 Protocole détaillé

Le protocole détaillé est décrit dans le tableau suivant

Dictionnaire $\underline{\mathbf{D}}_{ref}$	Union de deux bases orthonormées, initialisées aléatoirement selon une loi gaussienne, puis orthonormalisées par l'algorithme de Gram-Schmidt
Dictionnaire $\underline{\mathbf{D}}_{app}$	Union de deux bases orthonormées, de la même taille que $\underline{\mathbf{D}}_{ref}$
Taille des signaux L	4, 8, 16, 32 puis 64
Génération des signaux $\underline{\mathbf{s}}_q$	Chaque signal est généré par 1 à r atomes tirés aléatoirement selon une loi uniforme, et dont les amplitudes suivent une loi gaussienne.
Nombre d'atomes max r	$\lfloor \frac{1+\sqrt{L}}{2} \rfloor$, afin d'assurer que \mathbf{c}_q est le jeu de coefficients le plus parcimonieux générant le signal $\underline{\mathbf{s}}_q$ [GN04]
Nombre de signaux $Q = n \cdot L$	$n = 1/4, 1, 2, 5, 10, 20, 50$ et 100
Nombre d'expériences N_e	100, pour chaque taille de signal L
Seuil de décision ξ pour la précision et le rappel	0.99
Algorithme de calcul des coefficients	Seuillage doux alterné. Le paramètre de compromis λ est réglé selon la méthode indiquée dans [LGBB05] et décroît linéaire au fil des itérations de décomposition. Un seuillage dur, de moins en moins sévère au cours de l'apprentissage, est ensuite appliqué sur les coefficients
Initialisation de $\underline{\mathbf{D}}_{app}$ dans l'algorithme d'apprentissage	Tirage aléatoire selon une loi gaussienne, puis orthonormalisation de chacune des bases
Critère d'arrêt de l'algorithme d'apprentissage	La diversité des coefficients calculés est stable ou le nombre d'itérations est supérieur à 2 000

6.2.3.3 Résultats

Les résultats vont être présentés sous la forme suivante. Pour chaque valeur de la longueur des signaux L , nous afficherons le taux de rappel et de précision, en pourcentage, en fonction du coefficient $n = Q/L$. Cette abscisse permettra de tenir compte des différentes valeurs de L .

De façon intuitive, puisque le nombre de paramètres à estimer est de $L \cdot (L - 1)$ pour deux bases orthonormales, nous nous attendons à ce que les taux de rappel et de précision soient proches de 0% lorsque le nombre d'échantillons cumulé sur tous les signaux d'entraînement est inférieur à $L \cdot (L - 1)$, ce qui est le cas pour $n = 1/4$, et presque pour $n = 1$. De façon plus générale, nous pouvons penser qu'il faut que le

nombre de paramètres d'apprentissage ($Q \cdot L$) soit significativement plus grand que le nombre de paramètres libres pour que ceux-ci soient correctement estimés, et donc nous nous attendons à ce que les taux de rappel et de précision deviennent grands uniquement pour de grandes valeurs de n .

La figure 6.1 présente la courbe du taux de rappel en fonction du coefficient $n = Q/L$, pour les différentes valeurs de la taille des signaux L . Nous n'affichons pas la courbe du taux de précision car elle est quasiment identique. L'abscisse est présentée en échelle logarithmique pour être en conformité avec la méthode d'exploration, où n croît de façon exponentielle.

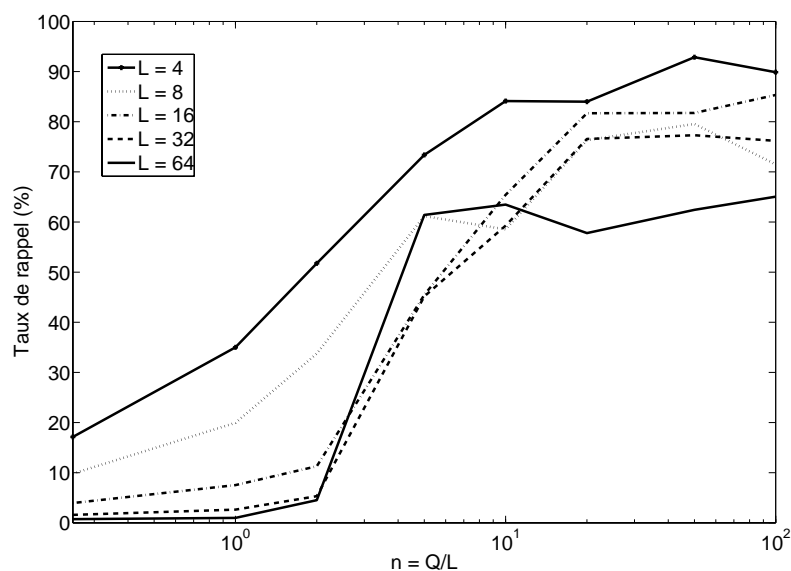


FIG. 6.1 – Taux de rappel moyen (sur 100 essais) en fonction du rapport entre le nombre de signaux d'apprentissage Q et la dimension des signaux L , pour différentes dimensions L

Nous voyons sur cette figure que globalement, le taux de rappel moyen augmente avec le nombre de signaux d'apprentissage, quelle que soit la dimension du signal. Cela signifie que plus la quantité de données d'apprentissage est élevée, plus le dictionnaire appris est proche du dictionnaire de référence. C'est un résultat logique pour les raisons données précédemment. D'un point de vue probabiliste, rappelons que nous avons défini le problème d'apprentissage comme la maximisation de la vraisemblance des données, en espérance. L'espérance est approchée par une moyenne sur Q signaux, il est donc intuitif, en suivant la loi des grands nombres, que la qualité de l'approximation de l'espérance augmente avec le nombre de signaux, et par suite que la qualité de l'apprentissage se trouve améliorée.

Il est remarquable que les courbes soient d'allure similaire, quelle que soit la dimension. Nous pouvons en conclure qu'il est important de régler le nombre de signaux Q

de façon proportionnelle à la dimension L , et que le facteur important est n .

De façon variable suivant les dimensions, il semble qu'un palier soit atteint vers $n = 20$, et qu'à partir de cette valeur, la performance de l'algorithme stagne. Nous pouvons en déduire que s'il est important que le nombre de signaux soit environ 20 fois supérieur à la dimension, en revanche, il n'est pas nécessaire d'aller plus loin. Du point de vue calculatoire, la complexité allant en grandissant comme nous le verrons en section 6.2.5, le choix de $Q = 20 \cdot L$ paraît judicieux.

Nous avons étudié l'influence de la taille de l'ensemble d'apprentissage sur la qualité de l'apprentissage, en supposant connaître la taille du dictionnaire à apprendre. Qu'en est-il lorsque le dictionnaire de référence et le dictionnaire appris ne sont pas de même taille ?

6.2.4 Influence du dimensionnement du dictionnaire

L'expérience précédente a été faite sous l'hypothèse forte que les dimensions du dictionnaire sous-jacent sont connues. En effet, le nombre d'atomes sous-jacents ne peut pas être connu à l'avance, et nous pouvons nous demander quelle est l'influence d'une méconnaissance du nombre d'atomes. Nous verrons qu'il est nécessaire de sur-dimensionner le dictionnaire pour espérer retrouver le dictionnaire de référence.

6.2.4.1 Principe de l'expérience

Nous proposons l'expérience suivante. L'objectif est toujours de retrouver un dictionnaire sous-jacent, à partir de données parcimonieuses artificiellement créées sur ce dictionnaire de référence. Nous testerons les performances de l'algorithme d'apprentissage, pour un nombre de bases donné dans le dictionnaire appris, en fonction du nombre de bases dans le dictionnaire de référence. La performance sera de nouveau mesurée en termes de rappel et de précision, et en moyennant sur un grand nombre d'expériences.

6.2.4.2 Protocole détaillé

Le protocole détaillé est décrit dans le tableau suivant

Nombre de bases orthonormées dans le dictionnaire $\underline{\mathbf{D}}_{ref}$	1, 2 ou 3
Initialisation des bases de $\underline{\mathbf{D}}_{ref}$	Tirage aléatoire des composantes selon une loi gaussienne, puis orthonormalisation des bases par l'algorithme de Gram-Schmidt
Dictionnaire $\underline{\mathbf{D}}_{app}$	Union de deux bases orthonormées
Taille des signaux L	32
Génération des signaux $\underline{\mathbf{s}}_q$	Chaque signal est généré par 1 à $r = 3$ atomes tirés aléatoirement selon une loi uniforme, et dont les amplitudes suivent une loi gaussienne.
Nombre d'atomes max r	$\lfloor \frac{1+\sqrt{L}}{2} \rfloor = 3$, afin d'assurer que \mathbf{c}_q est le jeu de coefficients le plus parmonieux générant le signal $\underline{\mathbf{s}}_q$ [GN04]
Nombre de signaux $Q = n \cdot L$	$n = 50$
Nombre d'expériences N_e	200
Seuil de décision ξ pour la précision et le rappel	0.99
Algorithme de calcul des coefficients	Seuillage doux alterné. Le paramètre de compromis λ est réglé selon la méthode indiquée dans [LGBB05] et décroît linéaire au fil des itérations de décomposition. Un seuillage dur, de moins en moins sévère au cours de l'apprentissage, est ensuite appliqué sur les coefficients
Initialisation de $\underline{\mathbf{D}}_{app}$ dans l'algorithme d'apprentissage	Union de deux fois la matrice identité : $\underline{\mathbf{D}}_{app} = [\underline{\mathbf{I}} \ \underline{\mathbf{I}}]$
Critère d'arrêt de l'algorithme d'apprentissage	La diversité des coefficients calculés est stable ou le nombre d'itérations est supérieur à 2 000

6.2.4.3 Résultats

Nous présenterons les résultats sous la forme d'un tableau contenant les taux moyens de rappel et de précision pour les trois configurations, à savoir modèle sur-dimensionné (deux bases au lieu d'une), bien dimensionné, et sous-dimensionné (deux bases au lieu de trois).

Nous pouvons nous attendre à ce que le modèle bien dimensionné donne le même type de résultat que dans le cas précédent. Pour le modèle sur-dimensionné, nous espé-

rons qu'au moins une des deux bases du dictionnaire appris retrouve la base de référence. L'indice important est le taux de rappel, qui compte le pourcentage d'atomes de référence correctement appris. En effet, le taux de précision, dans ce cas, peut être faible, jusqu'à 50%, sans que le dictionnaire soit mal appris, puisqu'il suffit qu'une base soit bien apprise, soit la moitié des atomes. Dans le cas sous-dimensionné, le taux de rappel ne peut pas être supérieur à 66%, puisque seuls 64 atomes sont appris alors que le dictionnaire de référence en compte 96. Nous verrons si le taux de rappel se rapproche de ce taux maximal, ce qui ne peut arriver que si le taux de précision s'approche de 100%, ou si l'algorithme prend le parti d'essayer de coder les 96 atomes par deux bases différentes, mais plus efficaces, que tout sous-ensemble de deux bases de référence.

Dictionnaire de référence $\underline{\mathbf{D}}_{ref}$	$\underline{\mathbf{D}}_{ref,1}$	$\underline{\mathbf{D}}_{ref,2}$	$\underline{\mathbf{D}}_{ref,3}$
Taux moyen de rappel	99.5%	93%	0.5%
Taux moyen de précision	56%	93%	1%

Les résultats sont meilleurs qu'attendu pour le dictionnaire bien dimensionné. Rappelons que nous avons changé un paramètre par rapport à l'expérience précédente, au niveau de l'initialisation du dictionnaire. Au lieu d'initialiser les bases aléatoirement, nous les avons initialisées avec la matrice identité. Le résultat est un taux moyen de rappel de 93% au lieu de 77% avec une initialisation aléatoire. Nous n'expliquons pas la différence de performance. Notons que parmi les 200 expériences lancées, deux cas de figure apparaissent pour expliquer le chiffre de 93%. Dans 8% des cas, aucun atome n'est correctement retrouvé, tandis que dans 92% des cas, tous les atomes sont parfaitement retrouvés. Nous supposons que l'initialisation joue un grand rôle, qui reste à comprendre.

Pour le dictionnaire sur-dimensionné, presque tous les atomes sont retrouvés, et 6% des atomes de référence sont présents dans les deux bases apprises, puisque 56% des atomes appris correspondent à un atome de référence. Parmi les 200 expériences, nous pouvons encore une fois distinguer deux comportements. Dans 55% des cas, une des deux bases de $\underline{\mathbf{D}}_{app}$ est exactement la base de référence, et dans 45% des cas, les atomes sont répartis dans les deux bases (82% dans la première, 17% dans la deuxième, et 1% n'est pas retrouvé).

Dans le cas sous-dimensionné, nous remarquons que pratiquement aucun atome n'est retrouvé correctement. Notons que parmi les 200 expériences, aucune n'a permis de retrouver plus de 15% des atomes. Il serait intéressant dans ce cas de regarder si les signaux sont plus parcimonieux dans le dictionnaire appris $\underline{\mathbf{D}}_{app}$ que dans n'importe quelle union de deux bases de $\underline{\mathbf{D}}_{ref}$, ce qui expliquerait le choix fait par l'algorithme.

Pour conclure sur cette expérience, lorsque l'objectif est de retrouver un dictionnaire sous-jacent, puisque l'algorithme n'est efficace que lorsque le nombre de bases apprises est au moins aussi grand que le nombre de bases de référence, il paraît indispensable de sur-dimensionner le dictionnaire, quitte à retirer ensuite des bases. Il faudrait avant cela vérifier que le comportement observé pour peu de bases se reproduit pour un plus grand nombre de bases.

Intéressons-nous maintenant à la vitesse de convergence de l'algorithme, en fonction de la dimension des signaux, afin d'évaluer la limite d'utilisation de l'algorithme.

6.2.5 Influence de la dimension des signaux

Après avoir étudié l'influence de la taille de l'ensemble d'apprentissage, et d'une erreur de modélisation, intéressons-nous à l'étude de la complexité de l'algorithme d'apprentissage. Nous allons montrer que cet algorithme ne peut être utilisé que pour des petites dimensions, ce qui limite son intérêt pour le traitement de données réelles.

6.2.5.1 Principe de l'expérience

Nous ne nous intéressons plus maintenant aux performances en termes de rappel et de précision de l'algorithme, mais uniquement à sa complexité algorithmique.

Pour cela, nous allons mesurer le temps de calcul et le nombre d'itérations nécessaires à la convergence, lorsqu'il converge, lors de l'apprentissage d'un dictionnaire sur un ensemble de signaux artificiels, générés comme pour les deux expériences précédentes à partir d'un dictionnaire de référence. Nous lancerons de nombreuses expériences, afin d'établir des statistiques fiables, pour différentes valeurs de la dimension L des signaux. L'algorithme est dit convergent lorsqu'il retrouve le dictionnaire exactement, c'est-à-dire avec un taux de rappel de 100%, pour un seuil ξ .

6.2.5.2 Protocole détaillé

Le protocole détaillé est décrit dans le tableau suivant

dictionnaire $\underline{\mathbf{D}}_{ref}$	Union de deux bases orthonormées
Initialisation des bases de $\underline{\mathbf{D}}_{ref}$	Tirage aléatoire des composantes selon une loi gaussienne, puis orthonormalisation des bases par l'algorithme de Gram-Schmidt
Dictionnaire $\underline{\mathbf{D}}_{app}$	Union de deux bases orthonormées
Taille des signaux L	4, 8, 16, 32, 64 et 128 pour les deux mesures ainsi que 256, 512, 1 024 et 2 048 pour la mesure du temps de calcul d'une itération
Génération des signaux $\underline{\mathbf{s}}_q$	Chaque signal est généré par 1 à r atomes tirés aléatoirement selon une loi uniforme, et dont les amplitudes suivent une loi gaussienne.
Nombre d'atomes max r	$\lfloor \frac{1+\sqrt{L}}{2} \rfloor = 3$, afin d'assurer que \mathbf{c}_q est le jeu de coefficients le plus parcimonieux générant le signal $\underline{\mathbf{s}}_q$ [GN04]
Nombre de signaux $Q = n \cdot L$	$n = 50$
Nombre d'expériences N_e	100
Seuil de décision ξ pour la précision et le rappel	0.99
Algorithme de calcul des coefficients	Seuillage doux alterné. Le paramètre de compromis λ est réglé selon la méthode indiquée dans [LGBB05] et décroît linéaire au fil des itérations de décomposition. Un seuillage dur, de moins en moins sévère au cours de l'apprentissage, est ensuite appliqué sur les coefficients
Initialisation de $\underline{\mathbf{D}}_{app}$ dans l'algorithme d'apprentissage	Union de deux fois la matrice identité : $\underline{\mathbf{D}}_{app} = [\underline{\mathbf{I}} \ \underline{\mathbf{I}}]$
Critère d'arrêt de l'algorithme d'apprentissage	La diversité des coefficients calculés est stable ou le nombre d'itérations est supérieur à 2 000

6.2.5.3 Résultats

Nous allons tout d'abord présenter sous forme graphique l'évolution du temps de calcul d'une itération de l'algorithme d'apprentissage, en fonction de la dimension des signaux. Il s'agira de mesurer selon quelle règle le temps de calcul augmente avec la dimension.

Dans un deuxième temps, nous présenterons une représentation statistique du nombre

d'itérations nécessaires à la convergence, pour chaque dimension (valeur pour chaque expérience, moyenne, écart type et régression linéaire).

La figure 6.2 représente l'évolution du temps de calcul d'une itération de l'algorithme d'apprentissage sur un Pentium 4.

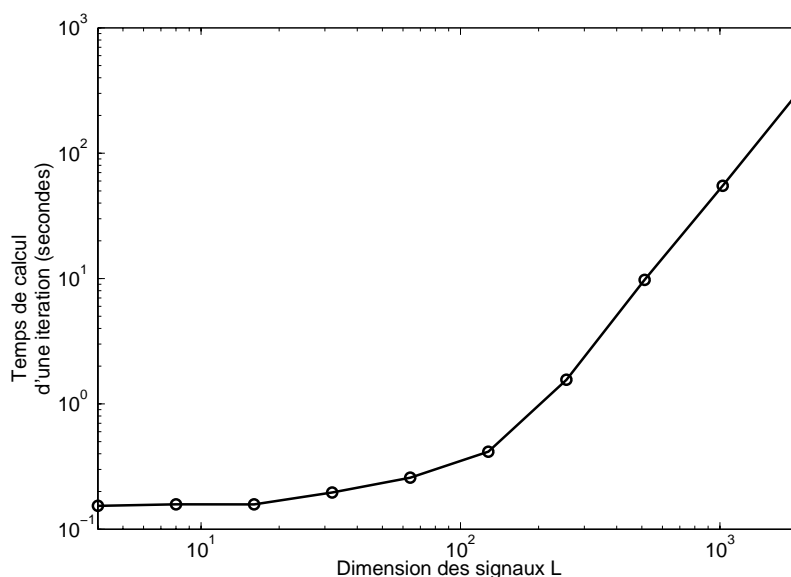


FIG. 6.2 – Évolution du temps de calcul d'une itération en fonction de la dimension des signaux.

Nous pouvons observer que si pour des petites dimensions, le temps de calcul ne varie pas vraiment en fonction de la dimension, en revanche, à partir de $L = 128$, le temps de calcul évolue comme une puissance de L (entre $O(L^2)$ et $O(L^3)$). Cela n'a rien d'étonnant puisque la dimension des matrices traitées est multiple de L^2 .

Si le temps de calcul d'une itération est de 10 s pour $L = 512$, il est de 324 s, soit environ 6 minutes, pour $L = 2048$. Évidemment, lorsqu'il faut faire des centaines d'itérations, comme nous allons le voir ensuite, ce n'est pas envisageable. Nous ne détaillons pas l'utilisation de la mémoire, mais grossièrement, les matrices de signaux, pour $L = 2048$ contiennent $2 \cdot 10^8$ échantillons, et pour les stocker en flottants double précision, soit 4 octets, il faut environ un 1 Go, ce qui correspond aux limites des machines actuelles.

Analysons maintenant le nombre d'itérations nécessaires à la convergence de l'algorithme en fonction de la dimension des signaux, qui est affiché sur la figure 6.3.

Nous observons que le nombre d'itérations moyen permettant la convergence de l'algorithme croît de façon linéaire avec la dimension L des signaux. L'utilisation de la moyenne est valable, puisque la variance des données est assez petite autour de cette moyenne pour voir clairement la dépendance linéaire entre la dimension L et le nombre d'itérations.

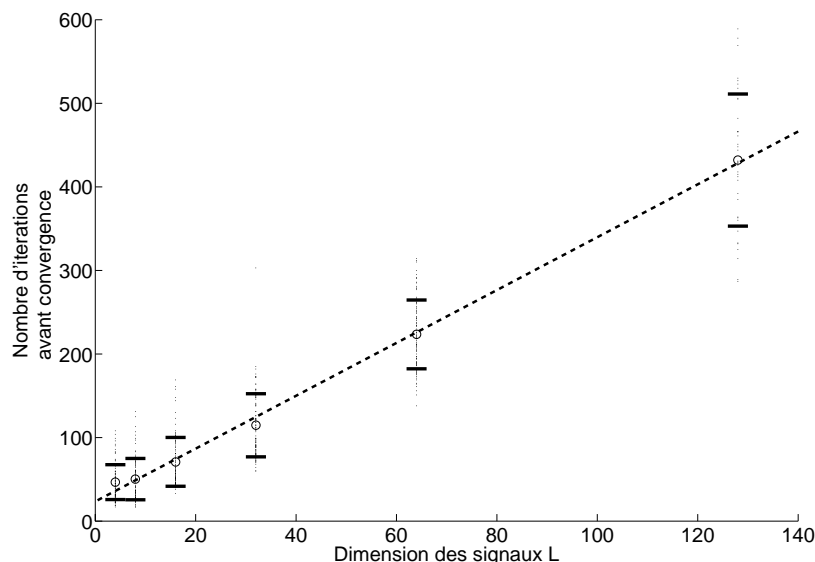


FIG. 6.3 – Évolution du nombre d'itérations nécessaires à la convergence de l'algorithme d'apprentissage en fonction de la dimension des signaux.

En nous rappelant l'évolution du temps de calcul d'une itération en fonction de la dimension, nous pouvons donc évaluer que la complexité de l'algorithme se situe entre $O(L^3)$ et $O(L^4)$, ce qui le disqualifie pour des dimensions de signaux dépassant la centaine d'échantillons. Or pour traiter des signaux audio, par exemple, la taille des trames de signal utilisées est en général d'environ 20 ms. Pour une fréquence d'échantillonnage de 44 100 Hz, une trame de 23.2 ms correspond à $L = 1\,024$, ce qui est impossible à traiter avec cet algorithme. Un challenge majeur pour permettre l'utilisation de cet algorithme est de l'adapter à de plus grandes dimensions.

6.3 Analyse critique

Nous avons présenté un algorithme permettant d'apprendre un dictionnaire, structuré comme une union de bases orthonormales, adapté à un ensemble de signaux, c'est-à-dire qui permet de les décomposer de façon parcimonieuse.

Nous avons montré ses bonnes propriétés de convergence, sous la condition d'un sur-dimensionnement du dictionnaire et d'un nombre de données d'apprentissage suffisamment grand. Nous avons également montré ses limites en terme de dimension des signaux traitables, qui ne peut guère être plus grande qu'une centaine d'échantillons par trame. Cette dernière critique peut être faite aux autres algorithmes d'apprentissage de dictionnaire non structurés, puisqu'ils utilisent les mêmes données. Cela plaide pour l'invariance par translation et plus largement pour le modèle d'atomes structurés que nous présentons dans le chapitre suivant et qui est mieux adapté aux signaux temporels

de grande dimension.

Il serait intéressant de comparer, sur des données réelles, la parcimonie atteinte par un dictionnaire structuré et par un dictionnaire non structuré, afin de comprendre si la contrainte d'orthogonalité est handicapante ou non. A noter, le fait d'imposer que le nombre d'atomes soit un multiple de L est aussi une contrainte forte. Nous pouvons nous attendre à ce qu'en pratique, le modèle structuré ait peu d'utilité pour les décompositions parcimonieuses, en revanche, il est intéressant à deux titres :

- il a permis de se pencher sur le problème de l'intégration d'une contrainte de structure dans la modélisation du dictionnaire, notamment pour son apprentissage,
- l'algorithme résultant, reposant sur une mise à jour alternée des différentes bases qui composent le dictionnaire et sur une décomposition en valeurs singulières, peut être vu comme une première ébauche de l'algorithme K-SVD [AEB05], que nous avons présenté en section 3.5.2 et que nous adaptons par la suite. Nous pouvons d'ailleurs trouver une référence à notre algorithme dans le nom K-SVD, puisque l'algorithme K-SVD n'effectue pas de DVS, mais exploite uniquement la composante principale, contrairement à notre algorithme qui utilise toutes les composantes.

Le modèle de dictionnaire utilisé dans cette section est un modèle matriciel, et nous avons exposé certaines de ses limites dans la section 5.2. Pour s'affranchir de la contrainte de tramage des signaux, notamment, nous proposons dans la section suivante un modèle d'atome généré par un motif subissant une déformation linéaire.

Chapitre 7

Structure fondamentale : dictionnaire généré par déformation linéaire de motifs

Dans ce chapitre, nous changeons de point de vue et de modèle de signaux. Nous avons pointé les manques du modèle de dictionnaire non-structuré : nécessité de tramer les signaux, redondance inutile du dictionnaire, pas de prise en compte des connaissances *a priori* sur les invariances du dictionnaire. Nous proposons ici de travailler sur des signaux entiers, c'est-à-dire non-tronqués et donc de dimension "infinie".

Nous nous focalisons sur les signaux temporels monodimensionnels (monocanaux) ou multidimensionnels (multicanaux). Cependant, le modèle que nous présentons ici est plus général, et peut être utilisé pour traiter des images ou des vidéos (nous utilisons le terme vidéo pour désigner une séquence d'images, et non pas un film contenant du son en plus des images).

Nous définissons le dictionnaire comme une collection d'atomes, qui sont des signaux du même type que les signaux traités. Un atome n'est pas paramétré par sa valeur à chaque échantillon sur l'axe du temps, mais doit être vu comme le résultat, par exemple, du placement d'un support, nommé motif, à un instant donné, ou bien du filtrage d'un signal d'excitation par un filtre à réponse impulsionnelle finie (RIF). Le dictionnaire est donc généré à partir d'un dictionnaire de motifs et de paramètres additionnels permettant de modifier et déformer ces motifs.

Nous exposons dans la section 7.1 les motivations qui ont conduit à changer de modèle pour les signaux et les atomes. Dans la section 7.2, nous présentons le modèle de signaux. Dans la section 7.3, nous détaillons le modèle d'atomes dans le cas intuitif de l'invariance par translation, qui a déjà été proposé par le passé [Ols00, SL05, BD06]. Dans la section 7.4, nous présentons le modèle d'atomes créés par des motifs déformés par une fonction linéaire. Ce modèle, ainsi que les algorithmes de décomposition parcimonieuse et d'apprentissage de dictionnaire, est la principale contribution de cette thèse.

7.1 Motivations

Nous avons exprimé dans les sections 5.2 et 6.3 les limites du modèle classique de décomposition de signal sur un dictionnaire parcimonieux représenté par une matrice.

7.1.1 Problèmes du découpage en trames

La principale restriction du modèle classique est que la représentation des signaux en tant que vecteurs de taille fixe L_t n'est pas appropriée au traitement de signaux temporels. Il faut noter que le modèle de dictionnaire non-structuré ne pose pas de problème pour traiter des données intrinsèquement sous la forme de suite de vecteurs \underline{s}_q de taille L_t , où la l^e composante est toujours de la même nature quel que soit le numéro q du vecteur. Le problème survient en revanche lorsque les vecteurs sont obtenus par un processus ayant une dimension arbitraire, comme par exemple le tramage de signaux temporels que nous expliquons ici.

La technique classique est de découper le signal temporel (d'une durée d'une minute, par exemple) en trames de longueur L égale à la longueur des atomes (par exemple de l'ordre de 100 ms). Le tramage doit être tel que le signal peut être reconstruit parfaitement à partir des trames.

Nous expliquons le principe classique de tramage sur la figure 7.1. Les tranches de signal, représentées par les rectangles noirs, sont multipliées par une fenêtre d'analyse w_a pour donner les Q trames \underline{s}_q . Deux tranches de signal successives sont décalées de D échantillons. Les trames, regroupées dans la matrice $\underline{\mathbf{S}}$, sont traitées, par exemple pour apprendre un dictionnaire puis être décomposées sur ce dictionnaire. Les trames résultant du traitement, notées $\tilde{\underline{s}}_q$, sont ensuite multipliées par une fenêtre de synthèse w_s , puis additionnées pour construire le signal résultant. Cette technique est appelée "overlap-add" (OLA).

Le schéma de tramage repose sur la condition de reconstruction parfaite, qui relie la fenêtre d'analyse w_a , la fenêtre de synthèse w_s et le décalage entre trames de signal D :

$$\sum_{q=0}^{Q-1} w_a(l - q \cdot D) \cdot w_s(l - q \cdot D) = 1, \forall l \quad (7.1)$$

Cette condition est vérifiée par exemple pour des fenêtres d'analyse et de synthèse identiques et égales à une demi-arche de cosinus, à condition que le décalage D soit égal à la moitié de la longueur L_t des trames.

Pour ce décalage d'une demi-trame, le dictionnaire a une mauvaise précision temporelle. En effet, les motifs récurrents du signal n'ont aucune raison d'être strictement alignés avec le début des trames, qui sont découpées de façon régulière. Ce problème est exposé schématiquement sur la figure 7.2. Les atomes appris à partir des trames ne peuvent donc pas refléter fidèlement ces motifs, et seule la tendance moyenne ressort, sans conserver les détails.

Il est possible d'appliquer le même schéma pour un décalage d'un échantillon entre les trames, ce qui permettrait d'être aligné à l'échantillon près avec les motifs du signal,

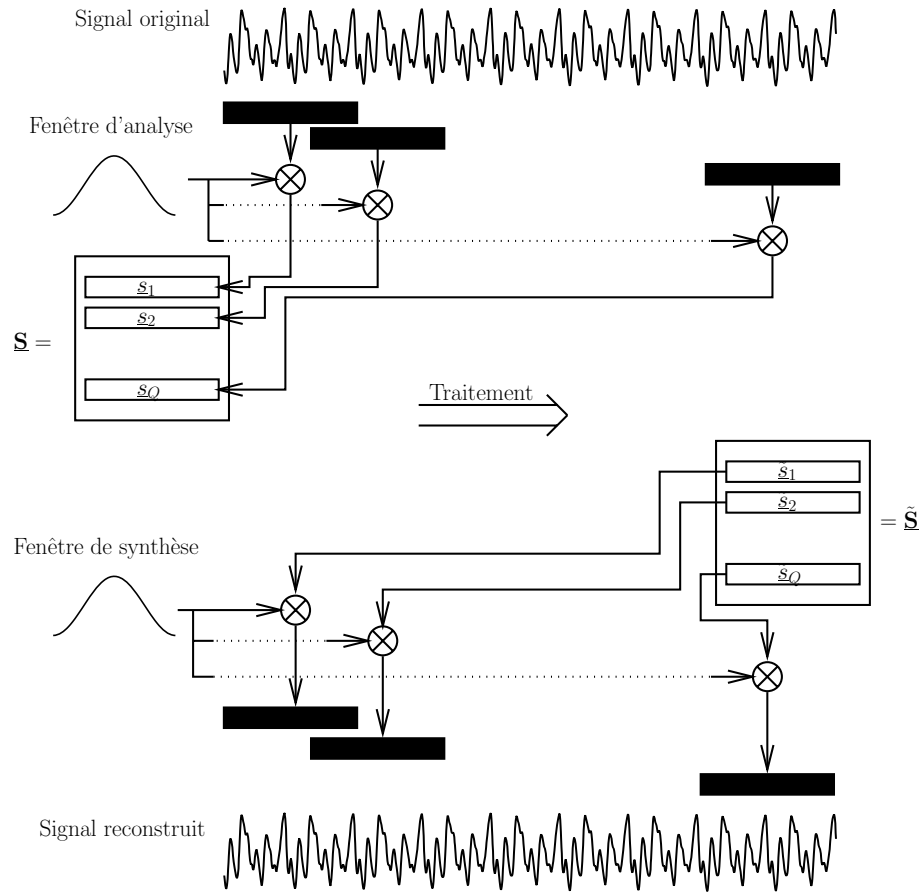


FIG. 7.1 – Principe de tramage d’un signal temporel monodimensionnel (schéma inspiré de [Bad]).

mais alors chaque trame correspond à un seul échantillon du signal, et il faut donc L (longueur du signal) trames de longueur L_t (longueur d’une trame), soit une matrice \mathbf{S} de taille $L \times L_t$, ce qui est énorme pour un long signal.

7.1.2 Problème de la redondance au sein du dictionnaire appris

Le deuxième argument justifiant de modifier le modèle non-structuré concerne la redondance résiduelle au sein du dictionnaire appris. Nous avons montré sur la figure 5.2 que le dictionnaire d’atomes appris sur des images naturelles en utilisant l’algorithme d’Olshausen (décrit en section 3.5.2) contient la même information dans plusieurs atomes et peut être avantageusement factorisé.

Ainsi, le même motif se retrouve dans plusieurs atomes du dictionnaire appris, et seule la localisation de ce motif au sein de l’atome change. Une deuxième forme de redondance tient au fait que ce même motif se retrouve dans plusieurs atomes, et cette

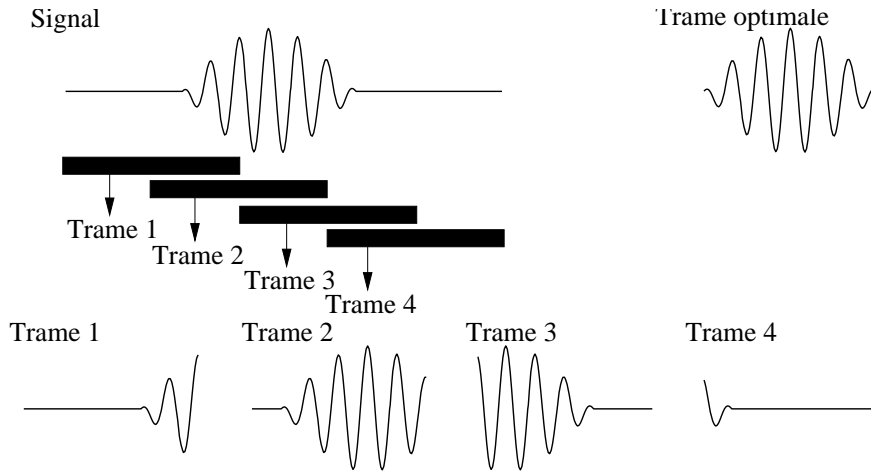


FIG. 7.2 – Des trames décalées de moitié (trames 1 à 4) ne permettent pas de s’aligner précisément avec le motif du signal, comme le ferait la trame optimale (en haut à droite).

fois seule la direction angulaire du motif change. Dans le même esprit, le même motif peut se retrouver dans plusieurs atomes avec un déphasage différent.

Le dictionnaire peut donc être factorisé par un dictionnaire de motifs élémentaires et un ensemble de paramètres décrivant la localisation, l’angle et la phase.

7.2 Modèle de signaux

Jusqu’ici, nous avons utilisé des signaux monocanaux \mathbf{s} modélisés par des vecteurs horizontaux de longueur L . Nous proposons maintenant d’utiliser des signaux de dimension infinie, et nous détaillons dans les deux sections qui suivent le modèle des signaux temporels monocanaux et des signaux temporels multicanaux, et dans la troisième section, nous donnons quelques conventions qui faciliteront le passage aux images et aux vidéos pour le lecteur voulant étendre le modèle à d’autres types de signaux.

7.2.1 Signaux temporels monocanaux

Nous modélisons désormais les signaux temporels mono-dimensionnels par des vecteurs horizontaux de dimension infinie, notés \underline{s} , et qui comprennent les échantillons du signal situés dans le temps entre $-\infty$ et $+\infty$:

$$\underline{s} = [\cdots \underline{s}(l-1) \underline{s}(l) \underline{s}(l+1) \cdots] . \quad (7.2)$$

L’ensemble de signaux temporels monocanaux de dimension infinie est noté \mathbb{S} .

En pratique, les signaux que nous traitons expérimentalement ont un support fini en dehors duquel les échantillons sont de valeur zéro. Nous notons $R_{T,L}$ la fonction qui

extrait d'un signal \underline{s} sa restriction au support situé à l'instant T et de longueur L :

$$\mathbb{S} \longrightarrow \mathbb{R}^{L \times 1}$$

$$R_{T,L} : \underline{s} \longrightarrow \mathbf{s} = \begin{bmatrix} \underline{s}(T) \\ \vdots \\ \underline{s}(T+L-1) \end{bmatrix} \quad (7.3)$$

Cette fonction restriction (par abus de langage, nous l'appellerons restriction, et non plus restriction au support) a évidemment une fonction adjointe, notée $R_{T,L}^*$, qui consiste à placer les composantes du vecteur \mathbf{s} entre l'instant T et l'instant $T+L-1$ d'un signal \underline{s} dont les autres échantillons sont nuls.

Nous définissons le produit scalaire entre deux signaux monocanaux \underline{s}_1 et \underline{s}_2 par :

$$\langle \underline{s}_1, \underline{s}_2 \rangle = \sum_l \underline{s}_1(l) \cdot \underline{s}_2(l) . \quad (7.4)$$

7.2.2 Signaux temporels multicanaux

Les signaux temporels multicanaux permettent par exemple de modéliser les mélanges sonores obtenus en sortie d'une rangée de microphones. Ils sont notés $\underline{\mathbf{S}}$, et peuvent être représentés par leurs M canaux (des signaux monocanaux) $\underline{s}_m \in \mathbb{S}$ sous la forme d'un vecteur de signaux :

$$\underline{\mathbf{S}} = \begin{bmatrix} \underline{s}_1 \\ \vdots \\ \underline{s}_M \end{bmatrix} . \quad (7.5)$$

L'ensemble de signaux temporels multicanaux de dimension infinie à M canaux est noté \mathbb{S}^M , et $\underline{\mathbf{S}}$ est un vecteur de cet espace.

Le signal multicanal $\underline{\mathbf{S}}$ peut être vu comme une matrice, dont les lignes sont les canaux, et dont les colonnes donnent la valeur des canaux à un instant donné.

Comme pour les signaux monocanaux, un signal multicanal a un support fini, qui est défini par les bornes des supports de ses canaux. En dehors de ce support temporel, tous les échantillons de tous les canaux sont nuls.

Nous définissons la fonction restriction $R_{T,L}$ par la succession des restrictions des canaux :

$$\mathbb{S}^M \longrightarrow \mathbb{R}^{M \cdot L \times 1}$$

$$R_{T,L} : \underline{\mathbf{S}} \longrightarrow \mathbf{s} = \begin{bmatrix} R_{T,L}[\underline{s}_1] \\ \vdots \\ R_{T,L}[\underline{s}_M] \end{bmatrix} \quad (7.6)$$

Nous avons choisi un ordre d'indexation particulier pour la fonction de restriction, mais les canaux auraient pu être entrelacés, ou rangés selon un ordre encore différent.

Il faut bien noter que **la restriction n'est pas une matrice de M lignes par L colonnes, mais bien un vecteur colonne de taille $M \cdot L$** . Cette convention permettra

de décrire les méthodes de décomposition de signal et d'apprentissage de motifs sous des notations générales sans *a priori* sur le caractère monocanal ou multicanal des signaux.

La fonction adjointe de $R_{T,L}$ est obtenue en plaçant les L premiers éléments de \mathbf{s} sur le premier canal de $\underline{\mathbf{S}}$, à partir de l'instant T , les L éléments suivants sur le deuxième canal, toujours à partir de l'instant T , et ainsi de suite jusqu'au dernier canal.

Nous définissons le produit scalaire entre deux signaux multicanaux $\underline{\mathbf{S}}_1$ et $\underline{\mathbf{S}}_2$ par :

$$\langle \underline{\mathbf{S}}_1, \underline{\mathbf{S}}_2 \rangle = \sum_{m=1}^M \sum_l \underline{s}_{1,m}(l) \cdot \underline{s}_{2,m}(l) . \quad (7.7)$$

7.2.3 Signaux image et vidéo

Bien que nous ne travaillions pas sur les images ni les vidéos, nous donnons un aperçu de la façon dont elles peuvent être traitées dans notre formalisme. Nous n'introduisons pas de nouvelle notation, et nous désignons les images par $\underline{s}(x, y) \in \mathbb{I}$ et les vidéos par $\underline{s}(x, y, l) \in \mathbb{V}$, où x et y sont l'abscisse et l'ordonnée, et l est le temps. Ces trois coordonnées sont à valeur dans \mathbb{Z} . Si de nombreuses recherches ont été effectuées pour apprendre des dictionnaires sur des images (à partir de 1995, avec Bell et Sejnowski [BS95]), les images animées, ou vidéos, n'ont été étudiées dans ce cadre que plus récemment par van Hateren et Ruderman en 1998 [vHR98].

Comme pour les autres types de signaux, nous supposons que les images et les vidéos ont un support fini sur chacune de leurs dimensions, et les fonctions de restriction transforment l'image ou la vidéo en un vecteur colonne (à noter, nous désignons les coordonnées de démarrage de la restriction par $T = (T_x, T_y)$ et la longueur par $L = (L_x, L_y)$ pour les images, avec l'ajout de T_t et L_t pour la vidéo) :

$$R_{T,L} : \mathbb{I} \longrightarrow \mathbb{R}^{L_x \cdot L_y \times 1} \quad \underline{s} \longrightarrow \mathbf{s} = \begin{bmatrix} \underline{s}(T_x, T_y) \\ \underline{s}(T_x + 1, T_y) \\ \vdots \\ \underline{s}(T_x + L_x - 1, T_y) \\ \underline{s}(T_x, T_y + 1) \\ \vdots \\ \underline{s}(T_x + L_x - 1, T_y + L_y - 1) \end{bmatrix} \quad (7.8)$$

Pour la vidéo, la fonction restriction est :

$$R_{T,L} : \mathbb{V} \longrightarrow \mathbb{R}^{L_x \cdot L_y \cdot L_t \times 1} \quad \underline{s} \longrightarrow \mathbf{s} = \begin{bmatrix} R_{\underline{s}(\cdot, \cdot, T_t)} \\ R_{\underline{s}(\cdot, \cdot, T_t + 1)} \\ \vdots \\ R_{\underline{s}(\cdot, \cdot, T_t + L_t - 1)} \end{bmatrix} \quad (7.9)$$

Encore une fois, la fonction restriction inverse est facile à obtenir. Il suffit de placer les éléments du vecteur \mathbf{s} dans l'image ou dans la vidéo selon l'ordre d'indexation, à partir des coordonnées de démarrage T_x, T_y , ainsi que T_t pour la vidéo.

Il faut noter que les images RGB (rouge-vert-bleu), ou YUV (luminance-chrominance), qui ont trois couches, sont aux images ce que les signaux multicanaux sont aux signaux monocanaux, et peuvent être traités en étendant le modèle de signaux de la même manière.

Nous définissons le produit scalaire entre deux images \underline{s}_1 et \underline{s}_2 par :

$$\langle \underline{s}_1, \underline{s}_2 \rangle = \sum_x \sum_y \underline{s}_1(x, y) \cdot \underline{s}_2(x, y), \quad (7.10)$$

et entre deux vidéos par :

$$\langle \underline{s}_1, \underline{s}_2 \rangle = \sum_x \sum_y \sum_l \underline{s}_1(x, y, l) \cdot \underline{s}_2(x, y, l). \quad (7.11)$$

D'une façon générale, tout type de signal discret ayant un support fini sur chacune de ses dimensions peut être traité dans le cadre du modèle que nous exposons dans cette thèse. Rappelons que les signaux ayant un support fini sont de carré intégrable et que l'espace peut être muni d'un produit scalaire.

Dans la section suivante, nous exposons le modèle d'atome généré par un motif translaté, correspondant aux modèles de signaux décrits dans cette section.

7.3 Modèle préliminaire : dictionnaire invariant par translation

Avant de présenter le modèle général de dictionnaire, qui est la principale contribution théorique de cette thèse, nous présentons un premier modèle plus simple de dictionnaire invariant par translation, qui a déjà été proposé dans la littérature pour l'apprentissage [Ols00, SL05, BD06].

Dans ce modèle simplifié, nous introduisons la séparation de la notion classique d'atome en deux notions distinctes : l'**atome** et le **motif**. Dans cette section, pour plus de simplicité, les signaux sont limités aux signaux temporels monocanaux.

7.3.1 Définition d'un atome et d'un motif

Un atome est un signal temporel monocanal, du même type que les signaux traités, et une décomposition atomique parcimonieuse consiste à représenter le signal comme somme de peu d'atomes :

$$\underline{s} = \sum_{i=1}^I \underline{d}_i + \underline{r}. \quad (7.12)$$

Dans le modèle d'invariance par translation, un **atome** $\underline{d} \in \mathbb{S}$ est constitué d'un **motif** $\underline{m}_k \in \mathbb{S}$, choisi dans un dictionnaire \mathcal{M} de motifs, positionné à un **instant** $t \in \mathbb{Z}$

donné, et multiplié par une **amplitude** $c \in \mathbb{R}$ donnée. La restriction du motif \underline{m} à son support est noté \mathbf{m} et est de taille L_m . Pour désigner le motif, nous utiliserons selon les cas :

- \mathbf{m} : le vecteur de L_m échantillons, ou
- $\underline{m} = R_{0,L_m}^*[\mathbf{m}]$: le signal dont le support, placé à l’instant zéro, correspond au motif \mathbf{m} .

Un atome \underline{d} s’écrit alors, pour tout l :

$$\underline{d}(l) = c \cdot \underline{m}_k(l - t) . \quad (7.13)$$

Il faut bien noter que contrairement au modèle de décomposition en vigueur jusqu’à maintenant, **l’amplitude est désormais incluse dans l’atome**. Le signal se décompose comme :

$$\underline{s}(l) = \sum_{i=1}^I c_i \cdot \underline{m}_{k_i}(l - t_i) + \underline{r}(l) . \quad (7.14)$$

Le principe consistant à pouvoir placer les motifs à n’importe quel instant est souvent appelé invariance par translation.

7.3.2 État de l’art

Un modèle de dictionnaire invariant par translation du type (7.14) a été proposé en 2003 par Olshausen. Les signaux traités ne sont pas des signaux mono-dimensionnels, mais des vidéos, et les motifs sont des vidéos très courtes (une succession de 7 images) localisées à un instant donné, et convoluées avec un filtre d’amplitudes.

Olshausen a proposé un algorithme pour décomposer une vidéo sur ces motifs, et un algorithme pour apprendre les motifs \underline{m}_k à partir du signal. Il faut noter que son modèle est invariant par translation dans le temps, mais pas au sein des images, qui sont découpées en trames de façon arbitraire.

Smith et Lewicki ont proposé le modèle (7.14) en 2005, pour traiter spécifiquement des signaux temporels mono-dimensionnels et proposent un algorithme pour apprendre les motifs. Blumensath et Davies ont également proposé un algorithme d’apprentissage en 2006.

Nous présentons tous les algorithmes d’apprentissage dans la section 9.1.

7.3.3 Schéma explicatif

La figure 7.3 décrit le modèle de décomposition parcimonieuse du signal \underline{s} sur un dictionnaire de motifs \underline{m}_k . Cela correspond à la formule :

$$\underline{s} = c_1 \cdot R_{t_1,L_{m_1}}^*[\mathbf{m}_1] + c_2 \cdot R_{t_2,L_{m_1}}^*[\mathbf{m}_1] + c_3 \cdot R_{t_3,L_{m_1}}^*[\mathbf{m}_2] + c_4 \cdot R_{t_4,L_{m_4}}^*[\mathbf{m}_3] + \underline{r} . \quad (7.15)$$

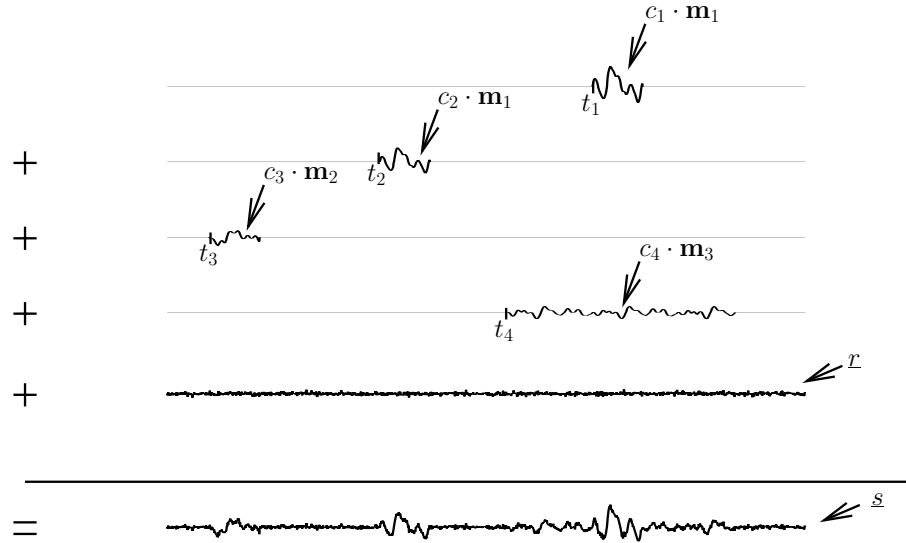


FIG. 7.3 – Modèle de la décomposition parcimonieuse d'un signal sur un dictionnaire invariant par translation.

7.3.4 Fonction de translation

De façon plus formelle, le modèle de génération d'un atome (7.13) peut s'écrire comme :

$$\underline{d} = A_c [T_t [\underline{m}]] , \quad (7.16)$$

où A_c est une fonction d'amplification, et T_t est un fonction de translation, qui sont définis par :

$$A_c : \underline{s} \longrightarrow c \cdot \underline{s} \quad (7.17)$$

et

$$T_t : \underline{s} \longrightarrow \underline{\delta}_t \star \underline{s} \quad (7.18)$$

avec $\underline{\delta}_t$ le signal de Dirac :

$$\underline{\delta}_t(l) = \begin{cases} 1 & \text{si } l = t \\ 0 & \text{si } l \neq t \end{cases} \quad (7.19)$$

et \star l'opérateur de convolution.

Cette formalisation peut paraître lourde, mais elle permet d'expliquer le processus de création des atomes comme une succession d'étapes, et de préparer aux fonctions de déformation présentées dans la section 7.4.

7.4 Modèle structuré : dictionnaire généré par déformation linéaire de motifs

Nous avons présenté dans la section précédente la génération d'atomes par le placement du support des motifs du dictionnaire à des instants donnés. Nous avons également présenté les atomes comme des motifs déformés par deux types de fonctions : une fonction de translation et une fonction d'amplification.

En s'inspirant de cette deuxième explication, nous pouvons étendre la définition des atomes en permettant l'utilisation d'autres fonctions de déformation à appliquer aux motifs. Ainsi, en contrepartie de quelques paramètres supplémentaires, l'ajout d'une déformation permet la génération de tout un ensemble d'atomes supplémentaires.

Le modèle d'atome généré par déformation linéaire d'un motif est :

$$\underline{d} = F_{\boldsymbol{\psi}, \phi}[\underline{m}] , \quad (7.20)$$

où

- $\underline{m} \in \mathbb{S}$ est le motif générateur, issu d'un dictionnaire \mathcal{M} de K motifs,
- $\underline{d} \in \mathbb{S}$ est l'atome généré et
- $F_{\boldsymbol{\psi}, \phi}$ est la fonction de déformation, paramétrée par $\boldsymbol{\psi}$ et ϕ .

Avant de donner des exemples concrets de telles déformations, précisons quelques points. Tout d'abord, nous imposons que les motifs $\underline{m}_k \in \mathcal{M}$, et non plus les atomes, soient de norme unité ($\|\underline{m}_k\|_2 = 1$). Cette contrainte a pour but l'identification des motifs dans les algorithmes de décomposition et d'apprentissage de dictionnaire. Pour l'apprentissage de dictionnaire, nous verrons notamment que l'algorithme K-SVD génère des motifs unitaires via la DVS.

Nous nous restreignons aux fonctions $F_{\boldsymbol{\psi}, \phi}$ linéaires par rapport au signal \underline{m} et par rapport aux paramètres $\boldsymbol{\psi}$, en raison des propriétés favorables des fonctions linéaires, telles que l'existence d'un adjoint, qui sera très utile pour les algorithmes présentés par la suite.

Les paramètres $\boldsymbol{\psi} = [\psi_1, \dots, \psi_{I_\psi}]^T \in \mathbb{R}^{I_\psi \times 1}$ sont continus et permettent d'exprimer la fonction $F_{\boldsymbol{\psi}, \phi}$ comme :

$$F_{\boldsymbol{\psi}, \phi} = \sum_{i=1}^{I_\psi} \psi_i \cdot F_\phi^i , \quad (7.21)$$

où les I_ψ fonctions F_ϕ^i sont des fonctions linéaires de \mathbb{S} dans \mathbb{S} . Ces fonctions forment une famille libre afin de pouvoir caractériser les paramètres ψ_i . En regroupant ces fonctions F_ϕ^i dans le vecteur \mathbf{F}_ϕ , que nous appelons base de fonctions, le modèle peut s'écrire sous forme vectorielle comme :

$$\underline{d} = \boldsymbol{\psi}^T \cdot \mathbf{F}_\phi[\underline{m}] . \quad (7.22)$$

La base de fonctions \mathbf{F}_ϕ est paramétrée par l'indice (discret) $\phi = (\phi_1, \dots, \phi_{J_\phi}) \in \mathbb{Z}^{1 \times J_\phi}$, qui permet de choisir parmi un ensemble discret de bases de fonctions $\mathcal{F} = \{\mathbf{F}_\phi\}_\phi$.

Présentons maintenant des exemples de déformations linéaires entrant dans le cadre du modèle de dictionnaire structuré. Ce sont les déformations linéaires qui nous semblent les plus intéressantes à appliquer à des signaux temporels, et notamment musicaux.

7.5 Déclinaison de diverses déformations

Nous proposons dans cette section quelques déformations que nous jugeons intéressantes pour le traitement des signaux temporels. Elles peuvent être enchaînées par composition, selon une règle que nous exposons dans la section 7.6, consacrée aux propriétés des fonctions de déformation.

7.5.1 Modification d'amplitude

La première des déformations est la plus simple, et correspond à la définition classique des atomes. Il s'agit de la modification d'amplitude A_c définie par :

$$A_c : \underline{S} \longrightarrow \underline{S} \\ \underline{m} \longrightarrow c \cdot \underline{m} \quad (7.23)$$

Il s'agit uniquement de multiplier le motif par un scalaire c . Le paramètre ψ est égal à c , et il n'y a pas de paramètre ϕ . La figure 7.4 montre diverses fonctions d'amplifications. Cette figure très simple permet de faire le parallèle avec les autres déformations.

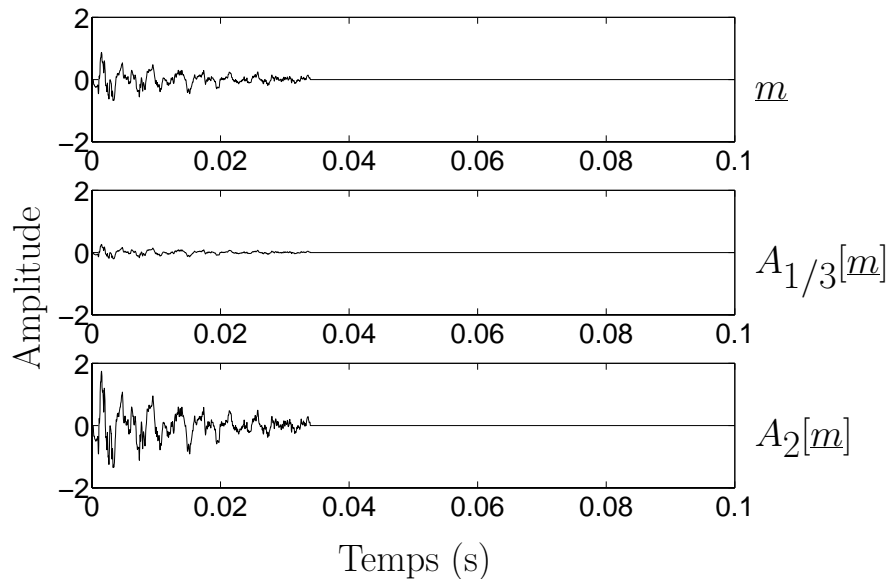


FIG. 7.4 – Déformation d'amplitude. Haut : le motif original, milieu : le motif divisé par 3, bas : le motif multiplié par 2.

L'adjoint de A_c est A_c elle-même.

7.5.2 Translation

La translation T_t a déjà été exposée à la section 7.3. Elle consiste à décaler le motif dans le temps de t échantillons :

$$T_t : \underline{m} \longrightarrow T_t[\underline{m}] \quad (7.24)$$

$$\text{avec } T_t[\underline{m}](l) = \underline{m}(l - t) \quad (7.25)$$

Le paramètre ϕ correspond au paramètre de translation $t \in \mathbb{Z}$, et il n'y a pas de paramètre ψ . La figure 7.5 montre diverses translations.

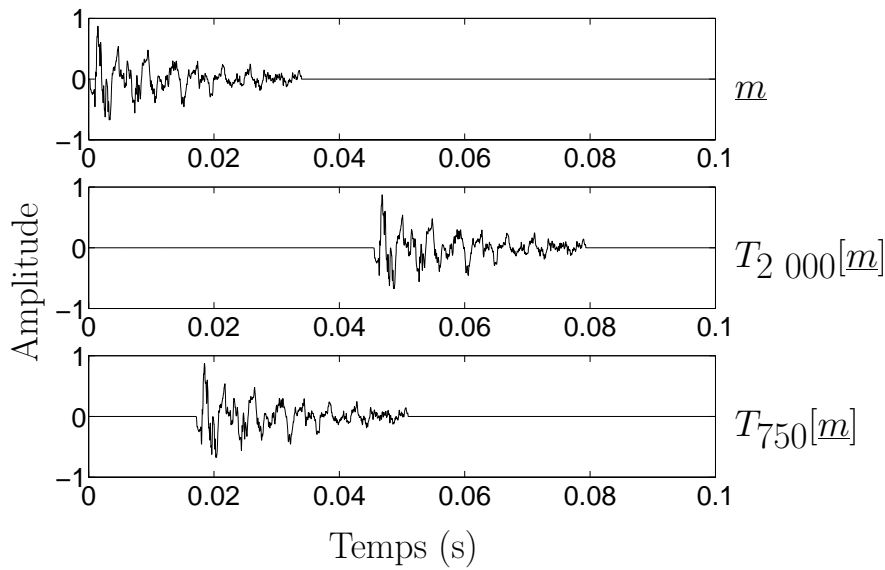


FIG. 7.5 – Translation du motif. Haut : le motif original, milieu : le motif translaté de 2 000 échantillons, soit 45 ms (fréquence d'échantillonnage de 44 100 Hz), bas : le motif translaté de 750 échantillons, soit 17 ms.

L'adjoint de la T_t est T_{-t} , c'est-à-dire la translation du même nombre d'échantillons dans le sens opposé.

7.5.3 Convolution par un filtre RIF

Le modèle de déformation linéaire comprend une déformation très importante : la convolution du motif \underline{m} par un filtre à réponse impulsionnelle finie \underline{e} . La modification d'amplitude en est un cas particulier. Elle est définie par :

$$C_e : \underline{m} \longrightarrow \underline{e}^T \cdot \mathbf{C}[\underline{m}] = R_{0, I_e}^*[\underline{e}] \star \underline{m} \quad (7.26)$$

Le paramètre ψ est le vecteur \mathbf{e} de taille I_e , et les fonctions de base sont $C^i = T_{i-1}$, les translations de T_0 à T_{I_e-1} . Il n'y a pas de paramètre ϕ . Notons $\underline{e} = R_{0,I_e}^*[\mathbf{e}]$ le signal dont la restriction au support, démarrnant à l'instant 0, est le vecteur \mathbf{e} . La déformation peut alors s'écrire de façon équivalente comme :

$$C_{\mathbf{e}}[\underline{m}] = \mathbf{e}^T \cdot \mathbf{C}[\underline{m}] = \sum_{i=1}^{I_e} e_i \cdot T_{i-1}[\underline{m}] , \quad (7.27)$$

ou comme

$$C_{\underline{e}}[\underline{m}](l) = [\underline{e} \star \underline{m}](l) = \sum_{n=-\infty}^{\infty} \underline{e}(n) \cdot \underline{m}(l-n) . \quad (7.28)$$

Nous notons la déformation $C_{\mathbf{e}}[\underline{m}]$ ou $C_{\underline{e}}[\underline{m}]$ de façon équivalente, où \mathbf{e} est la restriction au support de \underline{e} .

Il faut noter que les propriétés de la convolution font que $C_{\mathbf{e}}[\underline{m}] = C_{\underline{m}}[\underline{e}]$. Cela signifie alors que le signal \underline{e} est un signal d'excitation, filtré par le signal \underline{m} . Nous développerons cet aspect dans la section 7.5.6.

Notez que nous avons choisi la lettre \underline{e} en raison de l'interprétation de ce signal en tant qu'excitation. En aucun cas cette lettre ne représente une erreur, contrairement à la notation souvent utilisée.

La figure 7.6 montre diverses déformations par convolution du même motif \underline{m} .

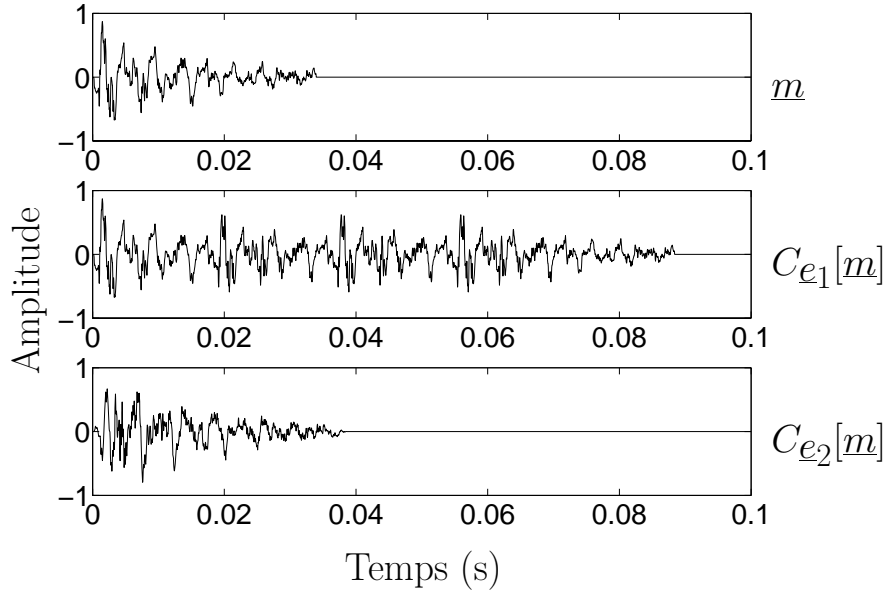


FIG. 7.6 – Déformation par filtrage. Haut : le motif original, milieu : le motif filtré par un train d'impulsions espacées de 500 échantillons, soit 11.3 ms : $\underline{e}_1 = \sum_{i=0}^3 1 \cdot \delta_{500-i}$, bas : le motif filtré par un bruit blanc \underline{e}_2 de 200 échantillons, soit 4.5 ms.

Nous introduisons la notion de signal inversé, noté \underline{s}^* et défini par :

$$\underline{s}^*(l) = \underline{s}(-l), \forall l \in \mathbb{Z}. \quad (7.29)$$

L'adjoint de la déformation $C_{\underline{e}}$ est la convolution par le signal inversé $C_{\underline{e}^*}$. Pour rester cohérent avec les conventions, l'adjoint s'écrit aussi $C_{\underline{e}}^* = T_{-I_e} \circ C_{\underline{e}^*}$, où \underline{e}^* est le vecteur \underline{e} retourné.

La modification d'amplitude A_c est un cas particulier de la déformation $C_{\underline{e}}$, où $\underline{e} = c$. Cela revient à convoluer le motif avec le signal \underline{e} nul partout, sauf en $l = 0$, où il vaut c ($\underline{e} = c \cdot \underline{\delta}_0$).

Le modèle de filtre présenté correspond à un filtre RIF dont les I_e coefficients sont libres. Afin de modéliser des filtres particuliers, ces coefficients peuvent être contraints. Par exemple, la structure suivante permet de modifier l'équilibre entre hautes et basses fréquences du motif :

$$F_{\psi} = \psi_1 \cdot T_0 + \psi_2 \cdot (T_{-1} + T_1) \quad (7.30)$$

$$= T_{-1} \circ C_{\underline{e}} \text{ avec } \underline{e} = \begin{bmatrix} \psi_2 \\ \psi_1 \\ \psi_2 \end{bmatrix}, \quad (7.31)$$

où le paramètre ψ_1 modifie le poids des basses fréquences du motif, tandis que le paramètre ψ_2 modifie le poids des hautes fréquences.

7.5.4 Déformation d'un paramètre de phase

Afin de modéliser plus finement les signaux temporels, il nous semble opportun de modéliser la phase du motif, en plus de sa forme d'onde réelle. Nous pouvons faire le parallèle avec les atomes de Gabor, qui sont caractérisés non seulement par leur localisation temporelle et leur fréquence, mais aussi par leur phase initiale. Cependant, si pour les atomes de Gabor, un seul paramètre de phase suffit, pour les signaux comprenant plusieurs fréquences, il convient de définir ce que nous appelons phase.

Pour décaler temporellement le motif d'un nombre non-entier d'échantillons, alors il faut autant de paramètres de phase que la longueur du support. En effet, un décalage temporel t correspond à la multiplication des coefficients de Fourier de la restriction au support par un facteur $e^{2i\pi ft}$ qui dépend de la fréquence f . Cette déformation n'est pas linéaire par rapport au paramètre t , et n'entre donc pas dans le cadre de notre modèle.

Ici, nous proposons un autre type de déformation de la phase, correspondant plutôt au "déroulement" des composantes sinusoïdales sur le support. Pour cela, nous définissons la déformation d'un paramètre de phase P_{ψ} par :

$$P_{\psi} : \begin{array}{ccc} \mathbb{S} & \longrightarrow & \mathbb{S} \\ \underline{m} & \longrightarrow & \psi_1 \cdot \underline{m} + \psi_2 \cdot H[\underline{m}] \end{array} \quad (7.32)$$

où $H[\underline{m}]$ est la partie imaginaire de la transformée de Hilbert du motif \underline{m} . Les signaux \underline{m} et $H[\underline{m}]$ sont en quadrature de phase, ce qui signifie qu'ils sont décalés d'un angle de phase de $\pi/2$. Nous appelons $H[\underline{m}]$ le motif en quadrature.

Notons \mathbf{m} la restriction de \underline{m} à son support, de longueur L , et $\check{\mathbf{m}}$, la restriction de $H[\underline{m}]$ à son support, de même taille. La déformation H peut être définie, dans le domaine de Fourier, par :

$$\hat{\mathbf{m}}(f) = \begin{cases} 0 & , \text{ si } f = 0 \\ -i \cdot \hat{\mathbf{m}}(f) & , \text{ si } f \in \llbracket 1, L/2 - 1 \rrbracket \\ 0 & , \text{ si } f = L/2 \\ +i \cdot \hat{\mathbf{m}}(f) & , \text{ si } f \in \llbracket L/2 + 1, L - 1 \rrbracket \end{cases} \quad (7.33)$$

où f désigne la fréquence, dans le cas où la taille du support de \underline{m} , L , est paire. Si L est impair, la troisième ligne, qui correspond à la fréquence de Nyquist dans le cas pair, disparaît, et les bornes $L/2 - 1$ et $L/2 + 1$ sont remplacées par $(L - 1)/2$ et $(L + 1)/2$.

Dans le domaine temporel, elle est définie par :

$$\check{\mathbf{m}} = \mathbf{H} \cdot \mathbf{m} \text{ avec } \mathbf{H} = \begin{bmatrix} & n & & & \\ & \downarrow & & & \\ & \vdots & & & \\ & \frac{2}{L} \sum_{k=1}^{L/2-1} \sin\left(\frac{2\pi k(n-l)}{L}\right) & \dots & & \\ & & & & \leftarrow l \end{bmatrix} \quad (7.34)$$

Nous montrons sur la figure 7.7 le même motif \underline{m} modifié pour différents paramètres de phase $\varphi = \arctan\left(\frac{\psi_2}{\psi_1}\right)$ et une amplitude $\|\boldsymbol{\psi}\|_2 = 1$. Il apparaît que l'enveloppe, qui correspond aux basses fréquences du motif, reste à peu près la même quelque soit le paramètre de phase, tandis que les hautes fréquences sont décalés dans le temps, proportionnellement à la fréquence.

Il faut noter que cette déformation correspond strictement à un déphasage si \underline{m} est une fréquence pure.

La déformation H annule les composantes moyenne et de Nyquist. Si le motif est à l'origine de moyenne nulle, et dans le cas de L pair, de composante fréquentielle de Nyquist nulle, alors la base de fonctions \mathbf{P} est unitaire, et conserve la norme.

L'adjoint de la déformation $P_{\boldsymbol{\psi}}$ correspond à l'angle de phase opposé : $P_{\boldsymbol{\psi}}^* = P_{\boldsymbol{\psi}^*}$,

avec $\boldsymbol{\psi}^* = \begin{bmatrix} \psi_1 \\ -\psi_2 \end{bmatrix}$.

7.5.5 Expansion multicanale

De façon surprenante, l'expansion multicanale, qui consiste à distribuer le motif sur différents canaux, entre également dans le cadre de ces déformations. Cette fonction de déformation permettra par la suite de faire un lien direct entre la séparation de sources et les décompositions parcimonieuses.

La fonction consiste à convoluer le motif avec un filtre différent sur chaque canal, ce qui correspond à la modélisation d'un mélange convolutif. Illustrons ce modèle avec l'exemple de l'enregistrement sonore. Le motif est émis par l'instrument, les canaux

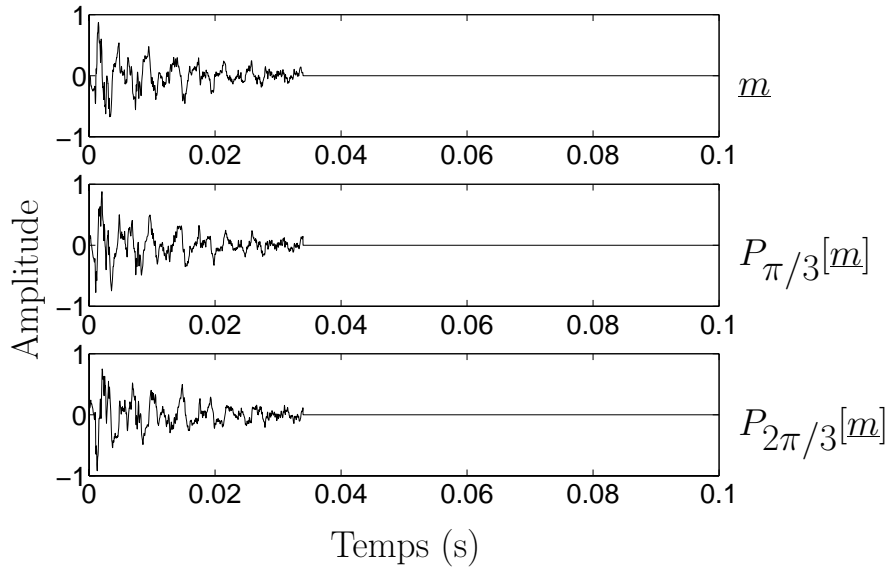


FIG. 7.7 – Déformation de phase. Haut : le motif original, milieu : le motif déformé par un paramètre de phase $\varphi = \pi/3$, bas : le motif déformé par un paramètre de phase $\varphi = 2\pi/3$.

correspondent aux enregistrements par différents microphones, et les filtres modélisent la distorsion subie par le motif pour traverser la salle et rejoindre chaque microphone.

Le modèle est le suivant :

$$E_{\mathbf{a}} : \underline{m} \longrightarrow \begin{bmatrix} C_{\mathbf{a}_1}[\underline{m}] \\ \vdots \\ C_{\mathbf{a}_M}[\underline{m}] \end{bmatrix} \quad (7.35)$$

avec \mathbf{a}_m le filtre correspondant au m^e canal, tous ces filtres étant regroupés dans le vecteur \mathbf{a} . L'expansion multicanale part de l'espace des signaux monocanaux \mathbb{S} pour arriver dans l'espace \mathbb{S}^M des signaux à M canaux.

En notant $I_{\mathbf{a}_m}$ le nombre de coefficients du filtre \mathbf{a}_m , le modèle peut s'écrire également comme :

$$E_{\mathbf{a}} = \sum_{m=1}^M \sum_{i_m=1}^{I_{\mathbf{a}_m}} a_{m,i_m} \cdot E_{i_m-1}^m, \quad (7.36)$$

où les composantes du filtre \mathbf{a}_m du canal m sont notées a_{m,i_m} , et où E_i^m correspond à

la translation du canal m de i échantillons :

$$E_i^m = \begin{bmatrix} \vdots \\ 0 \\ T_i \\ 0 \\ \vdots \end{bmatrix} \leftarrow m \quad (7.37)$$

Il faut noter que ce modèle englobe aussi les mélanges linéaires instantanés, correspondant à la simple multiplication par une amplitude sur chaque canal ($\mathbf{a}_m = c_m$), et les mélanges linéaires avec délai, où les filtres sont de simples retards, combinés avec la multiplication par une amplitude ($\underline{a}_m = c_m \cdot \underline{\delta}_{t_m}$, avec la notation équivalente sous forme de signaux).

La figure 7.9 représente le signal multicanal obtenu en simulant l'émission du motif \underline{m} dans une salle cubique et l'enregistrement avec trois microphones, placés à trois emplacements différents. La géométrie de la salle est représentée sur la figure 7.8. Notons que de très grandes dimensions ont été choisies pour la salle, afin de générer de longues réverbérations, mettant mieux en lumière le phénomène de réflexion sur les murs.

L'adjoint de la déformation est défini par :

$$E_{\mathbf{a}}^*[\underline{\mathbf{S}}] = \sum_{m=1}^M C_{\mathbf{a}_m}^*[\underline{\mathbf{s}}_m] . \quad (7.38)$$

7.5.6 Déformation par filtrage : interprétation pour la parole

La fonction de déformation par filtrage $C_{\underline{e}}$ a été exposée comme le filtrage du motif \underline{m} par un signal \underline{e} . Nous proposons de retourner le point de vue pour voir le motif \underline{m} comme un filtre (et le filtre \underline{e} comme le signal à filtrer). Dans ce cas, le signal $C_{\underline{e}}[\underline{m}]$ est obtenu par filtrage d'un signal d'excitation \underline{e} par le filtre $\underline{m} : C_{\underline{m}}[\underline{e}]$.

Ainsi, le cas particulier $A_1[\underline{m}] = C_{\underline{m}}[\underline{\delta}_0]$ correspond à la réponse impulsionnelle du filtre \underline{m} .

Ce modèle est particulièrement adapté à l'analyse des signaux de parole. Il existe deux modes principaux de génération de la parole, entraînant les sons voisés, et les sons non-voisés. Cette classification n'est pas stricte car des sons peuvent être générés selon les deux modes.

Les sons voisés (notamment les voyelles, et quelques consonnes comme 'm' ou 'b' par exemple) sont générés par des ondes glottiques, produites par la vibration des cordes vocales, qui sont modulées selon la forme du conduit vocal (gorge + bouche + lèvres). Les ondes glottiques ressemblent à un train d'impulsions dont la fréquence donne la hauteur du son. Le conduit vocal forme quant à lui le phonème prononcé. La figure 7.10 montre les formes d'ondes temporelles (haut) et spectrales (bas) du signal d'excitation (à gauche), du filtre du conduit vocal (au milieu) et du son produit (à droite).

Les sons non-voisés (consonnes comme 'p', 'ch', 's', ...) sont produits par un souffle filtré par le conduit vocal. Ce souffle est modélisé par du bruit. La figure 7.11 montre

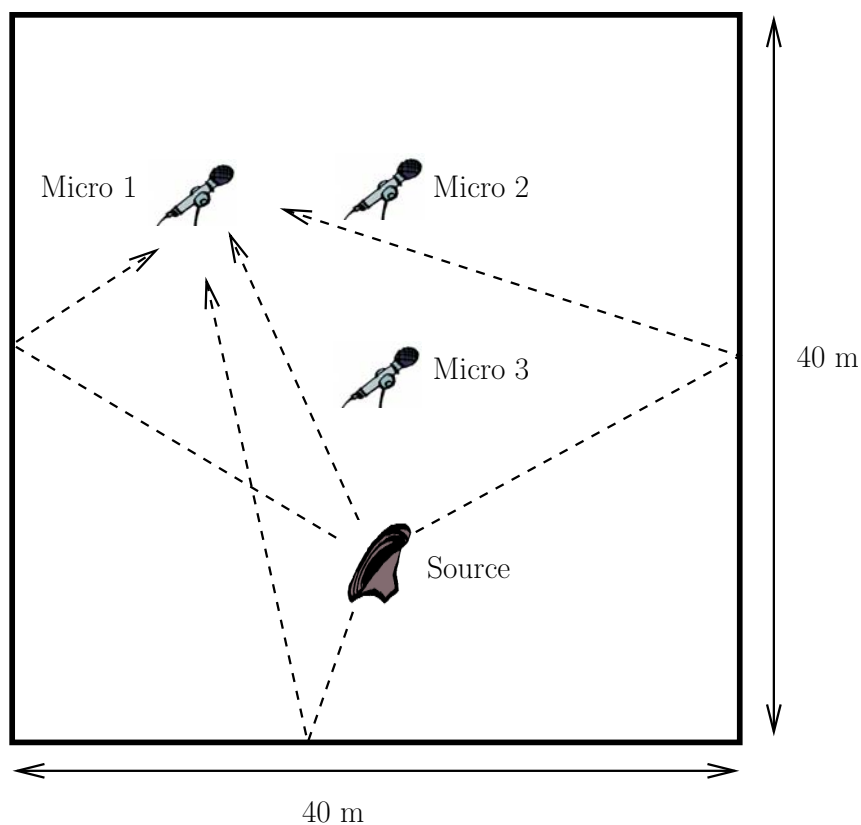


FIG. 7.8 – Géométrie de la salle, position de la source sonore et des trois microphones. Les flèches en tirets indiquent quelques uns des chemins menant les ondes de la source au micro 1. La différence de longueur entre deux chemins implique une différence de temps de parcours, et chaque réflexion sur un mur retient une partie de l'énergie de l'onde.

les formes temporelles (haut) et spectrales (bas) du signal d'excitation (à gauche) de type bruit turbulent, du filtre du conduit vocal (au milieu), qui correspond au motif m, et du son produit (à droite).

Nous pouvons remarquer sur la figure 7.10 que les deux modes ne sont pas discriminants, et que les ondes glottiques contiennent une part de bruit. C'est d'ailleurs la présence de ce bruit qui peut donner à la voix un caractère rocailleux et par exemple différencier Maria Callas de Dani.

En supposant posséder un dictionnaire de motifs qui représentent les formes de conduit vocal correspondant aux phonèmes possibles, alors la parole peut être décomposée en atomes, formés par filtrage d'un signal d'excitation (impulsions, ou bruit, selon le cas) par les filtres m.

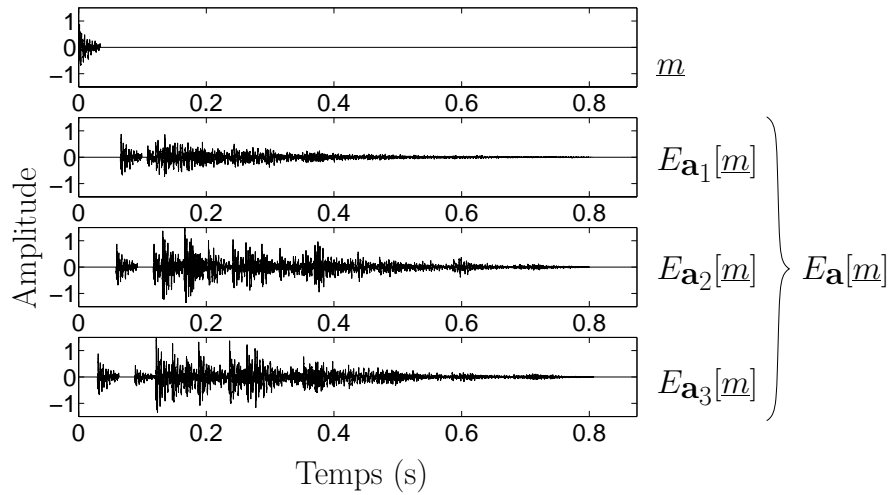


FIG. 7.9 – Expansion multicanale. Haut : le motif original, bas : le signal multicanal provenant des trois microphones représentés sur la figure 7.8. Le motif émis par la source est réfléchi sur les murs jusqu’à atténuation complète. Les réflexions se reflètent par les occurrences successives du motif, dont l’instant et l’amplitude dépendent de la géométrie de l’expérience

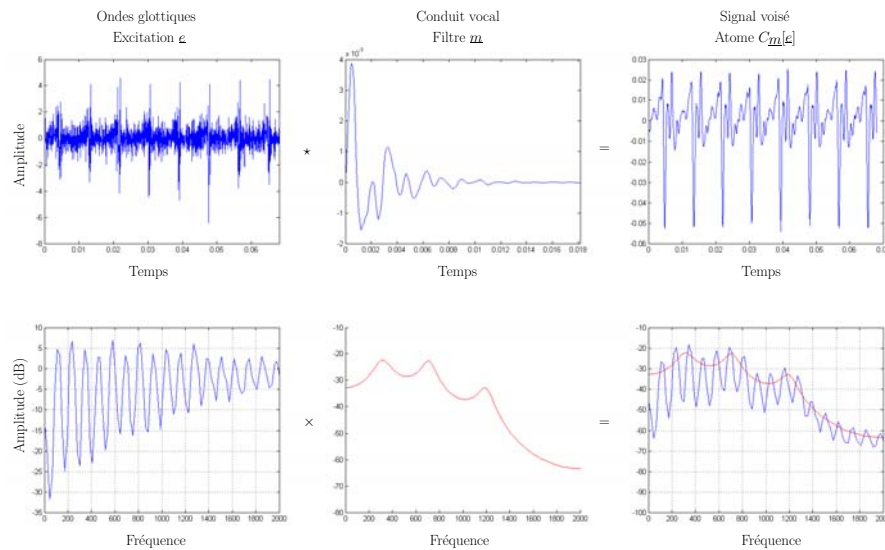


FIG. 7.10 – Formation d’un son voisé (illustration provenant de [Bad])

7.6 Propriétés des fonctions de déformation

Après avoir exposé les principales déformations entrant dans le modèle d’atomes structurés, nous détaillons dans cette section les propriétés des fonctions de déformation.

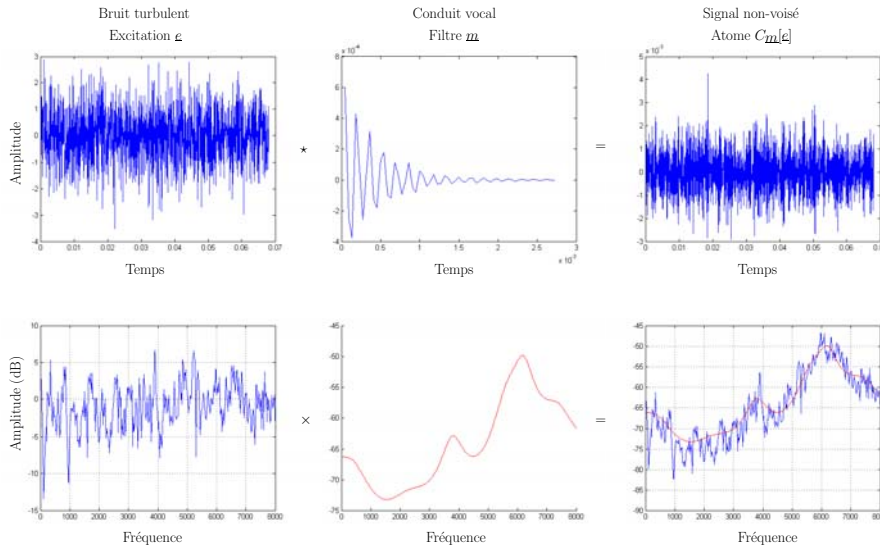


FIG. 7.11 – Formation d'un son non-voisé (illustration provenant de [Bad])

7.6.1 Paramètres continus ψ et paramètres discrets ϕ

Les paramètres de la fonction $F_{\psi,\phi}$ sont séparés en ψ et ϕ en raison de leur différence de nature.

Les paramètres ψ sont continus et ont vocation à être optimisés analytiquement lors du calcul de l'atome. Ils contiennent notamment l'amplitude de l'atome, mais peuvent servir également à modéliser l'angle d'une rotation, ou les coefficients d'un filtre. Dans le modèle précédemment exposé de l'invariance par translation, ψ ne comprend que le paramètre d'amplitude $c \in \mathbb{R}$.

Les paramètres ϕ ont eux des valeurs discrètes et leur optimisation requiert une recherche exhaustive. Ils peuvent modéliser des paramètres intrinsèquement discrets, comme le paramètre de translation $t \in \mathbb{Z}$ du modèle préliminaire, mais aussi des paramètres continus qui ont été tabulés au préalable. Par exemple, au lieu de laisser libre l'angle d'une rotation, nous pouvons le restreindre à prendre une valeur échantillonnée parmi $\{0, \pi/4, \pi/2, 3\pi/4\}$.

7.6.2 Les déformations $F_{\psi,\phi}$ sont linéaires par rapport au motif

Nous imposons aux fonctions de déformation d'être linéaires par rapport au motif \underline{m} . Le choix de cette restriction, qui permet cependant d'englober un grand nombre de déformations, est expliqué à la section 7.6.5 par l'existence d'un adjoint lorsque la déformation est linéaire. Nous verrons dans la suite que cet adjoint est très utile pour le calcul des coefficients ou pour l'apprentissage de dictionnaire.

La linéarité est également voulue dans un souci de cohérence avec le modèle classique des atomes, afin que l'opérateur de synthèse, qui génère un signal à partir des motifs et des coefficients, soit linéaire par rapport aux motifs comme il l'est par rapport aux

atomes du dictionnaire dans le modèle classique.

Rappelons que la fonction $F_{\boldsymbol{\psi},\phi}$ est linéaire si et seulement si, pour tout $\lambda \in \mathbb{R}$, et pour tous signaux $\underline{s}_1 \in \mathbb{S}$, et $\underline{s}_2 \in \mathbb{S}$:

$$F_{\boldsymbol{\psi},\phi}[\underline{s}_1 + \lambda \cdot \underline{s}_2] = F_{\boldsymbol{\psi},\phi}[\underline{s}_1] + \lambda \cdot F_{\boldsymbol{\psi},\phi}[\underline{s}_2] . \quad (7.39)$$

7.6.3 Simplifications lorsque la base de fonctions \mathbf{F}_ϕ est orthonormale

Si la base de fonctions \mathbf{F}_ϕ est orthonormale, alors les calculs qui seront exposés dans les chapitres suivants sont simplifiés. Certains problèmes sont alors résolus analytiquement, alors que dans le cas général, une stratégie d'optimisation coûteuse doit être mise en œuvre, et ne permet d'obtenir qu'une approximation de la solution.

Nous définissons qu'une fonction F est normée si et seulement si elle vérifie, pour tout $(\underline{s}_1, \underline{s}_2) \in \mathbb{S} \times \mathbb{S}$:

$$\langle F[\underline{s}_1] , F[\underline{s}_2] \rangle = \langle \underline{s}_1 , \underline{s}_2 \rangle , \quad (7.40)$$

et que deux fonctions F et G sont orthogonales si et seulement si elles vérifient, pour tout $(\underline{s}_1, \underline{s}_2) \in \mathbb{S} \times \mathbb{S}$:

$$\langle F[\underline{s}_1] , G[\underline{s}_2] \rangle = 0 . \quad (7.41)$$

La base de fonctions \mathbf{F}_ϕ est dite orthonormale si et seulement si les fonctions F_ϕ^j de cette base sont normées et orthogonales entre elles.

Il en résulte que la matrice :

$$\mathbf{G}_{k,\phi} = \begin{bmatrix} \vdots & & \\ \langle F_\phi^{j_1}[\underline{m}_k] , F_\phi^{j_2}[\underline{m}_k] \rangle & \dots & \\ \vdots & & \end{bmatrix} \leftarrow j_2 \quad (7.42)$$

qui est utilisée dans les calculs des chapitres suivants, est égale à la matrice identité \mathbf{I}_{J_ϕ} si la base de fonctions \mathbf{F}_ϕ est orthonormale, car nous avons imposé que le motif soit de norme unité.

De plus, pour la même raison, lorsque \mathbf{F}_ϕ est orthonormale, la norme du signal d'arrivée vaut :

$$\|F_{\boldsymbol{\psi},\phi}[\underline{m}]\|_2 = \|\boldsymbol{\psi}\|_2 \cdot \|\mathbf{F}_\phi[\underline{m}]\|_2 = \|\boldsymbol{\psi}\|_2 . \quad (7.43)$$

Précisons que nous n'imposons pas à toutes les bases de fonctions \mathbf{F}_ϕ d'être orthogonales.

7.6.4 Mesure de la parcimonie

De façon classique, la parcimonie d'une décomposition en atomes est mesurée à partir des amplitudes des atomes. Pour rester consistant avec les définitions de parcimonie, il suffit de définir l'amplitude c de chaque atome par sa norme $\|F_{\psi,\phi}[\underline{m}]\|_2$. La parcimonie au sens strict, qui mesure le nombre d'atomes, reste inchangée.

En revanche, pour des applications de codage, le poids de l'atome correspond au nombre de bits servant à le décrire. Dans le cas classique, il suffit de décrire le numéro k de l'atome, et l'amplitude c . Dans le modèle structuré, il faut décrire le numéro k du motif, les I_ψ paramètres contenus dans le vecteur ψ et les J_ϕ paramètres de ϕ . La façon de quantifier les différents paramètres dépend de l'implémentation du codeur. C'est un problème à part entière, que nous ne traitons pas dans ce travail.

7.6.5 Propriété fondamentale des déformations linéaires : l'existence d'un adjoint

Nous avons imposé aux déformations $F_{\psi,\phi}$ d'être linéaires par rapport au motif \underline{m} pour la raison suivante : la linéarité est équivalente à l'existence d'un adjoint $F_{\psi,\phi}^*$. L'adjoint est défini par :

$$\langle \underline{s}, F_{\psi,\phi}[\underline{m}] \rangle = \langle F_{\psi,\phi}^*[\underline{s}], \underline{m} \rangle, \quad (7.44)$$

et cette propriété est très utile pour tous les calculs incluant le produit scalaire, puisqu'elle permet de reporter la déformation sur le signal \underline{s} plutôt que sur le motif \underline{m} .

Nous verrons dans la section 8.1 que cette propriété permet d'utiliser l'algorithme Matching Pursuit pour calculer les paramètres ψ et ϕ , et donc les atomes, de façon rapide, et dans la section 9.3.3.3 que l'existence de l'adjoint permet d'appliquer l'algorithme K-SVD pour le calcul de motifs adaptés aux signaux d'apprentissage.

7.6.6 Composition de fonctions de déformation : stabilité et règle de construction

Le modèle d'atomes structurés présente $F_{\psi,\phi}$ comme une déformation donnée. Or, afin de donner plus de degrés de liberté au modèle en autorisant plusieurs types de déformations, ou pour construire des déformations complexes, il est intéressant de composer plusieurs fonctions $F_{\psi,\phi}$.

Malheureusement, le cadre du modèle (7.21) ne permet pas tout type d'association, pour des raisons de contrainte sur les paramètres ψ . Nous donnons ici les conditions sous lesquelles le modèle est stable par composition, et une règle de construction de la fonction de déformation résultante $F_{\psi,\phi}$.

7.6.7 Composition de deux fonctions

Appliquer successivement deux déformations, représentées par les fonctions G_{ψ^1,ϕ^1} et H_{ψ^2,ϕ^2} , revient à appliquer la composition $H_{\psi^2,\phi^2} \circ G_{\psi^1,\phi^1}$ des deux fonctions. Elle

s'écrit :

$$H_{\boldsymbol{\psi}^2, \phi^2} \circ G_{\boldsymbol{\psi}^1, \phi^1} = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \psi_{i_1}^1 \cdot \psi_{i_2}^2 \cdot \left(H_{\phi^2}^{i_2} \circ G_{\phi^1}^{i_1} \right). \quad (7.45)$$

Étant donnée la multiplication entre les paramètres $\psi_{i_1}^1$ et $\psi_{i_2}^2$, ceux-ci ne peuvent être connus qu'à un facteur près. En effet, les jeux de paramètres $\boldsymbol{\psi}^1$ et $\boldsymbol{\psi}^2$ sont équivalents aux jeux de paramètres $\lambda \cdot \boldsymbol{\psi}^1$ et $1/\lambda \cdot \boldsymbol{\psi}^2$, avec $\lambda \in \mathbb{R} \setminus \{0\}$. Par conséquent, le nombre de paramètres libres du modèle (7.45) est $I_1 + I_2 - 1$. Le retrait d'un paramètre peut se faire en normalisant $\boldsymbol{\psi}^1$ ou $\boldsymbol{\psi}^2$, ou en imposant une valeur à un paramètre (par exemple $\psi_1^1 = 1$).

En notant $F_{\phi}^i = H_{\phi^2}^{i_2} \circ G_{\phi^1}^{i_1}$, et

$$F_{\boldsymbol{\psi}, \phi} = \sum_{i=1}^I \psi_i \cdot F_{\phi}^i, \quad (7.46)$$

nous voulons savoir sous quelles conditions il est possible, après avoir calculé les paramètres $\boldsymbol{\psi} \in \mathbb{R}^I$ de ce modèle, d'identifier les paramètres $\psi_{i_1}^1$ et $\psi_{i_2}^2$.

Les paramètres $\boldsymbol{\psi}^1$ et $\boldsymbol{\psi}^2$ sont identifiables à partir de $\boldsymbol{\psi}$ si le nombre de paramètres libres est égal au nombre de paramètres à identifier :

$$I = I_1 + I_2 - 1. \quad (7.47)$$

Pour cela, il faut se trouver dans un des ces deux cas :

1. la famille des fonctions $H_{\phi^2}^{i_2} \circ G_{\phi^1}^{i_1}$ est libre. Dans ce cas, $I = I_1 \cdot I_2$, et les paramètres sont identifiables si et seulement si $I_1 = 1$ ou $I_2 = 1$, car :

$$I_1 \cdot I_2 = I_1 + I_2 - 1 \iff (I_1 - 1) \cdot (I_2 - 1) = 0. \quad (7.48)$$

2. la famille des fonctions $H_{\phi^2}^{i_2} \circ G_{\phi^1}^{i_1}$ est liée, et engendre un espace de dimension $I_1 + I_2 - 1$. C'est par exemple le cas si deux déformations de type filtrage par un filtre RIF de dimension N se succèdent. En effet, elles forment un filtre RIF de dimension $2 \cdot N - 1$, et les paramètres des deux filtres sont identifiables. Dans d'autres cas, la réduction de dimension induite par la composition n'est pas suffisante ($I > I_1 + I_2 - 1$), et la composition des fonctions sort du modèle (7.21). Nous ne connaissons pas de conditions générales permettant de prédire l'identifiabilité des paramètres dans ce cas. Dans la règle de construction des fonctions de déformation donnée ensuite, nous évitons ce cas de figure.

Remarquons que le problème porte seulement sur les paramètres $\boldsymbol{\psi}^1$ et $\boldsymbol{\psi}^2$. Les paramètres ϕ sont définis par $\phi = (\phi_1^1, \dots, \phi_{I_1}^1, \phi_1^2, \dots, \phi_{I_2}^2)$ et ils sont calculés après une recherche exhaustive sur tous les n-uplets possibles.

Il faut bien sûr noter que les paramètres $\boldsymbol{\psi}^1$ et $\boldsymbol{\psi}^2$ peuvent être optimisés à partir de la formule (7.45), sans se plier aux contraintes énoncées précédemment. La déformation

résultante n'est alors pas dans le cadre de notre modèle, et il faut développer des stratégies alternatives pour calculer les paramètres de l'atome. Dans le chapitre 11 traitant des problèmes non résolus dans cette thèse, nous montrons comment calculer les paramètres dans un cas simple, et nous mettons en lumière le principal verrou empêchant sa généralisation.

7.6.8 Composition de N fonctions

Le résultat précédent s'étend à la composition de N fonctions $F^n \boldsymbol{\psi}^n_{\phi_n}$ définies par :

$$F^n \boldsymbol{\psi}^n_{\phi_n} = \sum_{i_n=1}^{I_n} \psi_{i_n}^n \cdot F^{n i_n}_{\phi_n} . \quad (7.49)$$

La composition des N fonctions est :

$$F^1 \boldsymbol{\psi}^1_{\phi_1} \circ \dots \circ F^N \boldsymbol{\psi}^N_{\phi_N} = \sum_{i_1=1}^{I_1} \dots \sum_{i_N=1}^{I_N} \left(\prod_{n=1}^N \psi_{i_n}^n \right) \cdot \left(F^{1 i_1}_{\phi_1} \circ \dots \circ F^{N i_N}_{\phi_N} \right) . \quad (7.50)$$

Pour les mêmes raisons qu'à la section précédente, le produit des paramètres impose d'en fixer $N - 1$, et le nombre de paramètres libres est donc $\left(\sum_{n=1}^N I_n \right) - (N - 1)$.

En notant $F_{\phi}^i = F^{1 i_1}_{\phi_1} \circ \dots \circ F^{N i_N}_{\phi_N}$, avec l'indice $i = (i_1, \dots, i_N)$ et $\phi = (\phi_1, \dots, \phi_N)$, et

$$F_{\boldsymbol{\psi}, \phi} = \sum_{i=1}^I \psi_i \cdot F_{\phi}^i , \quad (7.51)$$

le problème consiste à pouvoir identifier les paramètres $\psi_{i_n}^n$ à partir des paramètres ψ_i .

La famille des fonctions F_{ϕ}^i peut être liée, ce qui réduit la dimension de l'espace des fonctions $F_{\boldsymbol{\psi}, \phi}$ possibles. Nous ne traitons pas ce cas, et nous supposons que le nombre de paramètres libres $\boldsymbol{\psi}$ est maximum et donc égal à $\left(\prod_{n=1}^N I_n \right)$. Les paramètres $\psi_{i_n}^n$ sont identifiables si et seulement si :

$$\left(\prod_{n=1}^N I_n \right) - \left(\sum_{n=1}^N I_n \right) + (N - 1) = 0 , \quad (7.52)$$

ce qui n'est possible (preuve dans l'annexe A.5) que si toutes les dimensions I_n sont égales à 1, sauf une.

Cela signifie donc qu'il y a au plus un indice n tel que la composition s'écrit :

$$F_{\boldsymbol{\psi}, \phi} = \sum_{i=1}^{I_n} \psi_i^n \cdot \left(F^1_{\phi_1} \circ \dots \circ F^{n-1}_{\phi_{n-1}} \circ F^{n i}_{\phi_n} \circ F^{n+1}_{\phi_{n+1}} \circ \dots \circ F^N_{\phi_N} \right) . \quad (7.53)$$

7.6.9 Règle d'enchaînement de déformations

Afin d'intégrer plusieurs déformations au processus de création de l'atome en étant certain de rester dans le cadre du modèle (7.21), nous proposons de respecter le processus de création suivant :

1. application de $n - 1$ déformations tabulées $F^1_{\phi^1}, \dots, F^{n-1}_{\phi^{n-1}}$, paramétrées par $\phi^1 \dots \phi^{n-1}$, puis
2. application d'une (et une seule) déformation $F^n_{\psi, \phi^n} = \sum_{i=1}^I \psi_i \cdot F^{ni}_{\phi^n}$ paramétrée par ψ et ϕ^n , enfin
3. application de $N - n$ déformations tabulées $F^{n+1}_{\phi^{n+1}}, \dots, F^N_{\phi^N}$ paramétrées par $\phi^{n+1} \dots \phi^N$.

La déformation résultante s'écrit :

$$F_{\psi, \phi} = \psi^T \cdot \mathbf{F}_{\phi} , \quad (7.54)$$

avec $\phi = (\phi^1, \dots, \phi^N)$, et $F_{\phi}^i = F^1_{\phi^1} \circ \dots \circ F^{n-1}_{\phi^{n-1}} \circ F^{ni}_{\phi^n} \circ F^{n+1}_{\phi^{n+1}} \circ \dots \circ F^N_{\phi^N}$.

Chapitre 8

Algorithmes de décomposition sur un dictionnaire structuré

Nous avons exposé aux chapitres 1 et 2 la nécessité de recourir à une stratégie élaborée pour approcher un signal par les atomes d'un dictionnaire redondant. Dès lors que la fonction de translation, par exemple, est utilisée dans le modèle présenté au chapitre 7, alors le dictionnaire généré par les motifs translétés est redondant, et l'approximation du signal ne peut se faire qu'avec une stratégie algorithmique complexe. Parmi les algorithmes présentés au chapitre 2, nous avons choisi d'adapter Matching Pursuit au modèle de dictionnaire structuré, pour des raisons exposées dans la section 8.1.

Nous présentons dans la section 8.2 le principe de l'algorithme Matching Pursuit adapté aux dictionnaires de motifs déformés. Dans les sections 8.3, 8.4, 8.5 et 8.6, nous donnons les détails de MP pour diverses fonctions de déformation, et nous expliquons comment implémenter MP de manière efficace.

Dans la section 8.7, nous exposons une étude expérimentale comparant la qualité de décomposition d'un signal sonore, à parcimonie égale, sur un dictionnaire non-structuré (recourant au tramage), et sur un dictionnaire invariant par translation. Cette étude qui montre le net avantage du dictionnaire invariant par translation permet de conforter notre intérêt pour le modèle de dictionnaire structuré.

8.1 Pourquoi choisir Matching Pursuit ?

Pour étudier les dictionnaires formés par des motifs déformés, il est nécessaire de disposer d'un algorithme de décomposition des signaux sur ces dictionnaires. Une première solution est de construire explicitement le dictionnaire contenant tous les atomes que le dictionnaire de motifs et les fonctions de déformations peuvent générer. Il suffit alors d'utiliser un des algorithmes présentés dans le chapitre 2. Cette solution ne peut cependant pas être retenue pour deux raisons. Tout d'abord, si les fonctions de déformation contiennent des paramètres continus (par exemple le paramètre de phase), il n'est pas possible de construire explicitement la matrice de dictionnaire. Ensuite, nous souhaitons intégrer la translation afin de pouvoir traiter des signaux temporels de taille

significative, et les dimensions sont alors incompatibles avec la longueur. Par exemple, pour une minute d'enregistrement sonore échantillonné à 8 000 Hz, soit 480 000 échantillons, si on ne désire pas effectuer de tramage, la matrice du dictionnaire doit contenir 480 000 lignes et environ $480\,000 \cdot K$ colonnes (avec K le nombre de motifs), ce qui est bien entendu impossible à traiter.

La deuxième solution consiste donc à adapter un des algorithmes présentés dans le chapitre 2 au modèle de motifs déformés. Nous avons choisi d'adapter l'algorithme Matching Pursuit pour les raisons suivantes :

- simplicité d'intégrer un nouveau type d'atome : il suffit de savoir calculer le produit scalaire entre l'atome et le signal, et de savoir soustraire l'atome au signal. C'est sur ce point que le choix de fonctions de déformation linéaires, et donc ayant un adjoint, va permettre de simplifier les calculs et de rendre l'algorithme utilisable.
- rapidité et passage à l'échelle : si MP n'est pas assuré de converger en un nombre d'itérations fini, il permet néanmoins de calculer très rapidement les atomes sélectionnés jusqu'à atteindre un taux d'erreur acceptable. Cela permet de pouvoir utiliser des signaux et des motifs de plus grande dimension.
- choix automatique et successif des atomes : le comportement de l'algorithme est monotone et permet de s'arrêter lorsqu'une approximation acceptable est obtenue.
- preuve de convergence : même s'il est globalement moins parcimonieux que Basis Pursuit ou OMP, il est prouvé que MP approche le minimiseur du même critère sous certaines conditions énoncées à la section 2.1.3.
- maîtrise de l'outil au sein de l'équipe : une bibliothèque de fonctions pour le calcul rapide de MP, nommée MPTK pour Matching Pursuit ToolKit, est développée au sein de l'équipe [KG06].

Parmi les autres algorithmes présentés au chapitre 2, OMP pose le problème théorique et au niveau du temps de calcul de l'orthogonalisation après chaque itération. L'algorithme FOCUSS et les algorithmes de programmation quadratique nécessitent de construire explicitement la matrice du dictionnaire et donc ne peuvent être utilisés comme tels qu'en ayant découpé les signaux en trames. Enfin, les algorithmes de seuillage doux n'ont pour le moment été proposés que pour la structure particulière d'un dictionnaire union de bases orthonormées.

Il est possible que les algorithmes cités précédemment puissent être adaptés au modèle de motifs déformés, mais au début de nos travaux, MP nous a semblé le plus à même d'être adapté, pour les raisons évoquées plus haut. Nous utiliserons donc l'algorithme MP par la suite.

8.2 Matching Pursuit sur un dictionnaire structuré

L'algorithme Matching Pursuit a été présenté à la section 2.1.1 dans sa version originale. Rappelons que l'objectif est la décomposition parcimonieuse du signal \underline{s} comme :

$$\underline{s} = \sum_{i=1}^I \underline{d}_i + \underline{r}_I \quad (8.1)$$

$$= \sum_{i=1}^I F_{\psi_i, \phi_i}[\underline{m}_{k_i}] + \underline{r}_I . \quad (8.2)$$

Le résidu original est fixé à $\underline{r}_0 = \underline{s}$. A chaque itération i , les paramètres de l'atome $F_{\psi, \phi}[\underline{m}_k]$ sont la solution du problème suivant :

$$(k_i, \phi_i, \psi_i) = \arg \min_{k, \phi} \min_{\psi} \|\underline{r}_{i-1} - F_{\psi, \phi}[\underline{m}_k]\|_2^2 , \quad (8.3)$$

et l'algorithme continue jusqu'à ce que la convergence soit atteinte.

La minimisation par rapport à k et par rapport à ϕ est effectuée par recherche exhaustive sur toutes les valeurs possibles de ces deux paramètres. En revanche, le paramètre ψ optimal se calcule de façon analytique. Ainsi, à k et ϕ fixés, le problème devient :

$$\psi(k, \phi) = \arg \min_{\psi} \|\underline{r}_{i-1} - \psi^T \cdot \mathbf{F}_{\phi}[\underline{m}_k]\|_2^2 , \quad (8.4)$$

ce qui peut également s'écrire :

$$\psi(k, \phi) = \arg \max_{\psi} 2 \cdot \psi^T \cdot \mathbf{v}_{k, \phi} - \psi^T \cdot \mathbf{G}_{k, \phi} \cdot \psi , \quad (8.5)$$

en notant $\mathbf{v}_{k, \phi} = [\dots, \langle \underline{r}_{i-1}, F_{\phi}^j[\underline{m}_k] \rangle, \dots]^T$ le vecteur d'intercorrélation entre le résidu \underline{r}_{i-1} et les différentes déformations du motif \underline{m}_k sur la base de fonctions \mathbf{F}_{ϕ} , et en notant $\mathbf{G}_{k, \phi}$ la matrice d'autocorrélation, ou matrice de Gram, du vecteur des déformations du motif \underline{m}_k par les fonctions de la base \mathbf{F}_{ϕ} : $\mathbf{G}_{k, \phi} = \left(\langle F_{\phi}^{j_1}[\underline{m}_k], F_{\phi}^{j_2}[\underline{m}_k] \rangle \right)_{1 \leq j_1, j_2 \leq I_{\psi}}$.

Le vecteur de paramètres $\psi(k, \phi)$ optimal est alors le suivant (preuve dans l'annexe A.6) :

$$\psi(k, \phi) = \mathbf{G}_{k, \phi}^{-1} \cdot \mathbf{v}_{k, \phi} . \quad (8.6)$$

En remplaçant le vecteur ψ par sa valeur optimale $\psi(k, \phi)$ dans (8.3), le problème devient alors :

$$(k_i, \phi_i) = \arg \max_{k, \phi} \mathbf{v}_{k, \phi}^T \cdot \mathbf{G}_{k, \phi}^{-1} \cdot \mathbf{v}_{k, \phi} , \quad (8.7)$$

et se résout par recherche exhaustive sur les paramètres k et ϕ . Le vecteur de paramètres ψ a été éliminé du problème.

Concernant la matrice $\mathbf{G}_{k,\phi}$, plusieurs situations peuvent se présenter, selon la déformation utilisée. Tout d'abord, si ψ est de dimension 1 (une seule fonction de déformation dans la base \mathbf{F}_ϕ), alors la matrice $\mathbf{G}_{k,\phi}$ est réduite à un simple scalaire. La matrice peut également ne pas dépendre du paramètre ϕ , ce qui permet de la calculer avant de commencer l'algorithme, soit pour chaque motif \underline{m}_k , si elle dépend de k , soit une seule fois.

Le vecteur $\mathbf{v}_{k,\phi}$ est quant à lui source de simplification des calculs, grâce à l'**adjoint** de la fonction de déformation. En effet, chaque produit scalaire $\langle \underline{r}_{i-1}, F_\phi^j[\underline{m}_k] \rangle$ peut s'écrire $\langle F_\phi^{j*}[\underline{r}_{i-1}], \underline{m}_k \rangle$, ce qui permet de réduire la dimension du produit scalaire. En effet, le produit scalaire entre signaux devient un produit scalaire entre vecteurs de dimension L_m , puisque le motif est de support fini de longueur L_m . La complexité peut également être réduite si les signaux $F_\phi^{j*}[\underline{r}_{i-1}]$ sont pré-calculés.

Dans les quatre sections suivantes, nous allons tout d'abord décrire l'algorithme MP appliqué à trois déformations (translation, déformation de phase, et expansion multicanal), en donnant la règle de calcul des atomes, mais aussi les détails de l'implémentation. Ces déformations ont été programmées et font l'objet d'expérimentations dans cette thèse. La quatrième des sections est consacrée à l'application de MP à la convolution, que nous décrivons théoriquement mais que nous n'avons pas implémentée. L'évaluation de l'algorithme MP pour décomposer des signaux sur un dictionnaire d'atomes filtrés n'a donc pas eu lieu dans cette thèse et reste une perspective de travail.

8.3 Matching Pursuit invariant par translation

En structurant les atomes comme des motifs tradlatés et amplifiés, Matching Pursuit vise à décomposer le signal \underline{s} comme :

$$\underline{s} = \sum_{i=1}^I c_i \cdot T_{t_i}[\underline{m}_{k_i}] + \underline{r}_I . \quad (8.8)$$

Comme il n'y a qu'un seul paramètre $\psi_1 = c$, la matrice $\mathbf{G}_{k,\phi}$ est réduite au scalaire 1, et le vecteur $\mathbf{v}_{k,\phi}$ au scalaire $\langle \underline{r}_{i-1}, T_t[\underline{m}_k] \rangle$. Ces produits scalaires peuvent être calculés par la corrélation entre le résidu \underline{r}_{i-1} et le motif \underline{m}_k :

$$\langle \underline{r}_{i-1}, T_t[\underline{m}_k] \rangle = [\underline{r}_{i-1} * \underline{m}_k](t) = \sum_l \underline{r}_{i-1}(l) \cdot \underline{m}_k(l+t) . \quad (8.9)$$

Précisons que $*$ désigne la corrélation, alors que nous avons utilisé le symbole \star pour la convolution.

L'algorithme Matching Pursuit est alors le suivant :

Matching Pursuit sur des motifs tradlatés :

Initialisation du résidu $\underline{r}_0 = \underline{s}$ et du compteur $I = 0$.

Faire

1. Mise à jour du compteur : $I = I + 1$
2. Recherche du motif et du paramètre de translation :

$$(k_I, t_I) = \arg \max_{k,t} [\underline{r}_{I-1} * \underline{m}_k]^2(t) \quad (8.10)$$

3. Calcul de l'amplitude :

$$c_I = [\underline{r}_{I-1} * \underline{m}_{k_I}](t_I) \quad (8.11)$$

4. Mise à jour du résidu :

$$\underline{r}_I = \underline{r}_{I-1} - c_I \cdot T_{t_I}[\underline{m}_{k_I}] \quad (8.12)$$

Tant que la convergence n'est pas atteinte

Du point de vue de la complexité, il est intéressant d'utiliser la corrélation entre le motif et le résidu, puis de calculer le carré des coefficients de la corrélation, au lieu de calculer le carré du produit scalaire de chaque motif, à chaque instant, avec le résidu (équation (8.10)).

En effet, si le résidu a une longueur L et le motif une longueur $L_m \ll L$, alors le calcul du carré de tous les produits scalaires revient à $O(L \times L_m)$ opérations. En calculant la corrélation, par la technique d'overlap-add [RS78], le nombre d'opérations tombe à $O(L \cdot \log(L_m))$.

Pour gagner beaucoup de temps, il est également utile de ne recalculer, à l'étape I , que les produits scalaires entre les motifs et la partie du résidu \underline{r}_{I-1} qui a été modifiée à l'étape précédente par le retrait de l'atome. Au lieu de calculer L produits scalaires, il suffit donc d'en calculer $2 \cdot L_m$. Évidemment, à la première étape, il faut calculer tous les produits scalaires.

Enfin, une troisième optimisation peut être appliquée pour la recherche du produit scalaire maximum. En effet, normalement, la recherche du maximum se fait en parcourant toutes les valeurs de la corrélation entre le résidu et chaque motif. Cette recherche a donc un coût de l'ordre de $O(L \cdot K)$. En utilisant une technique de dichotomie, la complexité de la recherche du maximum peut être ramenée à $O(\log(L) \cdot K)$.

Toutes ces optimisations sont détaillées dans l'article [KG06] qui est consacré à l'implémentation rapide de Matching Pursuit invariant par translation. Ces optimisations sont intégrées à la bibliothèque de fonctions MPTK, que nous utilisons pour les évaluations expérimentales.

8.4 Matching Pursuit invariant par translation et déformation de phase

Nous étudierons dans la suite l'apport de la déformation de phase dans la qualité et la parcimonie de la décomposition. Rappelons que la déformation de phase consiste,

associée à la translation des motifs, à modéliser les atomes comme :

$$\underline{d} = \psi_1 \cdot T_t[\underline{m}] + \psi_2 \cdot T_t \circ H[\underline{m}] . \quad (8.13)$$

Nous avons imposé que $\|\underline{m}\|_2^2$ est égal à 1. Nous imposons également que $\|H[\underline{m}]\|_2^2$ vaut 1, ce qui correspond à imposer que les motifs \underline{m} ont une composante moyenne nulle (la composante de Nyquist est aussi nulle si L_m est pair). Cette restriction permet en effet de simplifier l'expression de la matrice $\mathbf{G}_{k,\phi}$:

$$\mathbf{G}_{k,\phi} = \begin{bmatrix} \|\underline{m}\|_2^2 & 0 \\ 0 & \|H[\underline{m}]\|_2^2 \end{bmatrix} = \mathbf{I} . \quad (8.14)$$

La restriction sera de plus cruciale pour permettre l'apprentissage d'atomes de ce type par K-SVD (voir section 9.3.4).

Le jeu de paramètres $\boldsymbol{\psi}$ vaut $\begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix}$, et $\phi = t$. Le vecteur $\mathbf{v}_{k,\phi}$ vaut :

$$\mathbf{v}_{k,\phi} = \begin{bmatrix} \langle \underline{r}_{i-1}, T_t[\underline{m}_{k_i}] \rangle \\ \langle \underline{r}_{i-1}, T_t \circ H[\underline{m}_{k_i}] \rangle \end{bmatrix} . \quad (8.15)$$

En conséquence, l'algorithme Matching Pursuit se déroule comme :

Matching Pursuit sur des motifs tradatés et à phase déformée :

Initialisation du résidu $\underline{r}_0 = \underline{g}$ et du compteur $I = 0$.

Faire

1. Mise à jour du compteur : $I = I + 1$
2. Recherche du motif et du paramètre de translation :

$$(k_I, t_I) = \arg \max_{k,t} [\langle \underline{r}_{I-1}, T_t[\underline{m}_k] \rangle^2 + \langle \underline{r}_{I-1}, T_t \circ H[\underline{m}_k] \rangle^2] \quad (8.16)$$

3. Calcul de l'amplitude :

$$\begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} = \begin{bmatrix} \langle \underline{r}_{I-1}, T_{t_I}[\underline{m}_{k_I}] \rangle \\ \langle \underline{r}_{I-1}, T_{t_I} \circ H[\underline{m}_{k_I}] \rangle \end{bmatrix} \quad (8.17)$$

4. Mise à jour du résidu :

$$\underline{r}_I = \underline{r}_{I-1} - (\psi_1 \cdot T_{t_I}[\underline{m}_{k_I}] + \psi_2 \cdot T_{t_I} \circ H[\underline{m}_{k_I}]) \quad (8.18)$$

Tant que la convergence n'est pas atteinte

L'implémentation pour ce type d'atome diffère peu de celle de la section précédente. La seule différence est de calculer, en parallèle à la convolution entre le résidu et chaque motif \underline{m}_k , la convolution entre le résidu et le motif en quadrature $H[\underline{m}_k]$. Les motifs en quadrature $H[\underline{m}_k]$ peuvent être calculés une fois pour toutes, avant de démarrer Matching Pursuit.

L'implémentation de Matching Pursuit pour des motifs translétés, avec ou sans déformation de phase, ainsi que du calcul rapide de la convolution au sein de la bibliothèque MPTK de fonctions C++ a été une des principales contributions logicielle de ce travail. Notons que le “moteur” de Matching Pursuit était déjà implémenté avant ce travail de thèse, et permettait de calculer des décompositions sur des atomes de Gabor.

L'implémentation rapide de Matching Pursuit pour de tels atomes a été cruciale pour pouvoir effectuer des évaluations expérimentales sur des signaux de taille conséquente. Pour donner un ordre de grandeur, le calcul de la décomposition d'un signal musical de 1 024 512 échantillons (1 min 32 s à 11 025 Hz) jusqu'à un rapport signal à bruit de 30 dB, pour des atomes formés par la déformation de phase et la translation de motifs issus d'un dictionnaire de 40 motifs, prend 1 heure (la décomposition utilise $I = 300\,000$ atomes). Il n'aurait pas été possible de l'effectuer sans les quelques optimisations décrites plus haut.

8.5 Matching Pursuit multicanal et Demixing Pursuit

Nous exposons maintenant l'adaptation de l'algorithme Matching Pursuit à la décomposition de signaux multicanaux comme somme d'atomes formés par expansion multicanale de motifs monocanaux. Comme pour les sections précédentes, les motifs sont également translétés dans le temps.

Dans cette section, nous nous limitons au modèle où le motif est uniquement multiplié par une amplitude différente sur chaque canal, et non pas translété d'un délai différent sur chaque canal, ou convolué par des filtres RIF différents (le modèle convolutif a été décrit dans la section 7.5.5). En effet, le modèle simple est déjà très riche et nous verrons comment l'appliquer comme algorithme de séparation de sources multicanale, sous la dénomination Demixing Pursuit. De plus, les versions avec délai et filtre RIF n'ont pas été implémentées dans MPTK, contrairement au modèle simple.

Le modèle des atomes est :

$$\underline{\mathbf{d}} = E_{\mathbf{a}} \circ T_t[\underline{\mathbf{m}}] = \sum_{c=1}^M a_c \cdot E_t^c[\underline{\mathbf{m}}] , \quad (8.19)$$

avec :

$$E_t^c = \begin{bmatrix} \vdots \\ 0 \\ T_t \\ 0 \\ \vdots \end{bmatrix} \leftarrow c \quad (8.20)$$

la fonction qui envoie un signal monocanal entièrement sur le canal c , décalé de t . Nous avons noté l'atome $\underline{\mathbf{d}}$ avec une minuscule, bien que ce soit un signal multicanal, pour éviter la confusion avec le dictionnaire non-structuré $\underline{\mathbf{D}}$ que nous avons manipulé dans la première partie.

En se reportant au chapitre 4 traitant de la séparation de sources, dans le cas multicanal notamment, considérons que le motif est un signal émis à un certain endroit dans l'espace, tandis que les canaux correspondent à l'enregistrement du signal reçu à un autre endroit. Dans ce cas, les amplitudes a_c marquent l'importance de la réception du motif au capteur c . Le vecteur \mathbf{a} qui porte les amplitudes du motif sur chacun des canaux est alors assimilé à la direction spatiale d'émission de la source. Cette interprétation a également déjà été évoquée à la section 7.5.5.

Le modèle recouvre deux réalités différentes, qui dépendent de la connaissance ou non des coefficients a_c . Dans le premier cas, les coefficients a_c sont inconnus (la direction est inconnue) et ils sont les paramètres ψ_c . Dans le deuxième cas, les coefficients a_c sont supposés connus à un facteur près. Cela correspond à connaître la direction donnée par le vecteur \mathbf{a} et à supposer que seuls la norme et le signe du vecteur \mathbf{a} sont inconnus.

Le premier cas mène à l'algorithme Matching Pursuit multicanal, le deuxième cas mène à l'algorithme Demixing Pursuit.

8.5.1 Matching Pursuit multicanal

L'algorithme Matching Pursuit multicanal a pour objectif de décomposer un signal multicanal comme somme de motifs monocanaux placés au même instant avec une amplitude différente sur chaque canal. Dans un contexte d'enregistrement spatial, cela correspond à estimer librement l'endroit d'émission du motif pour chaque atome.

Le jeu de paramètres ψ vaut donc $[a_1 \dots a_{J_{\mathbf{a}}}]^T$, et $\phi = t$. La matrice $\mathbf{G}_{k,t}$ est la matrice identité $\mathbf{I}_{J_{\mathbf{a}}}$, en raison de l'orthogonalité entre les fonctions E_t^c et du fait qu'elles préservent la norme. Cette dernière propriété signifie que pour tout motif $\underline{\mathbf{m}}$ de norme unité, sa version déformée est aussi de norme unité : $\|E_t^c[\underline{\mathbf{m}}]\|_2 = 1$.

Le vecteur $\mathbf{v}_{k,t}$ vaut :

$$\mathbf{v}_{k,t} = \begin{bmatrix} \langle \underline{\mathbf{R}}_{i-1}, E_t^1[\underline{\mathbf{m}}_k] \rangle \\ \vdots \\ \langle \underline{\mathbf{R}}_{i-1}, E_t^{J_{\mathbf{a}}}[\underline{\mathbf{m}}_k] \rangle \end{bmatrix} = \begin{bmatrix} \langle \underline{\mathbf{r}}_{1,i-1}, T_t[\underline{\mathbf{m}}_k] \rangle \\ \vdots \\ \langle \underline{\mathbf{r}}_{J_{\mathbf{a}},i-1}, T_t[\underline{\mathbf{m}}_k] \rangle \end{bmatrix}, \quad (8.21)$$

avec $\underline{\mathbf{r}}_{c,i-1}$ le c^e canal du résidu $\underline{\mathbf{R}}_{i-1}$.

L'algorithme Matching Pursuit multicanal a pour objectif la décomposition :

$$\underline{\mathbf{S}} = \sum_{i=1}^I E_{\mathbf{a}_i} \circ T_{t_i} [\underline{\mathbf{m}}_{k_i}] + \underline{\mathbf{R}}_I \quad (8.22)$$

Voilà sa description :

Matching Pursuit multicanal sur des motifs monocanaux translétés et amplifiés sur chaque canal :

Initialisation du résidu $\underline{\mathbf{R}}_0 = \underline{\mathbf{S}}$ et du compteur $I = 0$.

Faire

1. Mise à jour du compteur : $I = I + 1$
2. Recherche du motif et du paramètre de translation :

$$(k_I, t_I) = \arg \max_{k,t} \|\mathbf{v}_{k,t}\|_2^2 = \arg \max_{k,t} \sum_{c=1}^M |\langle \underline{\mathbf{r}}_{c,I-1}, T_t[\underline{\mathbf{m}}_k] \rangle|^2 \quad (8.23)$$

3. Calcul des coefficients a_c (direction spatiale et amplitude) :

$$\mathbf{a}_I = \mathbf{v}_{k_I, t_I} = \begin{bmatrix} \langle \underline{\mathbf{r}}_{1,I-1}, T_{t_I}[\underline{\mathbf{m}}_{k_I}] \rangle \\ \vdots \\ \langle \underline{\mathbf{r}}_{I_a,I-1}, T_{t_I}[\underline{\mathbf{m}}_{k_I}] \rangle \end{bmatrix} \quad (8.24)$$

4. Mise à jour du résidu :

$$\underline{\mathbf{R}}_I = \underline{\mathbf{R}}_{I-1} - E_{\mathbf{a}_I} \circ T_{t_I} [\underline{\mathbf{m}}_{k_I}] \quad (8.25)$$

Tant que la convergence n'est pas atteinte

L'implémentation de Matching Pursuit multicanal s'inspire directement de celle de Matching Pursuit sur des motifs translétés, présentée à la section 8.3. Au lieu de calculer la corrélation entre le résidu et chaque motif, il faut maintenant calculer la corrélation entre chaque canal du résidu et chaque motif, avec toutes les optimisations précédemment expliquées. La recherche du produit scalaire maximum (équation (8.23)) se fait par l'addition des carrés des corrélations sur les différents canaux.

8.5.2 Demixing Pursuit

L'algorithme Demixing Pursuit a pour objectif de décomposer un signal multicanal comme somme d'atomes multicanaux créés de la manière suivante. Un motif monocanal est réparti sur les différents canaux suivant les coefficients d'un vecteur \mathbf{a}_n normalisé, pris au sein d'un ensemble de vecteurs tabulés. Il est ensuite placé à un instant donné, et multiplié par un facteur d'amplitude.

Dans un contexte d'enregistrement spatial, cela correspond à estimer, parmi un ensemble de directions spatiales préétabli, dans quelle direction a été émis le motif, pour chaque atome. La recherche de la direction est limitée à cet ensemble, et seule la recherche de l'amplitude est libre.

Le jeu de paramètres ψ est donc réduit à l'amplitude c , et le paramètre d'indice comprend le numéro du vecteur de direction \mathbf{a}_n et l'instant de démarrage de l'atome : $\phi = (n, t)$. La matrice $\mathbf{G}_{k,(n,t)}$ est réduite au scalaire 1. Le vecteur $\mathbf{v}_{k,(n,t)}$ est également un scalaire :

$$\mathbf{v}_{k,(n,t)} = \langle \underline{\mathbf{R}}_{i-1}, E_{\mathbf{a}_n} \circ T_t[\underline{\mathbf{m}}_k] \rangle. \quad (8.26)$$

L'algorithme Demixing Pursuit se déroule comme suit :

Demixing Pursuit :

Initialisation du résidu $\underline{\mathbf{R}}_0 = \underline{\mathbf{S}}$ et du compteur $I = 0$.

Faire

1. Mise à jour du compteur : $I = I + 1$
2. Recherche du motif, du paramètre de translation et du numéro du vecteur de direction :

$$(k_I, t_I, n_I) = \arg \max_{k,t,n} |\langle \underline{\mathbf{R}}_{I-1}, T_t \circ E_{\mathbf{a}_n}[\underline{\mathbf{m}}_k] \rangle|^2 \quad (8.27)$$

$$= \arg \max_{k,t,n} |\langle \sum_{c=1}^M a_{c,n} \cdot \underline{\mathbf{r}}_{c,I-1}, T_t \circ \underline{\mathbf{m}}_k \rangle|^2 \quad (8.28)$$

3. Calcul de l'amplitude :

$$c_I = \langle \underline{\mathbf{R}}_{I-1}, T_{t_I} \circ E_{\mathbf{a}_{n_I}}[\underline{\mathbf{m}}_{k_I}] \rangle \quad (8.29)$$

$$= \langle \sum_{c=1}^M a_{c,n_I} \cdot \underline{\mathbf{r}}_{c,I-1}, T_{t_I} \circ \underline{\mathbf{m}}_{k_I} \rangle \quad (8.30)$$

4. Mise à jour du résidu :

$$\underline{\mathbf{R}}_I = \underline{\mathbf{R}}_{I-1} - T_{t_I} \circ E_{\mathbf{a}_{n_I}}[\underline{\mathbf{m}}_{k_I}] \quad (8.31)$$

Tant que la convergence n'est pas atteinte

La différence entre Demixing Pursuit et Matching Pursuit sur des motifs translétés, décrit à la section 8.3, réside uniquement dans l'ajout de la recherche de la direction \mathbf{a}_n , ce qui multiplie la complexité par N , le nombre de directions. En effet, en utilisant les formules (8.28) et (8.30), et en notant $\underline{y}_n = \sum_{c=1}^M a_{c,n} \cdot \underline{r}_c = \mathbf{a}_n^T \cdot \mathbf{R}$ la projection du résidu \mathbf{R} sur la direction \mathbf{a}_n , il suffit de calculer la corrélation de chacune de ces N projections avec chaque motif, afin de trouver la corrélation maximale (équation (8.28)).

La mise à jour (8.31) peut alors être effectuée directement sur les projections :

$$\underline{y}_{n,I} = \underline{y}_{n,I-1} - \langle \mathbf{a}_n, \mathbf{a}_{n_I} \rangle \cdot T_{t_I}[\underline{m}_{k_I}] \quad (8.32)$$

Cette technique d'optimisation est expliquée en détails dans [GN05]. Le nom Demixing Pursuit est apparu dans l'article [LGK06], qui fait suite à l'article [LGK05]. Dans ces deux articles, nous avons notamment comparé l'utilisation de Matching Pursuit multicanal et de Demixing Pursuit pour la séparation de sources, dans le cas multicanal lorsque les directions \mathbf{a}_n des sources sont connues.

8.6 Perspective : Matching Pursuit convolutif

Le modèle d'atome formé par translation et filtrage de motifs par un filtre RIF est le suivant :

$$\underline{d} = T_t \circ C\mathbf{e}[\underline{m}] = \sum_{i=1}^{I_e} e_i \cdot T_{t+i-1}[\underline{m}] . \quad (8.33)$$

Nous n'avons pas implémenté ce type d'atomes dans MPTK, par manque de temps, ce qui implique que nous n'avons pas pu évaluer leur apport en termes de parcimonie ou en vue d'une application en séparation de sources. Nous présentons ici le principe théorique et quelques détails permettant une implémentation rapide.

Le jeu de paramètres $\boldsymbol{\psi}$ vaut donc \mathbf{e} , et $\phi = t$. La matrice $\mathbf{G}_{k,t}$ et le vecteur $\mathbf{v}_{k,t}$

valent :

$$\mathbf{G}_{k,t} = \begin{bmatrix} \|\underline{m}\|_2^2 & \langle \underline{m}, T_1[\underline{m}] \rangle & \dots & \langle \underline{m}, T_{I_e-1}[\underline{m}] \rangle \\ \langle \underline{m}, T_{-1}[\underline{m}] \rangle & \|\underline{m}\|_2^2 & \backslash & \vdots \\ \vdots & \backslash & \backslash & \langle \underline{m}, T_1[\underline{m}] \rangle \\ \langle \underline{m}, T_{1-I_e}[\underline{m}] \rangle & \dots & \langle \underline{m}, T_{-1}[\underline{m}] \rangle & \|\underline{m}\|_2^2 \end{bmatrix} \quad (8.34)$$

$$\mathbf{v}_{k,t} = \begin{bmatrix} \langle \underline{r}_{i-1}, T_t[\underline{m}_k] \rangle \\ \langle \underline{r}_{i-1}, T_{t+1}[\underline{m}_k] \rangle \\ \vdots \\ \langle \underline{r}_{i-1}, T_{t+I_e-1}[\underline{m}_k] \rangle \end{bmatrix} \quad (8.35)$$

La matrice $\mathbf{G}_{k,t}$ est une matrice de Toeplitz (diagonales constantes) symétrique. Cette structure lui permet d'être inversée rapidement en utilisant la récurrence de Levinson[GL96]. Elle ne dépend pas du paramètre de translation t , et nous la notons \mathbf{G}_k .

Si les coefficients e_i du filtre sont libres de toute contrainte (comme par exemple exiger un peigne de Dirac, ou toute relation entre les coefficients), alors l'algorithme Matching Pursuit est le suivant :

Matching Pursuit sur des motifs tradatés et filtrés par un filtre RIF :

Initialisation du résidu $\underline{r}_0 = \underline{g}$ et du compteur $I = 0$.

Faire

1. Mise à jour du compteur : $I = I + 1$
2. Recherche du motif et du paramètre de translation :

$$(k_I, t_I) = \arg \max_{k,t} \mathbf{v}_{k,t}^T \cdot \mathbf{G}_k^{-1} \cdot \mathbf{v}_{k,t} \quad (8.36)$$

3. Calcul de l'amplitude :

$$\mathbf{e}_I = \mathbf{G}_{k_I}^{-1} \cdot \mathbf{v}_{k_I, t_I} \quad (8.37)$$

4. Mise à jour du résidu :

$$\underline{r}_I = \underline{r}_{I-1} - \left(\sum_{i=1}^{I_e} e_i \cdot T_{t_I+i-1}[\underline{m}_{k_I}] \right) \quad (8.38)$$

Tant que la convergence n'est pas atteinte

L'implémentation de cet algorithme peut être optimisée en calculant la corrélation, par overlap-add, entre le résidu et chaque motif. Chaque tranche de I_e coefficients de la corrélation correspond au vecteur $\mathbf{v}_{k,t}$ pour une valeur donnée de t . En ayant au préalable calculé et tabulé la matrice \mathbf{G}_k^{-1} pour chaque motif, il faut ensuite calculer le produit décrit à l'équation (8.36). Il est possible que l'écriture de ce produit sous une autre forme permette de faire émerger un nouveau type d'optimisation.

Comme pour les sections précédentes, il est évidemment inutile de recalculer les produits inchangés, et une méthode dichotomique permet d'accélérer la recherche du maximum, par rapport à une recherche linéaire.

8.7 Dictionnaires structuré et non-structuré : comparaison en termes de parcimonie

Dans cette section, nous allons montrer sur un exemple simple l'intérêt en termes de qualité de la décomposition d'utiliser un dictionnaire invariant par translation pour traiter les signaux temporels. Cela peut sembler évident, mais de façon classique les signaux sont découpés en trames avant traitement, bien que ce ne soit pas le plus efficace.

Définissons dans un premier temps la métrique servant à mesurer la qualité de la décomposition d'un signal dans un dictionnaire.

8.7.1 Métrique

Dans le cadre de l'approximation parcimonieuse de signaux sur un dictionnaire, l'objectif est en général de minimiser l'erreur de reconstruction et de maximiser la parcimonie des coefficients de la décomposition. Selon la formulation du problème, divers algorithmes peuvent être mis en œuvre pour le résoudre. Un algorithme donne en résultat, pour une parcimonie donnée, un unique jeu de coefficients, associé à une erreur de reconstruction. Nous pouvons donc associer, pour un algorithme donné, une courbe de l'erreur en fonction de la parcimonie à un couple signal-dictionnaire.

C'est cette courbe de l'erreur de reconstruction en fonction de la parcimonie des coefficients qui sera utilisée pour mesurer la qualité d'une décomposition. La comparaison de deux couples signal-dictionnaire se fera par la comparaison des deux courbes associées. La décomposition du signal est notée :

$$\mathbf{s} = \sum_{i=1}^I F_{\psi_i, \phi_i} [m_{k_i}] + \mathbf{r}_I . \quad (8.39)$$

Comme nous pourrions être amenés à comparer la décomposition de signaux de taille et d'énergie différentes, nous proposons d'utiliser une métrique indépendante de ces deux

paramètres, en remplaçant l'énergie par le RSB, et le nombre d'atomes par la **diversité relative**. Avant d'expliquer plus précisément cette métrique, comparons dès maintenant la représentation classique de l'erreur de reconstruction en fonction du nombre d'atomes utilisés (par MP) avec la représentation du RSB en fonction de la diversité relative. La première est située à gauche de la figure 8.1, et la seconde à droite.

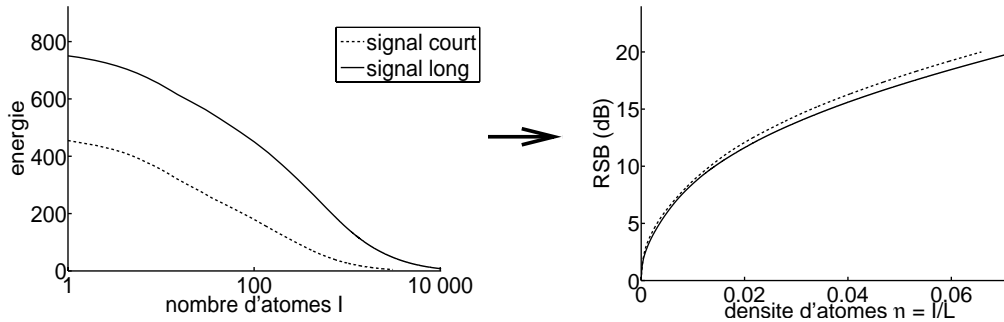


FIG. 8.1 – Affichage de la qualité de la décomposition d'un signal court et d'un signal long de même nature sur un même dictionnaire - gauche : énergie du résidu en fonction du nombre d'atomes - droite : rapport signal à bruit en fonction de la diversité relative

L'expérience a consisté à décomposer deux signaux sonores de même nature sur un même dictionnaire de Gabor. Le signal court correspond au premier tiers du signal long. La décomposition des deux signaux a été faite avec MP, jusqu'à obtenir un RSB de 20 dB entre le signal d'origine et le résidu.

Sur la figure de gauche, nous avons représenté les courbes de l'énergie du résidu en fonction du nombre d'atomes utilisés dans la décomposition. Nous avons représenté le nombre d'atomes en échelle logarithmique pour une meilleure lisibilité. Nous remarquons que l'énergie initiale des deux signaux est différente et que le nombre d'atomes utilisés est différent (environ trois fois plus d'atomes pour le signal long que pour le signal court). Il est difficile de comparer les décompositions à partir de ces deux courbes.

Sur la figure de droite, nous avons représenté les courbes du rapport signal à bruit entre le signal d'origine et le résidu, en fonction de la **diversité relative (DR)** η , qui correspond au nombre d'atomes par échantillon de signal :

$$\eta_l = \frac{l}{L}, \quad (8.40)$$

où l est le nombre d'atomes, et L la longueur du signal.

L'utilisation du RSB permet de gommer la différence d'énergie des signaux initiaux. Celui-ci vaut 0 dB lorsque le résidu est d'énergie égale à celle du signal, et vaut $+\infty$ lorsque le résidu est nul.

L'utilisation de la DR, quant à elle, permet de rendre la mesure indépendante de la longueur du signal traité. En effet, nous attendons à ce que la DR "locale" soit à peu près constante, quel que soit l'instant étudié, pour un signal temporel dont les propriétés sont constantes au cours du temps.

Ainsi, sur la figure de droite, les deux signaux présentent des courbes quasiment superposées, puisqu'en dehors de leur longueur et de leur énergie, les signaux sont de même nature et se décomposent de façon à peu près aussi parcimonieuse sur un même dictionnaire.

La DR est égale à 1 si le nombre d'atomes est égal à la longueur du signal. Une telle diversité est en général bien supérieure aux régimes classiquement étudiés, puisque la parcimonie serait dans ce cas très faible, et la décomposition serait tout sauf économique. Pour un régime classique, la DR vérifie $\eta_I \ll 1$.

La courbe servant à la comparaison est donc la courbe du RSB en fonction de la DR η . Plus le RSB est élevé, et meilleure est la décomposition. Une courbe placée au dessus d'une autre correspond à une meilleure décomposition, en termes d'erreur de reconstruction et de parcimonie.

Nous choisissons dans cette étude d'utiliser l'algorithme MP pour générer la courbe. Il en résulte que la courbe est composée de I points, où I est le nombre d'atomes sélectionnés par MP. Le RSB est défini, pour chaque point, par :

$$\text{RSB}_{\eta_i} = 10 \cdot \log_{10} \frac{\|s\|_2^2}{\|r_i\|_2^2} \quad (8.41)$$

Dans l'expérience qui suit, nous allons utiliser des dictionnaires invariants par translation. Nous utiliserons également des dictionnaires non-structurés, appliqués sur des trames de signaux, obtenues par tramage selon le principe de la figure 7.1. Pour ce cas, le RSB sera calculé sur le signal reconstruit après "détramage", et la DR η_I sera obtenue par le rapport entre le nombre d'atomes utilisés et la taille du signal reconstitué.

8.7.2 Intentions expérimentales

Nous souhaitons évaluer à travers cette expérience à quel point le modèle de dictionnaire invariant par translation est plus intéressant pour décomposer de façon parcimonieuse les signaux temporels que le modèle de dictionnaire classique, utilisant un découpage en trames des signaux.

8.7.3 Principe de l'expérience

Nous allons décomposer un signal audio de trois manières différentes sur un dictionnaire de Gabor, qui est en principe adapté à la décomposition parcimonieuse de signaux audio. Nous utiliserons une seule taille L_m de motifs de Gabor.

Dans le premier cas, le signal sera découpé en trames de taille L_m , puis décomposé sur le dictionnaire de motifs de Gabor (le même nombre d'atomes sera utilisé sur toutes les trames). Cela correspond à la décomposition d'un signal sur un dictionnaire classique. Dans le deuxième cas, le signal entier sera décomposé sur le dictionnaire de Gabor "partiellement invariant" par translation, avec un décalage entre atomes de Gabor égal au décalage entre les trames dans le premier cas (en revanche, le nombre d'atomes par trame sera variable). Enfin, dans le troisième cas, le signal sera décomposé sur le

dictionnaire de Gabor invariant par translation, avec un décalage entre atomes de Gabor de un échantillon.

Les deux premiers cas correspondent à un dictionnaire de même taille, tandis que le troisième cas utilise un dictionnaire plus grand. Il faut remarquer que ce troisième dictionnaire ne coûte pas plus cher à coder que les deux premiers, car il repose sur le même dictionnaire de motifs. En revanche, la description de la décomposition coûte plus cher, car le nombre d'atomes engendrés est plus grand et l'indice désignant l'atome choisi doit être codé avec plus de bits.

Nous comparerons les courbes du RSB de la décomposition du signal sur chaque dictionnaire, en fonction de la DR, ainsi que le temps de calcul, afin de comprendre quel gain de modélisation est obtenu en passant du premier au deuxième cas, puis du deuxième cas au troisième cas.

8.7.4 Protocole détaillé

Le protocole détaillé est décrit dans le tableau suivant

Signal \underline{s}	704 130 échantillons (1 m 03 s à 11 025 Hz), provenant de l'enregistrement "RWC-MDB-C-2001 No 44" de la base RWC [MGO02] : Le vol du bourdon, de Rimski-Korsakov, pour flûte et piano.
Dictionnaire de motifs \mathcal{M}	127 motifs de Gabor complexes, correspondant à 127 fréquences, ainsi qu'un motif constant (0 Hz) et qu'un motif correspondant à la fréquence de Nyquist (5 512.5 Hz, la moitié de la fréquence d'échantillonnage). Les motifs sont de taille $L_m = 256$ (23.2 ms à 11 025 Hz). Les motifs de Gabor complexes comptent pour deux motifs réels, ce qui explique que le nombre de fréquences soit 127 et pas 254.
Décalage entre trames de signal	$L_m/2$
Fenêtre	Arche de sinus
Critère d'arrêt de MP	$\eta = 0.8$
Dictionnaire \mathcal{D}_1	Dictionnaire de motifs \mathcal{M}
Dictionnaire \mathcal{D}_2	Dictionnaire "partiellement invariant" par translation où les atomes sont décalés de $L_m/2$ échantillons, utilisant le dictionnaire de motifs \mathcal{M} .
Dictionnaire \mathcal{D}_3	Dictionnaire invariant par translation où les atomes sont décalés de 1 échantillon, utilisant le dictionnaire de motifs \mathcal{M} .

8.7.5 Résultats attendus

Nous allons présenter les courbes de RSB en fonction de la DR des trois dictionnaires sur le même signal. Nous présenterons aussi le temps de calcul de la décomposition du signal sur chaque dictionnaire.

Nous pouvons nous attendre à ce que la courbe du dictionnaire \mathcal{D}_2 soit au dessus de celle du dictionnaire \mathcal{D}_1 , ce qui correspond à une meilleure décomposition, puisque le nombre d'atomes par trame n'est plus fixé. Ainsi, pour une DR $\eta = 3$, avec le dictionnaire \mathcal{D}_2 , une trame de signal peut recevoir cinq atomes et la suivante un seul, tandis qu'avec le dictionnaire \mathcal{D}_1 , les deux trames recevront exactement trois atomes.

Au niveau du temps de calcul également, le fait de chercher les atomes directement sur le signal, plutôt que de recourir à un tramage, devrait plaider en faveur du dictionnaire \mathcal{D}_2 .

La courbe du dictionnaire \mathcal{D}_3 , enfin, devrait être plus haute que pour les deux autres dictionnaires, puisque dans \mathcal{D}_3 , la contrainte de placer les atomes à un instant fixé est relâchée. Cela dit, le fait que les motifs soient de petite taille et que la phase de l'atome est optimisée peut limiter la différence de performance.

Le dictionnaire \mathcal{D}_3 devrait être plus gourmand en temps de calcul, puisqu'il faut calculer $L_m/2$ fois plus de produits scalaires que pour les deux autres dictionnaires, et rechercher le maximum parmi $L_m/2$ fois plus de valeurs.

Plus précisément, le dictionnaire \mathcal{D}_1 contient 709 500 atomes (le nombre de motifs de \mathcal{M} multiplié par le nombre de trames). Le nombre d'atomes est le même pour le dictionnaire \mathcal{D}_2 . Le dictionnaire \mathcal{D}_3 compte quant à lui 90 799 875 atomes, soit $L_m/2 = 128$ fois plus que les deux autres dictionnaires, aux effets de bord près.

8.7.6 Présentation des résultats

Le temps de calcul de la décomposition du signal sur le dictionnaire \mathcal{D}_2 est de 44 s, alors qu'il est de 2 304 s (38 m 24 s) pour le dictionnaire \mathcal{D}_3 . Le temps de calcul est environ 52 fois plus important pour le dictionnaire \mathcal{D}_3 que pour le dictionnaire \mathcal{D}_2 , ce qui est inférieur au ratio prévu de $L_m/2 = 128$.

En effet, les optimisations mises en œuvre dans MPTK [KG06] permettent de diminuer la complexité. Les principales optimisations sont l'utilisation de la transformée de Fourier rapide, qui ramène la complexité du calcul des produits scalaires de $O(L_m)$ à $O(\log(L_m))$, le calcul des produits scalaires uniquement pour les atomes situés à proximité du dernier atome sélectionné (les seuls produits scalaires à être modifiés), et la recherche du produit scalaire maximal par une méthode dichotomique plutôt que de façon linéaire, qui fait passer pour chaque itération la complexité de $O(K \cdot L)$ à $O(\log(K \cdot L))$.

Nous n'avons pas indiqué le temps de calcul de la décomposition pour le dictionnaire \mathcal{D}_1 car nous n'avons pas utilisé un algorithme optimisé. Nous nous attendons à ce que le temps de calcul soit légèrement inférieur à celui obtenu pour le dictionnaire \mathcal{D}_2 . La seule différence réside dans le traitement successif des trames, plutôt que de chercher la meilleure trame à chaque itération. Le maximum est donc recherché parmi K valeurs,

au lieu de $K \cdot L$. L'étape de calcul des produits scalaires, en revanche, est similaire pour les deux dictionnaires.

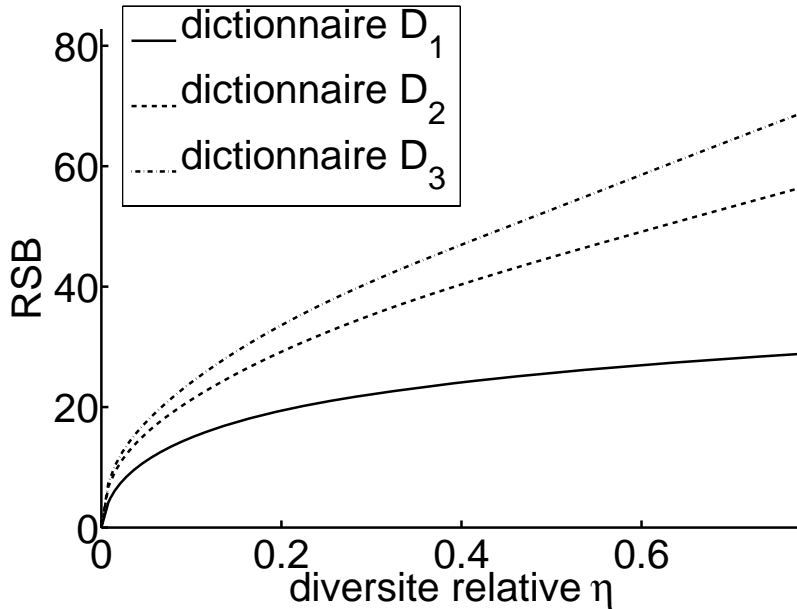


FIG. 8.2 – Courbe du RSB entre le signal et l'erreur résiduelle après décomposition, en fonction de la diversité relative η de la décomposition

Intéressons-nous maintenant à la qualité de la décomposition du signal sur les trois dictionnaires. La courbe du RSB en fonction de la DR est représentée pour chaque dictionnaire sur la figure 8.2.

Comme nous l'avons prévu précédemment, la meilleure décomposition (la courbe la plus haute) est obtenue avec le dictionnaire \mathcal{D}_3 , devant le dictionnaire \mathcal{D}_2 et le dictionnaire \mathcal{D}_1 .

Les courbes des dictionnaires \mathcal{D}_3 et \mathcal{D}_2 suivent une loi de progression à peu près similaire, alors que la courbe du dictionnaire \mathcal{D}_1 croît beaucoup plus lentement lorsque la diversité relative η devient élevée. Il en résulte que l'écart entre les courbes des dictionnaires \mathcal{D}_3 et \mathcal{D}_2 varie peu (le RSB du dictionnaire \mathcal{D}_2 passe de 90% à 82% du RSB du dictionnaire \mathcal{D}_3), tandis que l'écart devient très grand avec le dictionnaire \mathcal{D}_1 (le RSB de \mathcal{D}_1 passe de 65% à 42%).

Nous pouvons en déduire que plus la DR est grande, plus le gain obtenu en laissant libre le nombre d'atomes par trame est grand. C'est cette modification, différenciant les dictionnaires \mathcal{D}_1 et \mathcal{D}_2 , qui améliore le plus la qualité de la décomposition.

Le gain en RSB est moindre, mais tout de même conséquent (10% à 20%) pour l'autre modification, qui consiste à réduire le décalage entre deux atomes successifs de $L_m/2$ à 1. Rappelons que l'utilisation de motifs courts et de la déformation de phase réduit l'avantage d'une localisation très précise des atomes (dictionnaire \mathcal{D}_3) par rapport à une localisation plus approximative (dictionnaire \mathcal{D}_2).

Globalement, à \mathcal{M} constant, le rapport signal à bruit est multiplié par deux en passant d'une décomposition par tramage du signal, avec un nombre d'atomes constants par trame, à une décomposition sur un dictionnaire invariant par translation.

8.7.7 Conclusion

Nous venons de mesurer sur un exemple l'apport de l'invariance par translation sur la qualité de la décomposition d'un signal sur un dictionnaire. Le gain en performances, doublement du RSB, est corrélé avec une augmentation de la complexité de la décomposition. En utilisant des optimisations adéquates, le coût peut être réduit et peut permettre de traiter des signaux de grande taille.

De façon plus générale, nous avons montré dans ce chapitre que l'algorithme Matching Pursuit permet non seulement de décomposer un signal sur un dictionnaire invariant par translation, mais également sur toute une classe de dictionnaires correspondant aux déformations proposées dans le chapitre 7, comme par exemple la déformation de phase, ou la convolution.

Il permet également de décomposer des signaux multicanaux, soit avec une amplitude libre sur chaque canal (MP multicanal), soit selon des fonctions d'expansion multicanale pré-tabulées (Demixing Pursuit). Nous utiliserons ces deux implémentations particulières de MP pour la séparation de sources dans le cas de mélanges multicanaux, au chapitre 10.

Chapitre 9

Apprentissage de dictionnaire structuré

Nous avons déjà évoqué au chapitre 3 l'intérêt, pour les décompositions parcimonieuses, de pouvoir apprendre un dictionnaire $\underline{\mathbf{D}}$ adapté à un ensemble de signaux. Nous avons alors effectué un tour des principales méthodes de l'état de l'art.

Dans le cadre plus restreint que nous avons introduit dans la deuxième partie de ce document, les atomes du dictionnaire sont générés par des motifs déformés selon des fonctions bien définies et paramétrées. Apprendre un dictionnaire à partir de signaux revient alors à apprendre un dictionnaire de motifs \mathcal{M} , qui par suite engendre un dictionnaire $\underline{\mathbf{D}}$ adapté aux signaux, par le biais des fonctions de déformation.

Les principes des méthodes d'apprentissage présentées au chapitre 3 restent applicables dans le nouveau cadre, mais ils nécessitent quelques aménagements spécifiques, qui sont exposés tout au long de ce chapitre. Nous allons notamment revenir à la section 9.1 sur les premières contributions concernant l'apprentissage de motifs, dans un premier temps pour les dictionnaires invariants par translation, dans un deuxième temps pour d'autres types de déformations.

A la section 9.2 sera exposée la première approche que nous avons élaborée pour l'apprentissage de motifs générant un dictionnaire par translation. Nous y présentons l'algorithme appelé MoTIF, pour Matching of Time-Invariant Filters.

Nous exposons ensuite à la section 9.3 la deuxième approche essayant de marier les atouts de l'algorithme MoTIF et de l'algorithme K-SVD, présenté au chapitre 3. L'algorithme résultant permet d'apprendre des motifs déformés, et cela pour un grand nombre de fonctions linéaires. Sans trop entrer dans les détails, l'algorithme K-SVD permet d'approcher l'objectif de maximisation jointe des motifs et des paramètres plus près que les techniques traditionnelles d'apprentissage (de type Olshausen, par exemple).

Les trois dernières sections sont consacrées aux expériences menées sur l'algorithme K-SVD. Nous vérifions dans la section 9.4 le bon comportement de l'algorithme d'apprentissage sur des données artificiellement créées en suivant le modèle parcimonieux. Dans la section 9.5, nous étudions l'intérêt de la structuration du dictionnaire (notamment l'invariance par translation et par déformation de phase) sur des signaux

synthétiques. Dans la section 9.6 nous analysons la capacité de l'algorithme K-SVD à produire des dictionnaires généralisables à des signaux de même nature que les signaux d'apprentissage.

9.1 Algorithmes existants pour l'apprentissage de dictionnaire structuré

Le problème de l'apprentissage d'un dictionnaire invariant par translation est récent. Le premier algorithme permettant d'apprendre des motifs qui sont ensuite traduits date de 2000 [Ols00]. Des travaux ont également été menés pour l'apprentissage d'atomes générés par déformation (phase, rotation, fréquence, translation) de motifs, à partir de 2003 [WK03].

Nous présentons le modèle et le principe des algorithmes d'apprentissage de dictionnaire invariant par translation dans la section 9.1.1, et invariant par d'autres déformations dans la section 9.1.2.

9.1.1 Modèle de translation des motifs

Olshausen a été le premier à intégrer une contrainte d'invariance par translation aux atomes appris en 2000 [Ols00]. Ses signaux ne sont pas des signaux monodimensionnels temporels, mais des vidéos (au sens où nous les avons définies au chapitre 7). Smith et Lewicki [SL05], et Blumensath et Davies [BD06] ont appliqué cette contrainte d'invariance par translation aux signaux monocanaux temporels en 2005, en utilisant des techniques un peu plus avancées.

Quel que soit le type de signaux traités, le modèle de décomposition sur un dictionnaire \mathcal{M} de motifs traduits est le suivant :

$$\underline{s} = \sum_{k=1}^K \sum_l e_{k,l} \cdot T_l[\underline{m}_k] + \underline{r} . \quad (9.1)$$

Il faut remarquer que ce n'est pas tout à fait le même modèle que (7.14) où nous avons supposé qu'il existe un nombre I donné d'occurrences de tous les motifs. Cela correspond à supposer dans le modèle (9.1) que seuls I coefficients $e_{k,l}$ sont non-nuls. L'équation (9.1) peut aussi s'écrire :

$$\underline{s} = \sum_{k=1}^K \underline{e}_k \star \underline{m}_k + \underline{r} , \quad (9.2)$$

où $\underline{e}_k(l) = e_{k,l}$, et \star est la convolution.

Contrairement au modèle classique décrit à la section 3.5, il ne s'agit plus d'apprendre le dictionnaire d'atomes $\underline{\mathbf{D}}$, mais simplement le dictionnaire de motifs \mathcal{M} . Le principe des algorithmes d'apprentissage est de maximiser la log-vraisemblance moyenne des signaux d'entraînement sachant le dictionnaire de motifs :

$$\mathcal{L} = E\{\log p(\underline{s}|\mathcal{M})\} . \quad (9.3)$$

Cette vraisemblance est maximisée par rapport à chacun des motifs, et le gradient par rapport à un motif \underline{m}_k s'écrit :

$$\nabla_{\underline{m}_k} \mathcal{L} = E\{E\{\frac{\partial \log p(\underline{s}|\mathcal{M}, \underline{e})}{\partial \underline{m}_k}\}_{p(\underline{e}|\underline{s}, \mathbf{D})}\} . \quad (9.4)$$

Quelle que soit la méthode pour estimer la loi de probabilité $p(\underline{e}|\underline{s}, \mathbf{D})$ et pour calculer l'espérance, l'expression située à l'intérieur peut se calculer, en vertu de l'hypothèse de bruit gaussien centré :

$$\log p(\underline{s}|\mathcal{M}, \underline{e}) = \frac{1}{2 \cdot \sigma_r^2} \|\underline{s} - \sum_{k=1}^K \underline{e}_k \star \underline{m}_k\|_2^2 , \quad (9.5)$$

et donc :

$$\frac{\partial \log p(\underline{s}|\mathcal{M}, \underline{e})}{\partial \underline{m}_k} = -\frac{1}{\sigma_r^2} \cdot \underline{r} \star \underline{e}_k^* , \quad (9.6)$$

où rappelons-le, \underline{e}_k^* est le signal \underline{e}_k inversé par rapport au temps. La convolution entre \underline{r} et \underline{e}_k^* est aussi la corrélation entre \underline{r} et \underline{e}_k .

Notons bien que le gradient contient le motif \underline{m}_k dans le terme $\underline{r} = \underline{s} - \sum_{k=1}^K \underline{e}_k \star \underline{m}_k$.

9.1.1.1 Calcul alterné des coefficients et des motifs

Dans les méthodes proposées par Olshausen et par Smith et Lewicki, la probabilité *a posteriori* $p(\underline{e}|\underline{s}, \mathbf{D})$ est approchée par la $\underline{\delta}_{\underline{e}=\hat{\underline{e}}}$, où les coefficients $\hat{\underline{e}}$ ont été estimés précédemment. Les algorithmes sont alors une alternance de calcul des coefficients pour un dictionnaire de motifs donné, puis de calcul des motifs pour des coefficients donnés.

Le calcul des coefficients est effectué par une descente de gradient pour Olshausen, avec :

$$\nabla_{e_{k,l}} \mathcal{L} = -\frac{1}{\sigma_r^2} \cdot \underline{r} \star \underline{m}_k^* + f'(\underline{e}_k) , \quad (9.7)$$

où \underline{m}_k^* est le motif \underline{m}_k inversé par rapport au temps, et f' est la dérivée de la fonction de mesure de la parcimonie f . Au cours de cette thèse, nous avons généralement rencontré $f(e) = |e|$. De son côté, Olshausen choisit la fonction $f(e) = \beta \cdot \log(1 + (e/\sigma)^2)$ pour ses propriétés de parcimonie, mais certainement aussi parce qu'elle se dérive sans problème, contrairement à la valeur absolue. Le gradient utilise la valeur précédente des coefficients pour calculer les termes $f'(\underline{e}_k)$ et \underline{r} .

$$\underline{e}_k := \underline{e}_k + \mu \cdot \left[f'(\underline{e}_k) - \frac{1}{\sigma_r^2} \cdot \left(\underline{s} - \sum_{k=1}^K \underline{e}_k \star \underline{m}_k \right) \star \underline{m}_k^* \right] \quad (9.8)$$

Smith et Lewicki utilisent l'algorithme Matching Pursuit pour sélectionner un nombre I d'atomes actifs.

L'utilisation de deux algorithmes différents conduit à deux situations différentes. Dans le premier cas, les coefficients correspondant à toutes les translations de chaque motif sont calculés. Cela correspond à l'idée d'un filtre \underline{e} qui convolue chaque motif sur toute la durée du signal. Dans le deuxième cas, seuls les I coefficients les plus élevés sont calculés. L'idée sous-jacente est alors que chaque motif est actif uniquement à un nombre fini d'instantanés du signal, et que le reste du signal est vierge de toute contribution de ce motif. Notons que notre approche suit la deuxième idée, ce qui se reflète dans la description du modèle proposé (7.14).

La mise à jour de chaque motif se fait par descente de gradient dans les deux méthodes :

$$\underline{m}_k := \underline{m}_k + \mu \cdot E\left\{\frac{1}{\sigma_{\underline{r}}^2} \cdot \underline{r} \star \underline{e}_k^*\right\}, \quad (9.9)$$

et une étape de normalisation sur les motifs est appliquée après cette mise à jour. Rappelons une fois encore que le résidu \underline{r} est calculé à partir de la valeur précédente du motif \underline{m}_k .

9.1.1.2 Échantillonnage de Gibbs

Blumensath et Davies proposent dans [BD06] d'estimer l'espérance, pour le calcul du gradient, par échantillonnage de Gibbs qui est une méthode de Monte-Carlo. Le principe est le même que les méthodes d'apprentissage par Monte-Carlo dans le modèle classique, exposées à la section 3.5.4. Rappelons qu'il s'agit d'approcher le gradient de la façon suivante :

$$E\left\{\int \frac{\partial \log p(\underline{s}|\mathcal{M}, \underline{e})}{\partial \underline{m}_k} \cdot p(\underline{e}|\underline{s}, \mathbf{D}) d\underline{e}\right\} \approx \frac{1}{N} \sum_{n=1}^N E\left\{\frac{\partial \log p(\underline{s}|\mathcal{M}, \underline{e}_n)}{\partial \underline{m}_k}\right\}, \quad (9.10)$$

où \underline{e}_n sont des jeux de coefficients tirés selon la loi $p(\underline{e}|\underline{s}, \mathbf{D})$, et pour un nombre assez important de tirages N , l'échantillonnage de Gibbs assure une bonne convergence vers l'intégrale.

Dans [BD06], les auteurs effectuent un comparatif de méthodes d'apprentissage. Il en ressort que la modélisation plus fine permise par la méthode de Monte-Carlo présentée améliore la qualité de l'apprentissage par rapport à l'algorithme proposé par Smith et Lewicki. La qualité est mesurée par la corrélation entre des motifs de référence et les motifs appris sur des signaux générés selon le modèle parcimonieux (9.1) en utilisant les motifs de référence.

Il ressort également de l'étude que la méthode de mise à jour alternée des coefficients et du dictionnaire est beaucoup plus rapide que la méthode de Monte-Carlo.

9.1.2 Modèle de déformation des motifs

Des travaux ont été menés pour structurer le dictionnaire au delà de la simple invariance par translation. Les quatre travaux que nous mentionnons dans cette section concernent le traitement d'images.

Olshausen, Sallee et Lewicki [OSL00] ont proposé en 2000 de structurer le dictionnaire comme un dictionnaire d'ondelettes pour lequel les fonctions d'ondelette et la fonction d'échelle sont apprises à partir de signaux d'entraînement. Sans entrer dans les détails, à chaque atome du dictionnaire correspond une position, une bande de fréquences et un niveau de résolution. Les coefficients qui multiplient ces atomes sont parcimonieux et sont obtenus par une descente de gradient similaire aux autres algorithmes proposés par Olshausen (voir par exemple (9.7)). Chaque fonction d'ondelette ou d'échelle est mise à jour également par descente de gradient, après avoir calculé les coefficients, le long du gradient de la vraisemblance par rapport à la fonction. La structure en ondelettes implique que le gradient est une fonction récursive sur les niveaux de résolution, mais son calcul est rapide grâce à des algorithmes prenant en compte la structure en ondelettes. Il ressort de ce travail que l'adaptation des fonctions d'ondelette et d'échelle améliore la parcimonie, et donc le pouvoir de compression par rapport à des familles analytiques classiques d'ondelettes.

En 2005, Grimes et Rao [GR05] ont proposé un modèle de décomposition bilinéaire sur un dictionnaire, par rapport à des coefficients d'activation d'un côté, et à des coefficients de déformation de l'autre côté. Ils ne l'ont appliqué qu'à la translation qui est un cas particulier. Il faut noter que cela ne concerne que des petites translations au sein de trames découpées au hasard. Le problème de tramage n'est donc pas résolu. Berkes et Wiskott, en 2006 [BW06], ont restreint le modèle aux formes quadratiques (les termes d'activation et de déformation sont les mêmes). Ils obtiennent des motifs qui peuvent être invariants par décalage de phase, rotation, changement de fréquence, changement de courbure ou dilatation. Notons que les deux modèles sont différents de celui que nous exposons au chapitre 7. Ils proviennent de la communauté des réseaux de neurones, et ils reposent sur un tramage aléatoire des signaux.

Enfin, Wersing et Körner ont en revanche décrit en 2003 [WK03] un modèle de structuration très proche de celui que nous proposons au chapitre 7, basé sur la déformation de motifs de base par une fonction linéaire. Contrairement à notre modèle, ils n'imposent *a priori* que la fonction de déformation soit décomposable comme (7.21) en fonction des paramètres de déformation. S'ils ne traitent que le cas d'invariance par translation dans l'article [WK03], ils évoquent également l'extension de la modélisation à la rotation et à la dilatation. L'algorithme d'apprentissage, similaire à l'algorithme de Olshausen, consiste à effectuer une mise à jour alternée des coefficients de décomposition et des motifs, tous deux par descente de gradient.

9.2 MoTIF : notre première approche dans le cas invariant par translation

Après avoir constaté le besoin d'intégrer l'invariance par translation aux dictionnaires appris pour pouvoir traiter des signaux temporels avec une bonne précision, nous avons élaboré un algorithme d'apprentissage de motifs en collaboration avec Philippe Jost et Pierre Vandergheynst en 2005. Il a donné lieu à deux publications [JLVG05, JLVG06].

9.2.1 Principe

L'algorithme MoTIF cherche à apprendre un ensemble de K motifs \underline{m}_k à partir de signaux d'apprentissage, dans le but ultérieur de permettre une décomposition parcimonieuse des signaux avec les versions translatées et amplifiées de ces motifs.

Pour autant, l'algorithme ne consiste pas à minimiser un critère comprenant erreur de reconstruction et poids des coefficients, comme c'est le cas dans la plupart des algorithmes d'apprentissage précédemment exposés.

MoTIF est un algorithme heuristique qui génère les motifs successivement, sans jamais revenir sur les motifs précédemment appris. Le premier motif est le plus corrélé avec les signaux d'apprentissage, et correspond à la composante principale de trames spécifiquement choisies parmi les signaux. Pour les motifs suivants, l'objectif est de trouver le motif le plus corrélé avec les signaux, mais aussi le plus décorrélé avec les motifs précédents, afin de promouvoir la diversité dans le dictionnaire de motifs.

Chaque motif est contraint à être de norme unité. Le nom MoTIF signifie Matching of Time Invariant Filters.

Présentons en premier lieu l'apprentissage du premier motif. L'ensemble d'apprentissage est constitué de Q signaux \underline{s}_q dont le support est de taille supérieure au support du motif \underline{m}_1 , qui est fixé à L_m . Nous définissons le motif optimal comme le plus corrélé en moyenne avec les signaux d'apprentissage, soit :

$$\underline{m}_1 = \arg \max_{\|\underline{m}\|_2=1} \sum_{q=1}^Q \max_{t_q} |\langle \underline{s}_q, T_{t_q}[\underline{m}] \rangle|^2. \quad (9.11)$$

En supposant que $k-1$ motifs ont déjà été appris, le motif \underline{m}_k optimal est défini comme le plus corrélé en moyenne avec les signaux d'apprentissage, tout en étant le plus décorrélé avec les $k-1$ motifs précédents, soit :

$$\underline{m}_k = \arg \max_{\|\underline{m}\|_2=1} \frac{\sum_{q=1}^Q \max_{t_q} |\langle \underline{s}_q, T_{t_q}[\underline{m}] \rangle|^2}{\sum_{k'=1}^{k-1} \sum_t |\langle \underline{m}_{k'}, T_t[\underline{m}] \rangle|^2}. \quad (9.12)$$

Trouver la meilleure solution, dans les deux cas, est un problème difficile que nous simplifions en alternant la maximisation des paramètres de translation t_q à motif \underline{m} fixé, et la maximisation du motif pour des paramètres t_q fixés.

La première étape consiste à trouver pour chaque signal \underline{s}_q l'instant t_q correspondant au maximum de corrélation entre le motif et le signal.

La deuxième étape repose sur l'utilisation de l'adjoint de la translation qui permet de transformer le produit scalaire $\langle \underline{s}_q, T_{t_q}[\underline{m}] \rangle$ en $\langle T_{-t_q}[\underline{s}_q], \underline{m} \rangle = \langle \mathbf{s}_{q,-t}, \mathbf{m} \rangle$, où \mathbf{m} est la restriction au support du motif, et $\mathbf{s}_{q,-t} = R_{t,L_m}[\underline{s}_q]$ est la trame du signal \underline{s}_q de taille L_m démarrante à l'instant t .

Nous notons $\mathbf{A} = \sum_{q=1}^Q \mathbf{s}_{q,-t} \cdot \mathbf{s}_{q,-t}^T$ et $\mathbf{B}_k = \sum_{k'=1}^{k-1} \sum_t \mathbf{m}_{k',-t} \cdot \mathbf{m}_{k',-t}^T$. Les équations (9.11) et (9.12) peuvent alors s'écrire :

$$\mathbf{m}_k = \arg \max_{\|\mathbf{m}\|_2=1} \frac{\mathbf{m}^T \cdot \mathbf{A} \cdot \mathbf{m}}{\mathbf{m}^T \cdot \mathbf{B}_k \cdot \mathbf{m}}, \quad (9.13)$$

avec $\mathbf{B}_1 = \mathbf{I}_{L_m}$, pour le premier motif.

Le motif qui résout le problème (9.13) est le vecteur propre (de norme unité) associé à la plus grande valeur propre λ du problème aux valeurs propres généralisé suivant :

$$\mathbf{A} \cdot \mathbf{m} = \lambda \cdot \mathbf{B}_k \cdot \mathbf{m} . \quad (9.14)$$

Pour la première étape, le motif optimal est le vecteur propre associé à la plus grande valeur propre de la matrice \mathbf{A} . Le motif \underline{m}_1 est donc en quelque sorte la composante principale, au sens de l'ACP. La différence se situe au niveau de la gestion de la translation, car dans le cadre de l'ACP, les motifs et les signaux sont de même taille.

9.2.2 Algorithme

Présentons un résumé de l'algorithme MoTIF :

MoTIF : Algorithme d'apprentissage de motifs tradlatés :

Initialisation du compteur $k = 0$. L'ensemble des signaux d'apprentissage est $\{\underline{s}_q\}$.

Faire

- Incrément du compteur $k := k + 1$ et initialisation du motif \underline{m}_k
- $\mathbf{B}_k := \sum_{k'=1}^{k-1} \sum_t \mathbf{m}_{k',-t} \cdot \mathbf{m}_{k',-t}^T$
- **Faire**
 - pour chaque signal \underline{s}_q , trouver $t_q := \arg \max_t |\langle \underline{s}_q, T_t[\underline{m}_k] \rangle|$, l'instant où le motif \underline{m}_k et le signal \underline{s}_q sont le plus corrélés.
 - $\mathbf{A} := \sum_{q=1}^Q \mathbf{s}_{q,-t} \cdot \mathbf{s}_{q,-t}^T$
 - trouver :

$$\mathbf{m}_k := \arg \max_{\|\mathbf{m}\|_2=1} \frac{\mathbf{m}^T \cdot \mathbf{A} \cdot \mathbf{m}}{\mathbf{m}^T \cdot \mathbf{B}_k \cdot \mathbf{m}} , \quad (9.15)$$

correspondant au vecteur propre associé à la plus grande valeur propre du problème aux valeurs propres généralisé $\mathbf{A} \cdot \mathbf{m} = \lambda \cdot \mathbf{B}_k \cdot \mathbf{m}$.

Tant que la convergence n'est pas atteinte

Tant que le nombre de motifs appris n'est pas suffisant

9.2.3 Avantages et limites

MoTIF est très rapide par rapport à d'autres algorithmes d'apprentissage, car il ne comprend pas d'étape de calcul de coefficients \mathbf{c} , ni d'estimation des lois de probabilités par méthodes de Monte-Carlo, par exemple.

La figure 9.1 représente 19 motifs appris sur les mêmes images naturelles que Olshausen et Field [OF96a] avec l'algorithme MoTIF. La taille des motifs est 16×16 , soit un support de taille $L_m = 256$. Les motifs appris sont à comparer aux motifs appris

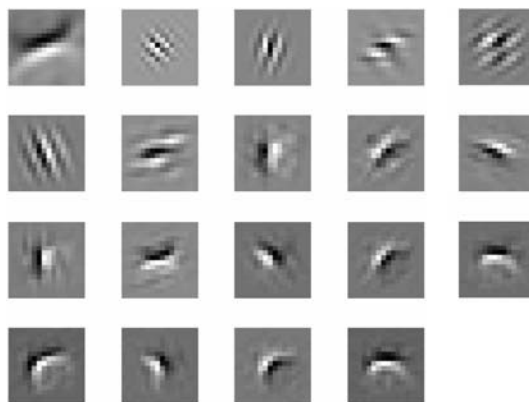


FIG. 9.1 – 19 motifs de taille 16×16 appris sur des images naturelles.

par l'algorithme de Olshausen et Field dans le cadre des dictionnaires classiques sans translation. Ils sont affichés sur la figure 5.2, dans le chapitre 5.

Les motifs appris par MoTIF sont centrés et orientés. Ils oscillent dans une direction différente de l'orientation et à des fréquences différentes. Les premiers motifs sont similaires à des atomes de Gabor, les suivants à des contours droits, et les derniers à des contours courbés.

Dans les motifs appris dans un modèle de dictionnaire classique, affichés sur la figure 5.2, nous pouvons trouver également des atomes de Gabor et des contours droits, mais pas de contours courbés. Ils ne sont pas centrés, mais au contraire localisés à différents endroits dans les motifs. La conséquence de cet éparpillement est que le même objet est appris plusieurs fois pour représenter les différentes positions à laquelle il peut se trouver. Cette redondance est évitée dans les motifs appris par MoTIF. Nous supposons que c'est cette place économisée qui a permis l'émergence de contours courbés, peut être moins représentatifs que les autres types de motifs, et donc absents des motifs appris dans le cas classique.

Ces résultats prometteurs doivent être nuancés par la présence des quelques motifs qui suivent le premier motif appris (ligne du haut), qui semblent plus influencés par la contrainte d'orthogonalité avec le premier motif que par la ressemblance aux signaux. En effet, ces motifs de haute fréquence semblent incongrus en tant que motifs les plus représentatifs des images naturelles. Les premiers motifs sont d'ailleurs très peu corrélés entre eux. Pour les motifs suivants, cet artefact semble s'atténuer pour donner des motifs plus attendus. Un traitement *a posteriori* visant à éliminer les motifs les moins représentatifs peut être envisagé.

L'absence de lien direct avec la parcimonie des coefficients fait également craindre un dictionnaire moins optimisé pour la parcimonie qu'en utilisant un algorithme minimisant classiquement l'erreur de reconstruction et le poids des coefficients de la décomposition.

L'algorithme MoTIF a été notre premier essai d'apprentissage de dictionnaire incluant la translation, et il nous a permis de remarquer le puissant intérêt que peut

apporter la propriété d'adjonction qui permet de reporter la translation (et plus largement la déformation linéaire) du motif au signal.

Afin de tirer le meilleur des deux approches, nous avons intégré les translations et les déformations linéaires par ce biais dans les algorithmes classiques minimisant l'erreur de reconstruction et le poids des coefficients de la décomposition. Nous exposons ainsi à la section suivante l'adaptation de l'algorithme K-SVD à la translation et aux déformations linéaires de motifs.

9.3 Algorithme de type K-SVD : notre principale contribution

Nous avons vu dans la section précédente que l'algorithme MoTIF ne repose que sur la corrélation entre les motifs et les signaux et sur la décorrélation entre les motifs. La parcimonie n'est pas prise en compte par MoTIF les motifs appris n'ont donc aucune raison de donner une décomposition très parcimonieuse des signaux d'apprentissage. Afin de remédier à ce manque, nous avons décidé d'élaborer une méthode d'apprentissage qui tienne compte de la parcimonie.

Nous avons opté pour l'adaptation de l'algorithme K-SVD, présenté à la section 3.5.2, au modèle d'atomes contraints (7.21). Rappelons que l'algorithme K-SVD vise à maximiser la vraisemblance des données connaissant le dictionnaire, et qu'il fait partie de la famille d'algorithmes qui décompose l'apprentissage en deux étapes : calcul des coefficients à dictionnaire fixé, et mise à jour du dictionnaire à partir des coefficients précédemment calculés. Rappelons encore que si pour la plupart des algorithmes de cette famille, le dictionnaire est mis à jour alors que les coefficients sont fixés, de son côté, l'algorithme K-SVD met à jour le dictionnaire mais aussi les coefficients lors de la deuxième étape. L'objectif de la deuxième étape est alors de minimiser l'erreur de reconstruction par rapport au dictionnaire et aux coefficients, sans hypothèse de parcimonie sur les coefficients.

9.3.1 Mise à jour alternée des paramètres et des motifs

Nous avons choisi un algorithme basé sur le calcul alterné des coefficients et du dictionnaire parce qu'il permet d'intégrer le modèle d'atomes déformés de façon simple. En effet, il suffit de savoir calculer d'une part les paramètres du modèle ψ , et ϕ , à dictionnaire fixé, et d'autre part la dérivée de l'erreur de reconstruction par rapport à chaque motif :

$$\frac{\partial \|\check{\underline{s}} - \sum_{i_k=1}^{I_k} F\psi_{i_k, \phi_{i_k}}[m_k]\|_2^2}{\partial \underline{m}_k} = \sum_{i'_k=1}^{I_k} F\psi_{i'_k, \phi_{i'_k}}^* \left[\check{\underline{s}} - \sum_{i_k=1}^{I_k} F\psi_{i_k, \phi_{i_k}}[m_k] \right], \quad (9.16)$$

avec $\check{\underline{s}} = \underline{s} - \sum_{k' \neq k} \sum_{i_{k'}=1}^{I_{k'}} F\psi_{i_{k'}, \phi_{i_{k'}}}[m_{k'}]$ le signal auquel ont été soustraites les contributions des autres motifs. Cette règle est équivalente à (9.6) lorsque la déformation est une convolution.

Les méthodes qui ont été proposées pour la translation dans la section 9.1.1 sont basées sur une descente de gradient. Dans une partie de l'article que nous avons coécrit sur l'apprentissage de dictionnaires structurés [LMG06], Mailhé et Gribonval proposent de calculer les motifs en minimisant le critère quadratique par rapport au motif, en gardant les coefficients fixés, sans recourir au gradient. La méthode élaborée par Mailhé et Gribonval est présentée à la section 9.3.2, tandis que la méthode issue de l'algorithme K-SVD, que nous proposons, est présentée à la section 9.3.3.

9.3.2 Algorithme voisin proposé par Mailhé et Gribonval

La méthode proposée par Mailhé et Gribonval consiste à mettre à jour les motifs un à un, alors que les paramètres $\boldsymbol{\psi}$ et ϕ sont calculés à l'étape précédente et fixés. L'objectif est de trouver le motif de norme unité minimisant l'erreur résiduelle quadratique :

$$\underline{m}_k^{\text{opt}} = \arg \min_{\|\underline{m}_k\|_2=1} \left\| \check{\underline{s}} - \sum_{i_k=1}^{I_k} F\boldsymbol{\psi}_{i_k, \phi_{i_k}} [\underline{m}_k] \right\|_2^2. \quad (9.17)$$

L'algorithme proposé repose sur l'hypothèse que les fonctions $F\boldsymbol{\psi}_{i_k, \phi_{i_k}}$ sont unitaires (elles préservent la norme), et que les atomes sont orthogonaux entre eux :

$$\langle F\boldsymbol{\psi}_{i'_k, \phi_{i'_k}} [\underline{m}_k], F\boldsymbol{\psi}_{i_k, \phi_{i_k}} [\underline{m}_k] \rangle = 0, \quad \forall i'_k \neq i_k. \quad (9.18)$$

Cela correspond par exemple au cas des dictionnaires invariants par translation, avec l'hypothèse que les atomes ne se recouvrent pas temporellement.

Le problème peut s'écrire :

$$\underline{m}_k^{\text{opt}} = \arg \min_{\|\underline{m}_k\|_2=1} \left[\|\check{\underline{s}}\|_2^2 + \sum_{i_k=1}^{I_k} \|F\boldsymbol{\psi}_{i_k, \phi_{i_k}} [\underline{m}_k]\|_2^2 - 2 \cdot \langle \check{\underline{s}}, \sum_{i_k=1}^{I_k} F\boldsymbol{\psi}_{i_k, \phi_{i_k}} [\underline{m}_k] \rangle \right] \quad (9.19)$$

$$= \arg \max_{\|\underline{m}_k\|_2=1} \left\langle \check{\underline{s}}, \sum_{i_k=1}^{I_k} F\boldsymbol{\psi}_{i_k, \phi_{i_k}} [\underline{m}_k] \right\rangle \quad (9.20)$$

$$= \arg \max_{\|\underline{m}_k\|_2=1} \left\langle \sum_{i_k=1}^{I_k} F\boldsymbol{\psi}_{i_k, \phi_{i_k}}^* [\check{\underline{s}}], \underline{m}_k \right\rangle. \quad (9.21)$$

Le passage de (9.19) à (9.20) s'explique par le fait que les deux premiers termes de la somme ne dépendent pas du motif. En effet, si c'est évident pour la norme du signal, le deuxième terme est également indépendant du motif, puisque par unitarité de la fonction de déformation, ce terme est égal à la norme du motif, qui vaut 1 selon la contrainte imposée.

Le passage de (9.20) à (9.21) correspond à l'application de l'adjoint de la fonction de déformation. Cet adjoint existe puisque la fonction de déformation est linéaire.

Le motif optimal est donc le vecteur de norme unité colinéaire au vecteur situé à gauche dans le produit scalaire de (9.21) :

$$\underline{m}_k^{\text{opt}} = \frac{\sum_{i_k=1}^{I_k} F^* \boldsymbol{\psi}_{i_k, \phi_{i_k}} [\check{\underline{s}}]}{\left\| \sum_{i_k=1}^{I_k} F^* \boldsymbol{\psi}_{i_k, \phi_{i_k}} [\check{\underline{s}}] \right\|_2}, \quad (9.22)$$

ce qui revient à une moyenne normalisée des “atomes duaux”, correspondant au signal $\check{\underline{s}}$ déformé par l’adjoint des fonctions de déformations utilisant les paramètres $\boldsymbol{\psi}_{i_k}$ et ϕ_{i_k} des atomes.

Dans le cas d’un dictionnaire invariant par translation où les atomes sont définis par $c_{i_k} \cdot T_{t_{i_k}}[\underline{m}_k]$, le motif optimal est la moyenne normalisée des trames de signal localisées au même instant t_{i_k} que les atomes, pondérées par l’amplitude c_{i_k} des atomes :

$$\underline{m}_k^{\text{opt}} = \frac{\sum_{i_k=1}^{I_k} c_{i_k} \cdot T_{-t_{i_k}}[\check{\underline{s}}]}{\left\| \sum_{i_k=1}^{I_k} c_{i_k} \cdot T_{-t_{i_k}}[\check{\underline{s}}] \right\|_2}. \quad (9.23)$$

Cette méthode a le mérite de la rapidité, puisqu’il s’agit uniquement d’une moyenne et d’une normalisation. En revanche, elle est restreinte par l’impossibilité de traiter des atomes non-orthogonaux entre eux. Cette restriction peut être contournée comme nous l’expliquons dans la section 9.3.4.1 pour l’algorithme K-SVD que nous exposons dans la section suivante.

9.3.3 Contribution majeure : algorithme de type K-SVD pour l’apprentissage de dictionnaire invariant par déformation linéaire

Nous avons présenté dans la section précédente les travaux de Maillé car ils sont très proches des nôtres et sont apparus dans la même période. Nous avons choisi une voie légèrement différente avec l’algorithme K-SVD. Il s’agit de minimiser l’erreur de reconstruction par rapport au motif \underline{m}_k mais aussi par rapport aux paramètres continus $\boldsymbol{\psi}_k$, tandis que les paramètres discrets ϕ_k calculés à l’étape précédente sont fixés. Nous faisons ce choix pour nous rapprocher de l’objectif premier, qui est l’optimisation jointe sur les motifs et sur tous les paramètres $\boldsymbol{\psi}$ et ϕ . Ce choix découle également de la chronologie du travail de thèse. En effet, nous avons d’abord élaboré l’algorithme d’apprentissage de dictionnaire formé d’une union de bases orthonormées, présenté au chapitre 6. Aharon et Elad ont ensuite créé l’algorithme K-SVD en suivant une ligne assez similaire. Nous avons donc naturellement décidé de fondre nos travaux sur l’invariance par translation et déformation avec cette famille d’algorithmes et notamment l’algorithme K-SVD.

L’algorithme que nous présentons dans cette section vise à maximiser la vraisemblance des données connaissant le dictionnaire. La tâche est divisée en deux : estimation des paramètres $\boldsymbol{\psi}$ et ϕ dans un premier temps, mise à jour des motifs \underline{m}_k et des paramètres $\boldsymbol{\psi}_i$ correspondant dans un deuxième temps. La première étape peut être avantageusement effectuée par Matching Pursuit (pour les raisons, voir la section 8.1).

La deuxième étape correspond à la minimisation suivante :

$$(\underline{m}_k, \{\psi_i\}_{k_i=k}) = \arg \min_{(\underline{m}, \{\psi_i\})} \left\| \underline{\check{s}} - \sum_{i / k_i=k} F_{\psi_i, \phi_i}[\underline{m}] \right\|_2^2 \quad (9.24)$$

Comme nous le montrons dans la section suivante, grâce à la linéarité de la déformation, et donc à l'existence d'un adjoint, le motif optimal \mathbf{m}^{opt} (représenté par son support) peut être exprimé comme la solution d'un problème du type :

$$\mathbf{m}_k = \arg \max_{\|\mathbf{m}\|_2} \mathbf{m}^T \cdot \mathbf{A}(\mathbf{m}) \cdot \mathbf{m} , \quad (9.25)$$

où la matrice $\mathbf{A}(\mathbf{m})$ est de taille $L_m \times L_m$ et symétrique.

Dans le cas où la matrice $\mathbf{A}(\mathbf{m})$ est indépendante du motif \mathbf{m} , le motif optimal est un des vecteurs propres de norme unité associés à la plus grande valeur propre de la matrice \mathbf{A} .

9.3.3.1 Détails de la résolution

Factorisons les paramètres des différents atomes sous la forme :

$$\psi_k^T = [\psi_1^T \ \dots \ \psi_{I_k}^T] \quad (9.26)$$

et les bases de fonctions par :

$$\mathbf{F}_{\phi_k}^T = [\mathbf{F}_{\phi_1}^T \ \dots \ \mathbf{F}_{\phi_{I_k}}^T] . \quad (9.27)$$

La somme des atomes issus du motif \underline{m}_k peut alors s'écrire :

$$\sum_{i / k_i=k} F_{\psi_i, \phi_i}[\underline{m}] = \psi_k^T \cdot \mathbf{F}_{\phi_k}[\underline{m}] . \quad (9.28)$$

En utilisant cette nouvelle écriture, les paramètres ψ_k peuvent être optimisés grâce à la règle (8.6) :

$$\psi_k^{\text{opt}} = \mathbf{G}_{k, \phi_k}^{-1} \cdot \mathbf{v}_{k, \phi_k} , \quad (9.29)$$

en suivant les notations définies à la section 8.2. En remplaçant les paramètres ψ_k par leur valeur optimale, le problème d'optimisation du motif devient alors :

$$\underline{m}_k = \arg \max_{\underline{m}} \mathbf{v}_{k, \phi_k}^T \cdot \mathbf{G}_{k, \phi_k}^{-1} \cdot \mathbf{v}_{k, \phi_k} . \quad (9.30)$$

Remarquons tout d'abord qu'en appliquant l'adjoint de la déformation, et en remplaçant le produit scalaire entre signaux en un produit scalaire entre les restrictions au support du motif, nous obtenons :

$$\langle \underline{\check{s}} , F_{\phi_k}^i[\underline{m}] \rangle = \langle F_{\phi_k}^{i*}[\underline{\check{s}}] , \underline{m} \rangle \quad (9.31)$$

$$= \left(R_{0, L_m}^* \circ F_{\phi_k}^{i*}[\underline{\check{s}}] \right)^T \cdot \mathbf{m} , \quad (9.32)$$

où $R_{0,L_m}^* \circ F_{\phi_k}^{i*}[\underline{s}]$ est un vecteur de $\mathbb{R}^{L_m \times 1}$, comme \mathbf{m} . La ligne i du vecteur \mathbf{v}_{k,ϕ_k} étant $\langle \underline{s}, F_{\phi_k}^i[\underline{m}] \rangle$, nous pouvons remplacer ce vecteur par le produit matriciel suivant :

$$\mathbf{v}_{k,\phi_k} = \mathbf{F}_{\phi_k}^*[\underline{s}] \cdot \mathbf{m} , \quad (9.33)$$

où les lignes de la matrice $\mathbf{F}_{\phi_k}^*[\underline{s}]$, représentée avec le symbole de l'adjoint par abus de notation, sont les vecteurs ligne $\left(R_{0,L_m}^* \circ F_{\phi_k}^{i*}[\underline{s}] \right)^T$. L'objectif de l'apprentissage devient la maximisation du critère suivant :

$$\mathbf{m}^T \cdot \left(\mathbf{F}_{\phi_k}^*[\underline{s}]^T \cdot \mathbf{G}_{k,\phi_k} \cdot \mathbf{F}_{\phi_k}^*[\underline{s}] \right) \cdot \mathbf{m} = \mathbf{m}^T \cdot \mathbf{A}(\mathbf{m}) \cdot \mathbf{m} . \quad (9.34)$$

Si la matrice d'autocorrélation \mathbf{G}_{k,ϕ_k} dépend du motif \mathbf{m} , alors la matrice

$$\mathbf{A}(\mathbf{m}) = \left(\mathbf{F}_{\phi_k}^*[\underline{s}]^T \cdot \mathbf{G}_{\phi_k}^{-1} \cdot \mathbf{F}_{\phi_k}^*[\underline{s}] \right) \quad (9.35)$$

dépend aussi du motif \mathbf{m} et nous ne savons pas résoudre ce problème d'optimisation de façon exacte en faisant appel à l'algèbre linéaire. Nous reviendrons sur ce problème difficile dans les perspectives au chapitre 11.

En revanche, si la matrice $\mathbf{A}(\mathbf{m})$ est indépendante du motif \mathbf{m} , alors le motif optimal peut être obtenu de façon analytique : le motif optimal \mathbf{m}_k est le vecteur propre, de norme unité, associé à la plus grande valeur propre de la matrice \mathbf{A} . La preuve de ce résultat est disponible dans l'annexe A.3.

9.3.3.2 Conditions sur la matrice \mathbf{G}_{k,ϕ_k}

Sans donner de conditions générales selon lesquelles la matrice \mathbf{G}_{k,ϕ_k} est indépendante du motif \mathbf{m} , nous pouvons décrire le cas particulier où elle est égale à l'identité.

Ce cas simple est obtenu lorsque les différentes occurrences du motif sont orthogonales entre elles et lorsque la base \mathbf{F}_ϕ de fonctions associées à la déformation est orthonormale.

Si par exemple la déformation comprend une translation et si ses fonctions de base sont orthonormales, alors il suffit que les occurrences des motifs ne se recouvrent pas pour que la matrice \mathbf{G}_{k,ϕ_k} soit la matrice identité.

9.3.3.3 Intérêt de la linéarité de la déformation

La propriété de linéarité de la déformation permet l'utilisation de l'adjoint de la déformation dans la formule (9.31), et par conséquent l'expression du critère à maximiser comme une forme quadratique du motif : $\mathbf{m}^T \cdot \mathbf{A}(\mathbf{m}) \cdot \mathbf{m}$. Rappelons que la linéarité permet également le calcul des paramètres ψ_k selon la règle (8.6).

9.3.3.4 Description de l'algorithme K-SVD pour l'apprentissage de dictionnaire invariant par déformation

Nous décrivons ici l'algorithme K-SVD destiné à l'apprentissage d'un dictionnaire de motifs, générant par déformation linéaire un dictionnaire d'atomes adapté aux signaux

d'apprentissage. Il faut noter que dans le modèle classique où les motifs sont les atomes du dictionnaire et sont de la taille des signaux, il était nécessaire d'utiliser un grand nombre Q de signaux d'apprentissage. Dans le modèle structuré, où nous supposons implicitement que la déformation contient la translation, c'est-à-dire que les motifs déformés peuvent être placés à n'importe quel endroit du signal, il n'est pas nécessaire de faire la distinction entre plusieurs signaux d'apprentissage, et pour alléger les notations, nous supposons qu'il n'y a qu'un signal d'apprentissage suffisamment long et divers (il peut être formé de la concaténation de signaux d'apprentissage).

Algorithme K-SVD pour l'apprentissage de dictionnaire invariant par déformation :

Initialisation du dictionnaire de motifs $\mathcal{M} = \{\underline{m}_k\}_{1 \leq k \leq K}$ (les motifs peuvent être de taille différente et sont normés).

Faire

1. Décomposition du signal d'entraînement \underline{s} sur le dictionnaire par Matching Pursuit :

$$\underline{s} = \sum_{i=1}^I F_{\psi_i, \phi_i}[\underline{m}_{k_i}] + \underline{r}_I \quad (9.36)$$

2. **Pour** tout motif \underline{m}_k , $k \in \llbracket 1, K \rrbracket$, du dictionnaire \mathcal{M}
Faire

- a) Retrait de la contribution des autres motifs dans le signal :

$$\check{\underline{s}} := \underline{s} - \sum_{i / k_i \neq k} F_{\psi_i, \phi_i}[\underline{m}_{k_i}] = \sum_{i / k_i = k} \psi_i^T \cdot \mathbf{F}_{\phi_i}[\underline{m}_k] + \underline{r}_I, \quad (9.37)$$

- b) Création des variables suivantes :

- l'ensemble des indices d'atomes générés à partir de \underline{m}_k : $I_k = \{i / k_i = k\}$, de taille N_k
- regroupement de tous les atomes issus de \underline{m}_k en un seul : $\psi_k^T \cdot \mathbf{F}_{\phi_k}$, avec $\psi_k^T = [\psi_{i_1}^T, \dots, \psi_{i_{N_k}}^T]$ et $\mathbf{F}_{\phi_k}^T = [\mathbf{F}_{\phi_{i_1}}^T, \dots, \mathbf{F}_{\phi_{i_{N_k}}}^T]$
- la matrice d'autocorrélation, \mathbf{G}_{k, ϕ_k} , qui doit être indépendante de \underline{m} , est de terme $(i_1, i_2) : \langle F_{\phi_k}^{i_1}[\underline{m}], F_{\phi_k}^{i_2}[\underline{m}] \rangle$.
- la matrice $\mathbf{F}_{\phi_k}^*[\check{\underline{s}}]^T = [R_{0, L_{m_k}}^* \circ F_{\phi_k}^1[\check{\underline{s}}], \dots, R_{0, L_{m_k}}^* \circ F_{\phi_k}^{N_k}[\check{\underline{s}}]]$

- c) Calcul de la matrice $\mathbf{A} = \left(\mathbf{F}_{\phi_k}^*[\check{\underline{s}}]^T \cdot \mathbf{G}_{k, \phi_k}^{-1} \cdot \mathbf{F}_{\phi_k}^*[\check{\underline{s}}] \right)$

- d) Mise à jour du motif \underline{m}_k : son support \mathbf{m}_k est un des vecteurs propres unitaires associés à la plus grande valeur propre de la matrice \mathbf{A} .

e) Mise à jour des paramètres $\boldsymbol{\psi}_k$:

$$\boldsymbol{\psi}_k := \mathbf{G}_{k,\phi_k}^{-1} \cdot \mathbf{F}_{\phi_k}^* [\underline{\check{s}}] \cdot \mathbf{m}_k \quad (9.38)$$

f) Mise à jour du résidu :

$$\underline{r} := \underline{\check{s}}_q - \boldsymbol{\psi}_k^T \cdot \mathbf{F}_{\phi_k} [\underline{m}_k] . \quad (9.39)$$

Tant que la convergence n'est pas atteinte

9.3.4 Détail pour la translation et la déformation de phase

Les déformations linéaires détaillées dans la section 7.5 n'entrent pas toutes dans le cadre de l'algorithme K-SVD, à cause de la condition d'indépendance de la matrice \mathbf{G}_{k,ϕ_k} au motif \underline{m} . Ainsi, si l'amplification, la déformation de phase et l'expansion multicanale ne posent pas de problème, et la translation non plus, à condition que les occurrences ne se recouvrent pas, en revanche la convolution ne convient pas dans le cas général.

Nous donnons ici les déclinaisons de l'algorithme K-SVD pour apprendre les dictionnaires invariant par translation et invariant par translation et déphasage.

9.3.4.1 Dictionnaire invariant par translation

Nous avons mentionné que la translation n'entre dans le cadre de K-SVD que si les occurrences d'un même motif ne se recouvrent pas. Supposons dans un premier temps que c'est le cas. Nous donnerons en fin de section une adaptation de l'algorithme permettant de se placer dans les bonnes conditions.

Rappelons que pour un dictionnaire invariant par translation, les atomes sont définis par $c_i \cdot T_{t_i} [\underline{m}_{k_i}]$, où c_i est l'amplitude, t_i le décalage temporel, et \underline{m}_{k_i} le motif générateur.

Les atomes ne se recouvrant pas temporellement, la matrice \mathbf{G}_{k,ϕ_k} est la matrice identité (grâce à la normalisation des motifs). La matrice \mathbf{A} peut alors s'écrire :

$$\mathbf{A} = \sum_{i \in I_k} R_{t_i, L_{m_k}}^* [\underline{\check{s}}]^T \cdot R_{t_i, L_{m_k}} [\underline{\check{s}}] , \quad (9.40)$$

où les vecteurs $R_{t_i, L_{m_k}}^* [\underline{\check{s}}]$ sont les trames du signal $\underline{\check{s}}$ de longueur L_{m_k} situées aux instants t_i .

Le motif est le vecteur propre (ou un des vecteurs propres) associé à la plus grande valeur propre de la matrice \mathbf{A} . Dans le cas de l'invariance par translation, cela signifie que le motif choisi est le plus corrélé en moyenne avec les trames du signal $\underline{\check{s}}$ situées aux instants t_i .

Cette optimisation correspond à la première étape de l'algorithme MoTIF défini à la section 9.2. Rappelons que cette étape était effectuée une seule fois dans MoTIF, et que les autres motifs étaient calculés différemment. Dans K-SVD, en revanche, tous les motifs sont appris avec cette règle et chaque motif est mis à jour plusieurs fois jusqu'à convergence, alors que dans MoTIF, les motifs sont calculés en une fois.

En comparaison, l'algorithme proposé par Mailhé et Gribonval effectue une moyenne de ces trames, pondérées par les coefficients d'amplitude calculés par Matching Pursuit. Les algorithmes du gradient, quant à eux, déplacent le motif dans la direction de la moyenne pondérée des trames du résidu.

Revenons à l'algorithme K-SVD. Les coefficients d'amplitude sont calculés directement à partir du nouveau motif \underline{m}_k par :

$$c_i = \langle \underline{\check{x}}, T_{t_i}[\underline{m}_k] \rangle. \quad (9.41)$$

Intéressons-nous maintenant au cas où les occurrences des motifs se recouvrent. Si l'algorithme Matching Pursuit est utilisé pour calculer la décomposition, et donc les paramètres t_i et c_i , alors pour un signal et des motifs quelconques, à partir d'un certain nombre d'itérations, il est certain que les occurrences des motifs commenceront à se chevaucher.

Dans ce cas, seule la mise à jour du motif pose problème, car la mise à jour des coefficients d'amplitude, regroupés dans le vecteur ψ_k , peut se faire par la règle générale :

$$\psi_k := \mathbf{G}_{k,\phi_k}^{-1} \cdot \mathbf{F}_{\phi_k}^*[\underline{\check{x}}] \cdot \mathbf{m}_k \quad (9.42)$$

où la matrice \mathbf{G}_{k,ϕ_k} tient compte des recouvrements. Nous nous intéressons donc maintenant uniquement à la façon de calculer le motif \underline{m}_k .

Les atomes peuvent être divisés en deux catégories : ceux qui ne recouvrent temporairement aucun autre atome formé par le même motif et tiré avant, et ceux qui recouvrent une occurrence précédente du même motif.

Une première stratégie basique serait d'approcher la matrice \mathbf{G}_{k,ϕ_k} par l'identité, mais serait assez éloignée de la réalité lorsque de nombreux atomes se recouvrent puisque les termes non-diagonaux correspondant ne sont pas nuls. Une deuxième approche consisterait à arrêter la décomposition au premier atome qui recouvre une autre occurrence du même motif, mais cela risque de restreindre la diversité des trames sélectionnées et de mener à un motif peu adapté au signal. Nous avons choisi une troisième approche qui consiste à n'utiliser que les atomes du premier groupe pour le calcul du motif, mais à tous les utiliser pour la mise à jour des coefficients d'amplitude et du résidu.

9.3.4.2 Dictionnaire invariant par translation et déphasage

Nous détaillons maintenant l'algorithme lorsque les atomes sont créés par translation et déformation de phase d'un motif : $T_{t_i} \circ P_{\psi_i}[\underline{m}_{k_i}] = \psi_{i,1} \cdot T_{t_i}[\underline{m}_{k_i}] + \psi_{i,2} \cdot T_{t_i} \circ H[\underline{m}_{k_i}]$. Rappelons qu'afin de rendre la déformation unitaire (et donc de préserver la norme), nous avons imposé dans la section 8.4 que les motifs sont de moyenne et de composante fréquentielle de Nyquist nulles.

Nous ne revenons pas sur la gestion des atomes qui se recouvrent, le problème et les solutions restent les mêmes. En supposant donc que les occurrences d'un même motif ne se recouvrent pas, la matrice \mathbf{G}_{k,ϕ_k} est de nouveau égale à l'identité. En effet, alors que les produits scalaires entre occurrences non recouvrantes sont évidemment nuls, les

produits scalaires diagonaux sont égaux à 1, puisque l'identité et la partie imaginaire de la transformation de Hilbert conservent la norme (pour une moyenne nulle, rappelons-le). Enfin, les produits scalaires entre un motif et la partie imaginaire de sa transformée de Hilbert sont nuls, par propriété de la transformée de Hilbert (démonstration en annexe A.7).

La matrice \mathbf{A} est donc :

$$\mathbf{A} = \sum_{i \in I_k} \left(R_{t_i, L_{m_k}}^* [\underline{\check{s}}] \right)^T \cdot \left(R_{t_i, L_{m_k}}^* [\underline{\check{s}}] \right) + \left(H \circ R_{t_i, L_{m_k}}^* [\underline{\check{s}}] \right)^T \cdot \left(H \circ R_{t_i, L_{m_k}}^* [\underline{\check{s}}] \right), \quad (9.43)$$

où les vecteurs $R_{t_i, L_{m_k}}^* [\underline{\check{s}}]$ sont les trames du signal $\underline{\check{s}}$ de longueur L_{m_k} situées aux instants t_i .

Le motif optimal est le vecteur propre de \mathbf{A} correspondant à la plus grande valeur propre, ce qui signifie qu'il est le plus corrélé en moyenne avec les sous-espaces engendrés par les trames du signal $\underline{\check{s}}$ et la partie imaginaire de leur transformée de Hilbert.

Les paramètres sont obtenus par :

$$\psi_{i,1} = \langle \underline{\check{s}}, T_{t_i}[\underline{m}_k] \rangle \quad (9.44)$$

$$\psi_{i,2} = \langle \underline{\check{s}}, T_{t_i} \circ H[\underline{m}_k] \rangle. \quad (9.45)$$

9.4 Evaluation de la capacité à favoriser la parcimonie

Nous nous intéressons maintenant à l'étude expérimentale de l'algorithme de type K-SVD présenté dans la section précédente. Nous évaluerons au cours des trois sections à venir, dans le cadre de l'apprentissage de dictionnaire utilisant cet algorithme :

1. la capacité à apprendre un dictionnaire parcimonieux (section 9.4),
2. l'intérêt de structurer le dictionnaire (section 9.5),
3. les conditions pour éviter le sur-apprentissage (section 9.6),

Le premier point est du même ordre que l'évaluation faite pour l'apprentissage de bases orthonormées, en section 6.2. Il s'agira d'apprendre un dictionnaire à partir de données artificielles créées parcimonieusement à partir d'un dictionnaire de référence. Nous évaluerons, en termes de rappel et de précision, la proximité du dictionnaire appris avec le dictionnaire de référence, lorsque les signaux sont très parcimonieux dans le dictionnaire de référence. Nous verrons également comment l'algorithme peut faire le choix d'un dictionnaire éloigné du dictionnaire de référence si la parcimonie des signaux peut y gagner.

Le deuxième point va plus loin que l'expérience de la section 8.7, qui montrait uniquement que le tramage est néfaste pour la décomposition de signaux temporels, et que pour un même dictionnaire de motifs, la décomposition peut être améliorée en augmentant l'invariance par translation. Ici, nous allons montrer sur des signaux synthétiques que le tramage casse complètement les motifs réellement présents dans le signal. Nous montrerons que l'invariance par translation permet de remonter à des motifs plus typiques du signal, et ainsi d'augmenter la parcimonie de la décomposition. Nous

montrons également l'apport de l'invariance par phase sur la qualité du dictionnaire appris.

Le troisième point concerne la généralisation du dictionnaire pour traiter des signaux de même nature que les signaux d'entraînement, mais différents. Nous montrerons sous quelles conditions sur les tailles respectives du dictionnaire et des signaux d'apprentissage nous pouvons raisonnablement assurer que l'algorithme n'a pas sur-appris les formes des signaux d'apprentissage, et que le dictionnaire appris pourra décomposer d'autres signaux de la même nature à peu près aussi parcimonieusement.

Dans cette section, nous étudions le premier point.

9.4.1 Métrique

Dans cette expérience où nous allons mesurer la ressemblance entre un dictionnaire appris \mathcal{D}_{app} et un dictionnaire de référence \mathcal{D}_{ref} , nous allons utiliser le corrélogramme, et les mesures de rappel et de précision qui en découlent, comme à la section 6.2. Nous devons cependant redéfinir le corrélogramme, car le modèle d'atome structuré ne permet plus d'utiliser la définition classique.

Dans la définition classique pour des dictionnaires non-structurés, les atomes sont les motifs et le corrélogramme représente les produits scalaires entre tous les atomes. Ainsi, la case (i, j) du corrélogramme contient le produit scalaire entre les atomes \underline{d}_i^{ref} et \underline{d}_j^{app} , ce qui permet de mesurer la proximité entre les deux espaces qu'ils engendrent, via l'amplitude qui leur est associée.

Dans cette section, les atomes sont engendrés par amplification et translation des motifs. Dans ce cadre des atomes structurés, il ne s'agit plus de mesurer la ressemblance entre tous les atomes, mais uniquement entre les K espaces engendrés par les motifs.

Nous définissons alors la case (i, j) du corrélogramme comme le maximum de corrélation entre deux atomes engendrés par les motifs \underline{m}_i^{ref} et \underline{m}_j^{app} :

$$\mathbf{C}_{i,j}(\mathcal{D}_{ref}, \mathcal{D}_{app}) = \max_t |\langle T_t[\underline{m}_i^{ref}], \underline{m}_j^{app} \rangle| \quad (9.46)$$

La mesure est indépendante du positionnement de la forme d'onde à l'intérieur du motif appris. Par exemple, supposons qu'une forme d'onde typique du signal est de taille 16 et que les motifs appris sont de taille 32. Supposons également que deux motifs ont appris cette forme d'onde, qu'elle est placée sur les 16 premiers échantillons du premier motif, et sur les 16 derniers échantillons du deuxième motif, le reste des motifs étant nul. Alors que les deux motifs sont orthogonaux entre eux, la mesure de corrélation est de 1, ce qui est satisfaisant puisqu'ils engendrent les mêmes atomes.

Remarquons que la mesure dépend uniquement de l'écart de translation entre deux atomes, et non pas de la translation absolue de chaque atome. Cette mesure est donc globale au sens où elle ne dépend pas d'une paire particulière de paramètres de translation.

Pour rendre compte de l'expérience, nous présenterons des mesures de type rappel et précision entre un dictionnaire de référence et un dictionnaire appris. Pour le rappel, nous afficherons, pour chaque motif de référence, la corrélation maximale avec un

motif appris. Pour la précision, nous afficherons pour chaque motif appris la corrélation maximale avec un motif de référence. En plus d'afficher ces nuages de points, nous les résumerons grâce à la valeur médiane parmi tous les motifs. Nous n'afficherons pas la corrélation, mais la distance de la corrélation à la corrélation maximale, 1. Une distance proche de 0 signifiera donc deux motifs très corrélés. Une échelle logarithmique permettra d'analyser plus précisément la distance à la corrélation maximale.

Dans l'expérience suivante, nous mesurerons également la qualité de la décomposition des signaux d'entraînement sur le dictionnaire appris et sur le dictionnaire de référence, grâce à la courbe du RSB en fonction de la densité relative, pour voir si l'algorithme d'apprentissage favorise la parcimonie, et pour détecter à quel moment le dictionnaire appris s'éloigne du dictionnaire de référence pour décomposer le signal avec une plus grande qualité.

Nous introduisons également la notion de densité relative pondérée (DRP) qui pénalise la longueur des atomes, par rapport à la DR. Elle est définie par :

$$\mu_I = \frac{\sum_{i=1}^I L_{d_i}}{L}, \quad (9.47)$$

où L_{d_i} désigne la longueur du support de l'atome i . Si tous les atomes ont un support de même taille que le motif qui les a générés et que tous les motifs sont de même taille L_m , alors le numérateur est égal à $I \cdot L_m$. En revanche, si la déformation modifie la taille du support (comme la convolution, par exemple), L_{d_i} est différente de L_m . Si d'un autre côté, les atomes sont générés avec des motifs de taille différente, alors le numérateur est $\sum_{k=1}^K I_k \cdot L_{m_k}$, c'est-à-dire le nombre d'atomes générés par chaque motif, multiplié par la longueur du motif.

La mesure correspond au nombre d'échantillons des atomes rapporté au nombre d'échantillons du signal. Elle est particulièrement intéressante pour comparer des dictionnaires appris utilisant des motifs de taille différente. Elle permet de donner une indication sur l'utilité d'apprendre de grands motifs, plus consommateurs de paramètres libres que les petits motifs.

9.4.2 Intentions expérimentales

L'objectif de cette expérience est de vérifier la capacité de l'algorithme de type K-SVD à apprendre des motifs permettant de décomposer parcimonieusement les signaux d'apprentissage.

Nous regarderons notamment, sur des signaux artificiels, comment se passe la transition entre signaux parcimonieux et signaux non-parcimonieux sur le dictionnaire de référence. Normalement, le dictionnaire appris devrait retrouver le dictionnaire de référence si les signaux ont été générés de façon très parcimonieuse. Et au contraire, le dictionnaire appris devrait être différent du dictionnaire appris, et plus efficace, si les signaux ont été générés de façon non parcimonieuse.

Nous voulons également comprendre quelle est l'influence du dimensionnement du dictionnaire sur la qualité de l'apprentissage.

9.4.3 Principe de l'expérience

Des signaux sont générés artificiellement à partir d'un dictionnaire de référence, pour des parcimonies différentes, alors que la taille du signal reste identique. Pour chacun des signaux générés, nous apprenons plusieurs dictionnaires, de taille plus ou moins grande par rapport au dictionnaire de référence.

Pour chaque dictionnaire appris, nous mesurons la ressemblance avec le dictionnaire de référence à l'aide du corrélogramme, ainsi que les courbes du RSB en fonction de la DR et de la DRP pour le dictionnaire d'apprentissage, que nous comparons à celles du dictionnaire original.

9.4.4 Protocole détaillé

Le protocole détaillé est décrit dans le tableau suivant

Dictionnaire des motifs de référence \mathcal{M}_{ref}	10 motifs de taille 16, générés avec une distribution gaussienne centrée, et normés.
Dictionnaire \mathcal{D}_{ref}	Atomes générés par translation et amplification des motifs de \mathcal{M}_{ref} : $A_c \circ T_t[\underline{m}^{ref}]$. Rappelons que tous les signaux sont générés avec le même dictionnaire \mathcal{D}_{ref} .
Taille des signaux d'entraînement et de test	10 000 échantillons
Parcimonie des signaux d'entraînement et de test	entre 30 et 6 000 occurrences des motifs de référence, soit une DR η comprise entre 0.003 (très parcimonieux) et 0.6 (peu parcimonieux).
Génération des signaux	100 signaux générés avec un nombre d'occurrences croissant entre 30 et 6 000 selon une échelle logarithmique. Les occurrences sont générées en utilisant les motifs successivement, et en tirant l'instant t selon une loi uniforme, et l'amplitude c selon une loi gaussienne.
Nombre de dictionnaires appris pour chaque signal	4.
Dictionnaire de motifs \mathcal{M}_{app}^1	10 motifs de taille 16.
Dictionnaire de motifs \mathcal{M}_{app}^2	10 motifs de taille 32.
Dictionnaire de motifs \mathcal{M}_{app}^3	20 motifs de taille 16.
Dictionnaire de motifs \mathcal{M}_{app}^4	20 motifs de taille 32.
Critère d'arrêt de MP	lorsque la DRP μ a atteint la DRP μ_{gen} des signaux et le nombre d'atomes a atteint le nombre d'atomes générateurs.
Paramètres de l'algorithme K-SVD	100 itérations, et pour chaque itération, 5 mises à jour de chaque motif, dans un ordre aléatoire de distribution uniforme.
Initialisation des motifs	tirage aléatoire gaussien, puis normalisation

9.4.5 Résultats attendus

Nous allons présenter deux figures, la première montrant à quel point les dictionnaires appris sont proches du dictionnaire de référence, et la seconde comparant la par-

cimonie de la décomposition du signal d'apprentissage sur les différents dictionnaires.

Sur la première figure sont affichées les nuages de points de type rappel et précision. Elle est divisée en huit sous-figures, correspondant aux quatre types de dictionnaires appris et aux deux mesures rappel et précision. Sur chaque sous-figure sont affichés les nuages de points correspondant aux dictionnaires appris sur les 100 signaux d'apprentissage, de parcimonie croissante.

Cette figure permettra d'analyser comment la taille du dictionnaire de motifs appris influe sur le fait de retrouver les motifs de référence, ceci en fonction de la parcimonie intrinsèque des signaux d'apprentissage.

La deuxième figure représente la qualité de la décomposition des signaux d'apprentissage sur les quatre dictionnaires appris et sur le dictionnaire de référence, pour trois signaux d'apprentissage (très parcimonieux sur le dictionnaire de référence, très peu parcimonieux, et entre les deux). La qualité de la décomposition est affichée grâce à la courbe du RSB en fonction de la DR, mais aussi en fonction de la DRP, ce qui permettra d'analyser la pertinence de sur-dimensionner le dictionnaire appris.

Les deux figures seront à analyser conjointement, car les interprétations seront liées. Nous nous attendons à ce que les corrélations de rappel et précision soient proches de 1 (ou que la distance soit proche de 0, sur la courbe) pour des signaux d'apprentissage très parcimonieux sur le dictionnaire de référence. En correspondance, si les dictionnaires sont proches, les courbes de RSB seront très proches pour le dictionnaire de référence et pour les dictionnaires appris.

En revanche, pour les signaux d'apprentissage très peu parcimonieux dans le dictionnaire de référence (η_{gen} élevée), l'algorithme d'apprentissage devrait calculer des dictionnaires plus performants en termes de parcimonie que le dictionnaire de référence. Cela devrait se traduire par une augmentation de la distance entre les motifs de référence et les motifs appris.

Nous analyserons également la transition entre ces deux phases, en termes de rapidité de passage entre les deux comportements, et en regardant si les dictionnaires appris sont toujours meilleurs en termes de parcimonie que le dictionnaire de référence.

9.4.6 Présentation des résultats

La figure 9.2 représente la distance entre les motifs de référence et les motifs appris, en fonction de la parcimonie intrinsèque des signaux d'apprentissage.

Le premier enseignement de ces courbes (nuages de points et médianes) vient de leur allure générale. Toutes les courbes sont à peu près constantes pour une DR η_{gen} des signaux d'apprentissage inférieure à 0.01, et elles montent, ce qui signifie que la distance augmente, pour des DR supérieures, c'est-à-dire pour des signaux générés de façon moins parcimonieuse avec le dictionnaire de référence.

Lorsque les signaux sont générés parcimonieusement, les dictionnaires appris sont proches du dictionnaire de référence, et ils s'en éloignent quand les signaux sont générés moins parcimonieusement. Cela vient du fait que les motifs appris, différents des motifs de référence, sont plus efficaces pour coder les signaux non parcimonieux, comme nous pouvons le voir sur la figure 9.3.

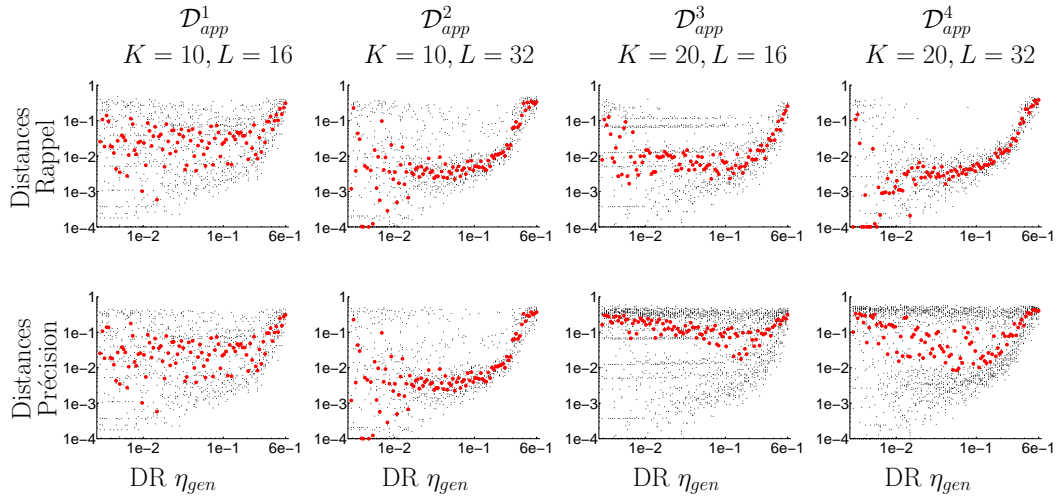


FIG. 9.2 – Distance entre les motifs appris et les motifs de référence (distance de la corrélation entre motifs à la corrélation maximale), pour quatre dictionnaires appris, en fonction de la parcimonie naturelle des signaux d'apprentissage.

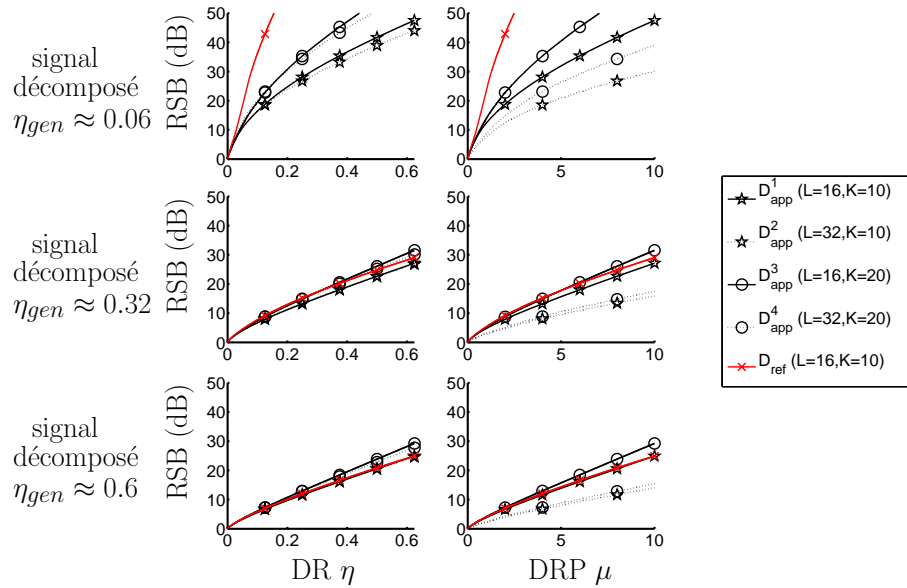


FIG. 9.3 – Qualité de la décomposition des signaux d'apprentissage (naturellement très (haut), moyennement (milieu) et peu (bas) parcimonieux) sur les quatre dictionnaires appris et sur le dictionnaire de référence, en fonction de la diversité relative (gauche) ou de la diversité relative pondérée (droite)

Cette figure représente la qualité de la décomposition en RSB en fonction de la densité relative (colonne de gauche), pour trois DR de génération. Pour le signal parcimonieux ($\eta_{gen} \approx 0.06$), le dictionnaire de référence décompose beaucoup mieux le signal

que les dictionnaires appris. En revanche, pour les signaux moyennement ($\eta_{gen} \approx 0.32$) et peu ($\eta_{gen} \approx 0.6$) parcimonieux, les dictionnaires appris décomposent les signaux aussi bien que le dictionnaire de référence.

Une première conclusion est donc que l'algorithme d'apprentissage ne permet pas de retrouver exactement le dictionnaire de référence, même pour des signaux très parcimonieux dans le dictionnaire de référence. Ces dictionnaires sont donc moins performants que le meilleur dictionnaire qui est le dictionnaire de référence. Dans le cas de signaux peu parcimonieux, les dictionnaires appris deviennent très éloignés du dictionnaire de référence, et doivent certainement coder des combinaisons d'atomes récurrentes non présentes dans le dictionnaire de référence, ce qui leur permet de décomposer le signal aussi bien que le dictionnaire de référence.

Intéressons-nous maintenant à la question de la taille du dictionnaire appris. Nous avons fait varier le nombre de motifs K et la taille des motifs L_m . Le dictionnaire \mathcal{D}_{app}^1 est de la même taille que le dictionnaire \mathcal{D}_{ref} . Les dictionnaires \mathcal{D}_{app}^2 et \mathcal{D}_{app}^3 doublent respectivement la taille des motifs et le nombre de motifs, et le dictionnaire \mathcal{D}_{app}^4 double la taille et le nombre de motifs.

Sur la figure 9.2, nous voyons que le doublement de la taille des motifs (\mathcal{D}_{app}^1 à \mathcal{D}_{app}^2) diminue la distance entre les motifs de référence et appris d'environ un ordre de grandeur (de 0.1 à 0.01, en gros) pour les signaux naturellement parcimonieux (gauche de l'abscisse). En revanche, pour les signaux naturellement peu parcimonieux, le comportement ne varie pas. Remarquons que les courbes de rappel et de précision sont presque identiques en raison du nombre égal de motifs dans le dictionnaire de référence et dans les dictionnaires appris.

Au niveau de la qualité de la décomposition (figure 9.3), nous voyons que le doublement de la taille des motifs augmente légèrement la taille du RSB pour le signal naturellement parcimonieux (haut, gauche) et n'a pas d'influence sur les signaux naturellement moins parcimonieux (milieu et bas, gauche). Sur la colonne de droite, nous voyons que la faible différence de qualité entre les deux dictionnaires devient presque une différence du simple au double au niveau du RSB, puisque les motifs longs sont deux fois plus pénalisés que les motifs courts. Cela signifie que les 16 paramètres libres supplémentaires ne sont pas aussi utiles que les 16 premiers paramètres pour l'apprentissage des motifs récurrents du signal. Rien d'anormal, puisque la taille des motifs de référence est de 16.

Le doublement du nombre de motifs conduit à deux remarques différentes. Tout d'abord, si l'amélioration de la distance de type "rappel" est à peu près la même que pour le doublement de la taille des motifs, il semble que pour la précision, environ la moitié des motifs appris s'approchent des motifs de référence, tandis que l'autre moitié est très différente, avec une distance à la corrélation maximale de l'ordre de 0.3 à 0.5. Nous pouvons supposer que ces motifs sont affectés à la description des formes complémentaires aux motifs de référence les plus représentatives (peut être des superpositions récurrentes de certains motifs).

La qualité de la décomposition est améliorée de manière plus nette que pour le doublement de la taille des motifs, très fortement pour le signal naturellement parcimonieux, et assez distinctement pour les signaux moins parcimonieux. L'utilisation de la DRP au

lieu de la DR en abscisse ne change rien, puisque les motifs sont de même taille que les motifs de référence. Cette mesure peut sembler discutable puisque le nombre de paramètres libres a aussi été multiplié par deux. Rappelons que la DRP ne s'intéresse pas au nombre de paramètres du dictionnaire mais uniquement au nombre d'échantillons des atomes utilisés par rapport au nombre d'échantillons du signal. Cela signifie qu'il faut moins d'échantillons d'atomes pour décrire le signal avec la même précision lorsque le nombre de motifs appris est doublé.

Nous pouvons déduire de ces constatations que les paramètres supplémentaires ont été utilisés de meilleure façon qu'en doublant la taille des motifs, ce qui paraît logique puisqu'un motif de taille 32 ne pourra pas représenter à la fois le motif de référence et une superposition particulière de motifs, tandis que c'est possible en utilisant deux motifs de taille 16.

Enfin, le doublement de la taille des motifs entre le dictionnaire \mathcal{D}_{app}^3 et le dictionnaire \mathcal{D}_{app}^3 apporte à peu près les mêmes améliorations qu'entre le dictionnaire \mathcal{D}_{app}^1 et le dictionnaire \mathcal{D}_{app}^2 . Les motifs appris sur les signaux naturellement parcimonieux sont plus proches des motifs de référence (mesure de rappel), la qualité de décomposition (RSB) augmente de quelques dB, et l'utilisation de la DRP en abscisse divise presque par deux les performances, ce qui confirme que l'apport du doublement de la taille des motifs n'est pas très grand.

9.4.7 Conclusion

Cette expérience met en lumière la bonne capacité de l'algorithme K-SVD à retrouver les motifs générateurs d'un signal naturellement parcimonieux, et à promouvoir la parcimonie. Le sur-dimensionnement du dictionnaire semble plus efficace, à nombre de paramètres libres égal, en ajoutant des motifs qu'en augmentant leur taille, car cela permet de coder des enchaînements récurrents d'atomes.

9.5 Evaluation de la qualité de l'apprentissage selon les déformations utilisées

Nous avons exposé dans la section précédente à quel point l'algorithme K-SVD favorise la parcimonie de la décomposition de son signal d'apprentissage. Nous nous intéressons dans cette section à savoir comment l'utilisation des fonctions de translation et de déformation de phase agissent sur la qualité de l'apprentissage. Nous comparerons la décomposition des signaux d'apprentissage sur des dictionnaires invariants par translation et sur des dictionnaires "partiellement" invariants par translation, revenant à découper les signaux en trames.

Nous comparerons les décompositions à l'aide de la mesure déjà définie de RSB en fonction de la densité relative.

9.5.1 Principe de l'expérience

Deux dictionnaires à la structure différente vont être étudiés (analogues aux dictionnaires \mathcal{D}^2 et \mathcal{D}^3 de la section 8.7), et nous les apprendrons à partir d'un même signal audio synthétique (MIDI). Nous utiliserons un signal synthétique, et non plus un signal artificiel généré à partir d'un dictionnaire de référence. En effet, les dictionnaires appris n'ayant pas le même modèle de structure, le choix du modèle du dictionnaire de référence favoriserait l'un ou l'autre dictionnaire appris.

Décrivons rapidement ce que sont les signaux MIDI. Le format MIDI est un format informatique pour décrire la musique qui ressemble à une partition. Pour générer un son à partir d'un fichier au format MIDI, il faut utiliser un logiciel de synthèse (synthétiseur), qui, grossièrement, attribue à chaque note de la partition le son correspondant, en utilisant le timbre de l'instrument indiqué sur la partition. Chaque synthétiseur utilise ses propres sons, et un même fichier MIDI n'aura donc pas forcément le même rendu sonore sur deux ordinateurs différents.

Nous utilisons des signaux de type MIDI plutôt que des signaux musicaux enregistrés parce que le modèle parcimonieux semble plus approprié pour ce type de signaux que pour des signaux réels où la variabilité est très grande. En effet, la même note jouée deux fois produira à peu près le même son avec un fichier MIDI, tandis que pour un enregistrement réel, le son peut être très différent, et même s'il semble identique à l'oreille, la suite d'échantillons du signal peut être très différente.

Afin d'éviter que le déroulement de la phase soit toujours le même à chaque occurrence de chaque note dans le fichier MIDI, nous générons le fichier MIDI avec une fréquence d'échantillonnage élevée, puis nous sous-échantillons le signal, d'un facteur entier d'un part, et d'un facteur non-entier d'autre part. Nous analyserons alors pour chacune des deux fréquences d'échantillonnage l'apport de la déformation de phase sur la qualité de l'apprentissage. Logiquement, il devrait être plus grand pour la fréquence d'échantillonnage obtenue par une division non-entière de la fréquence d'échantillonnage originale, car les relations de phase devraient être plus chamboulées.

Le premier dictionnaire, \mathcal{D}_1 , est "partiellement" invariant par translation, car les atomes qu'il contient correspondent à des occurrences des motifs décalées d'un grand nombre d'échantillons (analogue au découpage en trames pour le modèle de dictionnaire non-structuré). Le deuxième dictionnaire, \mathcal{D}_2 , est invariant par translation, car le décalage entre occurrences de motifs est de 1 échantillon. Les deux dictionnaires reposent sur un dictionnaire de motifs de même taille, afin de pouvoir comparer en toute objectivité.

Précisons que les paramètres de l'algorithme d'apprentissage donnés dans le tableau suivant ont été réglés pour permettre un apprentissage à la fois rapide et performant. Pour ne pas alourdir le document, nous ne présentons pas les expériences qui nous mènent à choisir ces paramètres. Le point le plus important pour la qualité d'apprentissage est d'arrêter l'algorithme MP quand un certain niveau de RSB est atteint, et non pas quand un certain nombre d'atomes est utilisé. Il est préférable de ne pas mettre à jour plusieurs fois les motifs après chaque étape de MP, car le gain de performance n'est pas très grand par rapport à l'augmentation du temps de calcul.

9.5.2 Protocole détaillé

Le protocole détaillé est décrit dans le tableau suivant

Signal original \underline{g}	Signal audio MIDI - Concertino op. 26 pour clarinette et orchestre (réduction pour clarinette et piano), Weber : uniquement la première piste de clarinette. Échantillonné à 16 000 Hz et de durée 1 m 33 s.
Signaux d'apprentissage $\underline{g}_{app}^{8\ 000}$ et $\underline{g}_{app}^{11\ 025}$	sous-échantillonnage de \underline{g} pour donner respectivement 743 039 et 1 024 000 échantillons.
Taille des dictionnaires des motifs	$K = 40$ motifs centrés de taille 46.4 ms, soit $L_m = 512$ à 11 025 Hz et $L_m = 372$ à 8 000 Hz. Nombre de paramètres : 20 480 et 14 480.
Dictionnaires appris	8 dictionnaires $\mathcal{D}_{fs,P}^i$ correspondant aux combinaisons des trois paramètres : – \mathcal{D}_{\dots}^1 : dictionnaires “partiellement” invariants par translation/ \mathcal{D}_{\dots}^2 : dictionnaires invariants par translation. – $\mathcal{D}_{8\ 000}^i$ appris sur le signal à 8 000 Hz, et $\mathcal{D}_{11\ 025}^i$ appris sur le signal à 11 025 Hz – \mathcal{D}_{fs}^i sans déformation de phase et $\mathcal{D}_{fs,P}^i$ avec déformation de phase
Décalage entre occurrences de motifs pour \mathcal{D}_{\dots}^1	respectivement $L_m/2 = 256$ et $L_m/2 = 186$
Critère d'arrêt de MP, dans l'algorithme K-SVD	lorsque le RSB atteint 20 dB.
Paramètres de l'algorithme K-SVD	30 itérations, et pour chaque itération, une seule mise à jour de chaque motif, sans mise à jour des coefficients ni de l'erreur.
Initialisation des motifs	tirage aléatoire gaussien, puis normalisation

9.5.3 Présentation des résultats

Avant de présenter les résultats de l'apprentissage de dictionnaire, présentons le signal sur lequel sont appris les motifs.

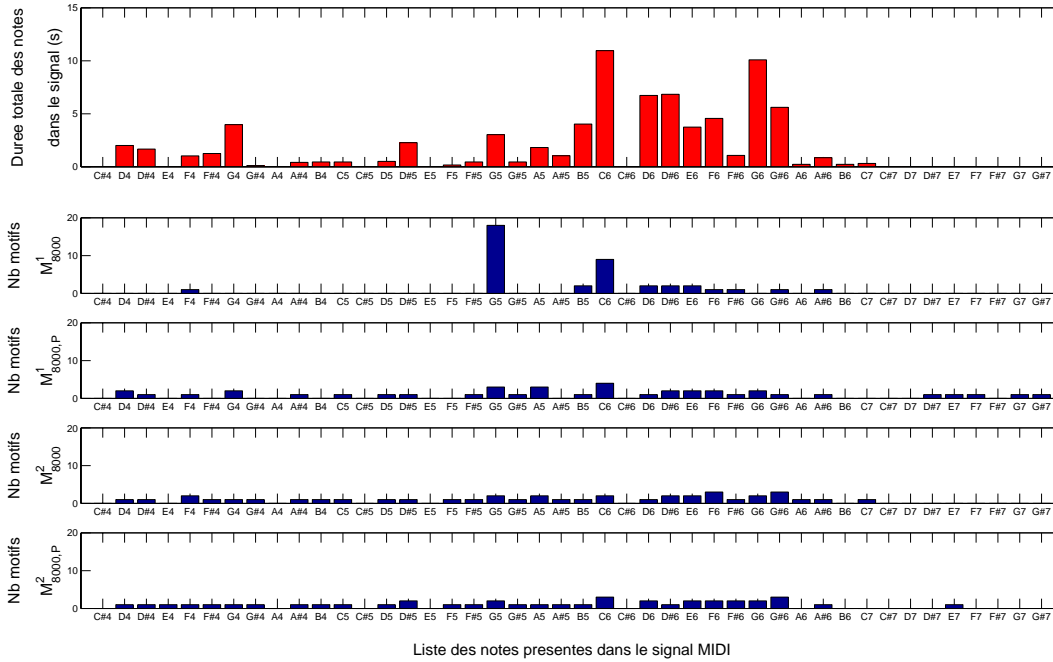


FIG. 9.4 – Histogramme des notes présentes dans le signal $\underline{s}_{app}^{8\,000}$ (haut), l’ordonnée correspond à la durée totale des occurrences de chaque note. Histogramme de la répartition des motifs des dictionnaires $\mathcal{M}_{8\,000,\dots}^i$ appris sur $\underline{s}_{app}^{8\,000}$ (bas), l’ordonnée correspondant au nombre de motifs associés à chaque note

9.5.3.1 Analyse du signal

La figure 9.4 contient cinq histogrammes, mais nous ne regardons que celui du haut pour l’instant. Cet histogramme présente la répartition des notes dans le signal d’apprentissage : à chaque note (par exemple, le LA 440 Hz, ou **A#4**, en notation anglaise) est associée la durée totale pendant laquelle la note apparaît dans le morceau. Cette donnée est facilement disponible puisque le fichier MIDI, que nous pouvons comparer à une partition, fournit toutes les informations nécessaires sous forme symbolique.

Sur cet histogramme, 30 notes sont présentes, avec des durées cumulées allant de 110 ms à presque 11 s. La répartition est inégale, et en comparant avec la durée des motifs appris (512 échantillons, soit 46.4 ms), 19 notes ont une durée cumulée totale plus longue que 10 fois la longueur des motifs, et parmi elles, 10 notes ont une durée cumulée plus longue que 50 fois la longueur des motifs. Cette mise en relation permet de supposer que les notes les plus fréquemment utilisées présentent une redondance suffisante par rapport à la taille des motifs pour être bien apprises tandis que les autres notes seront certainement moins bien retrouvées. Précisons que parmi toutes les notes jouées dans ce signal, l’occurrence la plus courte dure 100 ms. La durée des motifs (46.4 ms) se justifie

donc, en supposant que les notes sont à peu près stationnaires, ce qui permet de les coder en utilisant plusieurs fois le même motif.

Nous pouvons maintenant analyser les autres histogrammes, correspondant aux motifs appris sur le signal.

9.5.3.2 L'invariance par translation est primordiale pour retrouver les notes

Les quatre histogrammes situés en dessous de l'histogramme des notes du signal correspondent aux quatre dictionnaires appris (décalage de $L_m/2$ ou de 1 entre atomes, et utilisation ou non de la déformation de phase). Chaque motif appris est associé à une note (grâce à la fréquence dominante du spectre du motif), et les histogrammes sont construits en représentant pour chaque note le nombre de motifs appris qui lui sont associés.

Cette présentation permet de voir la diversité des motifs appris et de corrélérer l'apprentissage avec les notes présentes dans le morceau MIDI. Ainsi, nous nous attendons à ce que les notes les plus jouées dans le morceau soient retrouvées par les dictionnaires et que les notes peu jouées ou absentes du signal n'aient pas de motifs associés.

Pour comparer les dictionnaires, nous afficherons également la courbe du RSB de la décomposition en fonction de la densité relative η . Elle permettra de comparer les performances des différents dictionnaires du point de vue de la parcimonie de la décomposition.

Nous afficherons enfin la forme d'onde temporelle et le spectre de plusieurs motifs, afin de comprendre ce qu'ils contiennent et donc ce que l'algorithme a appris.

Les histogrammes de la figure 9.4 indiquent que les motifs appris selon le modèle invariant par translation (dictionnaire $\mathcal{M}_{8\ 000}^2$, quatrième ligne) retrouvent mieux les notes présentes dans le signal que les motifs appris selon le modèle "partiellement invariant par translation" (dictionnaire $\mathcal{M}_{8\ 000}^1$, deuxième ligne). En effet, les 40 motifs du dictionnaire $\mathcal{M}_{8\ 000}^2$ ont été affectés à 29 notes parmi les 30 notes jouées (seule la note **B6** n'a pas été retrouvée). Dans le même temps, les 40 motifs du dictionnaire $\mathcal{M}_{8\ 000}^1$ ont été affectés à seulement 11 notes.

Parmi ces 11 notes, 9 motifs ont été affectés à la note la plus jouée (**C6**) et 18 motifs à la note **G5**. Cette note n'est que la dixième note la plus jouée dans le signal, mais elle est surtout la note située un octave en dessous de la note **G6**, qui est la deuxième note la plus jouée. Cela peut s'expliquer par le fait qu'au niveau fréquentiel, la note **G6** correspond à la moitié des harmoniques de la note **G5** et peut donc être codée, moyennant la présence de certaines harmoniques (ainsi que la fréquence fondamentale) indésirables, par la note **G5**.

La figure 9.5 représente trois motifs du dictionnaire appris $\mathcal{M}_{8\ 000}^1$. Nous pouvons voir que l'harmonique **G6** est importante dans le motif associé à la note **G5**. En plus de la présence de ces deux notes, nous voyons que le motif contient également les fréquences correspondant aux notes **C6**, **F6**, **F#6** et **C7**. La même diversité se retrouve pour les deux autres motifs. Cela signifie que ces motifs codent en majorité les notes auxquelles ils sont affectés dans l'histogramme, mais qu'ils codent également en partie d'autres notes.

La figure 9.6 montre, quant à elle, les trois mêmes motifs appris au sein du diction-

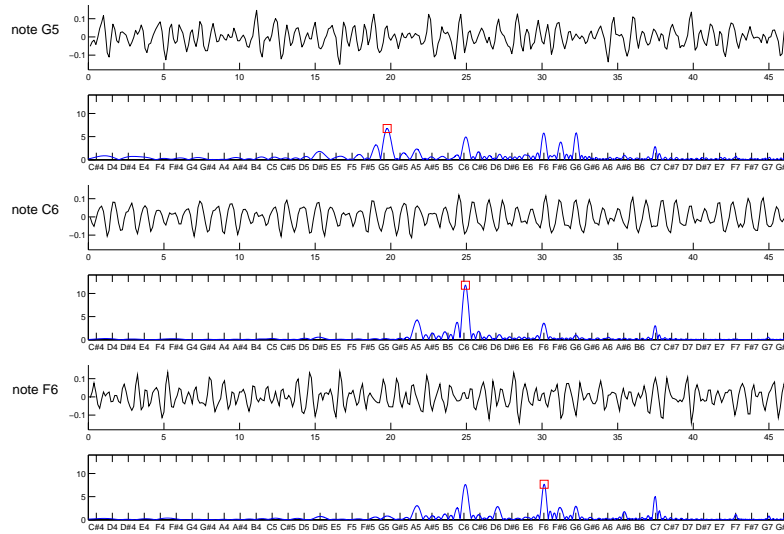


FIG. 9.5 – Trois motifs du dictionnaire appris $\mathcal{M}_{8\ 000}^1$, correspondant aux notes C6, F6 et G6. Affichage de la forme d’onde (haut) et du spectre (bas). Les fréquences du spectre sont en échelle logarithmique correspondant aux notes de la gamme.

naire $\mathcal{M}_{8\ 000}^2$, c’est-à-dire avec le modèle d’invariance par translation. Nous pouvons alors nous rendre compte que les motifs appris sont presque des sinusoïdes pures, où les seules fréquences présentes sont la fondamentale de la note associée et les harmoniques. Notez que les lobes secondaires sont dus à une étape de “zero-padding” permettant une plus grande précision du spectre.

La différence entre les motifs de $\mathcal{M}_{8\ 000}^1$ et de $\mathcal{M}_{8\ 000}^2$ est frappante. Nous pouvons l’interpréter comme le fait que les trames d’apprentissage du dictionnaire $\mathcal{M}_{8\ 000}^1$ sont découpées de façon arbitraire, et peuvent se trouver à cheval sur deux notes différentes, tandis que pour le dictionnaire $\mathcal{M}_{8\ 000}^2$, les trames sont découpées de façon précise à l’endroit où les notes sont jouées, menant à une meilleure focalisation sur une seule note.

Du point de vue de la qualité de la décomposition, la figure 9.7 nous montre que le modèle invariant par translation est plus efficace que le modèle “partiellement” invariant par translation.

Les lignes en pointillés représentent la performance du dictionnaire de motifs $\mathcal{M}_{8\ 000}^1$ appris suivant le modèle de tramage arbitraire. Les lignes continues correspondent au dictionnaire $\mathcal{M}_{8\ 000}^2$ appris avec l’invariance par translation. La figure de gauche correspond à la décomposition selon le modèle de tramage arbitraire, tandis que la figure de droite correspond à la décomposition sur un dictionnaire invariant par translation. A droite comme à gauche, le même modèle est appliqué sur tous les dictionnaires de motifs appris, indépendamment du modèle utilisé pour l’apprentissage.

De façon cohérente avec l’expérience présentée au chapitre 8, le modèle invariant

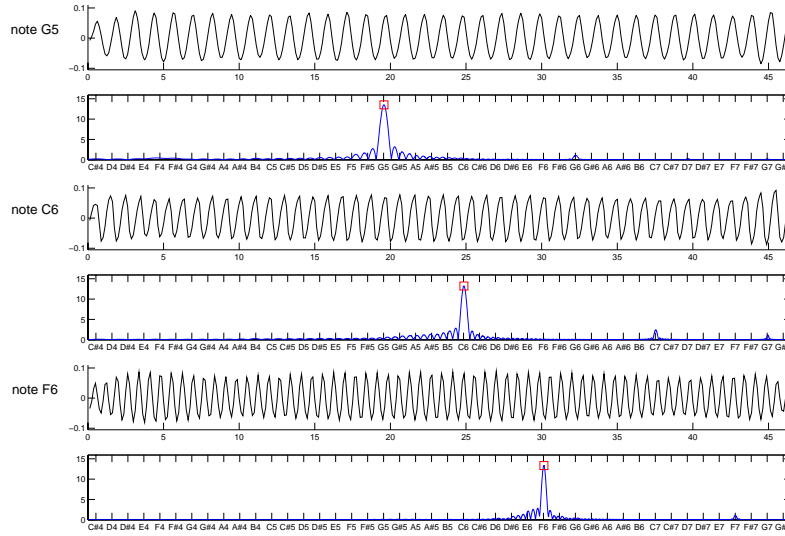


FIG. 9.6 – Trois motifs du dictionnaire appris $\mathcal{M}_{8\ 000}^2$, correspondant aux notes C6, F6 et G6. Affichage de la forme d'onde (haut) et du spectre (bas). Les fréquences du spectre sont en échelle logarithmique correspondant aux notes de la gamme.

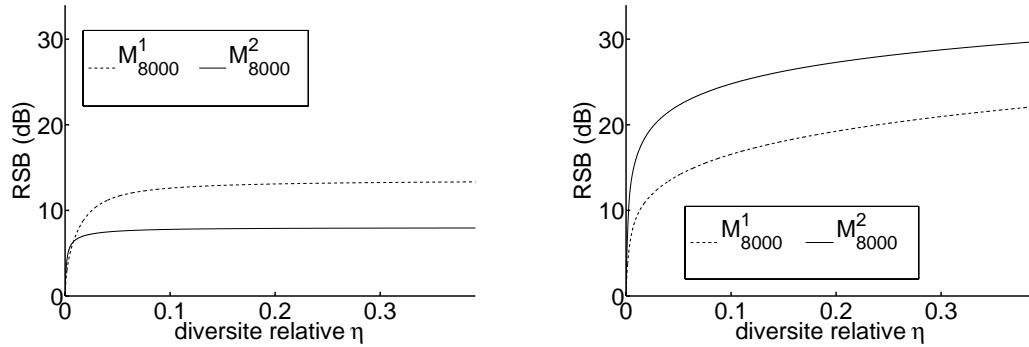


FIG. 9.7 – Mesure de la qualité de décomposition du signal $\underline{s}_{app}^{8\ 000}$ sur les quatre dictionnaires de motifs appris, utilisés dans le modèle avec décalage de $L_m/2$ entre atomes (gauche), et dans le modèle avec décalage de 1 entre les atomes (droite).

est le plus efficace pour la décomposition, quel que soit le dictionnaire de motifs utilisé. Ainsi, pour une densité relative d'environ 0.4 atome par échantillon de signal, utiliser le dictionnaire de motifs $\mathcal{M}_{8\ 000}^1$ dans le modèle avec lequel il a été appris (gauche), donne une décomposition moins performante d'environ 8 dB en termes de RSB qu'en utilisant ce même dictionnaire de motifs dans un modèle invariant par translation, qui n'est pourtant pas le modèle dans lequel les motifs ont été appris.

Pour faire abstraction de cette différence, nous comparons les dictionnaires de motifs appris sur le même modèle de décomposition. Ainsi, pour le modèle “partiellement” invariant par translation, le dictionnaire appris suivant le même modèle permet une meilleure décomposition (5 dB) que le dictionnaire appris avec invariance par translation, ce qui est une vérification rassurante sur le fait que l’algorithme apprend un bon dictionnaire pour le modèle utilisé.

Sur la figure de droite, nous voyons que pour le modèle de décomposition invariant par translation, le dictionnaire $\mathcal{M}_{8\ 000}^2$, appris sous le même modèle, donne une décomposition meilleure de 7 dB au dictionnaire $\mathcal{M}_{8\ 000}^1$, appris suivant l’autre modèle. Ce résultat indique qu’une stratégie consistant à apprendre le dictionnaire selon un modèle “partiellement” invariant pour réduire le temps de calcul, puis à utiliser le modèle invariant par translation pour la décomposition, donnerait des résultats bien inférieurs en termes de performance de décomposition. Cela n’empêche pas de réfléchir à des stratégies intermédiaires où l’écart entre les trames, sans être aussi grand que $L_m/2$, serait supérieur à 1 échantillon.

9.5.3.3 L’invariance par translation permet d’apprendre des motifs localisés en temps

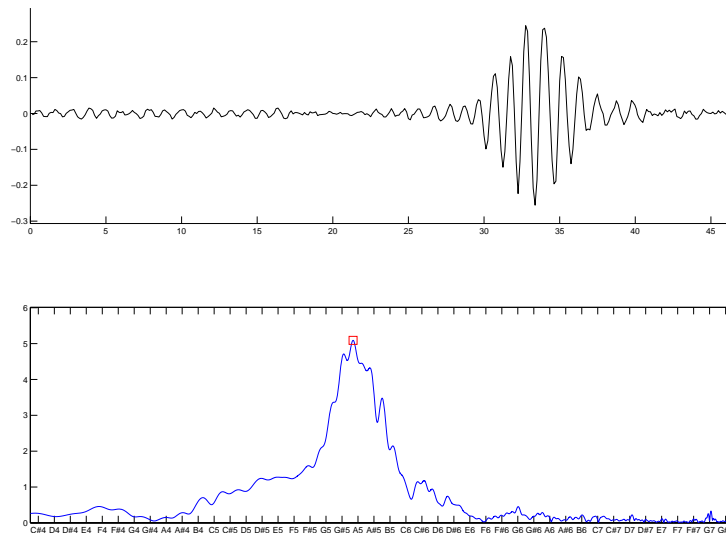


FIG. 9.8 – Un motif, issu du dictionnaire appris $\mathcal{M}_{8\ 000}^2$, localisé en temps et ressemblant à un motif de Gabor.

Nous montrons sur la figure 9.8 un exemple particulier de motif appris avec le modèle invariant par translation. Ce motif ressemble beaucoup à un motif de Gabor (voir la figure 1.3, par exemple) puisqu’il est localisé à un instant donné du motif, le reste du

motif étant presque nul, il n'est composé que d'une fréquence pure (environ 880 Hz, la fréquence fondamentale de la note **A5**) et son amplitude suit une fenêtre de type Hamming. Parmi le dictionnaire $\mathcal{M}_{8\ 000}^2$, seuls quelques autres motifs sont localisés, tandis que la plupart des autres motifs ressemblent aux quasi-sinusoides présentées sur la figure 9.6.

Le mini-motif de Gabor présenté sur la figure 9.8 est de taille 8 ms, bien inférieure à la taille du motif (46.4 ms), et à la taille de la seule note **A5** présente dans le morceau (1,8 s). Il faudrait analyser le résultat de la décomposition pour voir où se situent les atomes utilisant ce motif, et en déduire ce qui a pu engendrer un motif de Gabor. Pour cela, une solution serait d'utiliser un "atomogramme" qui, à l'instar d'un "spectrogramme", représenterait le signal en échelle temps-motifs, et permettrait de suivre les zones d'activité des différents motifs. Pour des raisons de temps, nous n'avons pas développé un tel outil, mais nous pensons qu'il serait d'une aide précieuse pour analyser les décompositions parcimonieuses.

Nous pouvons supposer qu'en apprenant des motifs plus longs, de taille supérieure aux différentes occurrences des notes dans le morceau, les motifs appris seraient plus nombreux à être localisés et que la taille de leur support serait calquée sur la taille des notes communément jouées.

9.5.3.4 Signaux MIDI : invariance par modulation ?

Les spectres typiques des motifs appris avec invariance par translation, représentés sur la figure 9.6, fait émerger l'idée d'une nouvelle déformation de motifs : l'invariance par modulation de fréquence. En effet, les trois spectres représentés semblent correspondre à un même motif fréquentiel, placé à trois fréquences différentes. Cette impression est due au fait que les motifs appris sur le signal MIDI sont très proches de sinusoides, lesquelles sont toutes, de façon théorique, des translations fréquentielles d'une seule raie. Rappelons également que l'échelle des fréquences est logarithmique, ce qui explique que nous voyions les motifs comme des simples translations, ce qui ne serait pas le cas en échelle linéaire.

Il faut remarquer qu'une déformation par modulation serait assez proche du principe de génération des signaux MIDI, puisque ces signaux sont créés à partir de quelques motifs (ici 5) qui subissent une modification de "pitch" pour atteindre la note demandée. Nous ne connaissons pas le détail de l'algorithme présent dans le synthétiseur que nous avons utilisé, mais connaissant le principe du modèle de génération, il semble logique d'apprendre des motifs à l'allure spectrale proche, placés à des fréquences différentes.

9.5.3.5 La déformation de phase est surtout utile pour de grands écarts entre atomes

Analysons maintenant les résultats de l'apprentissage utilisant la déformation de phase. La figure 9.9 représente, pour le modèle "partiellement" invariant par translation (gauche) et pour le modèle invariant par translation (droite), les courbes du RSB en fonction de la densité relative pour deux dictionnaires. Les courbes sans marqueurs

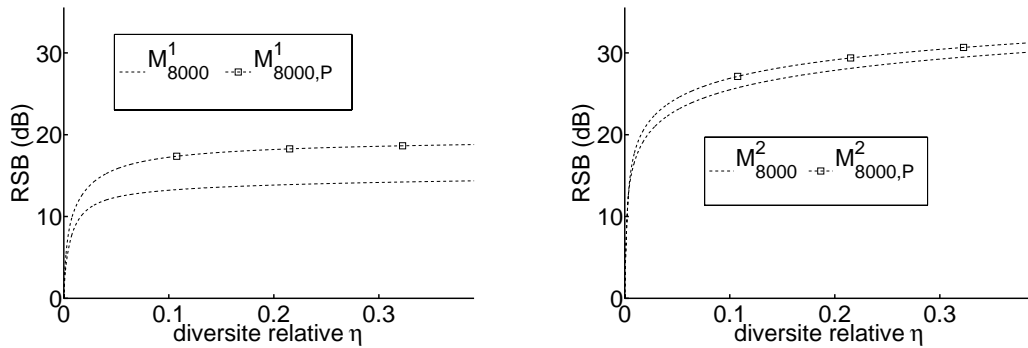


FIG. 9.9 – Mesure de la qualité de décomposition du signal $\underline{s}_{app}^{8000}$ sur les deux dictionnaires de motifs appris avec ou sans déformation de phase, utilisés dans le modèle avec décalage de $L_m/2$ entre atomes, avec déformation de phase (gauche), et dans le modèle avec décalage de 1 entre les atomes, avec déformation de phase (droite).

correspondent aux dictionnaires appris sans déformation de phase, et les courbes avec marqueurs carrés aux dictionnaires appris avec déformation de phase. La décomposition est faite, dans tous les cas, en utilisant le modèle avec déformation de phase, ce qui est naturel pour les courbes avec marqueurs, et qui correspond à un changement de modèle pour les courbes sans marqueurs.

Nous pouvons voir en premier lieu, par rapport aux valeurs lues sur la figure 9.7, que l’ajout de la déformation de phase apporte un gain de 1.2 dB en utilisant le même dictionnaire \mathcal{M}_{8000}^1 , avec invariance par translation “partielle” (gauche), et un gain de 0.35 dB en utilisant le même dictionnaire \mathcal{M}_{8000}^2 , avec invariance par translation (droite).

Au delà de ce gain attendu en raison des paramètres libres supplémentaires, regardons le gain en performances que permet l’utilisation de la déformation de phase lors de l’apprentissage. Pour le modèle avec invariance par translation (droite), l’apport est d’environ 1 dB, tandis qu’il monte à 4.4 dB pour le modèle avec invariance par translation “partielle” (gauche).

Ce résultat n’a rien d’étonnant puisque la déformation de phase compense en partie le manque de précision temporelle dû au grand décalage entre les atomes.

La déformation de phase avait été établie par analogie avec le déphasage (permutation circulaire) d’une sinusoïde. Dans le cas général, une permutation circulaire requiert $L_m/2$ paramètres (le nombre de coefficients de Fourier non-nuls dans le spectre), et pour limiter le nombre de paramètres, nous n’en avons gardé qu’un.

Toutefois, nous pouvons voir sur les figures 9.5 et 9.6 que les motifs appris ne comportent que peu de fréquences significatives. Grâce à cette “parcimonie” du spectre, la déformation de phase utilisant un seul paramètre permet donc de se rapprocher du déphasage, ce qui explique en partie le gain en performances pour la décomposition.

Le spectre des motifs du dictionnaire appris avec invariance par translation est le plus “parcimonieux”, mais cela ne mène pas à une plus grande amélioration lors de l’utilisation

de la déformation de phase, puisqu'une grande partie de l'effet du déphasage a été réglé par la translation. Grossièrement, il ne reste plus, pour la déformation de phase, qu'à régler les décalages inférieurs à un échantillon. Évidemment, pour une invariance par translation "partielle", la marge de manoeuvre de la déformation de phase est plus grande, ce qui explique le gain supplémentaire.

Cette interprétation trouve un appui supplémentaire à la lecture de la figure 9.4. En regardant les histogrammes deux à deux, nous pouvons voir que l'utilisation de la déformation de phase permet, dans le cas "partiellement" invariant par translation, de passer de 11 notes retrouvées ($\mathcal{M}_8^1{}_{000}$, deuxième ligne) à 22 notes retrouvées ($\mathcal{M}_8^1{}_{000,P}$, troisième ligne). Il faut noter que pour le dictionnaire $\mathcal{M}_8^1{}_{000,P}$, 5 motifs sont associés à des notes qui ne sont pas directement présentes dans le signal original (D#7, E7, F7, G7, G#7), mais correspondent à la première harmonique de notes parmi les plus présentes (D#6, E6, F6, G6, G#6).

Dans le cas de l'invariance par translation, l'augmentation du nombre de motifs retrouvés se transforme en une légère diminution, avec 27 notes retrouvées au lieu de 29, et 2 notes absentes du signal d'apprentissage. L'apport de la déformation de phase est donc là aussi plus important pour un dictionnaire appris seulement "partiellement" invariant par translation.

9.5.3.6 L'apprentissage est invariant à la fréquence d'échantillonnage

Pour terminer cette série d'expériences, nous nous intéressons à l'influence de la fréquence d'échantillonnage du signal. Nous rappelons que le signal a été généré à une fréquence de 16 000 Hz par le synthétiseur, et que nous avons sous-échantillonné ce signal à 8 000 Hz et à 11 025 Hz.

Nous avons supposé que l'amélioration due à l'ajout de la déformation de phase serait plus grande pour un sous-échantillonnage non-entier (11 025 Hz) que pour un sous-échantillonnage entier (8 000 Hz) parce que les relations de phase peuvent être davantage chamboulées.

Notez qu'il n'est pas possible de faire une comparaison en modifiant uniquement la fréquence, tout étant égal par ailleurs.

En effet, il est impossible de maintenir la même durée de signal, le même nombre de paramètres dans la motifs (nombre et longueur), et le même rapport entre la taille du signal et la taille des motifs. Nous avons choisi de conserver la même durée pour le signal d'apprentissage et les motifs, ce qui conduit à diminuer le nombre d'échantillons dans les motifs et dans le signal. Cela signifie une augmentation d'environ 38% du nombre d'échantillons pour décrire la même durée. Nous pouvons imaginer que la variabilité est plus grande lorsque le nombre d'échantillons par seconde augmente, et qu'un même nombre de motifs appris aura donc un peu plus de difficulté à représenter le signal.

Sur la figure 9.10, nous représentons les histogrammes des dictionnaires de motifs appris, avec ou sans déformation de phase, et avec invariance par translation partielle ou totale, pour le signal sous-échantillonné à 11 025 Hz. Cette figure est à comparer à la figure 9.4, qui concerne le signal sous-échantillonné à 8 000 Hz.

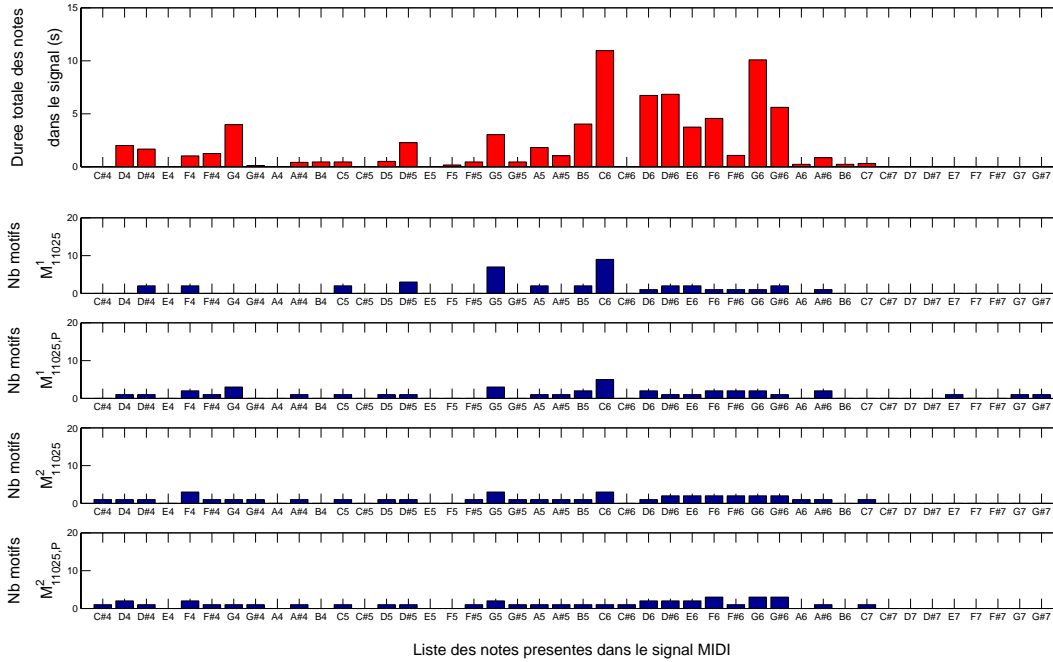


FIG. 9.10 – Histogramme des notes présentes dans le signal $\underline{s}_{app}^{11\ 025}$ (haut), l’ordonnée correspond à la durée totale des occurrences de chaque note. Histogramme de la répartition des motifs des dictionnaires $\mathcal{M}_{11\ 025,\dots}^i$ appris sur $\underline{s}_{app}^{11\ 025}$ (bas), l’ordonnée correspondant au nombre de motifs associés à chaque note

Nous pouvons remarquer que le nombre de notes retrouvées est à peu près du même ordre de grandeur :

- 16 notes retrouvées contre 11, pour \mathcal{M}_{fs}^1 ,
- 22 notes retrouvées pour les deux fréquences, et 3 absentes du signal original, contre 5, pour $\mathcal{M}_{fs,P}^1$
- 27 notes retrouvées et 1 absente, contre 29 notes retrouvées, pour \mathcal{M}_{fs}^2 , enfin
- 26 notes retrouvées et 2 absentes, contre 27 notes retrouvées et 2 absentes, pour $\mathcal{M}_{fs,P}^2$.

Regardons maintenant les performances de décomposition. La figure 9.11 représente les mêmes grandeurs que la figure 9.9, c’est-à-dire, pour les deux modèles d’invariance partielle ou totale par translation, la décomposition du signal sur un dictionnaire avec déformation de phase, utilisant le dictionnaire de motifs appris soit avec déformation de phase, soit sans déformation de phase. La différence avec la figure précédente est que nous représentons les courbes pour les deux fréquences 8 000 Hz et 11 025 Hz, et non plus seulement 8 000 Hz, et que l’abscisse n’est plus la densité relative (nombre d’atomes par échantillon de signal), mais le nombre d’atomes par seconde de signal.

Dans le cas “partiellement” invariant par translation (gauche), les courbes corres-

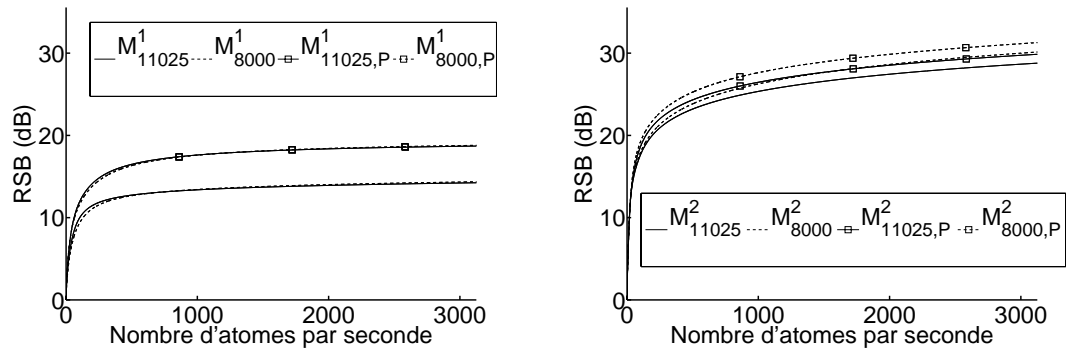


FIG. 9.11 – Mesure de la qualité de décomposition du signal $\underline{g}_{app}^{11\ 025}$ sur les deux dictionnaires de motifs appris avec ou sans déformation de phase, utilisés dans le modèle avec décalage de $L_m/2$ entre atomes, avec déformation de phase (gauche), et dans le modèle avec décalage de 1 entre les atomes, avec déformation de phase (droite).

pendant aux deux fréquences sont presque exactement superposées, ce qui signifie que la différence d'échantillonnage ne modifie pas réellement les performances d'apprentissage et de décomposition. Dans le cas invariant par translation (droite), la différence est de 1.5 dB en faveur de la fréquence 8 000 Hz, pour les deux courbes. En revanche, l'amélioration apportée par l'utilisation de la déformation de phase lors de l'apprentissage est presque la même pour les deux fréquences (1.1 dB). Ce constat, le même que pour le cas "partiellement" invariant par translation, met à mal l'hypothèse selon laquelle le sous-échantillonnage d'un facteur non-entier augmenterait le besoin d'un paramètre de phase pour compenser un bouleversement des relations de phase.

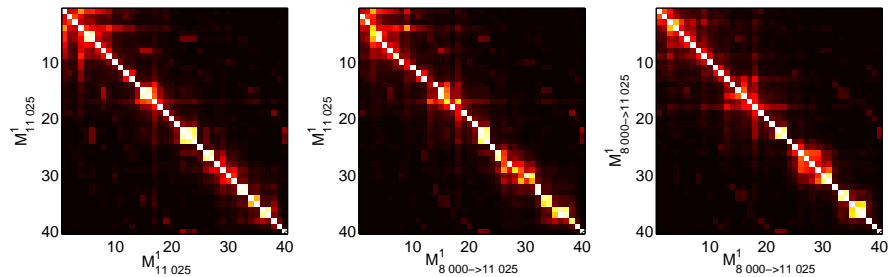


FIG. 9.12 – Les deux autocorrélogrammes et l'intercorrélogramme générés par les deux dictionnaires de motifs $\mathcal{M}_{11\ 025}^2$ et $\mathcal{M}_{8\ 000 \rightarrow 11\ 025}^2$. Ce dernier est le rééchantillonnage à 11 025 Hz de $\mathcal{M}_{8\ 000}^2$

La figure 9.12 présente les autocorrélogrammes des dictionnaires de motifs $\mathcal{M}_{11\ 025}^2$ et $\mathcal{M}_{8\ 000 \rightarrow 11\ 025}^2$, ce dernier étant le rééchantillonnage à 11 025 Hz du dictionnaire de motifs $\mathcal{M}_{8\ 000}^2$. Sur la figure est également représenté, au milieu, l'intercorrélogramme entre les deux dictionnaires de motifs. Les corrélogrammes ont été générés en classant

les motifs par fréquence fondamentale croissante.

Nous pouvons voir que les trois corrélogrammes sont très proches visuellement les uns des autres, ce qui signifie que les corrélations entre les motifs d'un même dictionnaire, ou les corrélations entre les motifs des deux dictionnaires sont assez proches. Nous pouvons remarquer en particulier que les corrélogrammes sont proches d'une matrice diagonale, ou bloc-diagonale, avec des blocs de un à trois motifs représentant la même note.

La principale différence entre les corrélogrammes vient du nombre de motifs dans ces blocs, ce qui donne des blocs rectangulaires, et non plus carrés, dans l'intercorrélogramme. En dehors de cela, les corrélogrammes sont très proches.

Pour conclure, à part la différence de qualité de la décomposition dans le cas invariant par translation, toutes les expériences plaident pour dire que les motifs appris sur le signal MIDI sont presque les mêmes, quelle que soit la fréquence d'échantillonnage, et donc que la différence d'échantillonnage n'a que peu d'influence sur l'apprentissage et la décomposition.

9.6 Influence de la taille de l'ensemble d'apprentissage sur la qualité de l'apprentissage

Nous venons d'exposer l'intérêt de l'invariance par translation sur des signaux réels, au niveau de la décomposition et au niveau de l'apprentissage. Nous nous intéressons maintenant au problème de la généralisation des dictionnaires appris sur un signal d'apprentissage à d'autres signaux de même nature (signaux de test).

Intuitivement, plus le signal d'apprentissage est long, plus le dictionnaire peut intégrer la diversité de ce signal, et donc décomposer efficacement d'autres signaux de même nature. Cette généralisation doit se faire au détriment de l'efficacité de la décomposition du signal d'apprentissage. Il faudra déterminer quelle redondance de données d'apprentissage est nécessaire pour atteindre une généralisation suffisante.

9.6.1 Métrique

Nous mesurerons l'efficacité de la décomposition du signal d'apprentissage et du signal de test à l'aide de la courbe du RSB en fonction de la densité relative. Nous mesurerons également le nombre de notes du signal d'apprentissage retrouvées par les motifs, et nous les comparerons aux notes présentes dans le signal de test.

9.6.2 Principe de l'expérience

Un signal audio synthétique (MIDI) de longue durée est découpé en deux sous-signaux. Le premier signal servira à l'apprentissage, et le deuxième à la généralisation, en prenant garde à ce que les deux signaux soient de la même nature. Nous ferons attention à laisser un long intervalle entre la partie du signal d'origine destinée à l'apprentissage et celle destinée à la généralisation pour que les signaux ne soient pas trop ressemblants.

Le signal d'apprentissage sera découpé en parties de plus en plus longues, et sur chacune de ces parties sera appris un dictionnaire, toujours de la même taille. Pour

chacun de ces dictionnaires, nous comparerons la qualité de la décomposition du signal d'apprentissage associé et du signal de test, ainsi que le nombre de notes retrouvées.

Comme dans l'expérience précédente, le fichier MIDI est généré avec une fréquence d'échantillonnage élevée, puis sous-échantillonné d'un facteur non-entier.

9.6.3 Protocole détaillé

Le protocole détaillé est décrit dans le tableau suivant

Signal	Signal audio MIDI - Concertino op. 26 pour clarinette et orchestre (réduction pour clarinette et piano), Weber : uniquement la première piste de clarinette. Échantillonné à 16 000 Hz et sous-échantillonné à 11 025 Hz.
Taille des dictionnaires	$K = 40$ motifs centrés de taille $L_m = 512$, soit 46.4 ms à 11 025 Hz. Nombre de paramètres : 20 480.
Signaux d'apprentissage	10 signaux de longueur 20 480 (1.86 s, autant d'échantillons que de paramètres dans le dictionnaire de motifs) à 2 048 000 (3 m 06 s, cent fois plus d'échantillons que de paramètres). Le nombre d'échantillons croît de façon exponentielle, et les échantillons sont pris à partir du début du signal.
Signal de test	Les 1 024 000 derniers échantillons du signal, soit 1 m 33 s.
Nombre de dictionnaires appris	20. Pour chaque signal d'apprentissage, deux dictionnaires invariants par translation sont appris, l'un avec invariance par déphasage et l'autre sans.
Critère d'arrêt de MP, dans l'algorithme K-SVD	lorsque le RSB atteint 20 dB.
Paramètres de l'algorithme K-SVD	30 itérations, et pour chaque itération, une seule mise à jour de chaque motif, sans mise à jour des coefficients ni de l'erreur.
Initialisation des motifs	tirage aléatoire gaussien, puis normalisation

9.6.4 Résultats

Avant d'exposer les résultats de l'expérience, décrivons quelques caractéristiques des signaux traités. Les dix signaux d'apprentissage contiennent de 1 note, pour le plus petit, à 34 notes pour le plus grand. Le signal de test compte également 34 notes. Parmi

ces 34 notes, 4 ne font pas partie d'aucun signal d'apprentissage, et parmi le signal d'apprentissage le plus long, 4 notes ne font pas partie du signal de test.

Cela signifie que, pour le signal le plus long, la plupart des notes (30 notes sur 34) sont communes avec le signal de test, ce qui atteste de la proximité de nature entre les deux signaux. En revanche, cette proximité n'est pas maximale, puisqu'il reste des notes que le dictionnaire appris sur le plus grand signal d'apprentissage ne pourra pas connaître. Les données reflètent bien la réalité de l'apprentissage où les données d'apprentissage sont proches, mais pas totalement similaires.

9.6.4.1 L'algorithme apprend les notes du signal d'apprentissage, indépendamment de la taille du signal

Intéressons-nous dans un premier temps à la qualité de l'apprentissage suivant la taille des signaux. La figure 9.13 représente, pour chaque signal d'apprentissage, le nombre de notes présentes dans le signal, le nombre de notes retrouvées par l'algorithme d'apprentissage, et le nombre de notes apprises absentes du signal d'apprentissage. Comme pour les histogrammes présentés sur la figure 9.4, par exemple, chaque motif est associé à la note dont la fréquence fondamentale est la plus proche de la fréquence prépondérante dans le spectre du motif.

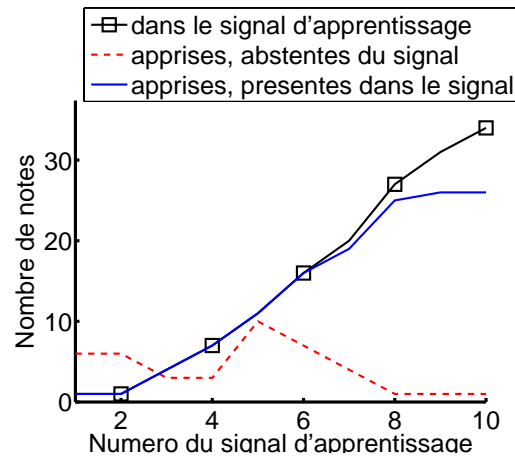


FIG. 9.13 – Comparaison du nombre de notes présentes dans les signaux d'apprentissage avec le nombre de notes retrouvées par les dictionnaires appris, et avec le nombre de notes apprises absentes des signaux d'apprentissage

Nous voyons sur la figure de gauche, consacrée au signal d'apprentissage, que le nombre de notes présentes dans le signal augmente au fur et à mesure que la taille du signal augmente (trait plein avec marqueurs), et que jusqu'au sixième signal (265 000 échantillons, soit 24 s à 11 025 Hz), toutes ces notes sont retrouvées par au moins un motif (trait plein sans marqueurs). A partir du septième signal, quelques-unes des notes supplémentaires ne sont pas retrouvées. En parallèle, nous pouvons voir qu'entre 3 et 8 notes apprises ne correspondent pas aux notes présentes dans le signal pour les 6

premiers signaux (ligne pointillée). A partir du huitième, seule une note apprise est absente du signal.

Nous pouvons interpréter ces résultats comme le fait que pour des petits signaux, le dictionnaire contient suffisamment de motifs pour coder d'autres formes d'ondes que les notes. Ainsi, le plus petit signal contient autant d'échantillons que le dictionnaire de motifs contient de paramètres, ce qui peut nous laisser penser que le dictionnaire apprend directement des tranches du signal. Nous verrons par la suite que ce n'est pas tout à fait le cas.

Pour les signaux les plus longs, il semble que le nombre de motifs soit insuffisant pour retrouver toutes les notes, puisque chaque motif, à l'exception d'un, est associé à une note présente dans le signal. Cela signifie que plusieurs motifs sont associés à la même note. Nous pourrions essayer d'augmenter le nombre de motifs pour voir si plus de notes sont retrouvées. Cela dit, il se peut qu'en augmentant le nombre de motifs, nous augmentions uniquement le nombre de motifs associés aux notes les plus fréquentes, sans pour autant coder les notes les moins fréquentes. Rappelons que l'algorithme MoTIF (section 9.2) reposait sur la contrainte de faible corrélation entre les différents motifs. Imposer une contrainte de ce type dans K-SVD pourrait permettre d'augmenter la diversité de formes dans le dictionnaire.

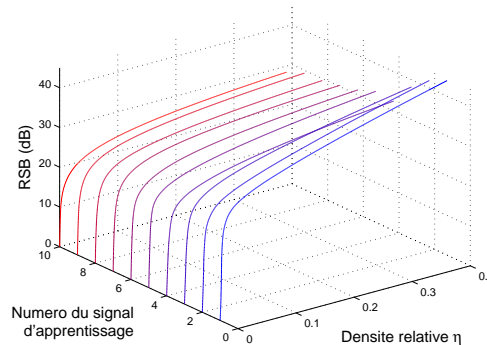


FIG. 9.14 – Qualité de la décomposition des dix signaux d'apprentissage sur les dictionnaires appris associés

Pour analyser la qualité de la décomposition, reportons-nous à la figure 9.14. Cette figure représente, pour chacun des dix signaux, la courbe du RSB de la décomposition du signal d'apprentissage en fonction de la diversité relative. Nous pouvons voir que plus le signal d'apprentissage est court, plus le RSB est élevé. Cela n'a rien d'étonnant, puisque lorsque le rapport entre taille du signal d'apprentissage et nombre de paramètres des motifs appris est petit, nous nous attendons à ce que ces motifs décrivent très précisément le signal. Quand le signal est plus grand, nous nous attendons au contraire à ce que les motifs reflètent la forme d'onde "moyenne" des notes, qui sont d'ailleurs plus nombreuses.

9.6.4.2 Seuls les longs signaux d'apprentissage permettent d'apprendre des dictionnaires généralisables

Nous allons maintenant analyser les résultats de l'expérience pour comprendre sous quelles conditions le dictionnaire appris se généralise avec succès, c'est-à-dire permet de décomposer parcimonieusement d'autres signaux de même nature que le signal d'apprentissage.

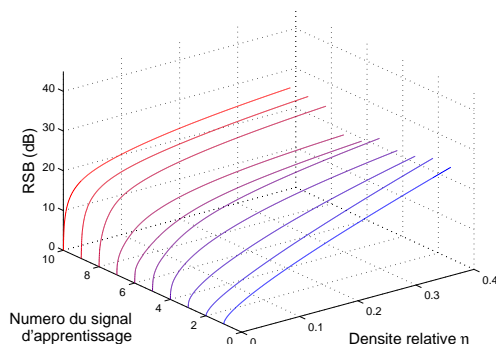


FIG. 9.15 – Qualité de la décomposition du signal de test sur les dix dictionnaires appris

La figure 9.15 représente, comme la figure 9.14 avant elle, la courbe du RSB en fonction de la diversité relative de la décomposition du signal sur chacun des signaux. Ici, le signal décomposé n'est plus celui qui a servi à apprendre le dictionnaire, mais le signal de test, qui est le même pour tous les dictionnaires. Rappelons qu'il contient 34 notes et dure 1 m 33 s.

Nous pouvons remarquer sur la figure que pour les dictionnaires appris sur des petits signaux (signaux 1, 2 et 3 notamment), la courbe du RSB croît linéairement en fonction de la diversité relative, et ceci dès les premiers atomes. C'est le mode de décomposition le plus lent qui existe en utilisant l'algorithme MP. En effet, il est prouvé que MP fait décroître l'énergie du résidu de façon exponentielle [DMZ94], ce qui se traduit dans notre cas par une augmentation linéaire du RSB en fonction de la densité relative. Le taux de décroissance de l'erreur, qui donne la pente de la droite, peut varier et donner de plus ou moins bonnes décompositions, mais le régime linéaire est le plus mauvais qui soit en utilisant MP.

De l'autre côté, pour les dictionnaires appris sur les plus longs signaux (signaux 8, 9 et 10 notamment), le RSB croît très vite avec la densité relative au début, puis la croissance ralentit lorsque la densité relative augmente. A titre de comparaison, pour une grande densité relative ($\eta = 0.3$), le dictionnaire appris sur le signal 1 (le plus petit) donne un RSB de 19.7 dB, alors que le dictionnaire appris sur le signal 10 (le plus grand) donne un RSB de 24.4 dB, soit un écart de 4.7 dB. En revanche, pour une faible densité relative ($\eta = 0.02$), correspondant aux premiers atomes, cet écart est de 11.5 dB (3.3 dB pour le dictionnaire appris sur le signal 1, et 14.8 dB pour le dictionnaire appris sur le signal 10).

Les dictionnaires appris sur de longs signaux permettent donc de décomposer avec une faible erreur le signal de test dès les premiers atomes, tandis que les dictionnaires appris sur les petits signaux ont besoin de beaucoup plus d'atomes pour parvenir au même niveau de RSB, ce qui signifie que la décomposition est beaucoup moins parcimonieuse.

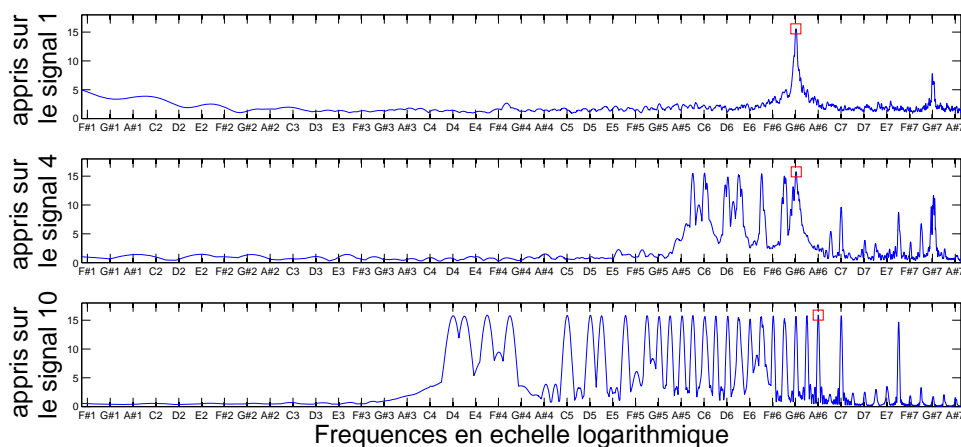


FIG. 9.16 – Amplitude crête parmi les spectres de tous les motifs, pour chacun des trois dictionnaires appris sur les signaux 1 (le plus petit), 4 et 10 (le plus grand)

Nous pouvons remarquer que pour de grandes valeurs de la densité relative, les dictionnaires appris sur les petits signaux parviennent au même niveau de RSB que les meilleurs dictionnaires. Nous l'expliquons par le fait que la largeur de bande des motifs appris sur des petits signaux est plus grande que pour les motifs appris sur des grands signaux, comme la figure 9.16 nous permet de le voir.

Cette figure représente, pour les dictionnaires appris sur les signaux 1 (petit), 4 et 10 (grand), la valeur maximale du spectre, à chaque fréquence, parmi tous les motifs du dictionnaire. Le résultat n'est pas un spectre en tant que tel, mais nous l'appellerons spectre par commodité. Nous voyons que pour le signal 10, de nombreux pics, correspondant aux notes apprises, sont présents sur des largeurs de bandes très petites. En dehors de ces pics, le spectre est très faible (une amplitude de l'ordre de 0.5), ce qui signifie que les fréquences correspondantes ne sont presque pas présentes, quelque soit le motif étudié.

En revanche, le spectre du dictionnaire appris sur le signal 1 comporte beaucoup moins de pics (uniquement pour la seule note présente dans le signal et son harmonique) et le reste du spectre a une valeur non négligeable, notamment en basses fréquences (une amplitude de 2 à 5). Pour le signal 4, le comportement est intermédiaire, avec un plus grand nombre de notes retrouvées et une amplitude de spectre un peu plus faible en dehors de ces pics.

L'interprétation de ce résultat est que les motifs du dictionnaire appris sur le signal 10 permettent de décrire très précisément les notes du signal de test qui ont été apprises, mais qu'ils sont presque incapables de représenter la partie fréquentielle située entre les

pics.

Ainsi, une première conséquence est que les quelques notes qui diffèrent peuvent être mal décomposées si ces notes se trouvent dans des creux. Notons que ce n'est pas le cas ici, puisque les notes uniquement présentes dans le signal de test sont **E4**, **C#5**, **D7** et **D#7**, pour lesquelles un petit pic d'amplitude est présent.

Le plus important est plutôt que toutes les fréquences éloignées des pics du spectre ont une amplitude très faible. Or, il est prouvé que pour certains types de dictionnaires, dont notamment le dictionnaire de Gabor, le résidu de MP converge vers un bruit blanc stationnaire [DMZ94]. Or, l'espérance de l'énergie de chaque fréquence est la même pour un bruit blanc. En supposant que le résultat est valable pour les dictionnaires que nous manipulons, nous voyons alors tout de suite qu'une partie du spectre du résidu ne peut pas être correctement décomposée à cause du manque des fréquences correspondantes dans les motifs appris. Nous pouvons en déduire que des motifs large bande seraient utiles, en complément, pour décomposer le résidu lorsque celui-ci est petit.

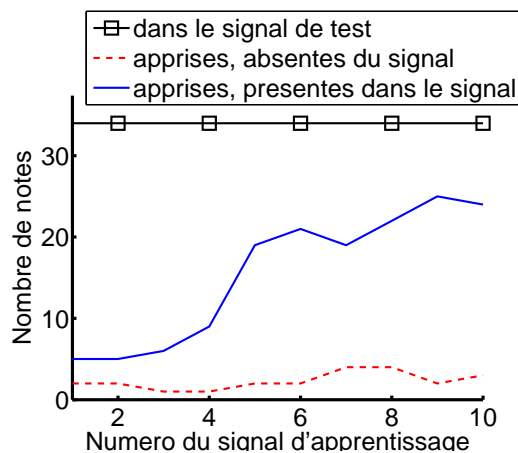


FIG. 9.17 – Comparaison du nombre de notes présentes dans le signal de test avec le nombre de notes retrouvées par les dictionnaires appris, et avec le nombre de notes apprises absentes du signal de test

Pour finir avec la discussion sur la généralisation, regardons la figure 9.17. Elle représente, à l'instar de la figure 9.13 pour les signaux d'apprentissage, la comparaison du nombre de notes présentes dans le signal de test et dans les dictionnaires de motifs appris. Pour chaque dictionnaire sont représentés le nombre de notes apprises présentes dans le signal de test et le nombre de notes apprises absentes du signal de test.

Nous voyons que le nombre de notes apprises présentes dans le signal de test augmentent avec la taille du signal d'apprentissage, ce qui est logique puisque le signal d'apprentissage contient lui aussi plus de notes, et que presque toutes sont communes avec le signal de test. Entre 2 et 4 notes apprises sont absentes du signal de test, quelle que soit la taille du signal d'apprentissage, ce qui paraît également normal, puisque 4 notes du signal de test sont absentes des signaux d'apprentissage. Le nombre de notes du signal de test retrouvées est inférieur, en proportion et en absolu, au nombre de notes

des signaux d'apprentissage retrouvées, ce qui est cohérent avec le fait que les signaux d'apprentissage soient mieux décomposés que le signal de test.

Chapitre 10

Séparation de sources : estimation des sources par dictionnaire structuré

Dans ce chapitre, nous revenons au sujet de la séparation de sources, déjà longuement évoqué au chapitre 4. Nous y avons notamment abordé les techniques de séparation de sources utilisant les décompositions parcimonieuses, dans le cadre du modèle classique de dictionnaire. Ces techniques reposent sur la différenciation des sources en fonction de leur direction spatiale, ou en fonction de la parcimonie de leur décomposition dans divers dictionnaires (analytiques).

Nous proposons ici de réunir les deux techniques en une méthode commune, et de les enrichir en remplaçant le choix des dictionnaires parmi une collection de dictionnaires analytiques connus, par l'apprentissage de dictionnaires adaptés aux sources et structurés (engendrés par des motifs déformés). L'algorithme de séparation est exposé à la section 10.2.

Rappelons les conditions du mélange que nous traitons. Nous nous plaçons dans le cas sous-déterminé, où le nombre de sources est supérieur au nombre de capteurs. Nous supposons également, lorsqu'il y a au moins deux capteurs, que les paramètres de mélange sont connus (les fonctions de transfert entre chaque source et chaque capteur). Enfin, nous supposons disposer d'exemples représentatifs de chacune des sources, permettant d'apprendre les dictionnaires.

Cela signifie que nous ne traitons pas le problème de séparation aveugle, pour lequel il faudrait estimer les paramètres de mélange et les modèles à partir du mélange, mais uniquement le problème de l'estimation des sources. Nous pouvons par exemple nous reporter à [Com98, VSI⁺04, FG06, AGB06] pour l'estimation des paramètres du mélange, ou à [Oze06] pour l'adaptation des modèles de source sur le mélange.

Nous présentons deux études expérimentales sur la qualité de la séparation obtenue suivant l'algorithme présenté en section 10.1, la première dans le cas multicanal (section 10.3) et la seconde dans le cas monocanal (section 10.4).

Dans la section 10.2, nous exposons le formalisme général de séparation lorsque

les fonctions de mélange suivent le modèle des fonctions de déformation présenté au chapitre 7. Ce cadre très général, qui inclut notamment les mélanges avec délai et les mélanges convolutifs, et permet de s'affranchir de l'existence de sources ponctuelles, n'a pas été expérimenté, en raison de l'investissement nécessaire à son implémentation. Il sera toutefois intéressant d'évaluer les performances de l'algorithme proposé pour des mélanges moins classiques que le mélange linéaire instantané, pour lequel il existe déjà de nombreuses méthodes performantes.

Nous tenons à préciser que la méthode de séparation de sources présentée ici doit être vue comme une application du travail présenté précédemment, et n'a pas constitué le cœur de ce travail de thèse. Lors des expériences, la méthode ne devra donc pas être considérée comme concurrente des méthodes les plus pointues de séparation de sources, mais plutôt comme une méthode préliminaire, dont les résultats peuvent encourager l'utilisation de la parcimonie, avec notamment l'apprentissage de dictionnaires structurés, pour la séparation de sources.

10.1 Cas particulier de la séparation d'un mélange linéaire instantané

Nous présentons dans cette section une méthode de séparation de sources dans le cas d'un mélange linéaire instantané. Cette méthode a été implémentée et expérimentée. Nous présentons ces expériences dans les sections 10.3 et 10.4.

La méthode de séparation se base sur la connaissance des paramètres de mélange, et sur la connaissance d'exemples des sources à séparer. Le modèle de mélange est le suivant :

$$\underline{\mathbf{X}} = \sum_{n=1}^N E_{\mathbf{a}_n}[\underline{\mathbf{s}}_n] , \quad (10.1)$$

ce qui signifie que les sources $\underline{\mathbf{s}}_n$ sont réparties sur les différents canaux par la fonction d'expansion $E_{\mathbf{a}_n}$ selon les coefficients du vecteur \mathbf{a}_n , qui indique la direction spatiale des sources dans le cas multicanal (voir 4.1.2), pour donner le mélange $\underline{\mathbf{X}}$. Dans le cas où le mélange est un signal monocanal, le vecteur \mathbf{a}_n est réduit à un simple scalaire et le mélange est une combinaison linéaire des sources.

Nous supposons que les vecteurs de direction \mathbf{a}_n sont connus, et nous supposons posséder des exemples $\tilde{\underline{\mathbf{s}}}_n$ des sources $\underline{\mathbf{s}}_n$. Le principe de la méthode de séparation, inspirée de l'ACM (Analyse en Composantes Morphologiques, voir la section 4.2.3.2), est le suivant. En premier lieu, pour chaque source $\underline{\mathbf{s}}_n$, un dictionnaire \mathcal{D}^n adapté est appris à partir de l'exemple $\tilde{\underline{\mathbf{s}}}_n$. Ensuite, tous les dictionnaires \mathcal{D}^n sont regroupés en un grand dictionnaire \mathcal{D} , après avoir été transformés par les fonctions $E_{\mathbf{a}_n}$. Enfin, une décomposition du mélange $\underline{\mathbf{X}}$ est effectuée sur ce grand dictionnaire \mathcal{D} par Matching Pursuit, et chaque atome est affecté à la source correspondant à son dictionnaire d'origine. Cette version de Matching Pursuit a été baptisée Demixing Pursuit (voir 8.5).

Détaillons les étapes.

- A) Le dictionnaire \mathcal{D}^n appris sur la source $\hat{\underline{s}}_n$ est constitué d'un dictionnaire de motifs \mathcal{M}^n et d'une fonction de déformation $F_{\psi, \phi}^n$. Il est appris par exemple en utilisant l'algorithme K-SVD présenté à la section 9.3.
- B) Le dictionnaire \mathcal{D} regroupe tous les dictionnaires \mathcal{D}^n appris et contient les atomes suivants :

$$E_{\mathbf{a}_n} \circ F_{\psi, \phi}^n[\underline{m}_k^n], \quad (10.2)$$

paramétrés par le numéro n de la source correspondante, le numéro k du motif et les paramètres ψ et ϕ de la fonction de déformation.

- C) La séparation est effectuée par l'algorithme Demixing Pursuit :

Séparation par Demixing Pursuit :

Initialisation du résidu $\underline{\mathbf{R}}_0 = \underline{\mathbf{X}}$, des sources estimées $\hat{\underline{s}}_n = \underline{\mathbf{0}}$ et du compteur $I = 0$.

Faire

- (a) Mise à jour du compteur : $I = I + 1$
 (b) Recherche de l'atome optimal dans le dictionnaire \mathcal{D} :

$$(n_I, k_I, \psi_I, \phi_I) = \arg \min_{n, k, \psi, \phi} \|\underline{\mathbf{R}}_{I-1} - E_{\mathbf{a}_n} \circ F_{\psi, \phi}^n[\underline{m}_k^n]\|_2^2 \quad (10.3)$$

- (c) Mise à jour du résidu :

$$\underline{\mathbf{R}}_I = \underline{\mathbf{R}}_{I-1} - E_{\mathbf{a}_{n_I}} \circ F_{\psi_I, \phi_I}^{n_I}[\underline{m}_{k_I}^{n_I}] \quad (10.4)$$

- (d) Mise à jour de la source $\hat{\underline{s}}_{n_I}$:

$$\hat{\underline{s}}_{n_I} = \hat{\underline{s}}_{n_{I-1}} + F_{\psi_I, \phi_I}^{n_I}[\underline{m}_{k_I}^{n_I}] \quad (10.5)$$

Tant que la convergence n'est pas atteinte

L'étape (10.3) a déjà été détaillée dans le chapitre 8, et consiste à trouver les paramètres ψ_I de façon analytique, et les paramètres n_I , k_I et ϕ_I par recherche exhaustive.

Il faut noter que la mise à jour de la source (10.5) est effectuée dans le domaine des sources \underline{s}_n , ce qui correspond à estimer les sources émises. Il serait également possible, en ajoutant l'atome total $E_{\mathbf{a}_{n_I}} \circ F_{\psi_I, \phi_I}^{n_I}[\underline{m}_{k_I}^{n_I}]$, qui a été enlevé au résidu, de reconstruire les sources dans le domaine du mélange, c'est-à-dire d'estimer les sources perçues par les capteurs.

Le fait de rester dans le domaine du mélange ne signifie bien sûr pas que la connaissance de la fonction $E_{\mathbf{a}_{n_I}}$ est inutile, car elle sert dans la recherche du meilleur atome.

Cette méthode d'estimation des sources est expérimentée dans les sections 10.3 et 10.4. Avant cela, présentons cette méthode dans un cadre plus général de mélanges linéaires.

10.2 Perspective : formalisme global de séparation utilisant des dictionnaires structurés

Nous élargissons le modèle de séparation de sources du cas linéaire instantané au cas plus général qui contient par exemple les mélanges avec délai ou avec convolution. Le modèle de mélange linéaire est le suivant :

$$\underline{\mathbf{X}} = \sum_{n=1}^N \underline{\mathbf{X}}^n . \quad (10.6)$$

Les sources sont notées $\underline{\mathbf{X}}^n$ en raison de leur appartenance au “domaine” du mélange, ce qui signifie qu’elles sont de même nature que le mélange $\underline{\mathbf{X}}$.

Ces sources $\underline{\mathbf{X}}^n$ peuvent avoir été générées selon la règle suivante :

$$\underline{\mathbf{X}}^n = M^n[\underline{\mathbf{s}}_n] , \quad (10.7)$$

où $\underline{\mathbf{s}}_n$ sont des signaux monocanaux représentant des sources ponctuelles, et M^n sont des fonctions de mélange linéaires envoyant les sources ponctuelles dans le domaine du mélange.

Si le mélange $\underline{\mathbf{X}}$ est monocanal, alors les fonctions M^n peuvent par exemple être l’identité, la translation d’un délai fixé, ou la convolution par un filtre donné. Si le mélange est multicanal, alors les fonctions M^n sont des fonctions d’expansion, qui appliquent au motif un filtre différent sur chaque canal (là encore, le filtre peut par exemple être un Dirac au temps 0, un Dirac positionné au délai, ou un filtre RIF).

Les sources $\underline{\mathbf{X}}^n$ peuvent également être considérées comme telles, et non plus comme résultant du modèle (10.7). Elles représentent le signal perçu par les capteurs lorsque toutes les autres sources sont muettes. **En considérant les sources au niveau des capteurs, nous nous affranchissons de l’hypothèse d’existence de sources ponctuelles.** Nous pouvons alors par exemple traiter des exemples plus réalistes de sources réparties et en champ proche.

Donnons un exemple d’un tel mélange. Sur la figure 10.1 est représenté un dispositif expérimental particulier. Deux microphones sont placés juste au dessus d’un xylophone. Supposons que les sources que nous souhaitons séparer sont les trois régimes du xylophone : grave (vert, source 3), médium (bleu, source 2) et aigu (rouge, source 1). Nous voyons bien que la note la plus aiguë et la note la plus grave de la source 1 ne sont pas situées dans la même direction spatiale, par rapport à n’importe quel microphone. La note la plus grave de la source 1 est même plus proche spatialement de la note la plus aiguë de la source 2, que de la note la plus aiguë de la source 1.

Il est donc impossible dans ce cas-ci de représenter les sources comme des objets ponctuels, d’autant plus que les microphones sont très proches des sources et que les notes sont générées à la fois par le coup de mailloche sur la lame et par la résonance dans le tube. Nous supposons toutefois que le mélange des sources est égal à la somme des sources jouant séparément (linéarité). Cette hypothèse est valide si les sources interfèrent suffisamment peu entre elles.

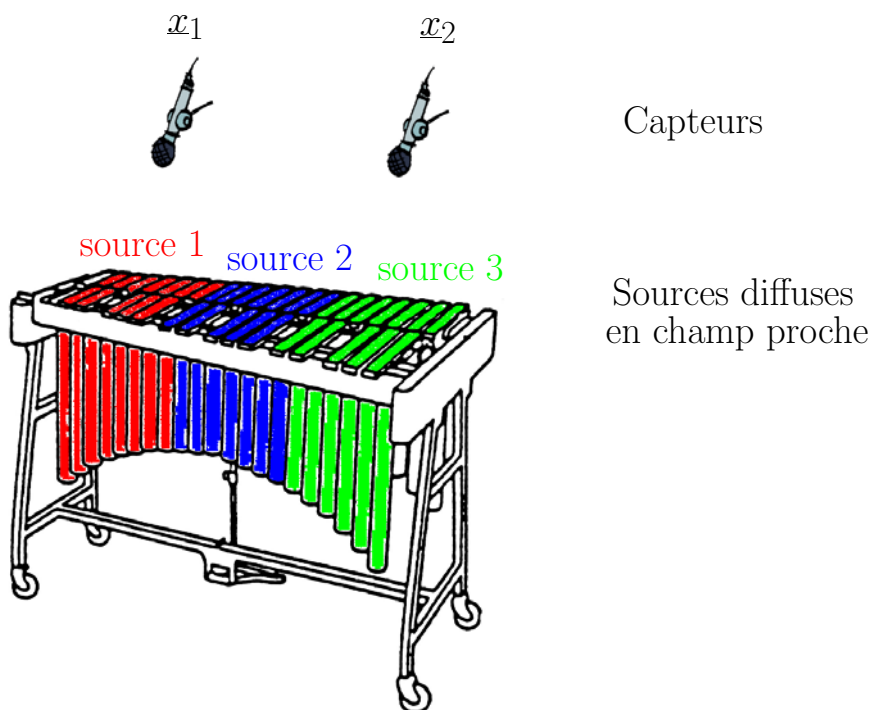


FIG. 10.1 – Exemple de mélange de sources diffuses en champ proche. Les trois sources correspondent aux notes aiguës, aux notes médium et aux notes graves du xylophone. Deux microphones placés juste au dessus de l’instrument enregistrent le mélange.

Dans cet exemple, les sources seront considérées au niveau des capteurs (sources notées $\underline{\mathbf{X}}^n$) et non pas au niveau d’une source ponctuelle qui, nous venons de le voir, n’existe pas.

Dans le cas où la source $\underline{\mathbf{X}}^n$ peut se factoriser comme $M^n[\underline{\mathbf{x}}_n]$, comme nous supposons que la fonction de mélange M^n est connue, il est équivalent de traiter les sources au niveau des capteurs. Si nous disposons d’exemples de sources $\underline{\tilde{\mathbf{x}}}_n$, nous disposons donc également des sources $\underline{\tilde{\mathbf{X}}}^n = M^n[\underline{\tilde{\mathbf{x}}}_n]$. Nous nous ramènerons à ce cas.

Le principe de l’algorithme d’estimation des sources repose alors sur la connaissance d’exemples de sources $\underline{\tilde{\mathbf{X}}}^n$. Ceux-ci peuvent par exemple être obtenus en enregistrant, au niveau des capteurs, la seule source n , pendant que les autres sont muettes.

Dans le cas du xylophone, cela signifie que l’exemple de source 1 est enregistré en jouant uniquement sur les lames aiguës, et en répétant la manœuvre pour les autres sources. Si la scène enregistrée est maintenant un orchestre, le “calibrage” des modèles de sources est effectué en enregistrant séparément au préalable chaque instrument, ou chaque groupe d’instruments.

Les modèles correspondent à des dictionnaires appris sur les exemples de sources. Si nous reprenons l’exemple du xylophone, où le mélange est stéréo, les motifs appris ne seront pas mono, mais bien stéréo (deux canaux) eux aussi. De tels motifs peuvent tou-

tefois être appris avec l'algorithme K-SVD, en utilisant simplement l'astuce d'entrelacer les canaux de droite et de gauche des sources d'apprentissage et des motifs, pour traiter tous les signaux en tant que signaux monocanaux. L'opération inverse peut ensuite être faite pour analyser les dépendances entre les deux canaux des motifs appris.

La séparation se fait ensuite en décomposant le mélange sur l'ensemble des dictionnaires appris, et en discriminant les atomes *a posteriori* selon le dictionnaire utilisé.

Le principe de cette séparation est simple et ne nécessite aucune connaissance sur la géométrie de la salle et des capteurs, puisque toute la différenciation entre les sources se situe dans la différence de perception par les capteurs. Il suffit d'effectuer une étape préalable d'entraînement des modèles, pour que ces modèles apprennent à distinguer les sources à partir des enregistrements.

Évidemment, en pratique, l'étape d'entraînement peut s'avérer lourde pour les musiciens, puisqu'il faut enregistrer une palette de sonorités suffisamment large pour couvrir l'ensemble des sons qui peuvent être produits par la source. De plus, nous nous attendons à ce que la séparation produise une qualité du même ordre que pour les cas multicanal et monocanal que nous présentons ensuite, et qui reste assez médiocre.

En revanche, la méthode a l'avantage de la simplicité et de pouvoir être appliquée à tout type de mélange.

Résumons les étapes de la méthode :

- A) Pour chaque source n , apprentissage d'un dictionnaire \mathcal{D}^n sur un exemple $\tilde{\mathbf{X}}^n$ suffisamment long, enregistré au niveau des capteurs et où seule la source n joue. Le dictionnaire est constitué de motifs \mathcal{M}^n et d'une fonction de déformation $F_{\psi, \phi}^n$. Il est appris par exemple en utilisant l'algorithme K-SVD présenté à la section 9.3.
- B) Les dictionnaires \mathcal{D}^n appris sont regroupés en un grand dictionnaire \mathcal{D} contenant tous les atomes $F_{\psi, \phi}^n [m_k^n]$ paramétrés par le numéro n de la source correspondante, le numéro k du motif et les paramètres ψ et ϕ de la fonction de déformation.
- C) La séparation du mélange enregistré \mathbf{X} est effectuée en le décomposant sur le dictionnaire \mathcal{D} par Matching Pursuit, pour donner :

$$\mathbf{X} = \sum_{i=1}^I F_{\psi_i, \phi_i}^{n_i} [m_{k_i}^{n_i}] + r_I \quad (10.8)$$

et en affectant ensuite chaque atome $F_{\psi_i, \phi_i}^{n_i} [m_{k_i}^{n_i}]$ à la source $\hat{\mathbf{X}}^{n_i}$ correspondante.

- D) Les sources estimées $\hat{\mathbf{X}}^n$ sont dans le domaine "mélange", c'est à dire que, par exemple, si deux microphones enregistrent le mélange, les sources sont stéréophoniques, comme l'est le mélange \mathbf{X} .

Précisons que cette méthode de séparation de sources n'a pas été implémentée pour plus qu'un capteur et qu'elle reste à valider expérimentalement pour des mélanges réels notamment. Dans les deux sections qui suivent, nous présentons la méthode de séparation par Demixing Pursuit, dans les cas linéaire instantané multicanal et monocanal. Ces deux expériences sont des cas particuliers de la méthode que nous venons de présenter (pour le mélange multicanal, les fonctions M^n sont les fonctions d'expansion $E_{\mathbf{a}_n}$, et pour le mélange monocanal, le nombre de capteurs est limité à un).

10.3 Séparation multicanale sur des mélanges synthétiques

Dans cette section, nous évaluons l'efficacité de la séparation de sources à base de dictionnaires structurés, en utilisant Demixing Pursuit, dans le cas multicanal. Nous étudierons la séparation d'un mélange multicanal de trois sources sonores synthétiques de type MIDI. Nous avons montré précédemment que les signaux MIDI peuvent être représentés efficacement à l'aide de dictionnaires structurés.

Rappelons que nous ne traitons que l'aspect d'estimation des sources, en supposant que les directions de sources sont connues parfaitement. La phase d'estimation des directions des sources nécessite un traitement particulier que nous ne traitons pas ici.

Nous souhaitons montrer dans un premier temps que la séparation de sources multicanale peut se faire à l'aide de décompositions parcimonieuses, sur un dictionnaire standard commun à toutes les sources.

Nous souhaitons également évaluer l'importance de l'apport de la connaissance des sources, en utilisant un dictionnaire différent par source, dans les performances de séparation. Pour cela, il est nécessaire de disposer d'exemples séparés des sources pour apprendre les dictionnaires nécessaires à la séparation. Le mélange à séparer est composé de signaux de la même nature, mais différents des signaux d'apprentissage.

Nous comparerons les performances de DP, avec ou sans connaissance *a priori* sur les sources, à deux méthodes classiques de type DUET.

10.3.1 Métrique

La qualité de la séparation sera mesurée en termes de distorsion des sources estimées. La distorsion d'une source est composée d'artefacts de calcul et d'interférences des autres sources. La mesure que nous présentons a été proposée par Vincent, Gribonval et Févotte [VGF06] en 2006, et nécessite de connaître les sources originales.

Les interférences correspondent à l'influence des autres sources dans l'estimation d'une source. Les artefacts correspondent quant à eux à tous les autres types de distorsions, et sont introduits lors du calcul. Une séparation linéaire ne produira par exemple aucune artefact. Vincent, Gribonval et Févotte ont également pris en compte le bruit présent sur le mélange dans l'élaboration de leurs mesures, mais comme nous travaillons dans un modèle sans bruit, nous présentons une version simplifiée des mesures.

Mathématiquement, en notant $\hat{\underline{s}}_n$ la source n estimée après séparation, et \underline{s}_n la source originale, la source estimée peut être décomposée comme :

$$\hat{\underline{s}}_n = \underline{s}_{cible} + \underline{r}_{interf} + \underline{r}_{artef} , \quad (10.9)$$

où \underline{s}_{cible} représente la portion de la source bien estimée, \underline{r}_{interf} représente l'erreur d'estimation due aux autres sources, et \underline{r}_{artef} représente l'erreur d'estimation due aux artefacts de calcul.

Ces trois signaux sont calculés de la façon suivante : \underline{s}_{cible} est la projection de la source $\hat{\underline{s}}_n$ sur la vraie source \underline{s}_n , \underline{r}_{interf} est la différence entre la projection de $\hat{\underline{s}}_n$ sur l'espace engendré par toutes les sources et la projection sur la vraie source \underline{s}_n . Enfin,

\underline{r}_{artef} est la différence entre $\hat{\underline{x}}$ et sa projection sur l'espace engendré par toutes les sources.

Trois mesures sont dérivées de cette décomposition. La première, notée RSD, pour rapport signal à distorsion, est définie par :

$$\text{RSD} = 10 \cdot \log_{10} \frac{\|\underline{s}_{cible}\|_2^2}{\|\underline{r}_{interf} + \underline{r}_{artef}\|_2^2}. \quad (10.10)$$

La deuxième, notée RSI, pour rapport signal à interférences, est définie par :

$$\text{RSI} = 10 \cdot \log_{10} \frac{\|\underline{s}_{cible}\|_2^2}{\|\underline{r}_{interf}\|_2^2}. \quad (10.11)$$

Enfin, la troisième, notée RSA, pour rapport signal à artefacts, est définie par :

$$\text{RSA} = 10 \cdot \log_{10} \frac{\|\underline{s}_{cible} + \underline{r}_{interf}\|_2^2}{\|\underline{r}_{artef}\|_2^2}. \quad (10.12)$$

Ces trois mesures sont du type RSB, avec une échelle en décibels, et plus elles sont élevées, meilleure est la séparation. Le RSD mesure la qualité globale de la séparation, et correspond à peu de choses près au minimum du RSI et du RSA. Le RSI permet de ne s'intéresser qu'à l'erreur due à la présence d'autres sources dans la source estimée. Le RSA mesure lui la dégradation de l'estimation due à la méthode de calcul.

10.3.2 Principe de l'expérience

Trois sources audio MIDI, jouant la même mélodie, sont mélangées selon une matrice de mélange qui les disposent dans l'espace stéréo. Les trois sources sont disponibles séparément, ce qui permettra de comparer les sources estimées aux vraies sources, mais aussi d'apprendre un dictionnaire par source.

Nous utilisons la première partie des sources pour l'apprentissage de dictionnaires adaptés et la deuxième partie pour le test de séparation. Nous utilisons des signaux d'apprentissage suffisamment longs pour ne pas poser de problème de sur-apprentissage.

Cinq méthodes d'estimation des sources seront comparées. Les deux premières méthodes viennent des algorithmes DUET et BZ (pour Bofill et Zibulevski [BZ01]), et consistent à attribuer les coefficients temps-fréquence obtenus par TFCT aux différentes sources, en fonction du rapport de ces coefficients sur le canal droit et le canal gauche.

Les trois autres méthodes utilisent l'algorithme DP, qui décompose le mélange sur des atomes formés par l'expansion multicanale, selon une des directions des sources, de motifs monocanaux. Les deux premières de ces trois méthodes consistent à utiliser le même dictionnaire de motifs pour toutes les directions, à savoir un dictionnaire de Gabor, "partiellement" invariant par translation (décalage entre atomes égal à la moitié de la longueur des motifs) pour la première méthode, et invariant par translation pour la deuxième méthode. La troisième méthode consiste à utiliser, pour chaque source, un

dictionnaire différent, appris sur des exemples de la source correspondante, et invariant par translation et par déformation de phase.

Pour pouvoir effectuer la comparaison de façon équitable, nous allons utiliser des tailles cohérentes pour les différents paramètres à régler. Cependant, il faut noter que les méthodes étant trop différentes, nous ne pouvons pas assurer l'équivalence de traitement.

Ainsi, pour les méthodes DUET et BZ, les atomes sont formés par un motif et une amplitude multicanale libre. Pour les méthodes DP, l'amplitude est proportionnelle à la direction de la source, et le nombre d'atomes n'est pas comparable à celui des méthodes précédentes (il est de mesure nulle dans l'espace des atomes à amplitude multicanale libre). Nous utiliserons le même nombre d'atomes pour DUET et BZ, en fixant le même décalage entre trames pour le calcul de la TFCT. En revanche, comme le nombre d'atomes est incommensurablement plus petit, nous permettrons aux méthodes DP d'utiliser un décalage d'un échantillon pour compenser la perte de liberté dans l'amplitude. Rappelons tout de même que cette perte de liberté correspond à connaître à l'avance la bonne direction, ce qui devrait être bénéfique.

Nous comparons ensuite les résultats en termes de RSD, RSI et RSA.

10.3.3 Protocole détaillé

Signaux sources	Signal audio MIDI de clarinette (source 1) et de piano (main droite, source 2 et main gauche, source 3). Concertino op. 26 pour clarinette et orchestre (réduction pour clarinette et piano), Weber. Échantillonné à 16 000 Hz puis sous-échantillonné à 11 025 Hz.
Signaux d'apprentissage (pour chaque source)	les 1 024 000 premiers échantillons (1 m 33 s à 11 025 Hz) échantillons
Sources de test	les 1 024 000 derniers échantillons (1 m 33 s à 11 025 Hz) échantillons
Matrice de mélange	Si les angles 0 et $\pi/2$ correspondent à droite et gauche, les sources sont placées à $\pi/8$, $\pi/4$ (milieu) et $3\pi/8$: $\mathbf{A} = \begin{bmatrix} 0.92 & 0.71 & 0.38 \\ 0.38 & 0.71 & 0.92 \end{bmatrix}.$
Paramètres de la TFCT pour DUET et BZ	Fenêtres de taille $L_m = 512$, décalées de $L_m/2 = 256$.
Dictionnaire de motifs de Gabor	255 motifs de Gabor complexes, correspondant à 255 fréquences, ainsi qu'un motif constant (0 Hz) et qu'un motif correspondant à la fréquence de Nyquist (5 512.5 Hz, la moitié de la fréquence d'échantillonnage). Les motifs sont de taille $L_m = 512$ (46.4 ms à 11 025 Hz). Les motifs de Gabor complexes comptent pour deux motifs réels, ce qui explique que le nombre de fréquences soit 255 et pas 510.
Dictionnaires de Gabor \mathcal{D}_{g_1} et \mathcal{D}_{g_2}	Décalage temporel de respectivement $L_m/2 = 256$ échantillons et de 1 échantillon entre atomes.
Dictionnaires \mathcal{D}_{a_n} appris sur les signaux	$K = 50$ motifs de taille $L_m = 512$, soit 46.4 ms à 11 025 Hz. Nombre de paramètres : 20 480. Le dictionnaire est invariant par translation et déformation de phase.
Critère d'arrêt de MP, dans K-SVD	lorsque le RSB atteint 20.
Paramètres de l'algorithme K-SVD	30 itérations, et pour chaque itération, une seule mise à jour de chaque motif, sans mise à jour des coefficients ni de l'erreur.
Initialisation des motifs	tirage aléatoire gaussien, puis normalisation

10.3.4 Résultats

Nous présentons les performances moyennes de séparation des algorithmes sur la figure 10.2. Nous représentons la valeur du RSI moyennée sur les trois sources, en fonction du RSA moyen, pour chacun des cinq algorithmes. Cette représentation graphique fait apparaître en un coup d’œil les avantages en termes d’interférences ou d’artefact des méthodes, en rappelant que les valeurs doivent être les plus grandes possibles, et donc que les meilleurs points se situent en haut à droite du graphique.

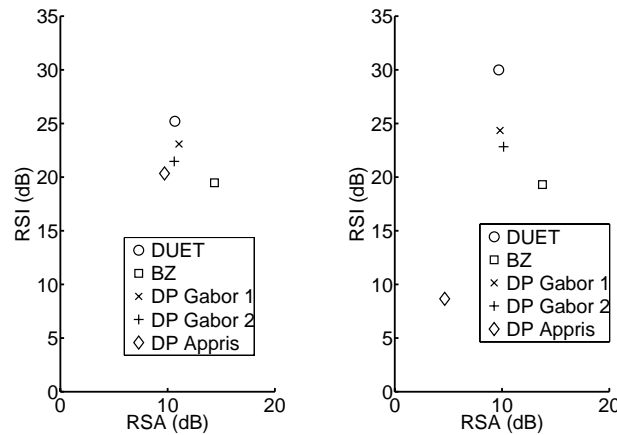


FIG. 10.2 – Performances de séparation du mélange d’apprentissage (gauche) et de test (droite).

Pour le graphique de gauche, le mélange est constitué des sources qui ont servi à apprendre les dictionnaires, tandis que pour le graphique de droite, le mélange est composé de sources de test. La nature des sources d’apprentissage et de test étant la même, les performances ne doivent être modifiées que pour l’algorithme DP utilisant les dictionnaires appris. Les quelques différences de performance des autres algorithmes entre les deux mélanges relèvent donc uniquement de la variabilité entre les signaux.

Précisons que dans le cas présent, résumer les performances des différentes sources par leur moyenne ne modifie pas les “rapports de force” entre les différents algorithmes.

En rappelant que des RSA et RSI élevés correspondent à des “taux” d’artefacts et d’interférences faibles, nous pouvons tout d’abord voir que l’algorithme DUET (rond) et l’algorithme BZ (carré) sont les meilleurs, chacun dans leur catégorie. DUET produit peu d’interférences et BZ peu d’artefacts. Cette différence s’explique par un détail des deux algorithmes. Chaque case temps-fréquence est assignée à une source, dans le cas de DUET et à deux sources dans le cas de BZ. Cela signifie, pour BZ, que les plans temps-fréquence des sources estimées contiennent en moyenne deux fois moins de “trous” qu’avec DUET. La reconstruction contient donc moins d’artefacts liés à ces trous. En revanche, obliger la séparation de chaque case temps-fréquence en deux sources augmente les chances d’attribuer de l’énergie à deux sources alors qu’une seule est active, et donc les interférences.

En utilisant le “petit” dictionnaire de Gabor \mathcal{D}_{g_1} , qui correspond à la TFCT utilisée pour les algorithmes précédents, l’algorithme Demixing Pursuit (croix) permet une séparation intermédiaire entre DUET et BZ, notamment au niveau des interférences. Pour les artefacts, le niveau est presque le même que DUET, ce qui se conçoit par le fait que chaque atome sélectionné par DP est également alloué à une seule source. Il faut aussi remarquer que la décomposition du mélange par DP ne couvre pas nécessairement tout le plan temps-fréquence, contrairement à DUET et BZ, ce qui peut engendrer des artefacts lorsque trop peu d’atomes sont sélectionnés. Le niveau d’interférences est plus bas que pour BZ, pour les mêmes raisons qu’avec DUET.

Nous remarquons que l’utilisation d’un “grand” dictionnaire de Gabor (plus), invariant par translation, ne produit pas une meilleure séparation qu’avec le “petit” dictionnaire de Gabor qui n’est que “partiellement” invariant par translation. Cela signifie que pour ces signaux MIDI, il ne semble pas nécessaire de dépenser l’important effort de calcul supplémentaire permettant d’assurer l’invariance par translation.

Pour finir, nous pouvons voir que l’utilisation d’un dictionnaire appris par source (losange) plutôt qu’un dictionnaire de Gabor commun n’apporte pas le gain de performances que nous pouvions espérer en raison de l’utilisation de la diversité de formes en plus de la diversité spatiale. Nous allons tenter d’en expliquer les raisons.

Tout d’abord, même lorsque la séparation est effectuée sur le mélange d’apprentissage, l’algorithme produit plus d’artefacts que toutes les autres méthodes, notamment DP avec dictionnaire de Gabor, et seul BZ produit plus d’interférences. Nous ne pouvons donc pas nous attendre à ce que cet algorithme concurrence les autres sur le mélange de test.

Les mauvaises performances sur le mélange d’apprentissage peuvent tout d’abord s’expliquer par la petite taille du dictionnaire par rapport au dictionnaire de Gabor (40 motifs contre 256). Les motifs de Gabor sont très bien adaptés à la décomposition de ces signaux MIDI qui sont très purs, ce qui est confirmé par le fait que les motifs appris sont en majorité très proches de sinusoïdes. Schématiquement, utiliser 256 sinusoïdes plutôt que 40 augmente les chances de reconstruire parcimonieusement le signal. Nous exposons deux motifs représentatifs, l’un appris sur la clarinette et l’autre sur le piano (main droite), sur la figure 10.3. Les motifs appris sur la clarinette sont proches de sinusoïdes, tandis que les motifs appris sur le piano représentent des accords (ici accord sol, do, ré), composés uniquement de quelques fréquences prépondérantes.

Nous ne parlons pour l’instant que du taux d’artefacts élevé, qui est dû à l’insuffisance de modélisation des dictionnaires appris par rapport au dictionnaire de Gabor. Nous avons notamment remarqué cette faiblesse en mesurant le RSA local (sur une fenêtre glissante de durée 46.4 ms, la taille des motifs). A un instant donné du mélange, seule la clarinette joue, et le piano est muet. Le RSA est alors environ de 65 dB pour BZ, de 60 dB pour DUET, de 35 dB pour DP avec un dictionnaire de Gabor, et de 25 dB pour DP avec des dictionnaires appris. Cela signifie que le dictionnaire appris sur la clarinette ne permet pas de coder aussi bien la clarinette que le dictionnaire de Gabor.

Au delà de l’insuffisance de modélisation, le faible intérêt de différencier les sources selon leur forme vient de la proximité entre les dictionnaires, pour les sources que nous

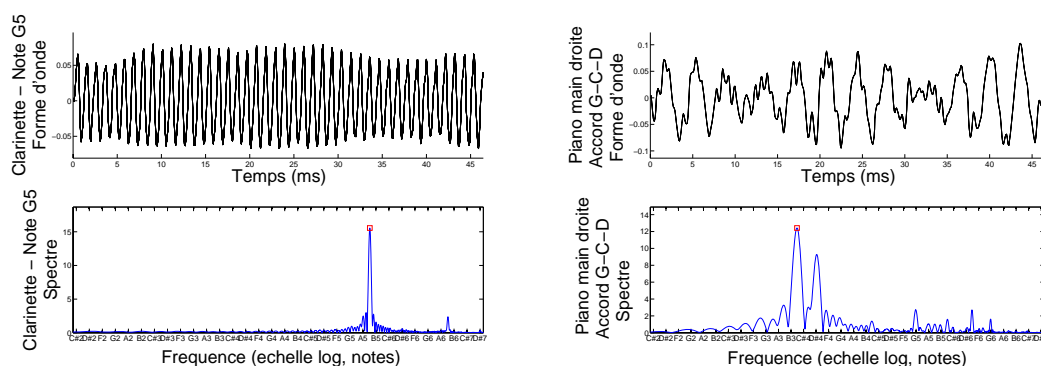


FIG. 10.3 – Un motif appris sur le signal de clarinette (gauche) et sur le signal de piano main droite (droite).

avons choisies. Les notes jouées par les différentes sources sont dans des registres assez séparés – le piano main gauche est en particulier plus grave que les deux autres instruments – comme nous le montre la figure 10.4.

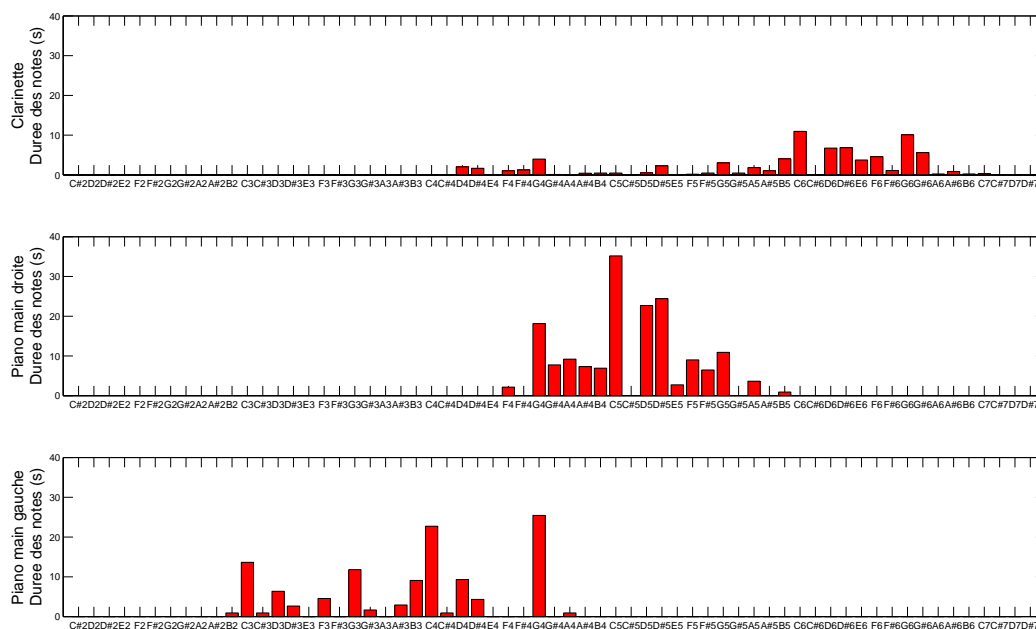


FIG. 10.4 – Histogramme des notes jouées dans les sources d'apprentissage (haut : clarinette, milieu : piano main droite, bas : piano main gauche)

En revanche, les quelques notes de piano communes à la main droite et à la main

gauche sont identiques et les notes du piano main droite sont très proches des harmoniques des notes du piano main gauche. Enfin, la proximité naturelle des instruments (plus grande en tout cas entre une clarinette et un piano qu'entre une clarinette et des percussions) et le son "artificiel" du synthétiseur font que les dictionnaires de clarinette et de piano main droite recouvrent les mêmes zones fréquentielles.

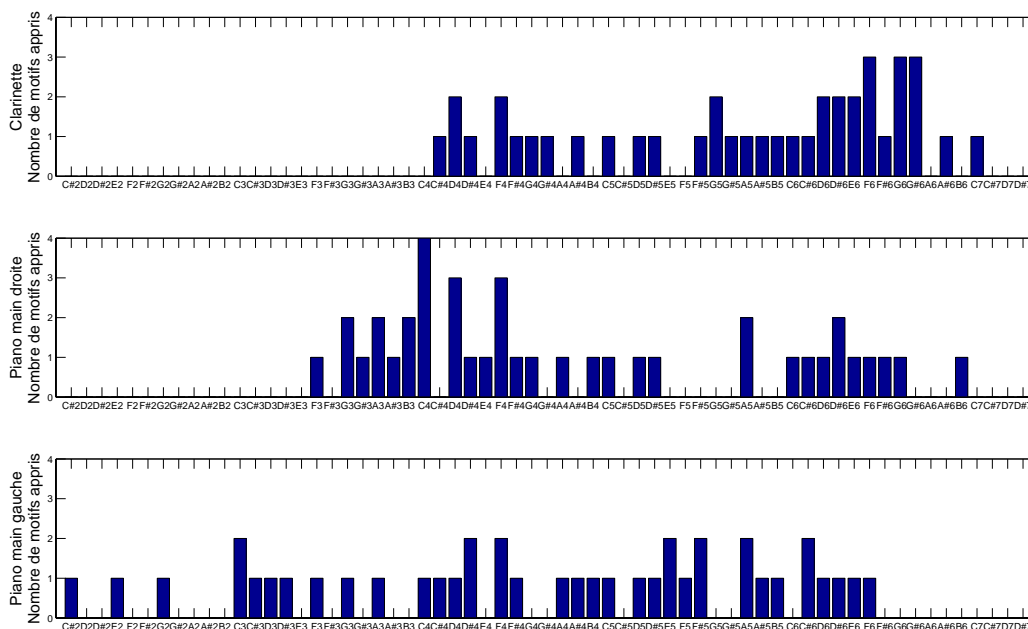


FIG. 10.5 – Histogramme des notes associées aux motifs appris sur les trois sources (haut : clarinette, milieu : piano main droite, bas : piano main gauche)

La figure 10.5 montre les notes associées aux motifs appris sur les sources. Même si l'attribution d'un motif à une note est discutable, puisque nous avons vu que les motifs de piano sont plutôt des accords que des notes isolées, nous remarquons que les instruments se superposent plus que dans les sources d'apprentissage. Cela peut être dû au fait que le nombre de motifs appris est trop important par rapport au nombre de notes réellement présentes (40 motifs pour 15 notes, pour le piano). La redondance de motifs permet alors de coder une plus grande diversité de formes, contribuant à une plus grande proximité entre les dictionnaires.

La proximité des dictionnaires conduit même à des confusions entre sources. Nous pouvons le voir sur la figure 10.6. Il s'agit d'un zoom sur une partie du mélange où la clarinette ne joue pas. En théorie, les motifs du dictionnaire de piano devraient servir à représenter le piano. En réalité, sur cet exemple, une partie du signal de piano a été décomposée avec des atomes formés par les motifs appris sur la clarinette, ce qui a conduit à attribuer cette partie du signal à la clarinette.

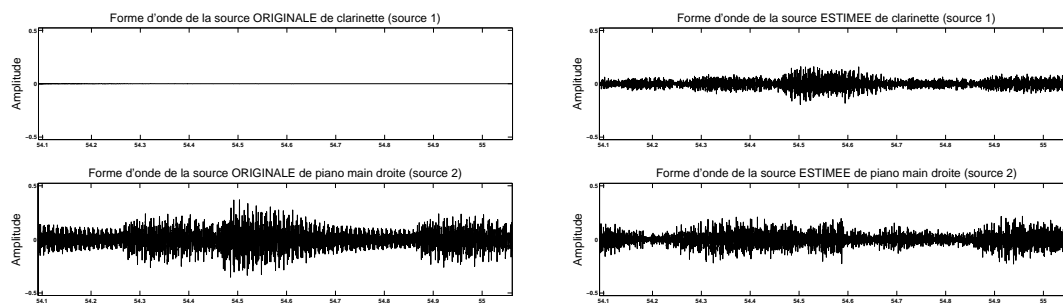


FIG. 10.6 – Confusion lors de l’estimation de la source de piano. Des atomes de clarinette sont choisis pour représenter la source de piano. A gauche sont représentés les formes d’ondes temporelles des sources originales de clarinette (haut) et de piano main droite (bas), et à droite les formes d’ondes des sources estimées.

10.3.5 Conclusion

Le dernier exemple montre que la décision binaire d’attribution de la totalité d’un atome à une seule source, dans l’algorithme DP, peut provoquer des interférences, alors que nous avons vu au contraire, avec DUET, que l’attribution à une seule source peut permettre un faible niveau d’interférences. L’avantage que possède DUET vient en partie d’un effet de lissage dû à l’utilisation de la TFCT. En effet, comme nous l’avons dit auparavant, toutes les cases temps-fréquence sont calculées et une décision, même binaire, sur une case peut être compensée par les cases environnantes. Ce n’est en revanche pas le cas avec les dictionnaires, puisqu’il n’existe pas forcément d’atomes voisins. Nous reviendrons dans la partie expérimentale de la section suivante sur ce problème de choix trop brut.

L’expérience précédente montre quelques limites des dictionnaires appris pour la séparation multicanale. Pour exploiter plus fortement le potentiel de l’apprentissage de dictionnaires, il sera intéressant d’essayer des mélanges de sources plus éloignées morphologiquement (percussions et instrument harmonique, par exemple). Il faudra également envisager d’améliorer la capacité de décomposition des sources sur leur dictionnaire associé. Cela peut être fait par augmentation du nombre de motifs (mais nous avons vu que cela peut augmenter du même coup la proximité entre dictionnaires de sources différentes), ou par utilisation de nouvelles déformations. Nous discuterons notamment dans les perspectives de modèles hybrides, contenant une part de modélisation statistique, pour les atomes.

Enfin, s’il est vrai que pour les mélanges linéaires instantanés multicanaux de signaux sonores, DUET et BZ effectuent une bonne séparation pour un coût calculatoire très faible, de nombreuses autres configurations ne permettent pas de les appliquer tels quels. Nous pouvons tout d’abord imaginer que pour des signaux peu harmoniques, la TFCT s’avère moins appropriée que pour les signaux que nous avons traités. Pour des mélanges convolutifs, DUET et BZ ne s’appliquent pas directement, en revanche le formalisme

proposé à la section 10.2 fournit un algorithme de séparation qu'il convient d'évaluer. Enfin, les algorithmes DUET et BZ étant basés sur la discrimination spatiale, il n'est pas possible de les mettre en œuvre pour des mélanges monocanaux. Nous évaluons justement dans la section suivante l'algorithme DP sur les mélanges monocanaux.

10.4 Séparation monocanale sur des signaux synthétiques

Dans cette section, nous évaluons l'efficacité de la séparation de sources dans le cas monocanal par Demixing Pursuit avec des dictionnaires structurés. Nous étudierons la séparation d'un mélange monocanal de deux sources sonores synthétiques de type MIDI.

Il est nécessaire de disposer d'exemples séparés des sources pour apprendre les dictionnaires nécessaires à la séparation. Le mélange à séparer est quant à lui effectué sur des signaux de la même nature, mais différents des signaux d'apprentissage. Nous déterminerons la taille des signaux d'apprentissage nécessaire à une bonne généralisation, comme nous l'avons fait précédemment à la section 9.6, mais en mesurant cette fois-ci la capacité de généralisation en termes de qualité de la séparation.

Nous comparerons la qualité de la séparation avec celle obtenue avec un algorithme de séparation monocanale performant, élaboré par Benaroya [Ben03] et repris par Ozerov [Oze06]. Cet algorithme est basé sur la modélisation par des mélanges de gaussiennes des densités spectrales de puissance des sources découpées en trames, et sur la séparation via un filtrage de Wiener adaptatif.

Pour comparer cette méthode avec DP, nous pouvons dire approximativement que la taille des trames correspond à la taille des motifs des dictionnaires appris, et que le nombre d'états dans le mélange de gaussiennes correspond au nombre de motifs appris.

10.4.1 Métrique

Nous utiliserons les mesures de RSD, RSI et RSA comme dans la section précédente.

10.4.2 Principe de l'expérience

Deux signaux audio synthétiques de type MIDI sont découpés en deux parties. La première partie est destinée à l'apprentissage des modèles (dictionnaires pour la méthode que nous proposons, paramètres de modèles de mélange de gaussiennes (MMG) pour la méthode de comparaison). La deuxième partie sert à créer le mélange de test.

Pour chaque source, les deux modèles concurrents sont appris pour différentes tailles du signal d'apprentissage, ce qui permettra de mesurer l'importance de cette taille pour la généralisation des modèles. Nous imposons que le dictionnaire et le MMG comptent autant de paramètres afin de permettre une comparaison équitable.

Nous appliquerons les deux algorithmes de séparation sur le mélange, pour chaque taille du signal d'apprentissage. Les performances de séparation sont ensuite évaluées en termes de distorsion, d'interférences et d'artefacts.

10.4.3 Protocole détaillé

Signaux sources	Signal audio MIDI de clarinette (source 1) et de piano (source 2). Concertino op. 26 pour clarinette et orchestre (réduction pour clarinette et piano), Weber. Échantillonné à 16 000 Hz puis sous-échantillonné à 11 025 Hz.
Dictionnaire	$K = 40$ motifs de taille $L_m = 512$, soit 46.4 ms à 11 025 Hz. Nombre de paramètres : 20 480.
Deux modèles de mélange de gaussiennes	Respectivement 32 et 64 états, et les signaux sont découpés en trames recouvrantes de taille 512, soit 46.4 ms à 11 025 Hz. Nombre de paramètres : respectivement 16 384 et 32 768.
Signaux d'apprentissage (pour chaque source)	10 signaux de durée croissant exponentiellement entre 1.86 s et 3 m 06 s (soit 20 480 à 2 048 000 échantillons, ou 1 à 100 fois le nombre de paramètres du dictionnaire).
Sources de test	1 m 33 s, soit 1 024 000 échantillons.
Mélanges	Somme des sources.
Critère d'arrêt de MP, dans K-SVD	lorsque le RSB atteint 20.
Paramètres de l'algorithme K-SVD	30 itérations, et pour chaque itération, une seule mise à jour de chaque motif, sans mise à jour des coefficients ni de l'erreur.
Initialisation des motifs	tirage aléatoire gaussien, puis normalisation

10.4.4 Résultats

Pour chacun des trois modèles appris sur les 10 signaux d'apprentissage (les dictionnaires et les MMG à 32 et 64 états), nous avons effectué la séparation du mélange monocanal. Précisons que le programme utilisé pour les MMG impose un nombre d'états égal à une puissance de 2. C'est pourquoi nous avons entouré le nombre de motifs (40) par les deux valeurs les plus proches (32 et 64).

Nous affichons sur la figure 10.7 les courbes moyennes du RSD, du RSI et du RSA, pour DP et les algorithmes de séparation à base de MMG. L'abscisse correspond aux différentes tailles de signaux d'apprentissage (en nombre d'échantillons). Cette figure permet d'évaluer la capacité de généralisation des algorithmes d'apprentissage, en termes de séparation. La généralisation est bonne si les performances de séparation ne sont pas trop dégradées en passant du mélange d'apprentissage, formé des sources d'apprentissage, au mélange de test.

Nous remarquons tout d'abord que les deux méthodes à base de MMG ont des

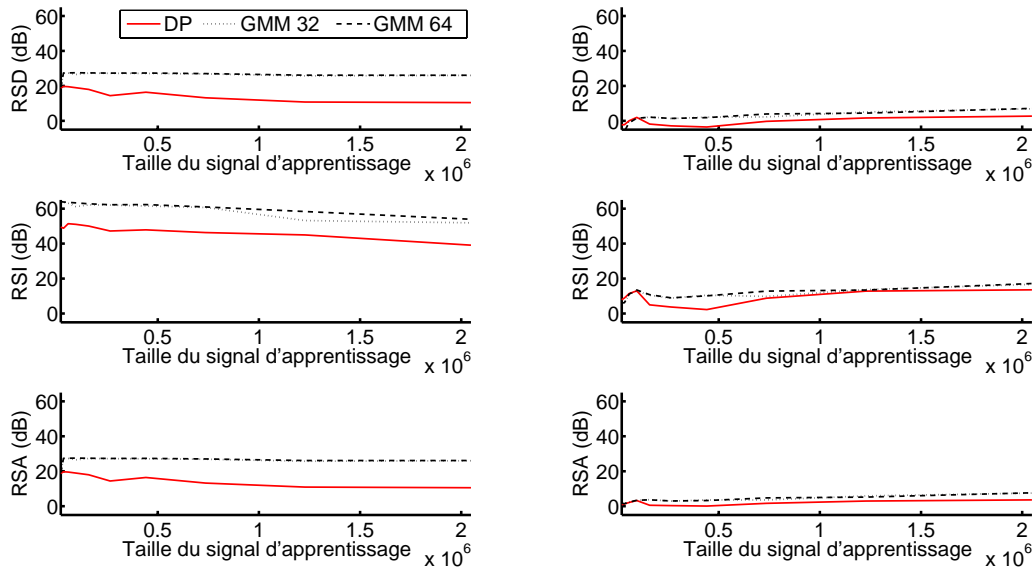


FIG. 10.7 – Performance de la séparation des 10 mélanges d'apprentissage (gauche) et de test (droite), mesurés en termes de RSD, RSI et RSA moyennés sur les deux sources.

courbes quasiment superposées. Nous verrons par la suite, sur un exemple particulier qu'elles peuvent différer localement. Globalement, en revanche, l'utilisation de 32 ou 64 états dans le MMG ne semble pas jouer beaucoup.

Pour les trois algorithmes, l'allure générale des courbes est logique, puisque la qualité de séparation du mélange d'apprentissage diminue lorsque la taille des signaux d'apprentissage augmente, et dans le même temps la qualité de séparation du mélange de test augmente. Cela signifie qu'utiliser des sources d'apprentissage trop restreintes spécialise les modèles appris sur ces données, alors qu'utiliser de longues sources d'apprentissage, plus diverses, augmente les chances d'être adapté à d'autres signaux de même nature (signaux de test).

Globalement, les distorsions sont dominées par les artefacts, puisque le RSA est inférieur au RSI (notez que le RSD est pratiquement égal au minimum des mesures RSI et RSA). Si les deux modèles (dictionnaire et MMG) semblent permettre une bonne séparation, en termes d'interférences, du mélange d'apprentissage, en revanche, ils semblent limités par les artefacts qu'ils produisent (RSA de 26 dB pour MMG, et de 10 dB pour DP, pour le plus grand signal d'apprentissage).

Cela rejoint, pour le modèle avec dictionnaires, l'analyse faite dans la section précédente, où nous supposons qu'augmenter le nombre de motifs et utiliser plus de paramètres libres dans les déformations permettrait de diminuer les artefacts de calcul. Remarquons que le MMG peut être vu, sans entrer dans les détails, comme un modèle de dictionnaire de motifs appris sur le spectre pour lesquels la phase est libre, ce qui

correspond à 512 paramètres libres dans notre cas. Le nombre de paramètres libres supplémentaire explique donc en partie l'écart de 16 dB entre les RSA des modèles avec dictionnaire et avec MMG.

Il est difficile de comparer les performances entre le mélange d'apprentissage et de test, puisqu'un écart d'un décibel entre deux RSD, par exemple, a plus d'importance si le RSD vaut 10 dB que s'il vaut 50 dB. Nous pouvons tout de même remarquer que l'écart entre les méthodes avec MMG et la méthode avec dictionnaires se réduit, pour passer d'environ 15 dB à environ 4 dB.

Pour analyser plus précisément les performances de DP sur le mélange de test, regardons la figure 10.8, où sont représentées les performances source par source, en plus de la moyenne.

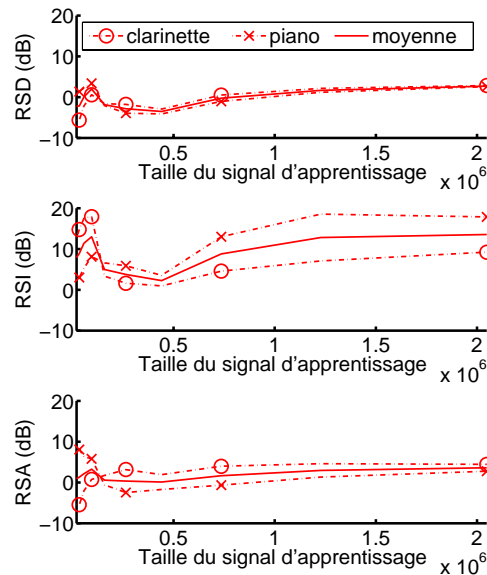


FIG. 10.8 – Performances de séparation du mélange de test par l'algorithme DP utilisant un dictionnaire appris pour chaque source.

Intéressons-nous à la partie droite des courbes, correspondant à des signaux d'apprentissage suffisamment longs pour assurer une généralisation suffisante. Nous voyons que la source de clarinette estimée (rond) contient beaucoup plus d'interférences que la source de piano estimée (croix), ce qui signifie qu'une partie du piano est attribuée à la clarinette. La courbe du RSA est au contraire plus élevée pour la clarinette que pour le piano, ce qui peut s'expliquer, en reprenant la discussion de la section précédente, par le fait que le plan temps-fréquence de la source de clarinette contient moins de "trous" que celui du piano. En effet, plus de la moitié des atomes sont attribués à la clarinette (ceux qui codent la partie clarinette, plus une partie de ceux qui codent la partie piano).

Nous illustrons le phénomène d'utilisation d'atomes de clarinette pour coder la partie

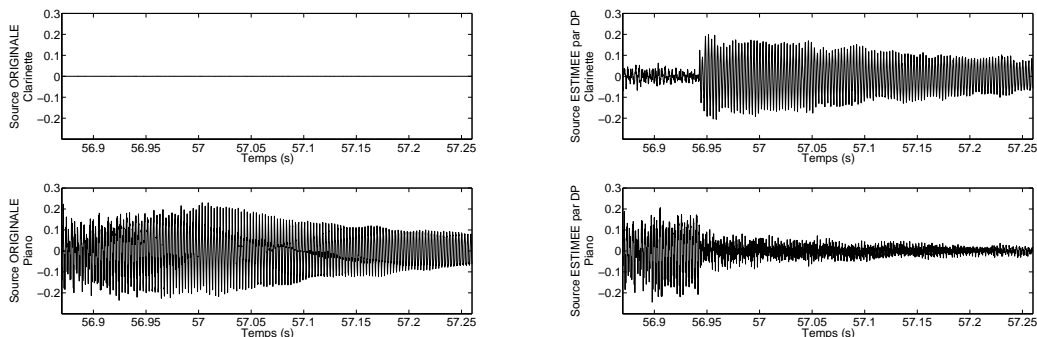


FIG. 10.9 – Formes d’onde des sources originales (gauche) et estimées par DP (droite) de clarinette (haut) et de piano (bas). La clarinette est muette sur cet extrait, et le piano est cependant codé en partie avec des atomes de clarinette.

de piano sur la figure 10.9. Sur cette figure sont représentées à gauche les formes d’onde des sources originales de clarinette (haut) et de piano (bas). Seul le piano joue sur cet extrait. A droite sont représentées les sources estimées avec l’algorithme DP.

Il est frappant de constater que la source estimée la plus énergétique est la source de clarinette. Ainsi, les 80 premières millisecondes, soit environ deux motifs successifs, ont été décomposées en majorité avec des atomes de piano, qui ont été attribués entièrement à la source de piano selon le principe de DP. Les 300 ms suivantes sont presque entièrement décomposées par des atomes de clarinette, qui sont donc attribués exclusivement à la source de clarinette.

Cela signifie qu’il existe des motifs de clarinette plus corrélés avec le signal de piano qu’aucun motif de piano. Une des raisons est que la note jouée, un fa (F5), n’est pas présente dans le dictionnaire de piano (ni celui de clarinette, d’ailleurs) – voir la figure 10.5 (les dictionnaires sont les mêmes). Rappelons de plus que les motifs de piano contiennent des accords plutôt que des notes isolées, contrairement à la clarinette, ce qui a peut-être mené à considérer cette note isolée comme de la clarinette.

Précisément, la note F5 est présente dans la source d’apprentissage de piano – voir figure 10.4 – et elle est présente également dans deux motifs de piano, dans lesquelles elle est cependant minoritaire par rapport à d’autres fréquences. Dans le dictionnaire de clarinette, la note F5 est présente dans deux motifs. Le premier est une sinusoïde correspondant à F4, et la note F5 est donc présente comme harmonique. Le deuxième est un motif contenant deux fréquences correspondant aux notes F5 et G#5, mais la deuxième fréquence est un peu plus marquée.

Ainsi, la clarinette est dotée de motifs permettant de représenter plus précisément la note F5 que les motifs de piano, ce qui explique qu’ils aient été choisis par DP. Mais puisqu’il existe tout de même des motifs de piano qui représentent en partie F5, il serait bon d’en tenir compte.

Une des principales perspectives d’évolution de DP est d’adoucir la décision, en remplaçant le choix binaire d’une source ou d’une autre, pour chaque atome, par un

choix tenant compte de la capacité des motifs de chaque source à représenter le mélange. Nous y reviendrons dans la conclusion.

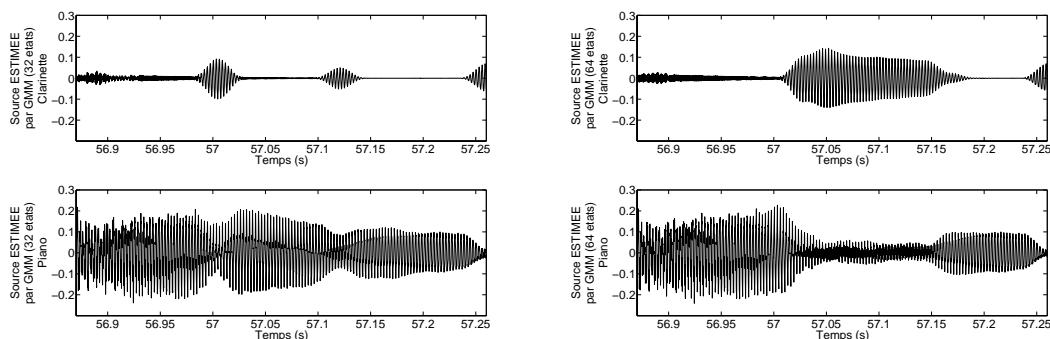


FIG. 10.10 – Formes d’onde des sources estimées avec le modèle de MMG à 32 états (gauche) et à 64 états (droite), pour la clarinette (haut) et le piano (bas). La clarinette est muette sur cet extrait (voir figure 10.9).

Cette perspective s’inspirera notamment du filtrage de Wiener utilisé dans l’algorithme de séparation à base de MMG. La figure 10.10 nous montre que ça ne permet pas de résoudre totalement le problème. En effet, pour le modèle à 32 états, les sources estimées, représentées à gauche, sont assez proches des sources originales (voir figure 10.9), et seuls deux “motifs” sont attribués en partie à la clarinette. En revanche, pour le modèle à 64 états, une longue partie du mélange est attribué presque exclusivement à la source de clarinette. Le filtrage de Wiener permet une décision douce, en partageant chaque trame entre les deux sources. Sur cet exemple, le partage est largement en faveur de la clarinette. Il ne faut donc pas tout attendre de l’adoucissement de la décision.

10.4.5 Conclusion et perspective

L’algorithme Demixing Pursuit utilisant un dictionnaire par source permet d’effectuer la séparation monocanale d’un mélange de deux sources. Les performances sont en deçà de celles de l’algorithme concurrent, basés sur un modèle de MMG et un filtrage de Wiener.

L’utilisation d’un modèle de mélange de gaussiennes permet d’augmenter le nombre de paramètres libres par rapport au modèle de dictionnaire, puisque tous les paramètres de phases sont libres. Parvenir à imiter d’une certaine manière cette liberté par rapport à la phase dans les dictionnaires, en allant plus loin que la simple déformation d’un paramètre de phase, permettra certainement d’améliorer le pouvoir de modélisation des sources et de diminuer les artefacts de calcul.

La différence entre les deux méthodes vient également de la manière d’attribuer les parties du mélange aux deux sources. Pour Demixing Pursuit, la décision est brutale et consiste à attribuer totalement chaque atome à la source correspondant au motif utilisé, tandis que pour l’autre méthode, le filtrage de Wiener attribue une partie de

chaque trame aux deux sources, proportionnellement à la probabilité que la trame ait été générée avec le modèle d'une source ou de l'autre.

A l'écoute, les sources estimées par l'algorithme DP contiennent un léger bruit, de type bruit blanc, qui vient certainement du rapport de phase imposé par le motif temporel. Elles contiennent également du "bruit musical", provenant de la décomposition de type Matching Pursuit qui fait apparaître et disparaître des atomes, produisant parfois des "accros", tandis que les sources estimées par la méthode à base de MMG sont plus lisses et plus épurées. Cette différence vient de l'utilisation de MP d'un côté, et de la TFCT de l'autre.

Une perspective, que nous pourrions appeler Soft Demixing Pursuit, consisterait à adoucir la phase d'attribution des atomes de la décomposition aux différentes sources. Ainsi, au lieu d'ajouter l'atome intégralement à la source correspondant au motif utilisé (par exemple, la première source) :

$$\underline{s}_1 := \underline{s}_1 + F_{\psi,\phi}[\underline{m}_k^1] \quad (10.13)$$

$$\underline{s}_2 := \underline{s}_2, \quad (10.14)$$

nous pouvons imaginer de partager l'atome entre les deux sources :

$$\underline{s}_1 := \underline{s}_1 + \alpha \cdot F_{\psi,\phi}[\underline{m}_k^1] \quad (10.15)$$

$$\underline{s}_2 := \underline{s}_2 + (1 - \alpha) \cdot F_{\psi,\phi}[\underline{m}_k^1], \quad (10.16)$$

où α , compris entre 0.5 et 1 pour privilégier la source associée au motif utilisé, est réglé en fonction de la corrélation entre le motif \underline{m}_k^1 et tous les motifs $\underline{m}_{k'}^2$ du dictionnaire associé à l'autre source. Si un motif du dictionnaire 2 est égal à \underline{m}_k^1 , α est réglé à 0.5, puisque les deux motifs ont autant vocation à représenter le mélange. En revanche, moins le motif \underline{m}_k^1 sera corrélé avec les motifs de l'autre dictionnaire, plus α sera proche de 1.

En augmentant la capacité des dictionnaires à représenter parcimonieusement les signaux sonores, via l'élaboration d'atomes moins rigides, et en adoucissant le processus d'attribution des atomes de la décomposition, l'algorithme Demixing Pursuit semble prometteur, notamment pour traiter des mélanges sous-déterminés complexes, de type mélange convolutif.

Troisième partie

Perspectives et conclusion

Chapitre 11

Problèmes ouverts et perspectives

Au cours du travail présenté dans ce document, de nombreuses pistes ont été explorées, et nous avons buté sur certains écueils qui nous ont obligés à restreindre le champ d'application des solutions trouvées. Nous présentons en section 11.1 le principal problème non résolu, qui concerne la prise en charge du recouvrement des trames lors de l'apprentissage de dictionnaire invariant par translation.

Une deuxième difficulté concerne l'estimation des coefficients lors de l'apprentissage de dictionnaire. Dans la plupart des algorithmes étudiés et proposés, les coefficients et le dictionnaire sont mis à jour alternativement, ce qui diminue la difficulté, mais rend l'algorithme sous-optimal. Une des perspectives de recherche est d'étudier s'il est possible d'optimiser le critère conjointement sur les coefficients et le dictionnaire dans le cas de l'apprentissage d'une base orthonormée ou d'une union de bases orthonormées. Nous en discutons dans la section 11.2.

Deux perspectives, présentées en sections 11.3 et 11.6, concernent des contributions théoriques du travail qui n'ont pas été expérimentées. En premier lieu, nous souhaiterions mesurer à quel point le modèle de filtres excités, exposé à la section 7.5.6, convient pour représenter la parole. La deuxième contribution que nous aimerions analyser expérimentalement concerne la séparation de sources, dans le cas d'un mélange convolutif, où les filtres sont connus. Il s'agit alors de mesurer les performances de la séparation par Matching Pursuit.

Enfin, toujours dans le domaine de la séparation de sources, deux perspectives semblent intéressantes. La première, présentée en section 11.4, consiste à élaborer et évaluer un algorithme de séparation de sources multicanale dans le cas où les conditions de mélange sont connues, et où les dictionnaires sont adaptés aux sources à partir du mélange, et non plus en amont sur des exemples de sources. Cela signifie que la séparation est effectuée sans connaissance *a priori* sur la nature des sources. La deuxième perspective, présentée en section 11.5, consiste à estimer les directions des sources, dans le cas instantané, ou plus largement les conditions du mélange, pendant la séparation à partir d'un algorithme de type K-SVD. Un objectif plus lointain est l'apprentissage conjoint des dictionnaires et des conditions de mélange, sous la seule connaissance du nombre de sources, et du fait qu'elles sont localisées dans l'espace.

11.1 Recouvrement des trames

Nous avons mentionné, au cours de la présentation de l'algorithme K-SVD destiné à l'apprentissage de dictionnaire invariant par translation (section 9.3), que la présence dans la décomposition du signal d'apprentissage d'occurrences d'un même motif qui se recouvrent temporellement est un obstacle pour l'apprentissage du motif en question. Pour la même raison, la déformation des motifs par convolution n'entre pas dans le cadre d'application de l'algorithme K-SVD.

Le problème peut être exprimé sur un exemple simple. Trouver une solution à ce problème donnera certainement de bonnes pistes pour la prise en charge du recouvrement des trames.

L'exemple est le suivant : supposons qu'un signal \underline{s} soit la somme de deux occurrences d'un même motif \underline{m} , espacées de t_1 , et d'amplitudes respectives c_1 et c_2 , auxquelles s'ajoutent un bruit résiduel \underline{r} . Le signal s'écrit :

$$\underline{s} = c_1 \cdot \underline{m} + c_2 \cdot T_{t_1}[\underline{m}] + \underline{r}, \quad (11.1)$$

et un schéma le représente sur la figure 11.1.

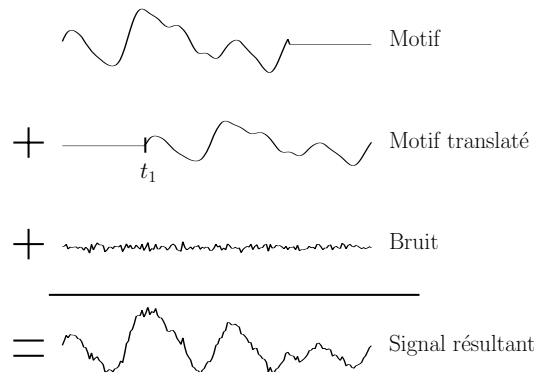


FIG. 11.1 – Construction du signal \underline{s} , comme somme de deux occurrences d'un même motif et d'un bruit.

Connaissant le décalage t_1 , l'objectif est de trouver le motif \underline{m} qui minimise le bruit résiduel :

$$\underline{m}^{\text{opt}} = \arg \min_{\underline{m}, c_1, c_2} \|\underline{s} - c_1 \cdot \underline{m} - c_2 \cdot T_{t_1}[\underline{m}]\|_2^2 \quad (11.2)$$

Nous n'avons pas trouvé de forme analytique simple pour résoudre ce problème. En utilisant les notations de la section 8.2, le vecteur \mathbf{v}_{t_1} et la matrice $\mathbf{G}_{t_1}(\underline{m})$ valent :

$$\mathbf{v}_{t_1} = \begin{bmatrix} \langle \underline{s}, \underline{m} \rangle \\ \langle \underline{s}, T_{t_1}[\underline{m}] \rangle \end{bmatrix} \quad (11.3)$$

et

$$\mathbf{G}_{t_1}(\underline{m}) = \begin{bmatrix} 1 & \langle \underline{m}, T_{t_1}[\underline{m}] \rangle \\ \langle \underline{m}, T_{t_1}[\underline{m}] \rangle & 1 \end{bmatrix}. \quad (11.4)$$

Les coefficients optimaux sont obtenus par :

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \mathbf{G}_{t_1}^{-1}(\underline{m}) \cdot \mathbf{v}_{t_1}, \quad (11.5)$$

et le problème se réécrit :

$$\underline{m}^{\text{opt}} = \arg \max_{\underline{m}} \mathbf{v}_{t_1}^T \cdot \mathbf{G}_{t_1}^{-1}(\underline{m}) \cdot \mathbf{v}_{t_1}, \quad (11.6)$$

soit :

$$\underline{m}^{\text{opt}} = \arg \max_{\underline{m}} \mathbf{m}^T \cdot \left(\mathbf{F}_{t_1}^{*T}[\underline{s}] \cdot \mathbf{G}_{t_1}^{-1}(\underline{m}) \cdot \mathbf{F}_{t_1}^*[\underline{s}] \right) \cdot \mathbf{m}, \quad (11.7)$$

où $\mathbf{F}_{t_1}^{*T}[\underline{s}] = [R_{0,L_m}[\underline{s}] \ R_{t,L_m}[\underline{s}]]$.

Cette dernière formulation montre qu'il est possible de trouver la solution si $\mathbf{G}_{t_1}^{-1}(\underline{m})$ est indépendant de \underline{m} . Dans ce cas, la solution est le vecteur propre associé à la plus grande valeur propre de la matrice $\left(\mathbf{F}_{t_1}^{*T}[\underline{s}] \cdot \mathbf{G}_{t_1}^{-1} \cdot \mathbf{F}_{t_1}^*[\underline{s}] \right)$, comme nous l'avons montré dans la section 9.3.3.

Or, $\mathbf{G}_{t_1}^{-1}(\underline{m})$ dépend du motif \underline{m} :

$$\mathbf{G}_{t_1}^{-1}(\underline{m}) = \frac{1}{1 - \langle \underline{m}, T_{t_1}[\underline{m}] \rangle^2} \cdot \begin{bmatrix} 1 & -\langle \underline{m}, T_{t_1}[\underline{m}] \rangle \\ -\langle \underline{m}, T_{t_1}[\underline{m}] \rangle & 1 \end{bmatrix}. \quad (11.8)$$

Nous ne savons donc pas trouver de solution analytique à ce problème. Étudions maintenant quelles pistes peuvent être explorées, en élargissant le point de vue.

En effet, le même problème se pose notamment lorsque plus de deux atomes se superposent, et dans ce cas, la dimension de la matrice carrée $\mathbf{G}_{t_1}(\underline{m})$ est égale au nombre d'atomes recouvrants. Le problème peut également englober d'autres déformations que les translations, et intervient dès qu'il existe un ensemble d'atomes issus du même motif qui ne sont pas orthogonaux entre eux.

Le problème peut donc s'écrire plus généralement :

$$\mathbf{m}^{\text{opt}} = \arg \max_{\underline{m}} \mathbf{m}^T \cdot \mathbf{A}(\mathbf{m}, \underline{s}) \cdot \mathbf{m}, \quad (11.9)$$

où $\mathbf{A}(\mathbf{m}, \underline{s})$ est une matrice carrée symétrique de taille $L_m \times L_m$ (L_m est la taille du support du motif), dépendant du motif \mathbf{m} et du signal \underline{s} .

Lorsque la matrice ne dépend que du signal et peut s'écrire $\mathbf{A}(\underline{s})$, alors le motif optimal \mathbf{m}^{opt} est un des vecteurs propres associés à la plus grande valeur propre de $\mathbf{A}(\underline{s})$. En revanche, si la matrice $\mathbf{A}(\mathbf{m}, \underline{s})$ dépend de \mathbf{m} , alors nous ne savons pas résoudre le problème analytiquement.

Une première question est de savoir s'il existe une solution au problème, et si oui, si elle est unique. Notons que dans le cas de la déformation de phase, par exemple, la solution optimale est connue à un paramètre de phase près, ce qui signifie que toutes ses versions déformées sont également solutions. Il est donc possible, selon les problèmes, de chercher l'unicité d'une solution, ou l'unicité d'un ensemble de solutions reliées entre elles.

Une deuxième question est de trouver comment tendre vers de telles solutions. Si les solutions ne peuvent pas être calculées de façon exacte, nous pouvons essayer d'élaborer des stratégies algorithmiques pour les approcher avec une précision suffisante. Selon la manière dont $\mathbf{A}(\mathbf{m}, \underline{\mathbf{s}})$ varie avec le motif \mathbf{m} , une méthode peut par exemple consister à trouver une équation de point fixe satisfaite par l'optimum \mathbf{m}^{opt} :

$$\mathbf{m}^{\text{opt}} = f\left(\mathbf{m}^{\text{opt}}\right), \quad (11.10)$$

et à en déduire un algorithme itératif de type point fixe :

1. Initialisation $\mathbf{m}^{(0)}$
2. Boucle sur i

$$\mathbf{m}^{(i+1)} = f\left(\mathbf{m}^{(i)}\right) \quad (11.11)$$

3. Jusqu'à convergence

Si un tel algorithme peut être conçu, il restera encore à étudier les conditions de convergence vers le point fixe et de stabilité autour de ce point, ainsi qu'à comprendre comment initialiser au mieux le motif.

Notons que nous avons essayé la méthode consistant à initialiser le motif par $\mathbf{m}^{(0)}$, puis à résoudre itérativement le problème (11.7) en fixant la matrice $\mathbf{A}(\mathbf{m}^{(i-1)}, \underline{\mathbf{s}})$ avec la valeur précédente du motif $\mathbf{m}^{(i-1)}$ et en calculant $\mathbf{m}^{(i)}$ comme le vecteur propre associé à la plus grande valeur propre de cette matrice. Nous avons vérifié expérimentalement que cette méthode est sous-optimale et dépend largement de l'initialisation.

Nous verrons en section 11.5 que le problème exposé ici se pose également lors de l'apprentissage de motifs multicanaux composés d'une direction spatiale et d'un motif temporel.

11.2 Apprentissage d'une union de bases orthonormées : optimisation conjointe sur les coefficients et les atomes

Dans le cadre de l'apprentissage d'une base orthonormée, présenté au chapitre 6, rappelons que l'objectif s'écrit :

$$(\underline{\mathbf{D}}^{\text{opt}}, \mathbf{C}^{\text{opt}}) = \arg \min_{\underline{\mathbf{D}}, \mathbf{C}} \|\underline{\mathbf{S}} - \mathbf{C} \cdot \underline{\mathbf{D}}\|_2^2 \text{ avec } \underline{\mathbf{D}} \text{ orthonormale}, \quad (11.12)$$

où $\underline{\mathbf{D}}$ est la base orthonormée, $\underline{\mathbf{S}}$ contient les signaux d'apprentissage \mathbf{s}_q et \mathbf{C} les coefficients de la décomposition \mathbf{c}_q .

Nous avons opté pour un algorithme alternant décomposition de chaque signal d'apprentissage \mathbf{s}_q sur la base $\underline{\mathbf{D}}$ courante, et mise à jour de la base $\underline{\mathbf{D}}$ en fixant les coefficients calculés précédemment.

Étant données les propriétés des bases orthonormées, nous nous demandons s'il est possible d'optimiser conjointement les coefficients et la base :

$$(\underline{\mathbf{D}}^{\text{opt}}, \mathbf{C}^{\text{opt}}) = \arg \min_{\underline{\mathbf{D}}, \mathbf{C}} \|\underline{\mathbf{S}} - \mathbf{C} \cdot \underline{\mathbf{D}}\|_2^2 + \lambda \cdot \|\mathbf{C}\|_1 \text{ avec } \underline{\mathbf{D}} \text{ orthonormale .} \quad (11.13)$$

En effet, les coefficients optimaux sont obtenus, à base $\underline{\mathbf{D}}$ fixée, par seuillage doux :

$$\mathbf{C} = \eta_\lambda(\underline{\mathbf{S}} \cdot \underline{\mathbf{D}}^T) , \quad (11.14)$$

et peuvent donc être intégrés dans le critère :

$$\underline{\mathbf{D}}^{\text{opt}} = \arg \min_{\underline{\mathbf{D}}} \|\underline{\mathbf{S}} - \eta_\lambda(\underline{\mathbf{S}} \cdot \underline{\mathbf{D}}^T) \cdot \underline{\mathbf{D}}\|_2^2 + \lambda \cdot \|\eta_\lambda(\underline{\mathbf{S}} \cdot \underline{\mathbf{D}}^T)\|_1 \text{ avec } \underline{\mathbf{D}} \text{ orthonormale .} \quad (11.15)$$

Toute la difficulté réside à trouver dans quelles conditions et avec quel algorithme il est possible de trouver la base $\underline{\mathbf{D}}$ optimale de ce problème, qui ne dépend plus que de la base et des signaux.

Une piste, en vue de construire un algorithme de résolution, consiste à utiliser les multiplicateurs de Lagrange pour imposer l'orthogonalité (ajouter $+\text{Tr}[(\underline{\mathbf{D}} \cdot \underline{\mathbf{D}}^T - \mathbf{I}_L) \cdot \underline{\mathbf{\Lambda}}]$ à la fin du critère dans (11.15) - voir annexe A.4). Nous pensons que le problème résultant n'a pas de solution analytique, en revanche, l'introduction de la contrainte dans le critère peut permettre d'élaborer un algorithme de point fixe à partir de l'équation d'annulation du gradient de ce critère. Il faudra vérifier que la solution optimale vérifie cette équation de point fixe et que le point fixe est unique.

S'il est possible de trouver la meilleure base, alors il sera intéressant d'essayer d'étendre la méthode de résolution au problème de l'apprentissage d'une union de bases orthonormées.

11.3 Intérêt du modèle convolutif pour modéliser la parole

Nous avons exposé à la section 7.5.6 la façon dont nous pensons pouvoir utiliser le modèle d'atomes formés par des motifs filtrés pour modéliser de la parole, le motif \underline{m} représentant le conduit vocal et le filtre \underline{e} représentant l'excitation venant des cordes vocales et du souffle, pour donner des atomes du type :

$$C_e[\underline{m}] = \underline{e} \star \underline{m} \quad (11.16)$$

Or, avant de décomposer la parole sur ce type de dictionnaire de motifs, il faut déjà connaître un dictionnaire de motifs. Nous avons vu qu'il est impossible d'utiliser l'algorithme K-SVD tel quel pour apprendre des dictionnaires invariants par convolution. Pour apprendre un dictionnaire de formes de conduit vocal, il faudra donc trouver la

manière d'adapter K-SVD aux motifs convolués, ou utiliser par exemple des algorithmes de type descente de gradient.

Nous n'avons pas mentionné les tailles respectives des supports des motifs \underline{m} et des filtres d'excitation \underline{e} dans la section 7.5.6. La durée de stationnarité de la parole, environ 30 ms, correspond à la durée pendant laquelle la configuration du conduit vocal est constante. Si le dictionnaire de motifs contient un motif pour chaque configuration du conduit vocal, le même motif peut être utilisé pour coder le signal pendant toute la durée de stationnarité.

La réponse impulsionnelle du conduit vocal, codée par le motif \underline{m} , est beaucoup plus courte que 30 ms, plutôt de l'ordre de quelques millisecondes. Pour générer le signal de parole, il faut donc un signal d'excitation \underline{e} durant environ 30 ms. Trois stratégies peuvent être envisagées. La première consiste à fixer la longueur de \underline{e} à environ 30 ms, ce qui peut s'avérer trop long dans certains cas, et conduire à utiliser trop de paramètres dans le modèle. La deuxième stratégie est de restreindre la taille de \underline{e} à quelques millisecondes, et produire une succession d'atomes provenant du même motif pour générer le signal. Enfin, la troisième solution est de laisser libre la longueur de l'excitation, mais dans ce cas, il faut définir une normalisation par rapport à la taille, pénalisant les longues excitations, afin de pouvoir choisir de façon équitable entre les atomes.

Si ce modèle s'avère utilisable pour la parole, alors il faudra comprendre dans quelle mesure il peut servir pour les principales tâches de traitement de la parole : codage, synthèse, reconnaissance de la parole et reconnaissance du locuteur.

11.4 Apprentissage des dictionnaires adaptés aux sources sur le mélange

Rappelons brièvement le contexte. Nous avons traité le modèle de mélange suivant :

$$\underline{\mathbf{X}} = \sum_{n=1}^N \underline{\mathbf{X}}^n, \quad (11.17)$$

où $\underline{\mathbf{X}}$ est le mélange observé (monocanal ou multicanal), tandis que $\underline{\mathbf{X}}^n$ sont les N composant le mélange. Selon les cas que nous avons détaillés aux sections 10.1 et 10.2, les sources $\underline{\mathbf{X}}^n$ peuvent s'exprimer différemment. Dans certains cas, comme celui de sources étendues, enregistrées en champ proche, les sources peuvent uniquement s'exprimer dans le domaine "mélange" comme $\underline{\mathbf{X}}^n$. Dans d'autres cas, les sources peuvent être considérées comme le filtrage d'une source ponctuelle monocanale :

$$\underline{\mathbf{X}}^n = M^n[\underline{s}_n], \quad (11.18)$$

où la fonction de mélange M^n reflète les conditions de mélange.

Si le mélange est monocanal, la fonction M^n est une simple multiplication par une amplitude dans le cas instantané, une convolution avec un filtre de Dirac dans le cas

d'un mélange avec délai, ou un filtrage plus compliqué, pour modéliser par exemple les différents chemins de propagation de la source jusqu'au capteur.

Si le mélange est multicanal, la fonction M^n transforme un signal monocanal en un signal multicanal, et peut être vue comme un vecteur de fonctions de mélange M_m^n (m est l'indice du canal) du même type que celles exposées dans le paragraphe précédent.

Quelle que soit la configuration du mélange, les hypothèses faites jusqu'à maintenant sont la connaissance des fonctions de mélange M^n , lorsqu'elles existent, et la connaissance d'exemples de sources $\tilde{\mathbf{s}}_n$, ou $\tilde{\mathbf{X}}^n$, similaires aux sources composant le mélange observé \mathbf{X} . Ces exemples de sources sont utilisés pour apprendre des dictionnaires de motifs \mathcal{M}^n servant à la séparation.

Connaître des exemples des sources suffisamment consistants pour que les dictionnaires appris permettent une séparation performante n'est pas toujours possible. Si les seules données disponibles sont le mélange observé et les éventuelles fonctions de mélange, alors il n'est plus possible d'apprendre les dictionnaires préalablement à la séparation.

Nous proposons dans ce cas d'effectuer l'apprentissage des dictionnaires adaptés aux sources directement sur le mélange, et non plus à partir de sources d'entraînement, qui sont parfois manquantes ou insuffisantes.

L'idée de la méthode proposée est la suivante. Il s'agit d'alterner une phase de séparation du mélange à dictionnaires fixés, et une phase d'adaptation de ces dictionnaires, à sources estimées fixées.

Oublions les fonctions de déformation pour simplifier, et exprimons les dictionnaires comme \mathcal{D}^n , sans nous pencher sur le détail des motifs et des fonctions de déformation qui les génèrent. En notant de façon générale \mathbf{S} l'algorithme de séparation, de type Matching Pursuit par exemple, et \mathbf{A} l'algorithme d'apprentissage de motifs, K-SVD par exemple, la méthode que nous aimerions étudier est la suivante :

Séparation de sources avec apprentissage des dictionnaires sur le mélange :

Le nombre de sources est connu. Initialisation des dictionnaires $\mathcal{D}^1, \dots, \mathcal{D}^N$

Faire

1. Séparation des sources à partir du mélange :

$$(\hat{\mathbf{X}}^1, \dots, \hat{\mathbf{X}}^N) = \mathbf{S}(\mathbf{X}, \mathcal{D}^1, \dots, \mathcal{D}^N) \quad (11.19)$$

2. **Pour** chaque source n

Faire l'apprentissage du dictionnaire \mathcal{D}^n sur la source précédemment estimée $\hat{\mathbf{X}}^n$:

$$\mathcal{D}^n = \mathbf{A}(\hat{\mathbf{X}}^n) \quad (11.20)$$

Tant que la convergence n'est pas atteinte

Nous n'avons pas mentionné les fonctions de mélange dans l'exposé de la méthode, et nous avons tout traité sous le même formalisme. Si le mélange est la somme de sources ponctuelles propagées jusqu'aux capteurs selon des fonctions de mélange connues, alors il est probable que l'information contenue dans ces fonctions (information spatiale) sera très utile pour discriminer les sources, notamment pour les premières itérations de l'algorithme où les dictionnaires ne reflètent pas encore la nature des sources.

Apprendre les dictionnaires sur le mélange dans le cas monocanal, par exemple, paraît beaucoup plus difficile, car alors la seule information utile pour la discrimination et disponible *a priori* concerne des contraintes imposées aux dictionnaires. En effet, pour que l'algorithme apprenne réellement des dictionnaires adaptés, il faut que les motifs appris pour un dictionnaire donné aient un certain lien entre eux, qui caractérise justement la source.

Cette nécessité de regroupement, ou d'étiquetage, peut être remplie par une contrainte encourageant la différence entre les dictionnaires, au sens d'une certaine mesure de distance, ou par une contrainte de proximité des motifs au sens d'un même dictionnaire (par exemple, pour la musique, des notes situées dans une tessiture commune). A moins de faire de telles hypothèses sur la nature des dictionnaires, il paraît impossible d'appliquer l'algorithme dans le cas où aucune information spatiale n'est disponible.

L'utilisation de contraintes de distance entre dictionnaires et de proximité au sein d'un même dictionnaire peuvent améliorer l'apprentissage et la séparation dans le cas où une information spatiale est disponible. Il reste cependant à voir quelles formes peuvent prendre de telles contraintes.

11.5 Apprentissage conjoint des conditions de mélange et des dictionnaires adaptés

Pour aller encore un peu plus loin que dans la section précédente, nous pouvons imaginer essayer d'apprendre conjointement les conditions de mélange et les dictionnaires à partir du mélange. Pour cela, nous supposons que le mélange s'écrit :

$$\underline{\mathbf{X}} = \sum_{n=1}^N M_{\mathbf{a}_n}^n[\underline{\mathbf{s}}_n], \quad (11.21)$$

où les sources $\underline{\mathbf{s}}_n$ sont des signaux monocanaux, et les fonctions de mélange $M_{\mathbf{a}_n}^n$ sont paramétrées par le vecteur \mathbf{a}_n (par exemple la direction spatiale, pour le mélange multicanal instantané).

Notre objectif est d'apprendre, pour chaque source n :

- un vecteur de paramètres \mathbf{a}_n
- et K_n motifs $\underline{\mathbf{m}}_k^n$

permettant de former les atomes :

$$M_{\mathbf{a}_n}^n \circ F_{\psi, \phi}^n[\underline{\mathbf{m}}_k^n], \quad (11.22)$$

où ψ et ϕ sont libres.

Pour effectuer l'apprentissage, il faut connaître à l'avance le nombre de sources et également contraindre les paramètres de mélange \mathbf{a}^n à être différents d'une source à l'autre. Il semble que les chances de réussite d'un tel apprentissage résident dans la capacité de la contrainte à éloigner significativement paramètres de mélange et dictionnaires des différentes sources. Une telle méthode semble toutefois difficile à mettre en œuvre.

Une méthode alternative, en deux étapes, est la suivante. Il s'agit d'apprendre dans un premier temps un ensemble de couples motifs-paramètres de mélange $(\underline{m}_k, \mathbf{a}_k)$ sur le mélange, sans contrainte. Cela correspond par exemple, dans le cas d'un mélange multicanal linéaire instantané, à apprendre des couples motif-direction spatiale. La deuxième étape consiste alors à traiter l'ensemble des K couples appris pour regrouper les couples proches par familles de similarité, de taille K_n , en les associant à une source $n : (\underline{m}_k^n, \mathbf{a}_k^n)$. La mesure et la méthode de regroupement restent à définir pour déterminer les dictionnaires et les fonctions de mélange permettant une séparation satisfaisante.

Illustrons la première étape de cette méthode sur l'exemple du mélange multicanal instantané – les fonctions de mélange sont $M_{\mathbf{a}_n}^n = E_{\mathbf{a}_n}$ – où les dictionnaires sont engendrés par un dictionnaire de motifs translétés.

Le dictionnaire initial est un dictionnaire multicanal, formé de K motifs multicanaux $E_{\mathbf{a}_k}[\underline{m}_k]$ qui peuvent être translétés et amplifiés.

En utilisant un algorithme de type K-SVD, les couples $(\underline{m}_k, \mathbf{a}_k)$ peuvent être appris sur le mélange :

Apprentissage d'un dictionnaire multicanal sur le mélange :

Initialisation des K directions \mathbf{a}_k et des K motifs \underline{m}_k correspondants. Motifs et vecteurs de directions sont de norme unitaire.

Faire

1. Décomposition du mélange par Matching Pursuit :

$$\underline{\mathbf{X}} = \sum_{i=1}^I c_i \cdot T_{t_i} \circ E_{\mathbf{a}_{k_i}}[\underline{m}_{k_i}] + \underline{r}_I \quad (11.23)$$

2. **Pour** chaque couple $(\mathbf{a}_k, \underline{m}_k)$

Faire

- (a) Retrait de tous les autres couples :

$$\underline{\check{\mathbf{X}}} = \sum_{i / k_i=k} c_i \cdot T_{t_i} \circ E_{\mathbf{a}_k}[\underline{m}_k] + \underline{r}_I \quad (11.24)$$

- (b) Minimisation de l'erreur par rapport au couple $(\mathbf{a}_k, \underline{m}_k)$ et aux paramètres c_i (voir la suite du texte pour les détails) :

$$(\mathbf{a}_k, \underline{m}_k, \{c_i\}) = \arg \min_{(\mathbf{a}_k, \underline{m}_k, \{c_i\})} \|\underline{\check{\mathbf{X}}} - \sum_{i / k_i=k} c_i \cdot T_{t_i} \circ E_{\mathbf{a}_k}[\underline{m}_k]\|_2^2 \quad (11.25)$$

(c) Mise à jour du résidu :

$$\underline{r}_I = \underline{\check{\mathbf{X}}} - \sum_{i / k_i=k} c_i \cdot T_{t_i} \circ E_{\mathbf{a}_k}[\underline{m}_k] \quad (11.26)$$

Tant que la convergence n'est pas atteinte

Dans cet algorithme d'apprentissage des motifs multicanaux, nous n'avons pas détaillé l'étape (11.25). Elle pose problème car il est impossible de mettre à jour conjointement la direction \mathbf{a}_k et le motif \underline{m}_k , pour des raisons du même ordre que celles exposées à la section 11.1, sans même parler du problème du recouvrement des atomes.

Détaillons le calcul. Nous supposons que les atomes sont de supports disjoints, et les coefficients c_i peuvent alors être calculés directement par :

$$c_i = \langle T_{-t_i}[\underline{\check{\mathbf{X}}}] , E_{\mathbf{a}_k}[\underline{m}_k] \rangle . \quad (11.27)$$

Le problème (11.25) peut donc s'écrire :

$$(\mathbf{a}_k, \underline{m}_k) = \arg \max_{(\mathbf{a}_k, \underline{m}_k)} \sum_{i / k_i=k} \langle T_{-t_i}[\underline{\check{\mathbf{X}}}] , E_{\mathbf{a}_k}[\underline{m}_k] \rangle^2 . \quad (11.28)$$

D'un côté, en écrivant le produit scalaire :

$$\langle T_{-t_i}[\underline{\check{\mathbf{X}}}] , E_{\mathbf{a}_k}[\underline{m}_k] \rangle = \mathbf{u}_i^T \cdot \mathbf{m}_k , \quad (11.29)$$

avec $\mathbf{u}_i = R_{0, L_{m_k}} \circ E_{\mathbf{a}_k}^* \circ T_{-t_i}[\underline{\check{\mathbf{X}}}]$ (L_{m_k} est la taille du support de \underline{m}_k), le problème (11.28) s'écrit :

$$\underline{m}_k = \arg \max_{\underline{m}_k} \mathbf{m}_k^T \cdot \sum_{i / k_i=k} \mathbf{u}_i \cdot \mathbf{u}_i^T \cdot \mathbf{m}_k , \quad (11.30)$$

et le motif optimal est le vecteur propre associé à la plus grande valeur propre de la matrice $\sum_{i / k_i=k} \mathbf{u}_i \cdot \mathbf{u}_i^T$, qui dépend de \mathbf{a}_k .

D'un autre côté, en notant le produit scalaire :

$$\langle T_{-t_i}[\underline{\check{\mathbf{X}}}] , E_{\mathbf{a}_k}[\underline{m}_k] \rangle = \mathbf{v}_i^T \cdot \mathbf{a}_k , \quad (11.31)$$

avec $\mathbf{v}_i^T = [\langle T_{-t_i}[\underline{\check{\mathbf{X}}}_1] , \underline{m}_k \rangle, \dots, \langle T_{-t_i}[\underline{\check{\mathbf{X}}}_M] , \underline{m}_k \rangle]$, le problème (11.28) s'écrit :

$$\mathbf{a}_k = \arg \max_{\mathbf{a}_k} \mathbf{a}_k^T \cdot \sum_{i / k_i=k} \mathbf{v}_i \cdot \mathbf{v}_i^T \cdot \mathbf{a}_k , \quad (11.32)$$

et la direction optimale est le vecteur propre associé à la plus grande valeur propre de la matrice $\sum_{i / k_i=k} \mathbf{v}_i \cdot \mathbf{v}_i^T$, qui dépend de \underline{m}_k .

Le problème est que le motif ne peut être optimisé qu'à direction fixée, et inversement que la direction ne peut être optimisée qu'à motif fixé, mais qu'en aucun cas nous ne

savons optimiser les deux conjointement en utilisant l'algèbre linéaire. Nous ne savons pas si ce problème a une solution analytique. Une solution approchée peut être envisagée en alternant la mise à jour du motif et de la direction, mais la question se pose de savoir sous quelles conditions cet algorithme converge vers les solutions optimales.

Le problème est difficile et peut se formuler également à l'aide de tenseurs. Ainsi, en retirant par souci de simplification les translations, l'équation (11.25) :

$$\min_{(\mathbf{a}, \underline{m})} \sum_i \min_c \|\check{\mathbf{X}} - c \cdot E_{\mathbf{a}}[\underline{m}]\|_2^2 \quad (11.33)$$

s'écrit aussi :

$$\min_{(\mathbf{a}, \mathbf{m}, \mathbf{c})} \sum_{i,j,l} (\tilde{x}_{i,j,l} - c_i \cdot a_l \cdot m_j)^2, \quad (11.34)$$

où i se réfère à l'atome, j à l'échantillon temporel de signal et l au canal. Le vecteur \mathbf{c} contient donc les amplitudes c_i des atomes. Le coefficient a_l du vecteur \mathbf{a} reflète la contribution du motif sur le canal l . Ensuite, m_j est le j^e échantillon de la restriction au support du motif \underline{m} . Enfin, $\tilde{x}_{i,j,l}$ est le j^e échantillon, situé sur le canal l , de la trame du mélange $\check{\mathbf{X}}$ correspondant à l'atome i .

L'équation (11.34) correspond à l'approximation aux moindres carrés d'un tenseur d'ordre 3, de composante $\tilde{x}_{i,j,l}$, **par un tenseur d'ordre 3 et de rang 1**, formé par la multiplication de trois vecteurs \mathbf{a} , \mathbf{m} et \mathbf{c} .

Nous avons mené des discussions avec Myriam Rajih et Pierre Comon, du laboratoire I3S situé à Sophia Antipolis, qui s'intéressent notamment aux factorisations de tenseurs et aux décompositions en valeurs singulières d'ordre supérieur (une généralisation aux tenseurs d'ordre supérieur de la DVS, que nous notons DVSOS). Nous avons également étudié les travaux de Lieven De Lathauwer (voir par exemple [DLV04]).

Sans entrer dans les détails, il ressort de ces échanges que, si dans le cas des matrices, la DVS permet de résoudre exactement le problème des moindres carrés en ne conservant que la plus grande valeur singulière, en revanche, dans le cas des tenseurs d'ordre supérieur, la même technique consistant à ne conserver que la plus grande valeur singulière issue de la DVSOS ne donne pas la solution optimale.

La DVSOS peut en revanche fournir une bonne initialisation pour les vecteurs \mathbf{a} , \mathbf{m} et \mathbf{c} en vue de l'algorithme itératif consistant à fixer à chaque itération un des trois vecteurs, et à mettre à jour les deux autres selon la méthode plusieurs fois exposée dans cette thèse utilisant la DVS.

Nous avons observé expérimentalement que cet algorithme converge généralement vers le minimum global, mais nous avons également trouvé des contre-exemples où l'algorithme ne converge que vers un minimum local.

Cette étude préliminaire nécessite des travaux complémentaires au niveau expérimental pour confirmer les résultats et au niveau théorique pour analyser les conditions de convergence de l'algorithme.

11.6 Séparation de sources par Matching Pursuit sur un mélange convolutif

Nous avons expliqué dans le chapitre 10 comment l'algorithme Matching Pursuit peut être appliqué pour la séparation de sources. Pour cela, il faut connaître les conditions du mélange (matrice de mélange dans le cas instantané, matrice de filtres dans le cas convolutif). Nous avons effectué une étude expérimentale concernant la séparation de mélanges multicanaux instantanés. Malheureusement, nous n'avons pas pu faire de même pour les mélanges convolutif car le temps a manqué.

La bibliothèque MPTK utilise notamment un algorithme de recherche rapide de l'atome le plus corrélé avec le signal. Le gain en complexité lié à l'utilisation de cet algorithme, ainsi que les autres accélérations précédemment détaillées ont justifié l'intérêt de s'impliquer dans le projet MPTK. Chronologiquement, nous avons d'abord eu l'idée de séparation de sources par Demixing Pursuit dans le cas multicanal instantané. Nous avons donc consacré les efforts nécessaires à l'implémentation de cet algorithme et à sa validation.

Les idées concernant la séparation de mélanges multicanaux convolutifs sont venues ensuite. Il reste à investir le temps pour coder l'algorithme de séparation afin de valider cette approche. Il faut noter que l'implémentation des deux perspectives exposées dans les sections précédentes nécessite également un temps non négligeable, que nous n'avons pas pris pour des raisons d'organisation du travail de recherche et qui explique qu'elles sont pour l'heure encore à l'état de perspectives.

Considérant les conclusions encourageantes de séparation dans le cas instantané, nous sommes impatients de voir quelle sera la qualité des sources estimées dans le cas convolutif. Il serait intéressant de combiner cet algorithme à des méthodes d'estimation des filtres convolutifs, afin d'évaluer notamment le rapport entre la qualité de la séparation et la rapidité du système total.

Chapitre 12

Perspectives pour d'autres applications

Au chapitre précédent, nous avons présenté les perspectives théoriques et expérimentales du travail de thèse. Nous exposons ici les perspectives au niveau applicatif. La section 12.1 est consacrée aux liens unissant notre travail au domaine du codage. Nous montrons dans la section 12.2 comment les dictionnaires structurés peuvent être utilisés pour le débruitage. Dans la section 12.3, nous exposons une méthode simple de classification basée sur les décompositions parcimonieuses, dont la pertinence reste à évaluer. Enfin, la section 12.4 présente le “remix”, qui peut être avantageusement effectué en utilisant les algorithmes de type Matching Pursuit.

12.1 Parcimonie et codage

Le codage est une application souvent mentionnée lorsqu'il est question de décompositions parcimonieuses. Cependant, il est généralement utopique de penser obtenir, à débit équivalent, une compression de meilleure qualité en transmettant les atomes de la décomposition qu'en utilisant les codeurs classiques (MP3, Ogg-Vorbis, par exemple pour l'audio). Nous expliquerons dans la deuxième partie de cette section les limites du modèle qui nous rendent prudents sur l'utilisation de dictionnaires appris pour le codage.

Avant cela, nous donnons dans la première partie de la section quelques considérations techniques sur le codage par décomposition parcimonieuse.

12.1.1 Considérations techniques sur le codage

Pour transmettre un signal \underline{s} en utilisant sa décomposition parcimonieuse sur un dictionnaire :

$$\underline{s} = \sum_{i=1}^I F_{\psi_{i,\phi_i}}[\underline{m}_{k_i}] + \underline{r}_I, \quad (12.1)$$

il suffit de transmettre le dictionnaire de motifs, puis les atomes de la décomposition, et pour du codage sans perte, le résidu.

Ce principe est du type codage par transformée, qui consiste à représenter un signal par les coefficients d'une transformation, à appliquer éventuellement une phase de réduction de dimension de ces coefficients puis à coder les coefficients. La compression peut être avec ou sans perte.

Dans notre cas, le coût de codage du dictionnaire est fixe, et comprend le poids de codage de tous les motifs, ainsi que la description des fonctions de déformation.

Ensuite, pour coder chaque atome, il faut coder l'indice k_i du motif dans le dictionnaire et les paramètres de déformation ψ_i et ϕ_i .

Enfin, pour reconstruire le signal de façon exacte, il faut transmettre le résidu \underline{r}_I qui est de même longueur que le signal \underline{g} , mais qui peut être de poids plus faible à pas de quantification égal si l'erreur est d'énergie faible. Pour du codage avec perte, le résidu peut être ne pas être transmis tel quel.

Le coût de codage dépend linéairement du nombre d'atomes, en supposant que le coût de description d'un atome est constant d'un atome à l'autre. Évidemment, à dictionnaire fixé, plus la qualité de codage exigée est élevée, plus il faut d'atomes dans la décomposition et plus le coût de codage est élevé.

Nous avons vu que plus le dictionnaire comporte de motifs, plus il peut coder un signal avec parcimonie. Dans le cas où le dictionnaire de motifs est connu à l'encodage et au décodage, alors seuls les paramètres (indice du motif et paramètres de déformation) des atomes sont à transmettre, et il semblerait intéressant d'utiliser un grand nombre de motifs. Il faut cependant remarquer que la taille du dictionnaire de motifs influe sur le coût de description de l'indice du motif. Grossièrement, pour coder l'indice du motif pour un dictionnaire de 8 motifs, il faut 3 bits, alors que pour un dictionnaire de 1 024 motifs, il en faut 10 ($\log_2(K)$, avec K le nombre de motifs). Ainsi, le poids étant multiplié par le nombre d'atomes, un sur-dimensionnement du dictionnaire de motifs peut coûter cher et un compromis est à trouver entre la taille du dictionnaire et le nombre d'atomes nécessaires pour atteindre une qualité donnée.

De même, les fonctions de déformations doivent être choisies à l'aune du rapport entre le gain de parcimonie apporté et le poids des paramètres supplémentaires.

Une étude plus poussée permettant d'évaluer la qualité du codage nécessiterait l'implémentation d'un codeur, et notamment du codeur entropique qui joue un rôle très important dans la qualité du codeur. Pour des raisons de priorités, de domaine de compétences et de temps, nous ne nous sommes pas lancés dans cette voie. Nous pouvons citer quelques travaux récents [LDT04, RD06] témoignant de l'intérêt pour les approches parcimonieuses dans un but de codage. Notez que ces travaux ne traitent pas de la question de l'apprentissage des dictionnaires et les dictionnaires utilisés sont analytiques.

Le codeur serait un outil intéressant pour évaluer la pertinence des choix de taille de dictionnaire, pour un algorithme d'apprentissage donné, ou pour comparer plusieurs techniques d'apprentissage, ou de décomposition parcimonieuse.

12.1.2 Limites de l'approche par apprentissage de dictionnaire pour le codage

Le codage par transformée est largement utilisé (par exemple dans les standards de type JPEG pour les images, MPEG pour les vidéos et l'audio). Pour qu'il soit efficace, il faut que le modèle de représentation des signaux soit bien adapté à la nature des signaux traités.

L'approche par décompositions parcimonieuses sur des dictionnaires appris présente certaines limites qui peuvent s'avérer critiques pour le codage. Expliquons pourquoi, en prenant l'exemple des signaux sonores.

Nous pouvons tout d'abord constater que le codage utilisant des dictionnaires structurés analytiques, par exemple le dictionnaire de Gabor, est très efficace. Sans entrer dans le détail, un dictionnaire voisin, non redondant, est utilisé dans le codage MP3, via la TDCM. La compression est principalement obtenue grâce à des techniques de masquage temps-fréquence basées sur des notions de perception auditive.

La TDCM a l'avantage de permettre des calculs rapides, et ne nécessite pas de transmission du dictionnaire, puisqu'il est décrit analytiquement. La description analytique a évidemment un poids mais ne nécessite pas de transmission de données brutes comme dans le cas des dictionnaires appris.

La TDCM est de plus très adaptée aux signaux audio, et plus particulièrement aux parties tonales du signal, qui peuvent être décomposées de façon parcimonieuse par la TDCM (de nombreux atomes sont de faible amplitude).

Face à ce dictionnaire analytique, l'intérêt des dictionnaires appris varie selon les circonstances. Si les signaux étudiés sont intrinsèquement parcimonieux, nous avons vu à la section 9.4, sur des données artificielles, qu'un algorithme d'apprentissage permet de trouver un dictionnaire satisfaisant cette grande parcimonie. Dans ce cas, l'utilisation de dictionnaires appris peut être envisagée pour le codage. Pour que les dictionnaires appris soient concurrentiels par rapport à la TDCM, il faut que la transmission du dictionnaire de motifs puisse être négligée, et que le temps de calcul n'entre pas en ligne de compte.

En revanche, les signaux sonores réels ne sont pas décomposables de façon très parcimonieuse en utilisant un dictionnaire de motifs lorsqu'ils recèlent une partie bruitée pour laquelle le modèle de décomposition atomique est mal adapté. Cette partie correspond aux sons comportant une part de génération aléatoire, non exactement reproductible. Dans nos expériences, nous avons utilisé des signaux MIDI qui sont synthétisés à partir de peu d'éléments et les variations entre les différentes occurrences d'une même note sont faibles. En revanche, les enregistrements de voix chuchotée, de flûte et de cymbales sont quelques exemples de signaux sonores comportant une part de signal non reproductible. Pour pouvoir coder ces derniers signaux de façon efficace, il faudrait construire des modèles de déformations plus fins que ceux que nous avons présenté dans ce travail.

Revenons sur les exemples de signaux sonores. Les deux premiers signaux peuvent être considérés schématiquement comme un bruit blanc filtré par le conduit vocal dans le premier cas, par le corps de la flûte dans le deuxième cas. Pour les cymbales, c'est le contraire. L'excitation, correspondant au coup de baguette, est nette et plus ou moins

reproductible. En revanche, la propagation de cette excitation génère des vibrations très bruitées.

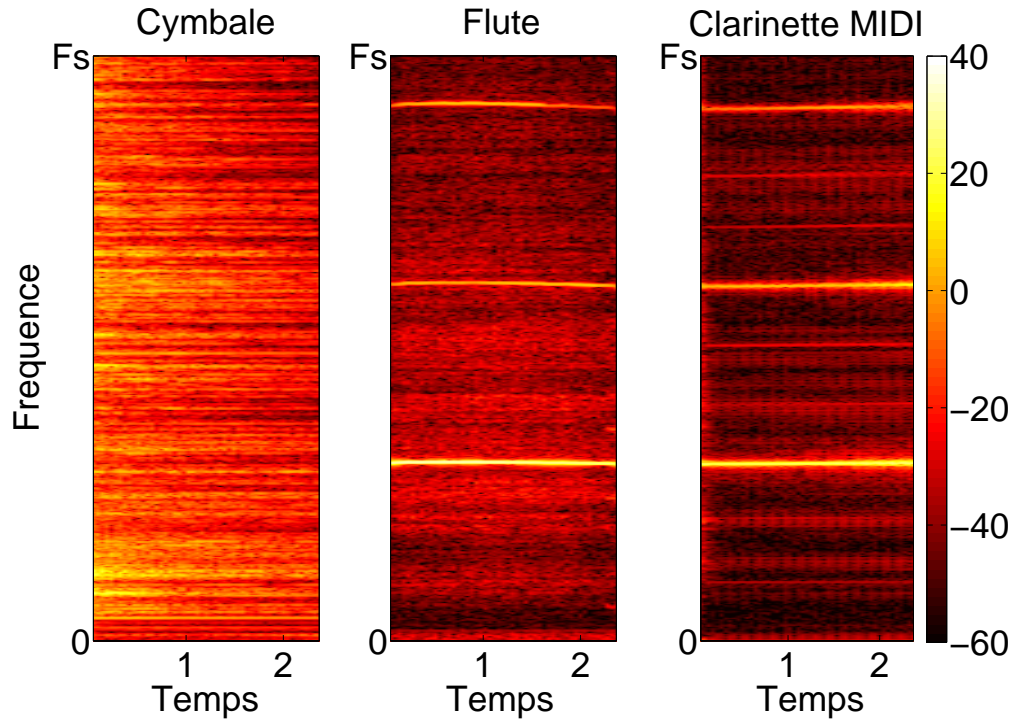


FIG. 12.1 – Spectrogramme d'une note de cymbale (gauche), de flûte (milieu) et de clarinette synthétique (droite). Les notes de cymbale et de flûte (instruments réels) contiennent beaucoup plus de bruit que la note synthétique de clarinette

Afin d'illustrer cette partie bruitée et aléatoire du signal, nous représentons sur la figure 12.1 les spectrogrammes de deux enregistrements réels : cymbale (gauche) et flûte (milieu), et le spectrogramme d'un signal synthétisé à partir d'un fichier MIDI : clarinette (droite). Nous avons pris soin d'utiliser la même note pour comparer la flûte et la clarinette (pour la cymbale, aucune fréquence ne se dégage).

Pour la cymbale, l'enveloppe temporelle de l'amplitude, pas très visible sur le spectrogramme, est une exponentielle décroissante. En revanche, pour le signal modulé par cette enveloppe, aucune structure harmonique, ni aucune autre structure fréquentielle ne ressort visuellement.

Pour la flûte et la clarinette, une fréquence fondamentale et deux harmoniques apparaissent très clairement. Les deux spectrogrammes se ressemblent mais contiennent toutefois une différence, dans la partie moins énergétique comprise entre les fréquences dominantes, qui nous intéresse et qui concerne la partie aléatoire que nous avons mentionné avant.

Dans le cas de la clarinette générée synthétiquement, les coefficients temps-fréquence ont une structure très périodique dans le temps, et le signal semble être formé d'une

concaténation de plusieurs variations d'un même motif. Cette période est de l'ordre de 150 ms. Nous ne connaissons pas le détail des algorithmes utilisés par le synthétiseur MIDI, mais nous savons que les notes sont générées en déformant cinq formes d'ondes brutes de clarinette enregistrée à différentes hauteurs, et une parmi elles dure 150 ms. Nous pouvons supposer que cette forme d'onde a été transposée à la bonne fréquence et répétée avec quelques variations. Évidemment, ce modèle rappelle fortement le modèle de décomposition sur un dictionnaire de motifs.

Dans le cas de la flûte, les coefficients de faible amplitude ne présentent pas du tout de structure périodique comme pour la clarinette, mais semblent plutôt répartis aléatoirement. Intuitivement, cette partie du signal sera plus difficile à représenter à l'aide d'un dictionnaire de motifs que la partie correspondante de la clarinette.

Cet exemple visuel mériterait d'être analysé plus finement pour vérifier l'analyse que nous venons d'en faire. Rappelons notamment que la différence visible entre la flûte et la clarinette et la flûte se situe au niveau des coefficients de faible amplitude. Il faudrait regarder si c'est également le cas pour les coefficients de forte amplitude (les trois raies horizontales).

Quoi qu'il en soit, l'exemple a permis d'illustrer le problème de modélisation des signaux réels à l'aide de motifs. Pour le codage, la description de la partie bruitée du signal réel sera trop consommatrice d'atomes et rend l'approche par apprentissage de dictionnaire non concurrentielle dans le cas général.

Une perspective pour améliorer la description des signaux réel vient du modèle de motifs convolués, dont nous avons déjà signalé qu'il peut être intéressant pour coder la parole. Le modèle convolutif est $\underline{e} \star \underline{m}$, où \underline{e} représente l'excitation (souffle, cordes vocales) dans le cas de la flûte ou de la parole, par exemple, et le motif \underline{m} représente le filtre qui modifie cette excitation (le corps de la flûte, le conduit vocal). Dans le cas de la cymbale, nous pouvons imaginer le contraire, à savoir un dictionnaire de motifs correspondant aux coups de baguette, et un filtre \underline{e} représentant le filtre de propagation de cette excitation.

En fusionnant le modèle de décomposition atomique avec un modèle statistique de type auto-régressif ou LPC (linear predictive coding) [RS78], par exemple, sur les coefficients libres \underline{e} du modèle $\underline{e} \star \underline{m}$, nous pouvons espérer représenter avec plus de parcimonie les parties bruitées du signal. Notons que cette approche ouvre également des perspectives d'utilisation des dictionnaires appris pour la séparation de signaux réels.

12.2 Débruitage par seuillage

Le débruitage de signaux est une application classique des décompositions parcimonieuses qui consiste à essayer de supprimer le bruit (souvent supposé gaussien) qui recouvre des données. Donoho et Johnstone [DJ94] ont proposé en 1994 une méthode de débruitage basée sur le seuillage des coefficients d'ondelettes. Elle consiste à annuler les coefficients d'ondelettes de poids inférieur à un certain seuil, et à diminuer le poids des autres coefficients, selon la règle de seuillage doux présentée à la section 2.3.1. La

difficulté consiste à trouver le seuil optimal.

Engan et Skretting [ES02] ont montré que cette technique peut s'adapter aux dictionnaires redondants appris sur les données. La méthode consiste alors à effectuer une décomposition parcimonieuse du signal bruité, puis à appliquer un seuillage doux. Cette méthode nécessite de régler le degré de parcimonie en plus du seuil.

Elad et Aharon ont amélioré l'approche dans des travaux récents [EA06]. La technique pourrait être étendue aux dictionnaires structurés, et la mesure de la qualité du débruitage, qui est en général la mesure du rapport signal à bruit, permettrait alors de donner un indicateur rapide de la qualité de l'adaptation d'un dictionnaire à un signal.

12.3 Classification

La classification est une tâche qui consiste, à partir d'un ensemble de classes de signaux prédéfinies, à déterminer à laquelle de ces classes rattacher un nouveau signal. Dans le domaine de la musique, chaque classe peut par exemple représenter un instrument, et la classification consiste alors à reconnaître l'instrument joué dans le nouveau signal.

Pour cette tâche, nous proposons une méthode simple à base de dictionnaires adaptés. En supposant disposer de nombreux signaux d'entraînement pour chaque classe, la première étape consiste à apprendre, pour chaque classe, un dictionnaire adapté aux signaux d'entraînement, par exemple par K-SVD. La deuxième étape, visant à reconnaître le nouveau signal, consiste à décomposer successivement ce signal de façon parcimonieuse sur chacun des dictionnaires. Le signal est alors attribué à la classe correspondant à la décomposition la plus parcimonieuse.

Ce principe simple reste toutefois à approfondir, et peut-être à améliorer en favorisant la distance entre les dictionnaires lors de l'apprentissage, afin d'augmenter le pouvoir de discrimination entre les classes.

12.4 Remix

Le "remix" est une tâche de post-production d'un document audio dont le but est de réaliser un mixage différent de celui précédemment effectué à partir de bandes originales. Ces bandes originales n'étant plus disponibles, la première étape du remix est une séparation du document audio en sources correspondant aux bandes originales. La deuxième étape est un mixage permettant de placer les sources dans l'espace sonore selon les effets désirés. Le remix permet notamment de transformer de vieux enregistrements stéréophoniques en son spatialisé 5.1 (6 enceintes). Le principe du remix est expliqué dans [VFG⁺03a] et [WPD06].

Les algorithmes de séparation destinés au remix ne visent pas forcément à obtenir une grande qualité sonore des sources séparées, car le mixage ultérieur des sources estimées permet de masquer en partie les erreurs d'estimation.

Un indice de la qualité d'une séparation de sources peut être donné par la qualité du remix. Le protocole suivant peut être envisagé. Deux mélanges différents $\underline{\mathbf{X}}_1$ et $\underline{\mathbf{X}}_2$ sont

effectués à partir des mêmes sources. L'algorithme de séparation effectue une estimation des sources à partir de la connaissance des conditions du premier mixage, puis les sources estimées sont mélangées suivant les conditions du deuxième mixage pour donner $\hat{\mathbf{X}}_2$. La qualité de la séparation est alors mesurée à l'aune du poids de l'erreur $\|\mathbf{X}_2 - \hat{\mathbf{X}}_2\|_2^2$.

Conclusion

Nous avons présenté dans cette thèse un nouveau modèle de dictionnaire structuré, dans le cadre des décompositions parcimonieuses de signaux sur un dictionnaire d'atomes. Un dictionnaire structuré est composé d'un dictionnaire de motifs et d'un ensemble de fonctions linéaires permettant de déformer ces motifs pour engendrer des atomes. Cette structure permet de refléter la nature des signaux analysés, et notamment les signaux temporels, grâce à la fonction de translation. Les motifs représentent les formes d'ondes les plus typiques des signaux traités, et les fonctions de déformation permettent de les tordre, de les tourner, de les déplacer...

Par rapport aux dictionnaires d'ondelettes ou de Gabor, qui utilisent quelques-unes de ces fonctions de déformation, l'apport de cette thèse est l'apprentissage des motifs à partir des données, ce qui permet d'obtenir des motifs plus adaptés au signal traité que les fonctions analytiques de type fréquence pure ou ondelette-mère.

En se plaçant d'un autre point de vue, cette thèse a permis, par rapport aux méthodes d'apprentissage de dictionnaires non-structurés, de factoriser le dictionnaire appris pour réduire sa redondance et d'influencer l'apprentissage avec des connaissances sur la nature des signaux traités et sur ses invariances.

Le cadre des dictionnaires structurés réunit diverses catégories d'atomes précédemment séparées. Ainsi, sous le cadre de motifs déformés se retrouvent les atomes non-contraints, les atomes translattés définis par Olshausen [Ols00], les atomes multicanaux destinés à la séparation de sources ou bien encore les atomes convolués par un filtre temporel.

Le passage à l'échelle des algorithmes de décomposition (Matching Pursuit) et d'apprentissage de dictionnaire (K-SVD) était une des motivations de la thèse, et sa réalisation permet de pouvoir traiter des signaux sonores de plusieurs minutes et d'envisager de traiter des vidéos. Nous avons également adapté ces algorithmes au formalisme général de dictionnaire structuré, ce qui est nouveau, pour MP, notamment pour les atomes convolués, et pour les atomes à déformation de phase.

Nous avons également introduit l'utilisation de plusieurs dictionnaires adaptés aux différentes sources pour la séparation de sources, dans les cas monocanal et multicanal. Les résultats encourageants sur la capacité des différents dictionnaires à séparer les sources selon leur morphologie ont également montré les limites du modèle de dictionnaire structuré dans son formalisme actuel.

En particulier, pour des raisons pratiques, le dictionnaire ne peut englober qu'un nombre restreint de motifs qui en conséquence ne peuvent représenter parcimonieuse-

ment qu'une classe restreinte de signaux.

Le modèle de dictionnaire structurés permet de faire en partie sauter ce verrou grâce à l'ajout de fonctions de déformation. Pour des fonctions bien choisies, l'allocation de paramètres libres pour décrire les déformations se révèle plus efficace que de les allouer pour des motifs supplémentaires ou plus grands.

Le manque de paramètres libres se fait pourtant sentir, notamment pour traiter des signaux réels contenant une part aléatoire difficile à coder par une superposition de formes d'ondes fixées, même lorsqu'elles sont déformées. Il se fait également sentir pour les applications de séparation de sources, où l'algorithme de séparation à base de dictionnaires structurés se révèle moins bon que l'algorithme de séparation basé sur la modélisation des densités spectrales de puissance du signal par des mélanges de gaussiennes et pour lequel tous les paramètres de phase sont libres. Rappelons que la différence de performance vient également d'une allocation moins brutale des parties du mélange aux sources.

Pour augmenter l'intérêt des dictionnaires appris pour la décomposition parcimonieuse de signaux réels, une perspective semble incontournable. Il s'agit de marier le modèle de dictionnaire structuré avec une modélisation statistique de certains des paramètres de déformation. Cette idée, déjà mentionnée pour représenter la parole, se base sur le modèle de motifs convolués. Le modèle hybride pourrait alors reposer sur un modèle statistique de l'excitation et sur un dictionnaire de motifs modélisant les filtres.

Un des défis de cette approche hybride est d'élaborer un modèle élargissant les capacités de représentation des signaux réels en mariant le cadre théorique des décompositions parcimonieuses et la modélisation probabiliste.

Annexe A

Annexes

A.1 Minimisation de l'erreur quadratique - Preuve du résultat (1.14)

Nous prouvons l'égalité $\arg \min_{\mathbf{c}} \|\underline{s} - \mathbf{c}^T \cdot \underline{\mathbf{D}}\|_2^2 = (\underline{s} \cdot \underline{\mathbf{D}}^+)^T$ présentée dans la formule (1.14). Rappelons que le nombre d'atomes K du dictionnaire $\underline{\mathbf{D}}$ est plus petit que la taille des signaux L .

Le problème de minimisation est convexe, et la solution globale correspond annule le gradient :

$$\nabla_{\mathbf{c}} \|\underline{s} - \mathbf{c}^T \cdot \underline{\mathbf{D}}\|_2^2 = -2 \cdot \underline{\mathbf{D}} \cdot (\underline{s} - \mathbf{c}^T \cdot \underline{\mathbf{D}})^T \quad (\text{A.1})$$

soit

$$\underline{\mathbf{D}} \cdot \underline{s}^T = \underline{\mathbf{D}} \cdot \underline{\mathbf{D}}^T \cdot \mathbf{c}^{\text{opt}} \quad (\text{A.2})$$

ou

$$\mathbf{c}^{\text{opt}} = (\underline{\mathbf{D}} \cdot \underline{\mathbf{D}}^T)^{-1} \cdot \underline{\mathbf{D}} \cdot \underline{s}^T = (\underline{s} \cdot \underline{\mathbf{D}}^+)^T, \quad (\text{A.3})$$

avec $\underline{\mathbf{D}}^+ = \underline{\mathbf{D}}^T \cdot (\underline{\mathbf{D}} \cdot \underline{\mathbf{D}}^T)^{-1}$.

A.2 Calcul du gradient de la vraisemblance par rapport au dictionnaire (équation (3.32))

Le gradient de la vraisemblance par rapport au dictionnaire $\nabla_{\underline{\mathbf{D}}} \mathcal{L}$ s'écrit :

$$\nabla_{\underline{\mathbf{D}}} \mathcal{L} = \frac{\partial E\{\log p(\underline{\mathbf{s}}|\underline{\mathbf{D}})\}}{\partial \underline{\mathbf{D}}} \quad (\text{A.4})$$

$$= E\left\{\frac{1}{p(\underline{\mathbf{s}}|\underline{\mathbf{D}})} \cdot \frac{\partial p(\underline{\mathbf{s}}|\underline{\mathbf{D}})}{\partial \underline{\mathbf{D}}}\right\} \quad (\text{A.5})$$

$$= E\left\{\frac{1}{p(\underline{\mathbf{s}}|\underline{\mathbf{D}})} \cdot \frac{\partial}{\partial \underline{\mathbf{D}}} \int p(\underline{\mathbf{s}}|\underline{\mathbf{D}}, \mathbf{c}) \cdot p(\mathbf{c}|\underline{\mathbf{D}}) d\mathbf{c}\right\} \quad (\text{A.6})$$

$$= E\left\{\frac{1}{p(\underline{\mathbf{s}}|\underline{\mathbf{D}})} \cdot \int \frac{\partial p(\underline{\mathbf{s}}|\underline{\mathbf{D}}, \mathbf{c})}{\partial \underline{\mathbf{D}}} \cdot p(\mathbf{c}|\underline{\mathbf{D}}) d\mathbf{c}\right\} \quad (\text{A.7})$$

$$= E\left\{\frac{1}{p(\underline{\mathbf{s}}|\underline{\mathbf{D}})} \cdot \int \frac{\partial \log p(\underline{\mathbf{s}}|\underline{\mathbf{D}}, \mathbf{c})}{\partial \underline{\mathbf{D}}} \cdot p(\underline{\mathbf{s}}|\underline{\mathbf{D}}, \mathbf{c}) \cdot p(\mathbf{c}|\underline{\mathbf{D}}) d\mathbf{c}\right\} \quad (\text{A.8})$$

$$= E\left\{\int \frac{\partial \log p(\underline{\mathbf{s}}|\underline{\mathbf{D}}, \mathbf{c})}{\partial \underline{\mathbf{D}}} \cdot p(\mathbf{c}|\underline{\mathbf{s}}, \underline{\mathbf{D}}) d\mathbf{c}\right\} \quad (\text{A.9})$$

$$= E\left\{E\left\{\frac{\partial \log p(\underline{\mathbf{s}}|\underline{\mathbf{D}}, \mathbf{c})}{\partial \underline{\mathbf{D}}}\right\}_{p(\mathbf{c}|\underline{\mathbf{s}}, \underline{\mathbf{D}})}\right\} \quad (\text{A.10})$$

L'équation (A.5) est obtenue par dérivation du logarithme. L'équation (A.6) vient de la marginalisation de la probabilité $p(\underline{\mathbf{s}}|\underline{\mathbf{D}})$ par rapport aux coefficients \mathbf{c} . L'équation (A.7) est obtenue par dérivation du produit, et par l'annulation du terme $\frac{\partial p(\mathbf{c}|\underline{\mathbf{D}})}{\partial \underline{\mathbf{D}}}$ car les coefficients sont supposés, *a priori*, indépendants du dictionnaire. L'équation (A.8) découle de l'écriture $p(\underline{\mathbf{s}}|\underline{\mathbf{D}}, \mathbf{c}) = \exp(\log p(\underline{\mathbf{s}}|\underline{\mathbf{D}}, \mathbf{c}))$. L'équation (A.9) vient de la loi de Bayes :

$$p(\underline{\mathbf{s}}|\underline{\mathbf{D}}, \mathbf{c}) \cdot p(\mathbf{c}|\underline{\mathbf{D}}) = p(\mathbf{c}|\underline{\mathbf{D}}, \underline{\mathbf{s}}) \cdot p(\underline{\mathbf{s}}|\underline{\mathbf{D}}) . \quad (\text{A.11})$$

Enfin, l'équation (A.10) est une simple convention de notation.

A.3 Solution optimale du problème (3.41)

Nous prouvons que la solution $\underline{\mathbf{d}}^{\text{opt}}$ du problème $\arg \max_{\underline{\mathbf{d}}} \underline{\mathbf{d}} \cdot (\underline{\mathbf{S}}^T \cdot \underline{\mathbf{S}}) \cdot \underline{\mathbf{d}}^T$, avec $\|\underline{\mathbf{d}}\|_2^2 = 1$, est un vecteur propre associé à la plus grande valeur propre de la matrice $\underline{\mathbf{S}}^T \cdot \underline{\mathbf{S}}$.

Pour cela, écrivons la matrice comme :

$$\underline{\mathbf{S}}^T \cdot \underline{\mathbf{S}} = \sum_{i=1}^I \lambda_i \cdot \underline{\mathbf{S}}_i^T \cdot \underline{\mathbf{S}}_i , \quad (\text{A.12})$$

où $0 \leq \lambda_I < \lambda_{I-1} < \dots < \lambda_1$ sont les I valeurs propres distinctes de la matrice, et où les matrices $\underline{\mathbf{S}}_i$ sont des bases des sous-espaces \mathcal{S}_i de vecteurs propres associés à chacune

des valeurs propres λ_i . La dimension de chaque sous-espace est égal à la multiplicité de la valeur propre associée, les sous-espaces sont distincts entre eux et leur union engendre l'espace des signaux $\mathbb{R}^{1 \times L}$.

Nous notons $P_{\mathcal{S}_i} \underline{d}$ la projection orthogonale de \underline{d} sur le sous-espace \mathcal{S}_i des vecteurs propres associés à la valeur propre λ_i . L'atome peut se décomposer comme :

$$\underline{d} = \sum_{i=1}^I P_{\mathcal{S}_i} \underline{d} . \quad (\text{A.13})$$

Notons que la projection peut s'écrire matriciellement comme :

$$P_{\mathcal{S}_i} \underline{d} = \underline{d} \cdot \underline{\mathbf{S}}_i^T \cdot \underline{\mathbf{S}}_i . \quad (\text{A.14})$$

A l'aide de ces éléments, la grandeur $\underline{d} \cdot (\underline{\mathbf{S}}^T \cdot \underline{\mathbf{S}}) \cdot \underline{d}^T$ vaut donc :

$$\underline{d} \cdot (\underline{\mathbf{S}}^T \cdot \underline{\mathbf{S}}) \cdot \underline{d}^T = \sum_{i=1}^I \lambda_i \cdot \underline{d} \cdot \underline{\mathbf{S}}_i^T \cdot \underline{\mathbf{S}}_i \cdot \underline{d}^T \quad (\text{A.15})$$

$$= \sum_{i=1}^I \lambda_i \cdot \langle P_{\mathcal{S}_i} \underline{d}, \underline{d} \rangle \quad (\text{A.16})$$

$$= \sum_{i=1}^I \lambda_i \cdot \|P_{\mathcal{S}_i} \underline{d}\|_2^2 . \quad (\text{A.17})$$

Or, par propriété de conservation de la norme :

$$\sum_{i=1}^I \|P_{\mathcal{S}_i} \underline{d}\|_2^2 = \|\underline{d}\|_2^2 = 1 , \quad (\text{A.18})$$

et évidemment les normes sont positives. La somme (A.17) est donc un barycentre des valeurs propres λ_i et sa valeur maximale est obtenue lorsque le poids affecté à la plus grande valeur propre λ_1 est égal à 1, et lorsque les autres poids sont égaux à 0.

Nous avons donc $P_{\mathcal{S}_1} \underline{d}^{\text{opt}} = 1$ et $P_{\mathcal{S}_i} \underline{d}^{\text{opt}} = 0$, pour tout $i > 1$, soit :

$$\underline{d}^{\text{opt}} = P_{\mathcal{S}_1} \underline{d}^{\text{opt}} , \quad (\text{A.19})$$

ce qui signifie que $\underline{d}^{\text{opt}}$ appartient à \mathcal{S}_1 et donc qu'il est un vecteur propre de norme 1 associé à la valeur propre λ_1 , soit la plus grande valeur propre de la matrice $\underline{\mathbf{S}}^T \cdot \underline{\mathbf{S}}$.

A.4 Apprentissage d'une base orthonormale - Preuve du résultat de 6.1.1

Nous donnons le détail du calcul du dictionnaire optimal, dans le cas de l'apprentissage d'une base orthonormale, qui est traité dans la section 6.1.1.

Rappelons que le critère à minimiser par rapport au dictionnaire $\underline{\mathbf{D}}$ est :

$$\mathcal{C}(\underline{\mathbf{D}}, \underline{\mathbf{A}}) = \|\underline{\mathbf{S}} - \underline{\mathbf{C}} \cdot \underline{\mathbf{D}}\|_2^2 + \text{Tr} [(\underline{\mathbf{D}} \cdot \underline{\mathbf{D}}^T - \underline{\mathbf{I}}_L) \cdot \underline{\mathbf{A}}] . \quad (\text{A.20})$$

A.4.1 Annulation des gradients par rapport au dictionnaire et aux multiplicateurs de Lagrange

Le dictionnaire optimal $\underline{\mathbf{D}}^{\text{opt}}$ vérifie nécessairement que les gradients $\nabla_{\underline{\mathbf{A}}} \mathcal{C}(\underline{\mathbf{D}}, \underline{\mathbf{A}})$ et $\nabla_{\underline{\mathbf{D}}} \mathcal{C}(\underline{\mathbf{D}}, \underline{\mathbf{A}})$ sont égaux à zéro :

$$\nabla_{\underline{\mathbf{A}}} \mathcal{C}(\underline{\mathbf{D}}^{\text{opt}}, \underline{\mathbf{A}}) = \underline{\mathbf{D}}^{\text{opt}} \cdot \underline{\mathbf{D}}^{\text{opt}T} - \mathbf{I}_L = 0 \quad (\text{A.21})$$

$$\nabla_{\underline{\mathbf{D}}} \mathcal{C}(\underline{\mathbf{D}}^{\text{opt}}, \underline{\mathbf{A}}) = -2 \cdot \mathbf{C}^T \cdot (\underline{\mathbf{S}} - \mathbf{C} \cdot \underline{\mathbf{D}}^{\text{opt}}) + (\underline{\mathbf{A}} + \underline{\mathbf{A}}^T) \cdot \underline{\mathbf{D}}^{\text{opt}} = 0 \quad (\text{A.22})$$

ou :

$$\underline{\mathbf{D}}^{\text{opt}} \cdot \underline{\mathbf{D}}^{\text{opt}T} = \mathbf{I}_L \quad (\text{A.23})$$

$$\mathbf{C}^T \cdot \underline{\mathbf{S}} = \left(\frac{\underline{\mathbf{A}} + \underline{\mathbf{A}}^T}{2} + \mathbf{C}^T \cdot \mathbf{C} \right) \cdot \underline{\mathbf{D}}^{\text{opt}} \quad (\text{A.24})$$

En notant $\underline{\mathbf{Z}} = \mathbf{C}^T \cdot \underline{\mathbf{S}}$ et $\mathbf{Y} = \left(\frac{\underline{\mathbf{A}} + \underline{\mathbf{A}}^T}{2} + \mathbf{C}^T \cdot \mathbf{C} \right)$, l'équation (A.24) s'écrit $\underline{\mathbf{Z}} = \mathbf{Y} \cdot \underline{\mathbf{D}}^{\text{opt}}$. Alors que $\underline{\mathbf{Z}}$ peut être calculé, car les signaux et les coefficients sont connus, \mathbf{Y} est inconnu et dépend des multiplicateurs de Lagrange.

L'équation (A.24) permet d'écrire :

$$\underline{\mathbf{Z}}^T \cdot \underline{\mathbf{Z}} = \underline{\mathbf{D}}^{\text{opt}T} \cdot \mathbf{Y}^2 \cdot \underline{\mathbf{D}}^{\text{opt}}, \quad (\text{A.25})$$

car la matrice \mathbf{Y} est symétrique. De plus, l'équation (A.23) permet d'écrire :

$$\underline{\mathbf{Z}} \cdot \underline{\mathbf{Z}}^T = \mathbf{Y} \cdot \underline{\mathbf{D}}^{\text{opt}} \cdot \underline{\mathbf{D}}^{\text{opt}T} \cdot \mathbf{Y}^T = \mathbf{Y}^2. \quad (\text{A.26})$$

En combinant ces deux résultats, le dictionnaire optimal doit vérifier :

$$\underline{\mathbf{Z}}^T \cdot \underline{\mathbf{Z}} = \underline{\mathbf{D}}^{\text{opt}T} \cdot \underline{\mathbf{Z}} \cdot \underline{\mathbf{Z}}^T \cdot \underline{\mathbf{D}}^{\text{opt}}. \quad (\text{A.27})$$

A.4.2 Introduction de la décomposition en valeurs singulières

Nous notons $\underline{\mathbf{Z}} = \mathbf{U}^T \cdot \underline{\mathbf{\Delta}} \cdot \mathbf{V}$ la décomposition en valeurs singulières (DVS) de $\underline{\mathbf{Z}}$. Il faut remarquer que si $\underline{\mathbf{Z}}$ n'est pas inversible, cette décomposition n'est pas unique, et entre deux DVS différentes de $\underline{\mathbf{Z}}$, les matrices \mathbf{U} et \mathbf{V} sont différentes, mais la matrice $\underline{\mathbf{\Delta}}$ est la même. Les matrices \mathbf{U} et \mathbf{V} sont orthonormales, et $\underline{\mathbf{\Delta}}$ est diagonale. Nous supposons que les éléments diagonaux de $\underline{\mathbf{\Delta}}$ (les valeurs singulières) sont ordonnés de façon décroissante. En notant L le nombre de valeurs singulières différentes et N_l l'ordre de multiplicité de la l^e valeur singulière λ_l , la matrice $\underline{\mathbf{\Delta}}$ s'écrit :

$$\underline{\mathbf{\Delta}} = \begin{bmatrix} \lambda_1 \cdot \mathbf{I}_{N_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_L \cdot \mathbf{I}_{N_L} \end{bmatrix} \quad (\text{A.28})$$

avec $\lambda_1 > \lambda_2 > \dots > \lambda_L \geq 0$.

L'équation (A.27) s'écrit alors :

$$\underline{V}^T \cdot \underline{\Delta}^2 \cdot \underline{V} = \underline{D}^{\text{opt}^T} \cdot \underline{U}^T \cdot \underline{\Delta}^2 \cdot \underline{U} \cdot \underline{D}^{\text{opt}} , \quad (\text{A.29})$$

ou, en multipliant à gauche par \underline{V} et à droite par $\underline{D}^{\text{opt}^T} \cdot \underline{U}^T$:

$$\underline{\Delta}^2 \cdot \underline{V} \cdot \underline{D}^{\text{opt}^T} \cdot \underline{U}^T = \underline{V} \cdot \underline{D}^{\text{opt}^T} \cdot \underline{U}^T \cdot \underline{\Delta}^2 . \quad (\text{A.30})$$

Nous notons $\underline{R}^{\text{opt}} = \underline{V} \cdot \underline{D}^{\text{opt}^T} \cdot \underline{U}^T$ et l'équation précédente s'écrit :

$$\underline{\Delta}^2 \cdot \underline{R}^{\text{opt}} = \underline{R}^{\text{opt}} \cdot \underline{\Delta}^2 . \quad (\text{A.31})$$

A.4.3 Transformation en un problème équivalent

Nous pouvons écrire le dictionnaire optimal comme

$$\underline{D}^{\text{opt}} = \underline{U}^T \cdot \underline{R}^{\text{opt}^T} \cdot \underline{V} , \quad (\text{A.32})$$

et pour trouver $\underline{D}^{\text{opt}}$, il suffit de trouver la matrice équivalente $\underline{R}^{\text{opt}}$. Nous allons montrer que $\underline{R}^{\text{opt}} = \underline{I}_L$ et $\underline{D}^{\text{opt}} = \underline{U}^T \cdot \underline{V}$ sont les solutions optimales.

A.4.3.1 La matrice équivalente est bloc-diagonale

Notons $\underline{R} = [\underline{R}_{i,j}]_{i,j}$ avec $\underline{R}_{i,j}$ une sous-matrice de \underline{R} de taille $N_i \times N_j$. Nous pouvons alors écrire :

$$\underline{R} \cdot \underline{\Delta}^2 = [\lambda_j^2 \cdot \underline{R}_{i,j}]_{i,j} \quad (\text{A.33})$$

$$\underline{\Delta}^2 \cdot \underline{R} = [\lambda_i^2 \cdot \underline{R}_{i,j}]_{i,j} . \quad (\text{A.34})$$

L'équation (A.31) donne l'égalité entre les deux termes précédents, et donc, pour tout $i \neq j$, $\lambda_i^2 \cdot \underline{R}_{i,j}^{\text{opt}} = \lambda_j^2 \cdot \underline{R}_{i,j}^{\text{opt}}$. Or, dans ce cas, $\lambda_i \neq \lambda_j$, et donc $\underline{R}_{i,j}^{\text{opt}} = \mathbf{0}$. La matrice $\underline{R}^{\text{opt}}$ est donc bloc-diagonale :

$$\underline{R}^{\text{opt}} = \begin{bmatrix} \underline{R}_{1,1}^{\text{opt}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \underline{R}_{L,L}^{\text{opt}} \end{bmatrix} . \quad (\text{A.35})$$

Comme la matrice $\underline{R}^{\text{opt}}$ est le produit de matrices orthonormales, elle est elle-même orthonormale. Comme elle est bloc-diagonale, chaque sous-matrice $\underline{R}_{i,l}^{\text{opt}}$ est orthonormale.

A.4.3.2 La matrice identité est la matrice équivalente qui minimise le critère

Le critère $\mathcal{C}(\underline{\mathbf{D}}^{\text{opt}}, \mathbf{\Lambda})$ s'écrit :

$$\mathcal{C}(\underline{\mathbf{D}}^{\text{opt}}, \mathbf{\Lambda}) = \|\underline{\mathbf{S}} - \mathbf{C} \cdot \underline{\mathbf{D}}^{\text{opt}}\|_2^2, \quad (\text{A.36})$$

car $\underline{\mathbf{D}}^{\text{opt}}$ est orthonormée. En notant que $\|\underline{\mathbf{S}} - \mathbf{C} \cdot \underline{\mathbf{D}}\|_2^2 = \|\underline{\mathbf{S}}\|_2^2 + \|\mathbf{C}\|_2^2 - 2 \cdot \text{Tr} [\underline{\mathbf{S}}^T \cdot \mathbf{C} \cdot \underline{\mathbf{D}}]$, minimiser $\mathcal{C}(\underline{\mathbf{D}}^{\text{opt}}, \mathbf{\Lambda})$ revient à maximiser $\text{Tr} [\underline{\mathbf{S}}^T \cdot \mathbf{C} \cdot \underline{\mathbf{D}}^{\text{opt}}]$.

Nous pouvons réécrire :

$$\text{Tr} [\underline{\mathbf{S}}^T \cdot \mathbf{C} \cdot \underline{\mathbf{D}}^{\text{opt}}] = \text{Tr} [\underline{\mathbf{Z}}^T \cdot \underline{\mathbf{D}}^{\text{opt}}] \quad (\text{A.37})$$

$$= \text{Tr} [(\underline{\mathbf{V}}^T \cdot \underline{\mathbf{\Delta}}^T \cdot \mathbf{U}) \cdot (\mathbf{U}^T \cdot \mathbf{R}^{\text{opt}T} \cdot \underline{\mathbf{V}})] \quad (\text{A.38})$$

$$= \text{Tr} [\underline{\mathbf{V}}^T \cdot \underline{\mathbf{\Delta}}^T \cdot \mathbf{R}^{\text{opt}T} \cdot \underline{\mathbf{V}}] \quad (\text{A.39})$$

$$= \text{Tr} [\underline{\mathbf{\Delta}}^T \cdot \mathbf{R}^{\text{opt}T}] \quad (\text{A.40})$$

$$= \text{Tr} [\mathbf{R}^{\text{opt}} \cdot \underline{\mathbf{\Delta}}] \quad (\text{A.41})$$

$$= \sum_{l=1}^L \lambda_l \cdot \text{Tr} [\mathbf{R}_{l,l}^{\text{opt}}]. \quad (\text{A.42})$$

Il s'agit de maximiser chacun des termes de la somme, et donc, pour tout l tel que $\lambda_l > 0$, il faut maximiser $\text{Tr} [\mathbf{R}_{l,l}^{\text{opt}}]$.

Comme tous les éléments diagonaux d'une matrice orthonormale sont inférieurs ou égaux à 1, la trace maximale est $\text{Tr} [\mathbf{R}_{l,l}^{\text{opt}}] = N_l$. Elle est atteinte uniquement si $\mathbf{R}_{l,l}^{\text{opt}} = \mathbf{I}_{N_l}$.

Si la dernière valeur singulière λ_L est strictement positive, alors $\mathbf{R}^{\text{opt}} = \mathbf{I}_L$ est la seule solution optimale.

Si $\lambda_L = 0$, alors $\mathbf{R}_{L,L}^{\text{opt}}$ peut être n'importe quelle matrice orthonormale de taille $N_L \times N_L$. La matrice $\mathbf{R}^{\text{opt}} = \mathbf{I}_L$ est alors une solution optimale mais pas la seule.

A.5 Condition d'identifiabilité des paramètres lors de la composition de N fonctions de déformation

Il s'agit de montrer que la proposition $P(N)$ est vraie pour tout N :

$$P(N) : \text{pour } I_n \in \mathbb{N} \setminus \{0\}, \text{ avec } 1 \leq I_N \leq I_{N-1} \leq \dots \leq I_1,$$

$$\prod_{n=1}^N I_n - \sum_{n=1}^N I_n + (N-1) \geq 0$$

avec égalité si et seulement si $I_N = \dots = I_2 = 1$

Nous le montrons par un raisonnement par récurrence. La propriété est vraie au rang $N = 2$. En effet,

$$I_1 \cdot I_2 - I_1 - I_2 + 1 = (I_1 - 1) \cdot (I_2 - 1) .$$

Cette expression est supérieure ou égale à 0 car $1 \leq I_2 \leq I_1$, et elle est égale à 0 si et seulement si $I_2 = 1$ ou $I_1 = 1$, ce dernier cas impliquant $I_2 = 1$.

Supposons maintenant la proposition $P(N)$ vraie au rang N et montrons que $P(N+1)$ est vraie. Rappelons que $1 \leq I_{N+1} \leq I_N \leq \dots \leq I_1$. L'expression

$$\prod_{n=1}^{N+1} I_n - \sum_{n=1}^{N+1} I_n + N$$

peut s'écrire

$$I_{N+1} \left(\prod_{n=1}^N I_n - 1 \right) - \sum_{n=1}^N I_n + N \geq \prod_{n=1}^N I_n - 1 - \sum_{n=1}^N I_n + N \quad (\text{A.43})$$

$$\geq 0 . \quad (\text{A.44})$$

L'inégalité (A.43) est obtenue par le fait que $\prod_{n=1}^N I_n - 1 \geq 0$, car tous les I_n sont supérieurs à 1, et par le fait que $I_{N+1} \geq 1$. L'inégalité se transforme en égalité si et seulement si $I_{N+1} = 1$ ou $\prod_{n=1}^N I_n = 1$, *i.e.* $I_1 = \dots = I_N = 1$, ce qui implique $I_{N+1} = 1$.

L'inégalité (A.44) vient du fait que la proposition $P(N)$ est vraie. L'égalité est obtenue si et seulement si $I_N = \dots = I_2 = 1$.

Nous avons donc prouvé que :

$$\prod_{n=1}^{N+1} I_n - \sum_{n=1}^{N+1} I_n + N \geq 0$$

et que l'égalité est atteinte si et seulement si $I_{N+1} = I_N = \dots = I_2 = 1$. $P(N+1)$ est donc vraie.

A.6 Calcul du paramètre ψ optimal dans Matching Pursuit

La norme de l'erreur $\|r_{i-1} - F\psi_{\phi}[\underline{m}_k]\|_2^2$, que nous cherchons à minimiser dans le problème (8.3), peut s'écrire :

$$\|r_{i-1}\|_2^2 - 2 \cdot \psi^T \cdot \mathbf{v}_{k,\phi} + \psi^T \cdot \mathbf{G}_{k,\phi} \cdot \psi , \quad (\text{A.45})$$

avec $\mathbf{v}_{k,\phi}$ et $\mathbf{G}_{k,\phi}$ respectivement le vecteur d'intercorrélation et la matrice d'autocorrélation définis dans la section 8.2.

Le paramètre ψ^{opt} minimisant l'erreur est obtenu par annulation du gradient suivant :

$$\nabla_{\psi} \|r_{i-1} - F\psi_{\phi}[\underline{m}_k]\|_2^2 = -2 \cdot \mathbf{v}_{k,\phi} + 2 \cdot \mathbf{G}_{k,\phi} \cdot \psi , \quad (\text{A.46})$$

ce qui équivaut à :

$$\boldsymbol{\psi}^{\text{opt}} = \mathbf{G}_{k,\phi}^{-1} \cdot \mathbf{v}_{k,\phi} . \quad (\text{A.47})$$

La norme de l'erreur s'écrit alors :

$$\|\underline{\mathcal{L}}_{i-1}\|_2^2 - 2 \cdot \mathbf{v}_{k,\phi}^T \cdot \mathbf{G}_{k,\phi}^{-1} \cdot \mathbf{v}_{k,\phi} + \mathbf{v}_{k,\phi}^T \cdot \mathbf{G}_{k,\phi}^{-1} \cdot \mathbf{G}_{k,\phi} \cdot \mathbf{G}_{k,\phi}^{-1} \cdot \mathbf{v}_{k,\phi} \quad (\text{A.48})$$

soit

$$\|\underline{\mathcal{L}}_{i-1}\|_2^2 - \mathbf{v}_{k,\phi}^T \cdot \mathbf{G}_{k,\phi}^{-1} \cdot \mathbf{v}_{k,\phi} , \quad (\text{A.49})$$

et la minimiser par rapport au motif \underline{m}_k et aux paramètres ϕ revient à maximiser

$$\mathbf{v}_{k,\phi}^T \cdot \mathbf{G}_{k,\phi}^{-1} \cdot \mathbf{v}_{k,\phi} . \quad (\text{A.50})$$

A.7 Un vecteur et la partie imaginaire de sa transformée de Hilbert sont orthogonaux

Nous prouvons que tout vecteur $\mathbf{m} \in \mathbb{R}^{L_m \times 1}$ est orthogonal à la partie imaginaire de sa transformée de Hilbert $\hat{\mathbf{m}}$.

Par égalité de Parseval, nous avons :

$$\langle \mathbf{m} , \hat{\mathbf{m}} \rangle = \frac{1}{L_m} \cdot \langle \hat{\mathbf{m}} , \hat{\mathbf{m}} \rangle \quad (\text{A.51})$$

$$= \frac{1}{L_m} \cdot \sum_{f=0}^{L_m-1} \hat{\mathbf{m}}(f) \cdot \overline{\hat{\mathbf{m}}(f)} , \quad (\text{A.52})$$

où $\overline{\hat{\mathbf{m}}(f)}$ représente le conjugué de $\hat{\mathbf{m}}(f)$.

En reprenant la définition de la transformée de Hilbert donnée par (7.33), les termes en $f = 0$ et $f = L_m/2$, si L_m est pair, disparaissent. Nous pouvons donc écrire de façon plus symétrique :

$$\langle \hat{\mathbf{m}} , \hat{\mathbf{m}} \rangle = \sum_{f=1}^{\lfloor (L_m-1)/2 \rfloor} \left[\hat{\mathbf{m}}(f) \cdot \overline{\hat{\mathbf{m}}(f)} + \hat{\mathbf{m}}(L_m - f) \cdot \overline{\hat{\mathbf{m}}(L_m - f)} \right] \quad (\text{A.53})$$

$$= \sum_{f=1}^{\lfloor (L_m-1)/2 \rfloor} \left[\hat{\mathbf{m}}(f) \cdot \overline{(-i \cdot \hat{\mathbf{m}}(f))} + \hat{\mathbf{m}}(L_m - f) \cdot \overline{(i \cdot \hat{\mathbf{m}}(L_m - f))} \right] , \quad (\text{A.54})$$

où $\lfloor . \rfloor$ représente la fonction partie entière. La dernière ligne est obtenue en remplaçant les coefficients $\hat{\mathbf{m}}(f)$ par leur expression. Le produit scalaire hermitien peut encore s'écrire comme :

$$\langle \hat{\mathbf{m}} , \hat{\mathbf{m}} \rangle = \sum_{f=1}^{\lfloor (L_m-1)/2 \rfloor} [i \cdot |\hat{\mathbf{m}}(f)|^2 - i \cdot |\hat{\mathbf{m}}(L_m - f)|^2] . \quad (\text{A.55})$$

Or, $\hat{\mathbf{m}}(L_m - f)$ est égal à $\overline{\hat{\mathbf{m}}(f)}$, pour $f \in \llbracket 1, \lfloor (L_m - 1)/2 \rfloor \rrbracket$, car le vecteur \mathbf{m} est réel (propriété de la transformée de Fourier). Le produit scalaire hermitien s'écrit donc :

$$\langle \hat{\mathbf{m}}, \hat{\check{\mathbf{m}}} \rangle = \sum_{f=1}^{\lfloor (L_m-1)/2 \rfloor} [i \cdot |\hat{\mathbf{m}}(f)|^2 - i \cdot |\hat{\mathbf{m}}(f)|^2] \quad (\text{A.56})$$

$$= 0 . \quad (\text{A.57})$$

Les vecteurs \mathbf{m} et $\check{\mathbf{m}}$ sont donc orthogonaux.

Bibliographie

- [AD05] F. Abrard and Y. Deville. A time-frequency blind signal separation method applicable to underdetermined mixtures of dependent sources. *Signal Processing*, 85(7) :1389–1403, july 2005.
- [AEB05] M. Aharon, M. Elad, and A. Bruckstein. The K-SVD algorithm. In *Proceedings of SPARS'05*, Rennes, France, November 2005.
- [AEB06] M. Aharon, M. Elad, and A. Bruckstein. The K-SVD : An algorithm for designing of overcomplete dictionaries for sparse representation. *to appear in the IEEE Trans. On Signal Processing*, 2006.
- [AEBLTAM⁺06] A. Aïssa-El-Bey, N. Linh-Trung, K. Abed-Meraim, A. Belouchrani, and Y. Grenier. Underdetermined blind separation of nondisjoint sources using time-frequency representation. accepté pour publication dans IEEE Transaction on Signal Processing, may 2006.
- [AGB06] S. Arberet, R. Gribonval, and F. Bimbot. A robust method to count and locate audio sources in a stereophonic linear instantaneous mixture. In *Proceedings of ICA conference*, pages 536–543, Charleston, SC, USA, march 2006.
- [Ama98] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2) :251–276, 1998.
- [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 Int Conf on Independent Component Analysis and Blind Signal Separation (ICA2001)*, pages 534–539, 2001.
- [Att99] H. Attias. Independent factor analysis. *Neural computation*, 11(4) :803–851, 1999.
- [Bad] R. Badeau. diapositives du cours modification temporelle et spectrale. http://www.tsi.enst.fr/~rbadeau/unrestricted/Fc_Transp.pdf, pages 8 et 11.
- [BD06] T. Blumensath and M.E. Davies. Sparse and shift-invariant representations of music. *IEEE Transactions on Audio, Speech and Language Processing*, 14(1) :50–57, january 2006.
- [Ben03] L.E. Benaroya. *Séparation de plusieurs sources sonores avec un seul microphone*. PhD thesis, Université de Rennes 1, IRISA, 2003.

- [BR05] R. Balan and J. Rosca. Sparse source separation using discrete prior models. In *Proceedings of SPARS'05*, pages 107–110, Rennes, France, November 2005.
- [BS95] A.J. Bell and T.J. Sejnowski. An information maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6) :1129–1159, 1995.
- [BS97] A.J. Bell and T.J. Sejnowski. The 'independent components' of natural scenes are edge filters. *Vision research*, 37(23) :3327–3338, 1997.
- [BW06] P. Berkes and L. Wiskott. On the analysis and interpretation of inhomogeneous quadratic forms as receptive fields. *Neural Computation*, 18(8) :1868–1895, 2006.
- [BZ01] P. Bofill and M. Zibulevsky. Underdetermined blind source separation using sparse representations. *Signal Processing*, 81 :2353–2362, 2001.
- [Car89] J.-F. Cardoso. Blind identification of independent components with higher-order statistics. In *Proc. Workshop on Higher-Order Spect. Anal.*, pages 157–160, Vail, Colorado, 1989.
- [Car91] J.-F. Cardoso. Super-symmetric decomposition of the fourth-order cumulant tensor. blind identification of more sources than sensors. In *In Proc. ICASSP*, pages 3109–3112, 1991.
- [Car92] J.-F. Cardoso. Iterative techniques for blind source separation using only fourth order cumulants. In *Proc. EUSIPCO*, Brussels, Belgium, 1992.
- [Car98] J.-F. Cardoso. Blind signal separation : statistical principles. *Proceedings of the IEEE, special issue on blind identification and estimation*, 90(8) :2009–2026, oct 1998.
- [CDS99] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1) :33–61, 1999.
- [Com89] P. Comon. Separation of stochastic processes. In *Proc. Workshop on Higher-Order Spectral Analysis*, pages 174–179, Vail, Colorado, 1989.
- [Com98] P. Comon. Blind channel identification and extraction of more sources than sensors. In *SPIE Conference on Advanced Algorithms and Architectures for Signal Processing*, pages 2–13, San Diego (CA), USA, july 1998.
- [CR01] S.F. Cotter and B.D. Rao. Application of tree-based searches to matching pursuit. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 6, pages 3933–3936, july 2001.
- [CR04] G. Casella and C.P. Robert. *Monte Carlo Statistical Methods*. Springer, 2nd edition, 2004.
- [CS93] J.-F. Cardoso and A. Soloumiac. Blind beamforming for non-gaussian signals. *IEE Proceedings-F*, 140(6) :362–370, december 1993.

- [CW92] R.R. Coifman and M.V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2) :713–718, 1992.
- [CW05] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Journal on Multiscale Modeling and Simulation*, 4(4) :1168–1200, nov 2005.
- [DDM04] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11) :1413–1457, 2004.
- [DE03] D. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization. *Proc. Nat. Acad. Sci.*, 100(5) :2197–2202, 2003.
- [DJ94] D.L. Donoho and I.M. Johnstone. Threshold selection for wavelet shrinkage of noisy data. In *Proceedings of the 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages A24–A25, Baltimore, USA, 1994.
- [DLD77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(B) :2–38, 1977.
- [DLV04] L. De Lathauwer and J. Vandewalle. Dimensionality reduction in higher-order signal processing and rank- (r_1, r_2, \dots, r_n) reduction in multilinear algebra. *Lin. Alg. Appl.*, 391 :31–55, nov 2004.
- [DM04] M. Davies and N. Mitianoudis. A simple mixture model for sparse overcomplete ica. *IEE proceedings in Vision, Image and Signal Processing*, 151(1) :35–43, february 2004.
- [DMZ94] G. Davis, S. Mallat, and Z. Zhang. Adaptive time-frequency decompositions. *SPIE Journal of Optical Engineering*, 33(7) :2183–2191, july 1994.
- [Don95] D.L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3) :613–627, 1995.
- [DT02] L. Daudet and B. Torr sani. Hybrid representations for audiophonic signal encoding. *Journal of Signal Processing, Special issue on Image and Video Coding Beyond Standards*, 82(11) :1595–1617, November 2002.
- [EA06] M. Elad and M. Aharon. Image denoising via learned dictionaries and sparse representation. In *Proceedings of the conference on Computer Vision and Pattern Recognition*, New York, june 2006.
- [EAH] K. Engan, S. Aase, and J. Husoy.
- [Ela] M. Elad. Why simple shrinkage is still relevant for redundant representations? *to appear in the IEEE Trans. On Information Theory*.

- [Eng00] K. Engan. *Frame based signal representation and compression*. PhD thesis, Stavanger university college, Norway, 2000.
- [ERKD99] K. Engan, B. Rao, and K. Kreutz-Delgado. Frame design using focuss with method of optimal directions (mod). In *Proc. Norwegian Signal Processing Symposium*, pages 65–69, Asker, Norway, september 1999.
- [ES02] K. Engan and K. Skretting. A novel image denoising technique using overlapping frames. In *Proc. of 2nd. IASTED Int. Conf. on Visualization, Imaging, and Image Processing*, pages 265–270, Malaga, Spain, september 2002.
- [FG06] C. Févotte and S.J. Godsill. A bayesian approach for blind separation of sparse sources. *IEEE Trans. on Speech and Audio Processing*, 2006.
- [FN03] M. Figueiredo and R. Nowak. An em algorithm for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 12(8) :906–916, 2003.
- [Fuc04] J-J Fuchs. On sparse representations in arbitrary redundant bases. *IEEE transactions on Information theory*, 50(6) :1341–1344, june 2004.
- [Fuc06] J.J. Fuchs. Communication personnelle, 2006.
- [GFV05] R. Gribonval, R. Figueras, and P. Vandergheynst. A simple test to check the optimality of sparse signal approximations. In *Proc. ICASSP 05*, pages 717–720, Philadelphia, USA, 2005.
- [Gir01] M. Girolami. A variational method for learning sparse and overcomplete representations. *Neural computation*, 13(11) :2517–2532, 2001.
- [GL96] G.H. Golub and C.F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.
- [GN03a] R. Gribonval and M. Nielsen. Approximation with highly redundant dictionaries. In *Proc. SPIE*, volume 5207 "Wavelets : Applications in Signal and Image Processing", pages 216–227, San Diego, USA, August 2003.
- [GN03b] R. Gribonval and M. Nielsen. Sparse representations in unions of bases. *IEEE Transactions on Information Theory*, 49(12) :3320–3325, december 2003.
- [GN04] R. Gribonval and M. Nielsen. On the strong uniqueness of highly sparse expansions from redundant dictionaries. In *Proc. Int Conf. Independent Component Analysis (ICA'04)*, sept 2004.
- [GN05] R. Gribonval and M. Nielsen. Beyond sparsity : recovering structured representations by ℓ_1 -minimization and greedy algorithms. – application to the analysis of sparse underdetermined ICA. *to appear in Journal of Advances in Computational Mathematics*, January 2005.
- [GR05] D.B. Grimes and R.P.N. Rao. Bilinear sparse coding for invariant vision. *Neural computation*, 17(1) :47–73, 2005.

- [Gri02] R. Gribonval. Sparse decomposition of stereo signals with matching pursuit and application to blind separation of more than two sources from a stereo mixture. In *Proc. Int. Conf. Acoust. Speech Signal Process*, Orlando, Florida, USA, 2002.
- [GV06] R. Gribonval and P. Vandergheynst. On the exponential convergence of matching pursuits in quasi-incoherent dictionaries. *IEEE Transactions on Information Theory*, 52(1) :255–261, 2006.
- [HCO99] A. Hyvarinen, R. Cristescu, and E. Oja. A fast algorithm for estimating overcomplete ica bases for image windows. In *In Proc. Int. Joint Conf. on Neural Networks*, pages 894–899, Washington, D.C., 1999.
- [HJA85] J. Hérault, C. Jutten, and B. Ans. Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé. In *Actes du Xème colloque GRETSI*, pages 1017–1022, Nice, France, may 1985.
- [HKO01] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley Interscience, 2001.
- [HO97] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural computation*, 9(7) :1483–1492, 1997.
- [Hot33] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24 :417–441, 498–520, 1933.
- [JH91] C. Jutten and J. Herault. Blind separation of sources, part i : An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24 :1–10, 1991.
- [JLVG05] P. Jost, S. Lesage, P. Vandergheynst, and R. Gribonval. Learning redundant dictionaries with translation invariance property : the MoTIF algorithm. In *Proc. SPARS'05*, Rennes, France, November 2005.
- [JLVG06] P. Jost, S. Lesage, P. Vandergheynst, and R. Gribonval. MOTIF : an efficient algorithm for learning translation invariant dictionaries. In *Proc. ICASSP'06*, Toulouse, France, May 2006.
- [Jon87] L.K. Jones. On a conjecture of huber concerning the convergence of projection pursuit regression. *Annals of statistics*, 15(2) :880–882, 1987.
- [Kar46] K. Karhunen. Zur spektraltheorie stochastischer prozesse. *Annales Academiae Scientiarum Fennicae, Series A1, Mathematica-Physica*, 34 :1–7, 1946.
- [KDMR⁺03] K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T.W. Lee, and T.J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15 :349–396, 2003.
- [KG06] S. Krstulović and R. Gribonval. MPTK : Matching pursuit made tractable. In *Proceedings of ICASSP'06*, Toulouse, France, May 2006. see <http://mptk.gforge.inria.fr>.

- [KMPY05] G.Z. Karabulut, L. Moura, D. Panario, and A. Yongacoglu. Flexible tree-search based orthogonal matching pursuit algorithm. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 4, pages 673–676, march 2005.
- [LBG80] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, pages 702–710, january 1980.
- [LDT04] S. Molla L. Daudet and B. Torrèsani. Towards a hybrid audio coder. In *Proceedings of the "International Conference Wavelet analysis and Applications"*, pages 13–24, Chongqing, China, 2004.
- [LGBB05] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya. Learning unions of orthonormal bases with thresholded singular value decomposition. In *Proc. of ICASSP'05*, Philadelphia, USA, March 2005.
- [LGK05] S. Lesage, R. Gribonval, and S. Krstulović. Séparation de sources dans le cas sous-déterminé : comparaison de deux approches basées sur des décompositions parcimonieuses. In *Proceedings GRETSI'05*, Louvain-la-Neuve, Belgique, 2005.
- [LGK06] S. Lesage, R. Gribonval, and S. Krstulović. Under-determined source separation : comparison of two approaches based on sparse decompositions. In *Proc. ICA'06*, pages 633–640, March 2006.
- [LLGS99] T-W Lee, M. Lewicki, M. Girolami, and T. Sejnowski. Blind source separation of more sources than mixtures using overcomplete representations. *Signal Processing Letters*, 6(4) :87–90, 1999.
- [LMG06] S. Lesage, B. Mailhe, and R. Gribonval. Apprentissage de motifs invariants par translation pour les décompositions parcimonieuses. In *Proc. Workshop TBICA 06*, Paris, France, july 2006.
- [LO99] M.S. Lewicki and B. Olshausen. A probabilistic framework for the adaptation and comparison of image codes. *Journal of the Optical Society of America*, 1999.
- [Loè55] M.M. Loève. *Probability theory*. Van Nostrand, Princeton, NJ, 1955.
- [LS00] M.S. Lewicki and T.J. Sejnowski. Learning overcomplete representations. *Neural computation*, 12(2) :337–365, 2000.
- [Mal98] S. Mallat. *A wavelet tour of signal processing*. Academic Press, San Diego, 1998.
- [MCW05] D.M. Malioutov, M. Cetin, and A.S. Willsky. Homotopy continuation for sparse signal representation. *IEEE ICASSP, Philadelphia, USA*, 5 :733–737, 2005.
- [Meh92] S. Mehrotra. On the implementation of a primal dual interior point method. *SIAM journal on optimization*, 2(4) :575–601, 1992.

- [MF06] S. Maria and J.J. Fuchs. Application of the global matched filter to stap data : an efficient algorithmic approach. *IEEE ICASSP, Toulouse, France*, 2006.
- [MGO02] T. Nishimura M. Goto, H. Hashiguchi and R. Oka. Rwc music database : Popular, classical, and jazz music databases. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pages 287–288, oct 2002.
- [MZ93] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12) :3397–3415, 1993.
- [Nat95] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24 :227–234, april 1995.
- [OF96a] B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381 :607–609, 1996.
- [OF96b] B. Olshausen and D. Field. Natural image statistics and efficient coding. *Network Computation in Neural Systems*, 7(2) :333–339, 1996.
- [OF97] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set : A strategy employed by v1? *Vision research*, 37 :3311–3325, 1997.
- [Ols00] B.A. Olshausen. Sparse coding of time-varying natural images. In *Int. Conf. Independent Component Analysis and Blind Source Separation (ICA)*, pages 603–608, Barcelona, Spain, september 2000.
- [OM00] B. Olshausen and K. Millman. Learning sparse codes with a mixture-of-gaussians prior. *Advances in Neural Information Processing Systems*, 12 :841–847, 2000.
- [OSL00] B.A. Olshausen, P. Sallee, and M.S. Lewicki. Learning sparse wavelet codes for natural images. *Proceedings of SPIE - The International society for optical engineering*, 4119(1) :200–207, 2000.
- [Oze06] A. Ozerov. *Adaptation des modèles pour la séparation de sources avec un seul capteur. Application à la séparation voix/musique dans les chansons populaires*. PhD thesis, Université de Rennes I, France, 2006.
- [Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11) :559–572, 1901.
- [PRK93] Y. Pati, R. Rezaifar, and P. Krishnaprasad. Orthogonal matching pursuit : Recursive function approximation with applications to wavelet decomposition. In *Proc. of 27th Annual Asilomar Conference on Signals, Systems, and Computers*, November 1993.
- [RD06] E. Ravelli and L. Daudet. Representations of audio signals in over-complete dictionaries : What is the link between redundancy factor

- and coding properties? In *Proc. Of 9th Int. Conf. on Digital Audio Effects (DAFX'06)*, pages 267–270, Montreal, september 2006.
- [REC⁺03] B.D. Rao, K. Engan, S.F. Cotter, J. Palmer, and K. Kreutz-Delgado. Subset selection in noise based on diversity measure minimization. *IEEE transactions on Signal processing*, 51(3) :760–770, 2003.
- [RKD99] B.D. Rao and K. Kreutz-Delgado. An affine scaling methodology for best basis selection. *IEEE Transactions on Signal Processing*, 47(1) :187–200, january 1999.
- [RS78] L.R. Rabiner and R.W. Schafer. *Digital Processing of Speech Signals*. Signal Processing Series. Prentice-Hall, 1978.
- [SBT00] S. Sardy, A. Bruce, and P. Tseng. Block coordinate relaxation methods for nonparametric wavelet denoising. *Journal of Computational and Graphical Statistics*, 9 :361–379, 2000.
- [SED04] J.-L. Starck, M. Elad, and D. Donoho. Redundant multiscale transforms and their application for morphological component analysis. *Advances in Imaging and Electron Physics*, 132, 2004.
- [SJ03] T. Strohmer and R. Heath Jr. Grassmannian frames with applications to coding and communications. *Appl. Comp. Harm. Anal.*, 14(3) :257–275, 2003.
- [SL05] E. Smith and M. Lewicki. Learning efficient auditory codes using spikes predicts cochlear filters. *Advances in Neural Information Processing Systems*, 17 :1289–1296, 2005.
- [TL02] F.J. Theis and E.W. Lang. Formalization of the two-step approach to overcomplete bss. In *Proc. 4th Intern. Conf. on Signal and Image Processing (SIP'02)*, pages 207–212, Hawaii, 2002.
- [TLP04] F.J. Theis, E.W. Lang, and C.G. Puntonet. A geometric algorithm for overcomplete linear ica. *Neurocomputing*, 56 :381–398, 2004.
- [Tro04] J. Tropp. Greed is good : Algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory*, 50(10) :2231–2242, october 2004.
- [VFG⁺03a] E. Vincent, C. Févotte, R. Gribonval, L. Benaroya, X. Rodet, A. Röbel, E. Le Carpentier, and F. Bimbot. Comment évaluer les algorithmes de séparation de sources audio? In *Proc. GRETSI'03*, pages 27–30, 2003.
- [VFG⁺03b] E. Vincent, C. Févotte, R. Gribonval, L. Benaroya, X. Rodet, A. Röbel, E. Le Carpentier, and F. Bimbot. A tentative typology of audio source separation tasks. In *Proc. ICA. NTT CS Labs*, pages 715–720, 2003.
- [VGF06] E. Vincent, R. Gribonval, and C. Févotte. Performance measurement in blind audio source separation. *IEEE Trans. Speech, Audio and Language Processing*, 14(4) :1462–1469, july 2006.

- [vHR98] J.H. van Hateren and D.L. Ruderman. Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proc.R.Soc.Lond. B*, (265) :2315–2320, 1998.
- [VR04] E. Vincent and X. Rodet. Underdetermined source separation with structured source priors. In *Proc. ICA*, pages 327–332, 2004.
- [VSI⁺04] L. Vielva, I.S. Santamaría, J. Ibáñez, D. Erdogmus, and J.C. Príncipe. On the estimation of the mixing matrix for underdetermined blind source separation in an arbitrary number of dimensions. In *Proc. ICA*, pages 185–192, Granada, Spain, may 2004.
- [WK03] H. Wersing and E. Körner. Learning optimized features for hierarchical models of invariant recognition. *Neural Computation*, 15(7) :1559–1588, 2003.
- [WPD06] J. Woodruff, B. Pardo, and R. Dannenberg. Remixing stereo music with score-informed source separation. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 314–319, Victoria, USA, 2006.
- [YR04] O. Yilmaz and S. Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, 52(7) :1830–1847, July 2004.
- [ZP01] M. Zibulevsky and B.A. Pearlmutter. Blind source separation by sparse decomposition in a signal dictionary. *Neural Computation*, 13(4) :863–882, 2001.
- [ZSC06] Y. Zhang, X. Shi, and C.H. Chen. Gaussian mixture model-based Bayesian analysis for underdetermined blind source separation. *Circuits systems and Signal processing*, 25(1) :81–94, 2006.

Résumé

Les décompositions parcimonieuses décrivent un signal comme combinaison d'un petit nombre de formes de base, appelées atomes. Le dictionnaire d'atomes, crucial pour l'efficacité de la décomposition, peut résulter d'un choix *a priori* (ondelettes, Gabor, ...) qui fixe la structure du dictionnaire ou d'un apprentissage à partir d'exemples représentatifs du signal. Nous proposons ici un cadre hybride combinant des contraintes structurelles et une approche par apprentissage. Les dictionnaires ainsi structurés apportent une meilleure adaptation aux propriétés du signal et permettent de traiter des volumes importants de données. Nous exposons les concepts et les outils qui étayent cette approche, notamment l'adaptation des algorithmes Matching Pursuit et K-SVD à des dictionnaires d'atomes constitués de motifs linéairement déformables, via une propriété d'adjonction. Nous présentons également des résultats de séparation de signaux monocanaux et multicanaux dans le cadre proposé.

Abstract

Sparse decompositions describe a signal as the combination of a few basis waveforms, called atoms. The dictionary of atoms, crucial for an efficient decomposition, may result from a prior choice (wavelets, Gabor atoms, ...) that fixes the dictionary structure or from a learning process on representative samples of the signal. Here, we propose a hybrid framework combining some structural constraints with a learning approach. Such structured dictionaries lead to a better adaptation to the properties of the signal and enable the handling of large amounts of data. We expose the concepts and tools that support this approach, notably the adaptation of the Matching Pursuit and K-SVD algorithms to dictionaries composed of linearly deformable motifs, via an adjunction property. Moreover, we present results of monochannel and multichannel separation using the proposed framework.

Mots clés

Décomposition parcimonieuse, dictionnaire, atome, motif, déformation, dictionnaire structuré, Matching Pursuit, K-SVD, apprentissage, séparation de sources, multicanal