Master 2 Sciences de l'ingénieur
Université Pierre et Marie Curie - Paris 6

Parcours ATIAM

Rapport de stage de fin d'étude

Extraction de la voix dans un signal audio par factorisation en matrices non-négatives semi-supervisée

Auteur
Damien Bouvier

Lieu du stage
Ircam - Equipe Analyse Synthèse

Responsable
Nicolas Obin - Ircam
Encadrants
Axel Roebel - Ircam
Marco Liuni - Ircam
(Prénom.Nom@ircam.fr)

2 août 2015

Résumé

Ce rapport de stage de 2ème année de master a pour sujet la factorisation en matrices non-négatives pour la séparation de sources audio appliquée à la voix. La contribution principale de ce stage est la réalisation d'une méthode semi-supervisée de séparation de la parole, basée sur un modèle source/filtre et l'intégration de contraintes pour informer la factorisation en matrices non-négatives sur les spécificités du signal de parole.
Tout d'abord, un état de l'art sur les méthodes de séparation de parole est présenté, et en particulier à propos des méthodes de factorisation en matrices non-négatives et de la spécification de connaissances sur la parole à l'intérieur de ce cadre. Puis, après un rappel des enjeux du stage, la méthode d'apprentissage du modèle de voix et les contraintes intégrées à la factorisation en matrices non-négatives sont présentées ainsi que la structure de l'algorithme proposé. Enfin, les contributions proposées sont évaluées et comparées aux méthodes de l'état de l'art sur une tâche de séparation de la parole en environnement bruité.

Mots-clés : Séparation de la parole, débruitage de la parole, factorisation en matrices non-négatives (NMF), modèle source/filtre, NMF sous contraintes. //

Abstract

This master thesis is on the use of non-negative matrix factorization for audio source separation applied to speech separation. The principal contribution of this work is the creation of a semi-supervised method for speech separation, based on a source/filter model and the integration of constraints in order to inform the non-negative matrix factorization with the specific properties of speech signal.
First, we present state-of-the-art speech separation methods, and emphasize on non-negative matrix factorization methods and the integration of a prori knowledge about speech in this framework. Then, after the goals of the stage are reminded, the speech model learning method and the integration of constraints in the non-negative matrix factorization are presented, as well as the structure of the proposed method. Finally, the contributions are evaluated and compared with state-of-the-art methods on a speech separation task in noisy environments.

Keywords : Speech separation, speech enhancement, non-negative matrix factorization (NMF), source/filter model, constrained NMF. //

Remerciements

Je tiens tout d'abord à remercier mon tuteur de stage Nicolas Obin, pour m'avoir proposé ce sujet intéressant et stimulant, mais surtout pour sa disponibilité et sa patience à mon égard. Ses nombreux conseils sur la rigueur scientifique, la rédaction, et ses astuces d'implémentation et de programmation ont été d'une aide précieuse.

Par ailleurs, ces travaux n'auraient jamais vu le jour sans l'aide continue d'Axel Roebel et de Marco Liuni. Je tiens à les remercier pour les nombreuses heures passées à partager leur connaissance du sujet.

Enfin, je tiens à remercier les autres étudiants ATIAM qui ont effectué leur stage à l'Ircam, Adrien, Axel, Hugo, Léo, Quentin et Xavier, pour pour tous ces bons moments, ainsi que tous les autres stagiaires et thésards de l'équipe Analyse-Synthèse, pour avoir fait de ce stage une expérience enrichissante sur tous les domaines.

Notations

Mathématiques

$\boldsymbol{X}$	Matrice
$\boldsymbol{X}_{i,j}$	Coefficient d'indices i,j de la matrice $\boldsymbol{X}$
$\boldsymbol{X}_j$	$j^{\text{ème}}$ colonne de la matrice $\boldsymbol{X}$
X_i	$i^{\text{ème}}$ ligne de la matrice $\boldsymbol{X}$
$\otimes$	Produit de Hadamard (terme à terme)
$(.)^T$	Transposé

Algorithmiques

$(.)^{(i)}$	Valeur à la $i^{\text{ème}}$ itération
n, N	Indice de trame temporelle, nombre de trames temporelles
f, F	Indice de point fréquentiel, nombre de points fréquentiels
k, K	Indice de composante élémentaire, nombre de composantes élémentaires
$\boldsymbol{V}$	Représentation temps-fréquence à valeurs positives ou nulles observée
$\boldsymbol{\Lambda}$	Représentation temps-fréquence à valeurs positives ou nulles estimée
$\boldsymbol{W}$	Matrice de dictionnaires spectraux à valeurs positives ou nulles
$\boldsymbol{H}$	Matrice d'activations temporelles à valeurs positives ou nulles

Acronymes

NMF	Factorisation en matrices non-négatives
EUC	Euclidienne, distance euclidienne
KL	Kullback-Leibler, divergence de Kullback-Leibler
IS	Itakura-Saito, divergence d'Itakura-Saito
TFCT	Transformée de Fourier à court terme

Table des matières

Résumé

Remerciements

Notations

Introduction **1**
Cadre . . . 1
Objectifs . . . 1

1 État de l'art sur la séparation de parole **3**
1.1 La factorisation en matrices non-négatives . . . 3
1.1.1 Fonctions de coût . . . 5
1.1.2 Mise à jour multiplicative . . . 6
1.1.3 Application à la séparation de sources . . . 7
1.2 Spécificitées de la voix . . . 7
1.2.1 Modèle source/filtre . . . 7
1.2.2 Intégration à la NMF . . . 7
1.3 Autres méthodes basées sur la NMF . . . 9

2 Contributions **12**
2.1 Apprentissage guidé par la physique . . . 12
2.1.1 Estimation du filtre par l'algorithme *True Envelope* . . . 12
2.1.2 Informations phonétiques . . . 13
2.2 Intégration de contraintes . . . 15
2.2.1 Contraintes et mises à jour multiplicatives . . . 16
2.2.2 Contraintes de la littérature . . . 16
2.2.3 Contrainte de cohérence source/filtre . . . 19
2.2.4 Contraintes adaptatives . . . 20
2.3 Structure de l'algorithme . . . 21

3 Évaluation expérimentale **23**
3.1 Bases de données . . . 23
3.2 Algorithmes et références . . . 24
3.3 Protocoles expérimentaux . . . 24
3.4 Résultats . . . 26

4 Conclusion **29**

A Formules de mises à jour multiplicatives **30**
A.1 Cas général de la β-divergence 30
A.2 Modèle source/filtre 30
A.2.1 Modèle de Durrieu 30
A.2.2 Modèle proposé 31
A.3 Gradient des contraintes 31
A.3.1 Contrainte de parcimonie 32
A.3.2 Contrainte de continuité temporelle 32
A.3.3 Contrainte de décorrélation 32
A.3.4 Contrainte de cohérence source/filtre 33

Bibliographie **35**

Liste des figures **39**

Liste des tableaux **39**

Introduction

La séparation de sources est un domaine de traitement du signal consistant à extraire d'un ensemble de données ses différentes constituantes. En audio, elle permet la détection, la localisation, et la séparation de sources sonores, et ses applications sont variées : indexation de contenu audio, analyse de scène sonore, démixage et remixage audio.
La cas particulier de la séparation de la parole est un outil de pré-traitement requis pour l'exploitation des technologies de la parole dans des environnements sonores réels. Ces technologies regroupent des outils comme la reconnaissance de la parole, la reconnaissance de locuteur, ou bien l'écoute artificielle de robots.
L'objectif de ce stage est d'informer la séparation de la parole par factorisation en matrices non-négatives à l'aide de connaissances a priori sur le signal de parole, ce qui s'est traduit par une semi-supervision de l'algorithme, l'intégration d'un modèle source/filtre et l'ajout de contraintes.

Cadre

Ce stage s'est déroulé du 2 mars au 31 juillet 2015 sous l'encadrement de Nicolas Obin, dans l'équipe Analyse et Synthèse des sons à l'Ircam. Cette équipe fait partie de l'UMR STMS (Sciences et Technologies de la Musique et du Son), qui associe le CNRS, l'UPMC, le ministère de la Culture et l'Ircam, et se focalise sur l'étude et la modélisation des phénomènes sonores et musicaux à des fins d'analyse, de transformation et de synthèse.
Les travaux effectués se positionnent dans une continuité de recherche sur l'exploitation des factorisations en matrices non-négatives pour l'indexation et la production de contenus audio multimédia (travaux menés sur le cinéma 3D dans le cadre du projet européen 3DTVS) et l'analyse de scène sonore (travaux à venir sur l'écoute artificielle d'un robot dans le projet ROUTE).

Objectifs

Dans ce stage, nous cherchons à développer une méthode de séparation de sources pour l'extraction de la voix parlée dans des enregistrements audio pouvant contenir tout type de bruit (stationnaire, non-stationnaire, fond musical, etc). Dans le cadre de nos recherches, nous nous limitons à l'étude de signaux monophoniques, et considérons que le mélange est linéaire et instantanée ; nous sommes donc face à un problème sous-déterminée, puisque nous cherchons à obtenir deux sources (la parole et le fond sonore) à partir d'un seul signal mono-canal.
Si la séparation de sources audio par factorisation en matrices non-négatives est par essence non-informée et non-supervisée, une tendance lourde est à l'intégration de connaissance a priori sur la nature des sources sonores. Pour la séparation de la parole, deux sources sonores uniquement sont considérées : la voix et le fond sonore (qu'on appellera par commodité bruit). Or, si de nombreux modèles acoustique et signal ont été développés pour la voix (spécifique à un locuteur

ou non), il n'existe pas à ce jour de modèle universel pour le fond sonore (sa variabilité rend impossible une modélisation exhaustive). Ce stage se concentrera donc sur la réalisation d'un modèle de voix dans un cadre semi-supervisé de séparation de la parole : un modèle de voix est spécifié et appris à partir d'une base de données de parole ; en revanche, le modèle de bruit est laissé libre, et n'est déterminé que lors de la séparation.

Le chapitre 1 dresse un état de l'art de la séparation de parole, et introduit les bases théoriques de la factorisation en matrices non-négatives. Le chapitre 2 détaille ensuite les principales contributions apportées au cours du stage. Enfin, le chapitre 3 décrit la procédure d'évaluation et les résultats obtenus.

Chapitre 1

État de l'art sur la séparation de parole

Historiquement, l'extraction de la voix fait partie de la thématique du rehaussement de la parole (*speech enhancement*), qui consiste à améliorer la qualité de la parole au sein d'un environnement sonore bruité. Par exemple, des méthodes comme la soustraction spectrale [1] ou les réseaux neuronaux profonds [2, 3] sont utilisées pour réaliser le débruitage de la parole. Plus récemment, les méthodes de séparation de sources audio ont été introduites en rehaussement de la parole, sous le nom de séparation de la parole (*speech separation*), pour séparer la voix du reste de l'environnement sonore. Il s'agit en fait d'une ré-interprétation des méthodes de séparation de sources audio pour le débruitage : il n'y a plus que deux sources audio d'intérêts, la voix et le reste (fond sonore, ou bruit).
Dans ce domaine de recherche, une des méthodes actuellement dominante est la factorisation en matrices non-négatives (NMF [1]), et c'est sur celle-ci que nous avons porté nos recherches.

Nous présentons ici tout d'abord les bases théoriques de la NMF, puis parlons du modèle source/filtre de la voix et de son intégration dans le modèle de la NMF ; enfin nous dressons un tableau (non exhaustif) des différentes méthodes de séparation de parole qui se basent sur la NMF.

1.1 La factorisation en matrices non-négatives

Le principe de la NMF, introduit tout d'abord par Paatero et Tapper [4] et popularisé par Lee et Seung [5], est d'approcher une matrice $\boldsymbol{V}$ de taille $F \times N$ à coefficients positifs ou nuls par :

$$\boldsymbol{V} \simeq \boldsymbol{\Lambda} = \boldsymbol{W}\boldsymbol{H} \tag{1.1}$$

où $\boldsymbol{W}$ et $\boldsymbol{H}$ sont deux matrices à coefficients positifs ou nuls, de taille respective $F \times K$ et $K \times N$. La NMF revient donc à chercher, pour un K préalablement choisi, la meilleure matrice approchée $\boldsymbol{\Lambda}$ de rang maximum K, selon une fonction de coût choisie ; chaque colonne de la matrice $\boldsymbol{W}$ est

1. Par souci de simplicité et de cohésion avec la littérature sur le sujet, nous adoptons l'acronyme anglophone *NMF* (pour *Non-negative Matrix Factorization*). De plus, pour éviter une possible confusion entre les termes *matrice positive* et *matrice non-négative*, nous précisons que ce dernier désigne ici une matrice à coefficients positifs ou nuls, et non une matrice semi-définie positive.

une base de la décomposition de $\boldsymbol{\Lambda}$, et $\boldsymbol{H}$ correspond à l'encodage de $\boldsymbol{\Lambda}$ selon cette décomposition.
L'idée principale des NMFs est la contrainte de positivité des matrices, qui oblige l'estimation $\boldsymbol{\Lambda}$ des données à se faire de manière additive, de sorte que l'annulation entre bases de $\boldsymbol{W}$ n'est pas possible (contrairement à l'analyse en composantes principales par exemple).

En audio, la matrice $\boldsymbol{V}$ est une représentation temps-fréquence du signal à analyser, le plus souvent la Transformée de Fourier à Court-Terme (TFCT) ou la transformée à Q-constant. Puisque l'on nécessite des coefficients positifs ou nuls, la matrice considérée est une puissance positive de l'amplitude, e.g. dans le cas de la STFT :

$$V_{fn} = |\texttt{TFCT}\{x\}_{fn}|^{\alpha} \quad , \alpha \in \mathbb{R}_{+}^{*} \tag{1.2}$$

Dans le cadre audio, $\boldsymbol{W}$ est un dictionnaire de sources sonores, chaque colonne étant un spectre de puissance α, et $\boldsymbol{H}$ représente les activations temporelles de ces sources. La FIGURE 1.1 montre un exemple de décomposition sur un signal simple ; chaque base de $\boldsymbol{W}$ représente une note et la ligne correspondante de $\boldsymbol{H}$ contient ses activations dans le temps. Généralement, une normalisation est appliquée sur les bases de $\boldsymbol{W}$ (et reportée dans $\boldsymbol{H}$) pour que l'énergie du système ne soit présente que dans une seule matrice. Dans le cas de l'exemple de la FIGURE 1.1, nous avons normalisé les spectres des bases par leur norme ℓ_2 ; la matrice $\boldsymbol{H}$ contient donc l'énergie des bases à chaque instant.

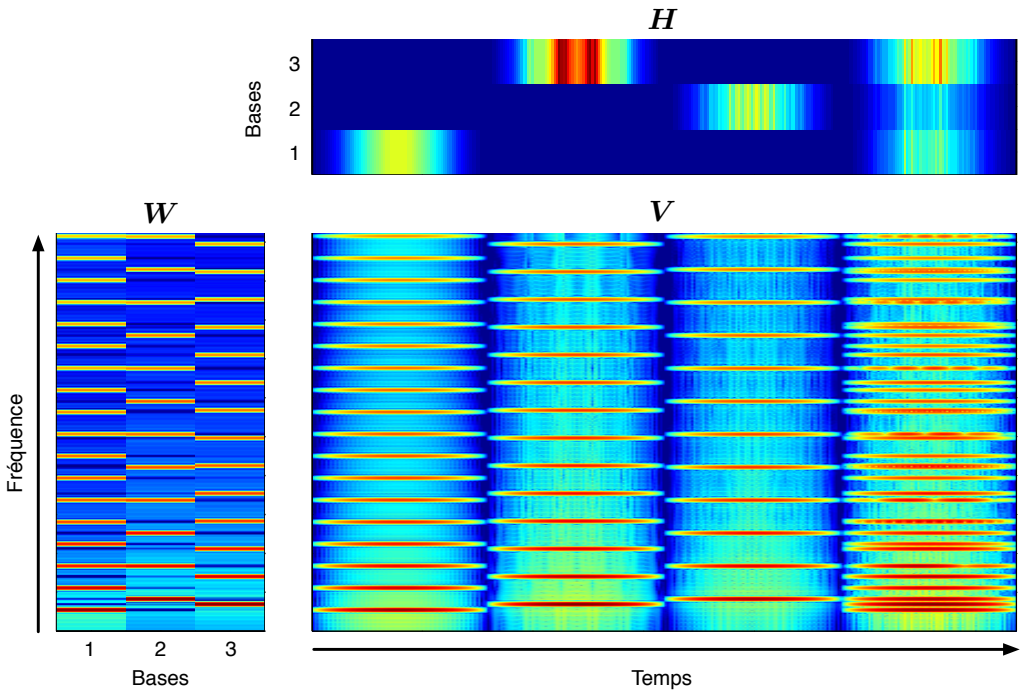


FIGURE 1.1 – Exemple d'une décomposition par NMF d'un signal synthétique composé de 3 notes joués successivement puis simultanément.

1.1.1 Fonctions de coût

Dans la littérature, l'estimation (1.1) se fait par la minimisation suivante :

$$\arg\min_{\boldsymbol{W},\boldsymbol{H}\geq 0} D(\boldsymbol{V}|\boldsymbol{W}\boldsymbol{H}) \tag{1.3}$$

où la fonction de coût $D(\boldsymbol{V}|\boldsymbol{W}\boldsymbol{H})$ est de la forme [2]

$$D(\boldsymbol{V}|\boldsymbol{W}\boldsymbol{H}) = \sum_{f=1}^{F}\sum_{n=1}^{N} d\left(\boldsymbol{V}_{fn}|(\boldsymbol{W}\boldsymbol{H})_{fn}\right) \tag{1.4}$$

où $d(x|y)$ est une fonction de coût scalaire. En audio, les plus couramment utilisées sont les divergences de Kullback-Leiber (KL) et d'Itakura-Saito (IS) [6, 7] ; ces divergences, avec la distance euclidienne (EUC), sont des cas particuliers de la β-divergence. Elles sont toutes rappelées Tableau 1.1.

β-divergence ($\beta \in \mathbb{R}\setminus\{0,1\}$)	$d_\beta(x\|y)$	$=$	$\frac{1}{\beta(\beta-1)}\left(x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}\right)$
Itakura-Saito	$d_{IS}(x\|y)$	$=$	$\frac{x}{y} - \log\frac{x}{y} - 1$
		$=$	$\lim_{\beta\to 0} d_\beta(x\|y)$
Kullback-Leiber	$d_{KL}(x\|y)$	$=$	$x\log\frac{x}{y} + (y-x)$
		$=$	$\lim_{\beta\to 1} d_\beta(x\|y)$
Distance euclidienne	$d_{EUC}(x\|y)$	$=$	$\frac{1}{2}(x-y)^2$
		$=$	$d_{\beta=2}(x\|y)$ avec

Tableau 1.1 – Formules de la β-divergence scalaire et de ses cas particuliers : IS, KL et EUC

Plusieurs propriétés font de la β-divergence une bonne mesure d'erreur de reconstruction [6] :

- $d_\beta(x|y)$ possède un unique minimum global au point $y = x$, et croît avec la distance $|y-x|$.
- $d_\beta(x|y)$ est convexe sur $\mathbb{R}_+$ si et seulement si $1 \leq \beta \leq 2$; pour le coût matriciel $D_\beta(\boldsymbol{V}|\boldsymbol{W}\boldsymbol{H})$, cette convexité ne se généralise pas par rapport au couple de variables $(\boldsymbol{W},\boldsymbol{H})$, mais est conservée par rapport à $\boldsymbol{W}$ (respectivement $\boldsymbol{H}$) lorsque $\boldsymbol{V}$ et $\boldsymbol{H}$ (respectivement $\boldsymbol{W}$) sont fixes.

De plus, le cas particulier de IS possède une propriété intéressante pour le signal audio :

$$d_{IS}(\lambda x|\lambda y) = d_{IS}(x|y), \lambda \in \mathbb{R}^* \tag{1.5}$$

Cette mesure donne donc le même poids aux coefficients de faible et forte amplitude, ce qui permet de prendre en compte tout ce qui est perceptiblement important, même à bas niveau.

2. Nous utilisons dans ce stage des notations d'indices partant de 1, comme utilisé dans MatLab®.

Mais cela peut affecter la stabilité numérique de l'algorithme, en cas de présence de valeurs nuls ou très petites dans la matrice observée $\boldsymbol{V}$. Roebel et *al.* [8] proposent donc d'utiliser un palier de bruit en ajoutant aux matrices observée $\boldsymbol{V}$ et estimée $\boldsymbol{\Lambda}$ une constante fixée, qui permet, outre d'assurer la stabilité, de définir la précision maximale voulue pour la reconstruction.

1.1.2 Mise à jour multiplicative

Il existe de nombreux algorithmes pour minimiser la fonction de coût (un inventaire exhaustif et leur comparaison est disponible dans Cichocki et *al.* [9]) ; parmi ceux-ci, les algorithmes multiplicatifs, introduits par Lee et Seung [5], proposent une mise à jour itérative de $\boldsymbol{W}$ et $\boldsymbol{H}$ adaptée au coût choisi, e.g. pour EUC [3] :

$$\begin{cases} \boldsymbol{W}^{(i+1)} \longleftarrow \boldsymbol{W}^{(i)} \otimes \dfrac{\boldsymbol{V}\boldsymbol{H}^T}{(\boldsymbol{W}\boldsymbol{H})\,\boldsymbol{H}^T} \\ \boldsymbol{H}^{(i+1)} \longleftarrow \boldsymbol{H}^{(i)} \otimes \dfrac{\boldsymbol{W}^T\boldsymbol{V}}{\boldsymbol{W}^T\,(\boldsymbol{W}\boldsymbol{H})} \end{cases} \tag{1.6}$$

où i est un entier indiquant l'itération.
Cette méthode reste largement utilisée grâce à sa facilité d'implémentation et à la conservation de la non-négativité à chaque itération. De plus, Lee et Seung donnent dans [10] la preuve de la non-croissance du coût sous les mises à jour proposées pour EUC et KL. Nakano et *al.* [11] proposent une formule générale à la β-divergence, ainsi que la preuve de sa non-croissance.
Il est important de préciser que cette preuve de non-croissance n'implique pas obligatoirement la convergence de l'algorithme vers un minimum local ; en effet, il peut tendre vers un point d'équilibre non minimal (point de selle), ou vers un bord du domaine de définition (e.g. des matrices avec valeurs nulles) qui peut ne pas être minimal.

Cette méthode peut être vu comme un cas particulier de la descente de gradient où le pas d'avancement serait déterminé automatiquement. Nommons $\mathcal{C}$ la fonction de coût choisie, alors l'algorithme de descente de gradient appliqué à $\boldsymbol{W}$ peut s'écrire :

$$\boldsymbol{W}^{(i+1)} \longleftarrow \boldsymbol{W}^{(i)} - \boldsymbol{\nu}\nabla_{\boldsymbol{W}}\mathcal{C} \tag{1.7}$$

où $\boldsymbol{\nu}$ est le pas d'avancement (ici matriciel). En le choisissant automatiquement comme

$$\boldsymbol{\nu} = \frac{\boldsymbol{W}^{(i)}}{\nabla^+_{\boldsymbol{W}}\mathcal{C}} \tag{1.8}$$

où $\nabla_{\boldsymbol{W}}\mathcal{C} = \nabla^+_{\boldsymbol{W}}\mathcal{C} - \nabla^-_{\boldsymbol{W}}\mathcal{C}$, la mise à jour (1.7) devient

$$\boldsymbol{W}^{(i+1)} \longleftarrow \boldsymbol{W}^{(i)} \otimes \frac{\nabla^-_{\boldsymbol{W}}\mathcal{C}}{\nabla^+_{\boldsymbol{W}}\mathcal{C}} \tag{1.9}$$

En faisant de même pour $\boldsymbol{H}$ et en prenant EUC ou KL comme coût $\mathcal{C}$, on retrouve les formules de mise à jour proposées par Lee et Seung [5, 10].
Cette méthode de construction d'un algorithme multiplicatif pour la NMF permet facilement le changement de divergence, l'ajout de pénalité dans la fonction de coût ou même le changement du modèle de factorisation ; mais il n'y a aucune preuve sur la convergence de la descente de gradient avec le pas d'avancement ainsi choisi, bien qu'elle soit toujours observée. Nous utiliserons par la suite cette méthode pour déterminer les mises à jour multiplicatives des algorithmes proposés.

3. Dans le cadre de ce stage, toutes les fractions incluant des matrices sont à considérer terme à terme.

1.1.3 Application à la séparation de sources

Bien que ce ne soit pas leur seul application, les NMFs sont très largement utilisées en séparation de sources.

Filtrage de Wiener Si l'on considère que l'on connait l'ensemble $\mathcal{K}_c$ des indices des bases $\boldsymbol{W}_k$ correspondant à une même source cible, alors il est possible d'obtenir un estimateur par moindres carrés du spectrogramme $\boldsymbol{X}^c$ de cette cible par filtrage de Wiener :

$$\boldsymbol{X}^c_{fn} = \frac{\sum_{k\in\mathcal{K}_c}(\boldsymbol{W}_{fk}\boldsymbol{H}_{kn})}{(\boldsymbol{W}\boldsymbol{H})_{fn}} \otimes \boldsymbol{X}_{fn}, \quad \forall(f,n) \tag{1.10}$$

où $\boldsymbol{X}_{fn}$ est le spectrogramme du mélange observé.
Malheureusement, le filtrage de Wiener ne prend pas en compte la phase. Une solution a été proposé par Le Roux et Vincent [12], ou encore Magron et *al.* [13], qui proposent des algorithmes basés sur la NMF intégrant l'information de la phase.

Supervision de la NMF La NMF est par essence une méthode non-informée et non-supervisée ; il n'y a aucune information sur ce que représente les bases de $\boldsymbol{W}$, ce qui est pourtant nécessaire pour pouvoir appliquer le filtrage de Wiener et effectuer la séparation des sources.
Pour remédier à cela, il est possible de superviser la séparation de sources par NMF. On obtient alors un algorithme ayant la structure suivante :

1. Apprentissage au préalable d'un dictionnaire pour chaque source à extraire (généralement en appliquant une NMF non-supervisée sur un signal propre représentatif de la source) ;
2. NMF sur le signal à séparer, en fixant $\boldsymbol{W}$ comme une concaténation des dictionnaires appris précédemment ;
3. Filtrage de Wiener pour extraire chaque source.

1.2 Spécificitées de la voix

1.2.1 Modèle source/filtre

La voix, comme une majorité des instruments de musiques, peut être modélisée par un système source/filtre. Dans ce modèle, un système excitateur crée un signal source qui est convolué par un système résonateur. Lorsque l'on passe dans le domaine de Fourier, cette convolution temporelle se simplifie en un produit entre le spectre de l'excitation et le spectre de la source ; la FIGURE 1.2 montre un exemple de ce produit spectral.
Dans le cas de la parole, la source d'excitation peut être soit la vibration périodique des plis vocaux, soit du bruit. Le filtrage, produit par le conduit vocal, est essentiel pour le timbre de la voix et va déterminer le phonème prononcé. La FIGURE 1.3 montre un exemple de la décomposition source/filtre d'un signal de parole.

1.2.2 Intégration à la NMF

La NMF, telle que proposé dans [5], se base sur la combinaison linéaire des bases spectrales d'un dictionnaire ; elle n'est donc pas adaptée à la variabilité des sons résultants d'un système source/filtre (puisqu'il faudrait une base par combinaison source/filtre possible). Pour remédier à cela, Durrieu et *al.* [14] proposent une factorisation en matrices non-négatives spécifique au

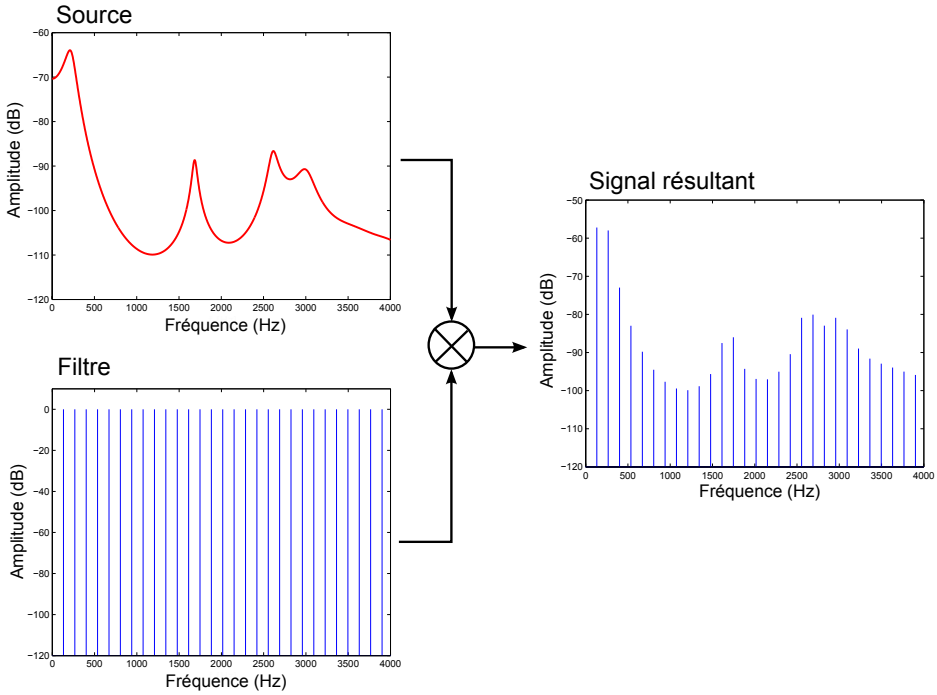


FIGURE 1.2 – Exemple du produit spectral du modèle source/filtre

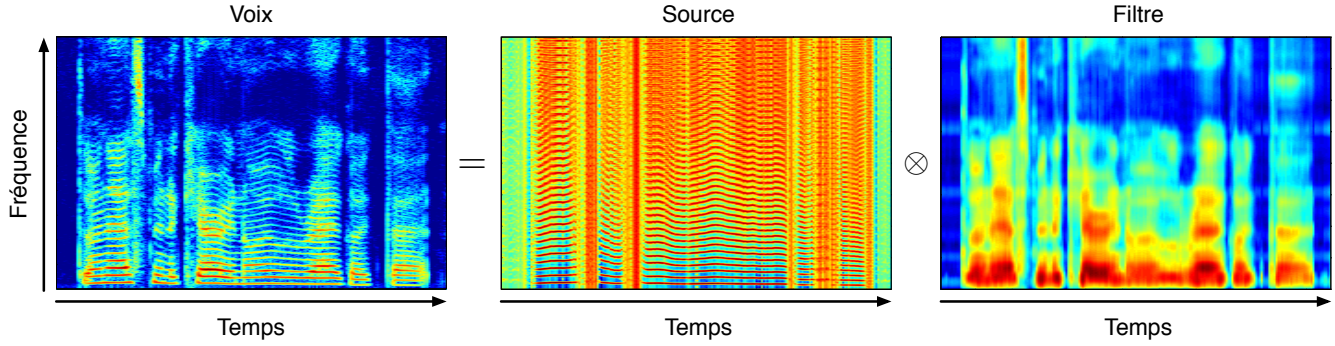


FIGURE 1.3 – Décomposition source/filtre du spectrogramme d'une voix

modèle source/filtre pour une tâche de séparation de l'instrument principal en musique [4].
Dans ce modèle le spectrogramme d'amplitude de la voix ou de l'instrument est représenté comme le produit d'un spectrogramme des sources $\boldsymbol{\Lambda}^{\text{ex}}$ par un spectrogramme des filtres $\boldsymbol{\Lambda}^{\Phi}$. La factorisation classique de la NMF est alors appliqué sur $\boldsymbol{\Lambda}^{\text{ex}}$ et $\boldsymbol{\Lambda}^{\Phi}$. Par ailleurs, chaque base du dictionnaire des filtres est représenté comme une combinaison linéaire de filtres élémentaires venant d'un banc de filtre pour assurer la continuité fréquentielle des filtres estimés. La FIGURE 1.4 illustre la décomposition dans son ensemble.

Tout ce qui est considéré comme bruit (i.e. l'accompagnement musical pour Durrieu et *al.*, le fond sonore pour nous) est représenté en utilisant la décomposition classique de la NMF. Le modèle général ainsi obtenu est nommée IMM (*Instantaneous Mixture Model*, ou Modèle de mélange instantanée) et s'écrit :

$$\begin{array}{rcccccc} \boldsymbol{\Lambda} & = & \boldsymbol{\Lambda}^{\text{ex}} & \otimes & \boldsymbol{\Lambda}^{\Phi} & + & \boldsymbol{\Lambda}^{N} \\ & = & (\boldsymbol{W}^{\text{ex}}\boldsymbol{H}^{\text{ex}}) & \otimes & (\boldsymbol{W}^{\Phi}\boldsymbol{U}^{\Phi}\boldsymbol{H}^{\Phi}) & + & (\boldsymbol{W}^{N}\boldsymbol{H}^{N}) \end{array} \tag{1.11}$$

où les matrices $\boldsymbol{H}^{\text{ex}}$, $\boldsymbol{H}^{\Phi}$ et $\boldsymbol{H}^{N}$ représentent respectivement les activations temporelles des excitations, des filtres et du bruit, $\boldsymbol{W}^{\text{ex}}$ est un dictionnaire d'excitations, $\boldsymbol{W}^{\Phi}$ est un banc de filtres, $\boldsymbol{U}^{\Phi}$ contient les combinaisons linéaires construisant les bases de filtres à partir de $\boldsymbol{W}^{\Phi}$, et $\boldsymbol{W}^{N}$ est le dictionnaire des bases de bruit. Les matrices $\boldsymbol{W}^{\text{ex}}$ et $\boldsymbol{W}^{\Phi}$ sont construites préalablement à la NMF, et restent fixes durant toutes les itérations ; $\boldsymbol{W}^{\text{ex}}$ contient à la fois les spectres des excitations harmoniques potentielles et les spectres des excitations bruitées. Le TABLEAU 1.2 résume le rôle des différentes matrices.
Les règles de mise à jour multiplicatives spécifiques à ce modèle, déduites à partir de (1.9), sont décrites en ANNEXE A.2.1.

Matrice	Taille	Rôle	Construction
$\boldsymbol{W}^{\text{ex}}$	$F \times K_e$	bases spectrales des excitations	imposé
$\boldsymbol{H}^{\text{ex}}$	$K_e \times N$	activations temporelles des excitations	libre
$\boldsymbol{W}^{\Phi}$	$F \times L$	banc de filtres	imposé
$\boldsymbol{U}^{\Phi}$	$L \times K_f$	gains formant les K_f filtres vocaux	libre
$\boldsymbol{H}^{\Phi}$	$K_f \times N$	activations temporelles des filtres	libre
$\boldsymbol{W}^{N}$	$F \times K_n$	bases spectrales du bruit	libre
$\boldsymbol{H}^{N}$	$K_n \times N$	activations temporelles du bruit	libre

TABLEAU 1.2 – Rôle des matrices dans le modèle IMM de Durrieu et *al.* [14]

1.3 Autres méthodes basées sur la NMF

Pour pouvoir mieux situer nos travaux par rapport à la littérature, nous dressons ici un bref descriptif des différents algorithmes de séparation de parole utilisant la NMF ou des méthodes qui en sont inspirées.

4. Ce n'est pas la seule intégration du modèle source/filtre dans la NMF ; on peut citer par exemple les travaux de Virtanen et Klapuri [15] qui proposent une nouvelle factorisation où chaque base spectrale est décomposée en un produit d'un spectre d'amplitude d'une excitation harmonique par celui d'un filtre.

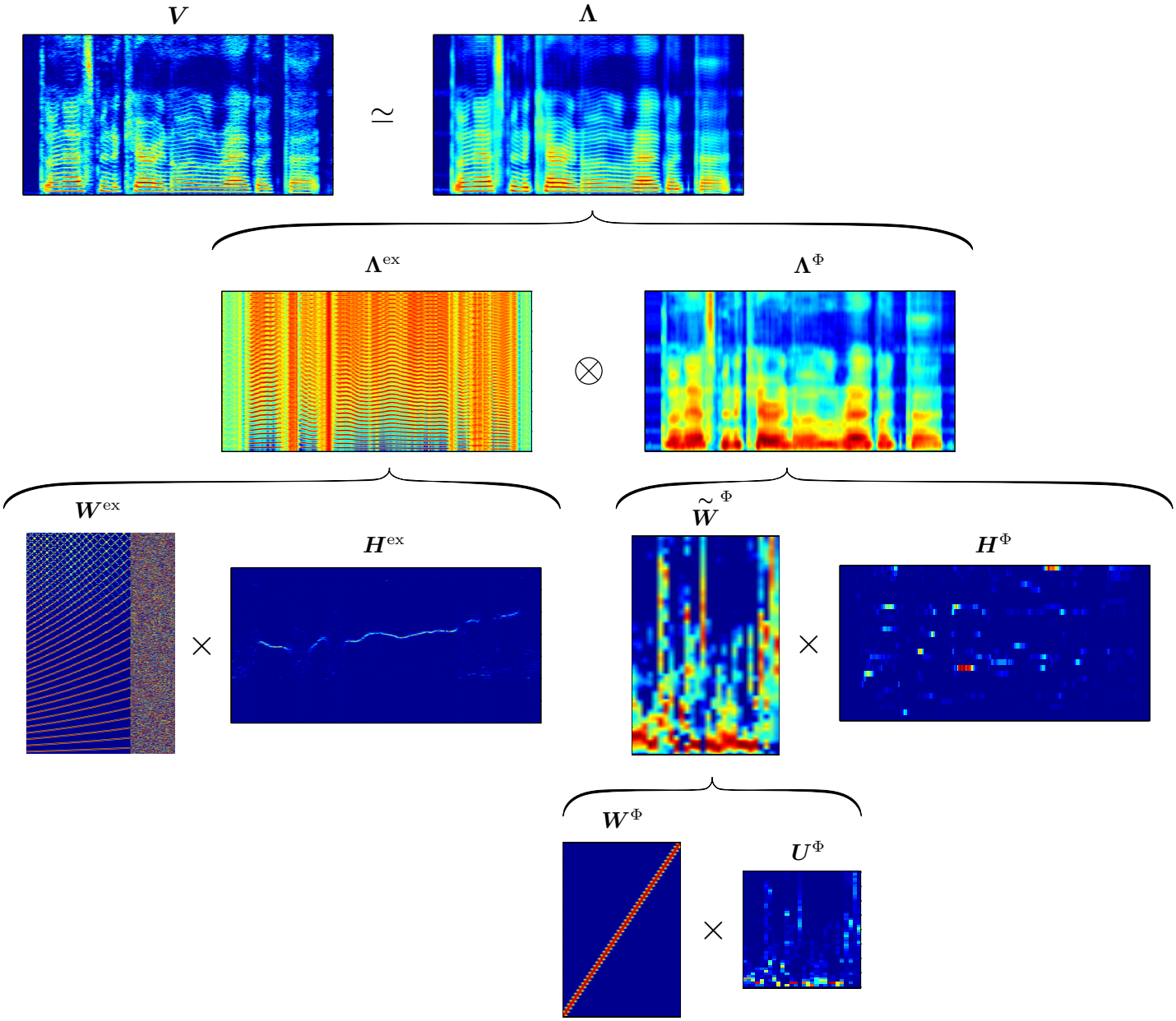


FIGURE 1.4 – Décomposition source/filtre dans le modèle IMM de Durrieu et *al.* [14]

La séparation de la parole par NMF s'inspire largement de l'expérience acquise en reconnaissance de la parole, notamment sur la modélisation de la structure du langage par HMM (chaînes de Markov cachées), la construction d'un modèle de voix indépendant du locuteur (modèle « moyen » ou « universel »), ou encore l'adaptation au locuteur.
Ainsi, Mysore et Smaragdis [16, 17] proposent la combinaison des NMFs avec un modèle de langage construit grâce à des chaînes de Markov cachées (HMM). Des petits dictionnaires sont appris pour chaque phonème, et puis des HMMs sont entraînés pour modéliser les transitions entre phonèmes. De plus, une macro-structure du langage sur la transition entre les mots est aussi construite à l'aide des HMMs. La séparation combine alors un modèle NMF classique pour le bruit avec le modèle NMF-HMM développée.
Par ailleurs, Sun et Mysore [18] proposent pour la NMF de construire un Modèle Universel de Parole (USM, pour *Universal Speech Model*) comme la concaténation de dictionnaires appris sur de nombreux locuteurs ; l'idée est que, si l'apprentissage est fait avec un assez grand nombre de voix différentes, il existera toujours des locuteurs dont les caractéristiques seront assez proches de celles du locuteur cible à séparer. De plus, les auteurs introduisent une contrainte de parcimonie de groupe qui permet d'empêcher la NMF de combiner des bases venant de locuteurs différents (ce qui laisserait une trop grande possibilité de reconstruction). Germain et Mysore [19] développe, en se basant sur l'USM, une méthode de séparation de la parole temps-réel, où les bases de bruit sont ré-adaptées en continu.
Ozerov et *al.* [20], proposent quant à eux une méthode où les modèles, une fois appris, peuvent s'adapter au locuteur lors de la séparation. Ils présentent un formalisme général dans le cadre de l'inférence bayésienne, et applique leur algorithme à la séparation de la voix dans la musique.
Enfin, Le Magoarou et *al.* [21, 22] proposent l'intégration de l'information textuelle dans la NMF. Pour cela, les auteurs synthétisent une voix parlée à partir du texte, et effectue une NMF conjointe sur la parole synthétisée et la voix bruitée. Une matrice de de transition leur permet de prendre en compte de petites différences temporelles ou fréquentielles entre la voix synthétisée et la parole à séparer. La contrainte de la connaissance a priori du texte est forte, mais cela est se vérifie dans certaines applications (e.g. le doublage ou le karaoké).
Il existe aussi des modèles liant NMF et réseaux neuronaux profonds (DNN pour Deep Neural Network). Williamson et *al.* [23] proposent par exemple d'estimer les activations de la NMF à l'aide d'un DNN. Le Roux et *al.* [24] présentent quant à eux un modèle de NMF profond, où les itérations de la NMF sont déployées à la manière d'un réseau neuronal.

Chapitre 2

Contributions

La séparation de la parole nécessite la modélisation de deux types de sources sonores : la voix, et le bruit. Or la création d'un modèle universel de bruit est problématique, et son apprentissage est généralement impossible dans le cadre d'une application réelle. Nos travaux se placent donc dans le cadre d'une séparation semi-supervisée de la parole, dans laquelle les bases de parole sont apprises en amont sur des bases de données, alors que la modélisation du bruit est déterminée de manière non-supervisée lors de la séparation. Nous nous limiterons au cas de la séparation de parole mono-locuteur, et nous ne traiterons pas de la question de l'adaptation au locuteur.

Notre contribution principale est l'intégration de connaissances spécifique à la voix pour la séparation de la parole. Notre contribution se base sur le modèle IMM [14], qui est un modèle source/filtre de la parole dans le cadre de la factorisation en matrices non-négatives, dans lequel nous proposons l'intégration de contraintes supplémentaires pour la séparation de la parole. La proposition repose sur la volonté de guider la modélisation de la parole par factorisation en matrices non-négatives pour mieux rendre compte de la réalité physique de la voix. Nous faisons l'hypothèse qu'une meilleure modélisation de la parole en permettra une meilleure séparation.

2.1 Apprentissage guidé par la physique

La méthode proposée dans [14] est non-supervisée : le dictionnaire de filtres vocaux, qui est encodée dans la matrice $\boldsymbol{U}^{\Phi}$, se construit lors de la séparation. Il nous apparaît donc vraisemblable que l'apprentissage au préalable des filtres vocaux du locuteur devrait améliorer la qualité de la séparation.

2.1.1 Estimation du filtre par l'algorithme *True Envelope*

La phase d'apprentissage sur une voix non bruitée se fait par NMF en utilisant le modèle décrit en (1.11). Mais il est possible que l'algorithme, puisque non-supervisée, n'effectue pas correctement la répartition des parties sources et filtres aux bonnes matrices. Même si cette possibilité est diminué par l'imposition du dictionnaire d'excitation $\boldsymbol{W}^{\text{ex}}$ et du banc de filtre $\boldsymbol{W}^{\Phi}$, la présence d'excitations bruitées dans $\boldsymbol{W}^{\text{ex}}$ (nécessaire à la bonne reconstruction du signal de voix) peut perturber cette séparation.

Pour s'assurer de la bonne séparation entre source et filtres lors de l'apprentissage, nous utilisons l'algorithme *True Envelope* [25], qui est un algorithme d'estimation d'enveloppe spectrale ;

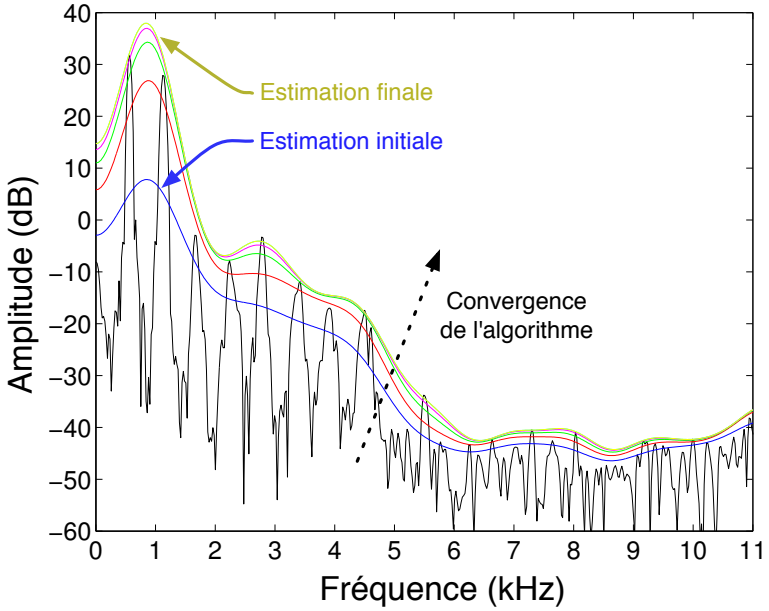


FIGURE 2.1 – Estimation d'une enveloppe spectrale avec l'algorithme *True Envelope* ; le spectre observé est en noir, et les lignes de couleur représentent les différentes itérations de l'estimation.

la FIGURE 2.1 en donne un exemple. Cet algorithme nous permet d'obtenir avec précision le spectrogramme d'amplitude des filtres vocaux, auquel nous pouvons appliquer une NMF utilisant la factorisation

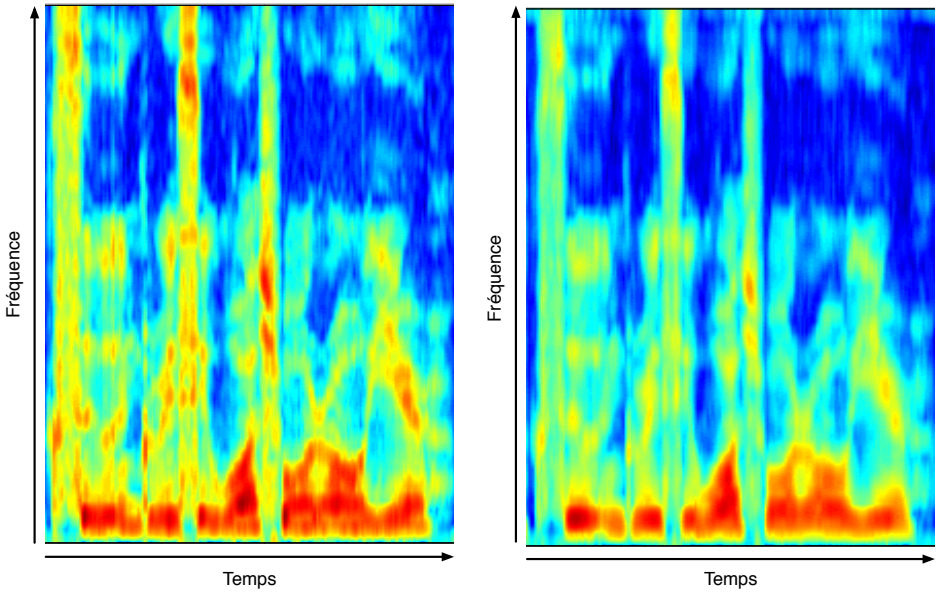
$$\boldsymbol{V}^{\Phi} \simeq \boldsymbol{\Lambda}^{\Phi} = \boldsymbol{W}^{\Phi}\boldsymbol{U}^{\Phi}\boldsymbol{H}^{\Phi} \tag{2.1}$$

en vue d'apprendre la matrice $\boldsymbol{U}^{\Phi}$ pour le locuteur visé ; la matrice $\boldsymbol{W}^{\Phi}$ est construite au préalable et gardée fixe. Les mises à jour multiplicatives de ce modèle sont données en ANNEXE A.2.2.
L'estimation a priori du filtre à partir du signal audio garantit la cohérence de la modélisation source/filtre par la NMF. La FIGURE 2.2 montre les matrices $\boldsymbol{\Lambda}^{\Phi}$ estimées avec et sans l'utilisation de l'algorithme *True Envelope* ; on peut voir que sans celui-ci, les filtres vocaux présentent de nombreux pics fréquentiels, ce qui tend à montrer que l'estimation des filtres vocaux ne se défait pas totalement de l'information venant des excitations.

2.1.2 Informations phonétiques

Dans une tâche visant à séparer deux signaux de paroles dans un signal mono-canal, Schmidt et Olsson [26] comparent l'efficacité de deux types d'apprentissage :

- dans le premier, le dictionnaire d'un locuteur est appris sur l'ensemble de sa base d'apprentissage ;
- dans le second, un petit dictionnaire est appris pour chaque phonème en regroupant toutes ses occurrences.



(a) Sans *True Envelope* (b) Avec *True Envelope*

FIGURE 2.2 – Comparaison des spectrogrammes des filtres $\mathbf{\Lambda}^{\Phi}$ estimées lors de l'apprentissage : (a) sans et (b) avec utilisation de l'algorithme *True Envelope*

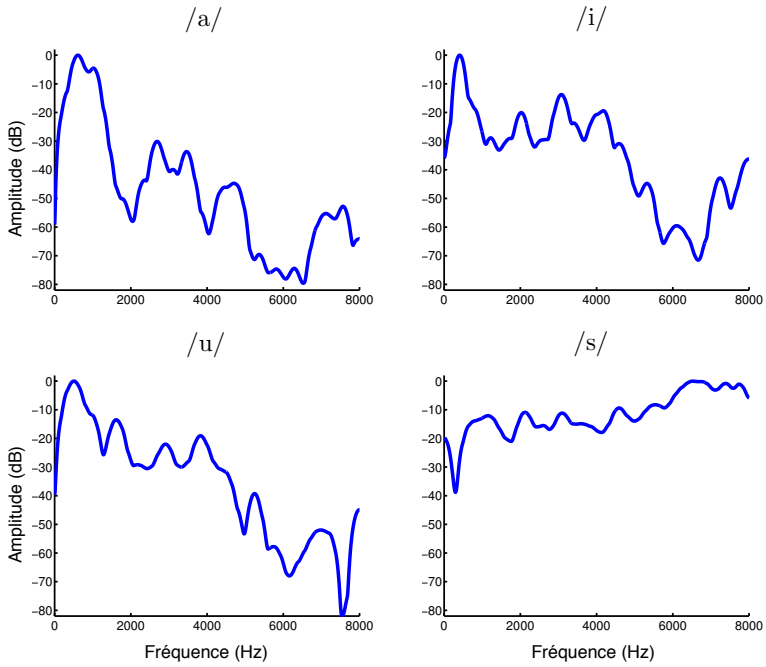


FIGURE 2.3 – Exemples de phonèmes appris à l'aide de la NMF sur le même locuteur : les 3 voyelles /a/, /i/ et /u/, et la chuintante /s/.

Leurs résultats montrent qu'il est plus intéressant d'effectuer un apprentissage distinct pour chaque phonème, ce que nous avons décidé d'appliquer durant notre phase d'apprentissage [5]. Dans notre cas, nous n'apprenons qu'une base par phonème, puisque nous effectuons l'apprentissage directement sur un spectrogramme de filtres vocaux. La FIGURE 2.3 montre quelques exemples de phonèmes appris.
Cet apprentissage se situe dans notre optique d'avoir une méthode traduisant la réalité physique du phénomène vocal puisqu'il nous permet d'être sûr que chaque base encodée dans $\boldsymbol{U}^{\Phi}$ représente un phonème distinct ; de plus, cela nous permet d'assurer une cohérence entre les sources d'excitations et les filtres vocaux (cf. section 2.2.3), et ouvre même la possibilité à la construction d'un modèle de langage (par chaînes de Markov par exemple).

La matrice $\boldsymbol{U}^{\Phi}$ ainsi apprise est utilisée lors de la phase de séparation dans une NMF par modèle IMM, où elle est gardée fixe. On se référera à cette méthode semi-supervisée par S-IMM, et ses règles de mises à jour multiplicatives sont données en ANNEXE A.2.2.

2.2 Intégration de contraintes

Un système source/filtre, de par sa nature, ne combine à un instant qu'une seule source d'excitation avec un seul filtre. Or la NMF permet l'activation simultanée de plusieurs excitations et de plusieurs filtres vocaux [6].
Pour résoudre cette incohérence, Durrieu et *al.* [14] utilisent, après une première NMF avec leur modèle IMM, un algorithme de Viterbi sur $\boldsymbol{H}^{\text{ex}}$ pour n'avoir, à chaque instant, que la source d'excitation la plus probable, avant de ré-estimer les filtres par NMF (toujours utilisant sur le modèle IMM). Cela correspond assez bien à leur tâche, la séparation de la voix ou de l'instrument principal dans un morceau, puisque on peut y voir une recherche de la mélodie dominante la plus probable à partir de $\boldsymbol{H}^{\text{ex}}$, qui serait alors la probabilité d'apparition de chaque fréquence à chaque instant.

Dans ce stage, nous voulons imposer un sens physique à la NMF ; il faut donc que nous soyons capable de forcer une unique activation par trame pour les sources d'excitations et les filtres. Il est possible d'utiliser un algorithme de Viterbi sur la matrice $\boldsymbol{H}^{\text{ex}}$, comme dans [14], et même de l'appliquer à $\boldsymbol{H}^{\Phi}$ en développant une mesure de transition entre phonèmes. Mais il nous a paru plus intéressant d'imposer ce sens physique en intégrant des contraintes à l'intérieur de la NMF, et ce pour plusieurs raisons :

- tout d'abord, cela nous permet d'imposer des contraintes ayant un certain sens physique, et donc de contribuer à une meilleure modélisation de la parole par la NMF ;
- de plus, cela nous permet de nous affranchir des possibles erreurs de l'algorithme de Viterbi ; il nous parait en effet intéressant de limiter le nombre de méthodes différentes en interaction pour éviter une accumulation des erreurs.

Mais l'utilisation de contraintes pose aussi des difficultés, comme le choix du poids à leur donner ou leur interaction avec la normalisation des matrices. Par souci de comparaison, nous avons

5. Dans le cadre du stage, nous avons pu travailler avec une base de données annotée phonétiquement. Dans le cas contraire, il faudrait, avant l'apprentissage, intégrer une méthode de reconnaissance de la parole ; malheureusement, les potentiels erreurs d'une méthode de ce type se répercuteraient alors sur l'efficacité de l'algorithme proposé.

6. Étant donné l'utilisation des mises à jour multiplicatives et le fait que les matrices ne sont pas initialisées avec des valeurs nulles, on peut même préciser que toutes les sources sont actives à tout moment, mais avec des gains différents.

intégré à notre méthode la possiblité d'utiliser Viterbi, sans pour autant approfondir son optimisation.
Il est à noter que la volonté de donner une direction physique à la NMF diffère de son fonctionnement classique, qui cherche à minimiser le coût de reconstruction sans s'intéresser au sens de celle-ci. Nous faisons l'hypothèse qu'en imposant un sens physique aux dictionnaires et aux activations, on améliorera sensiblement les résultats de la séparation.

2.2.1 Contraintes et mises à jour multiplicatives

De nombreux travaux sur la NMF concernent l'intégration et l'utilisation de contraintes pour orienter le comportement des dictionnaires ou des activations vers une distribution voulue. Cela se fait par le rajout d'une pénalité dans la fonction de coût :

$$\mathcal{C} = D_\beta(\boldsymbol{V}|\boldsymbol{\Lambda}) + \mu\mathcal{P}(\boldsymbol{\Theta}) \tag{2.2}$$

où D_β est la β-divergence matricielle, $\boldsymbol{V}$ et $\boldsymbol{\Lambda}$ respectivement les matrices observée et estimée, $\mathcal{P}$ est la pénalité que l'on veut appliquer,$\boldsymbol{\Theta}$ est la matrice sur laquelle on veut appliquer la contrainte (e.g. dans le cas du modèle IMM, $\boldsymbol{\Theta}$ peut représenter $\boldsymbol{H}^{\text{ex}}$,$\boldsymbol{U}^{\Phi}$,$\boldsymbol{H}^{\Phi}$,$\boldsymbol{W}^{N}$ ou $\boldsymbol{H}^{N}$), et μ est un coefficient réel positif permettant de contrôler le poids donnée à la contrainte. En suivant (1.9), on peut retrouver des règles de mises à jour multiplicatives :

$$\boldsymbol{\Theta}^{(i+1)} \longleftarrow \boldsymbol{\Theta}^{(i)} \otimes \frac{\nabla^{-}_{\boldsymbol{\Theta}} D_\beta + \mu\nabla^{-}_{\boldsymbol{\Theta}}\mathcal{P}}{\nabla^{+}_{\boldsymbol{\Theta}} D_\beta + \mu\nabla^{+}_{\boldsymbol{\Theta}}\mathcal{P}} \tag{2.3}$$

Il n'y pas de preuve de la non-croissance de $\mathcal{C}$ par itération de (2.3), mais en pratique la convergence est toujours observée tant que μ n'est pas excessivement grand. Toutes les formules de gradients des contraintes décrites ci-dessous sont données en Annexe A.3.

Comme nous l'avons mentionné, il est habituel de normaliser les bases des matrices dictionnaires $\boldsymbol{W}$ pour que les activations temporelles $\boldsymbol{H}$ représentent l'énergie du système. Puisque ces matrices évoluent à chaque itération, cette normalisation se fait après chaque mise à jour, et comme ce transfert d'énergie ne change pas la matrice estimée $\boldsymbol{\Lambda}$, le coût de reconstruction n'est pas affecté. Mais la normalisation peut affecter la valeur des contraintes, et donc modifier le coût total entre deux mises à jour, ce qui compromet la convergence de l'algorithme. Pour remédier à cela, il est possible de changer le coût utilisé pour prendre en compte la normalisation, comme le proposent Eggert et Körner [27] dans le cas de l'utilisation d'une contrainte de parcimonie.
Dans notre cas, ce problème est aisément évitable. En effet, puisque nos matrices $\boldsymbol{W}^{\Phi}$, $\boldsymbol{U}^{\Phi}$ et $\boldsymbol{W}^{\text{ex}}$ sont apprises et fixes, il n'est pas nécessaire de les normaliser à chaque itération, mais seulement antérieurement et postérieurement à la NMF. De plus, nous ne cherchons pas à appliquer de contraintes sur les matrices $\boldsymbol{W}^{N}$ et $\boldsymbol{H}^{N}$ représentant le bruit, ce qui nous laisse libre pour la normalisation de $\boldsymbol{W}^{N}$.

2.2.2 Contraintes de la littérature

Dans cette partie, nous allons décrire plusieurs contraintes de la littérature, et expliquer en quoi elles peuvent nous aider à imposer un sens physique au modèle S-IMM. Nous ne nous intéressons qu'à l'ajout de contraintes sur les matrices d'activations $\boldsymbol{H}^{\text{ex}}$ et $\boldsymbol{H}^{\Phi}$, et seulement lors de la phase de séparation.

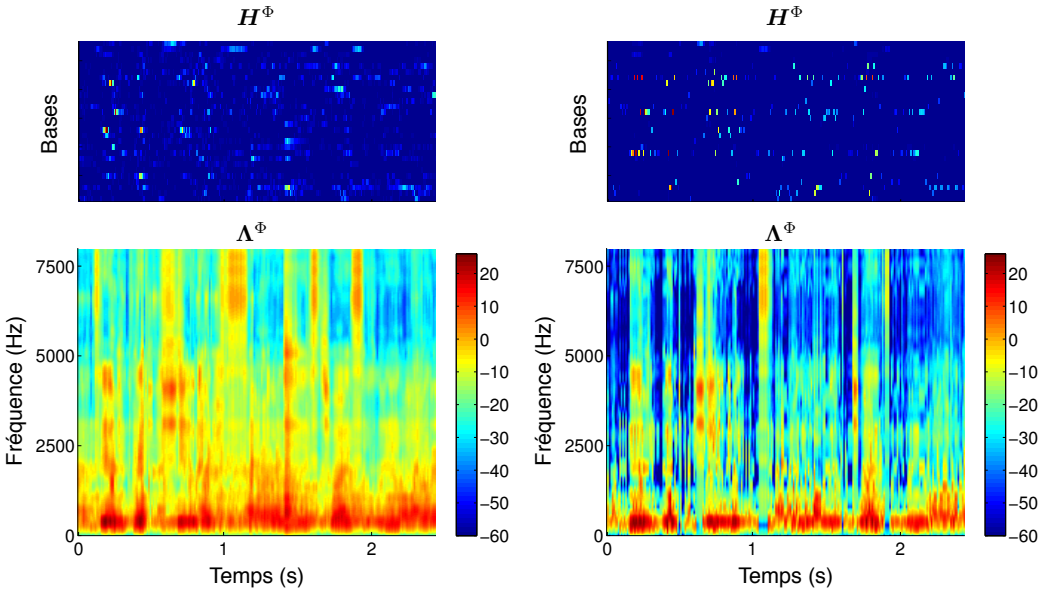
Contrainte de parcimonie La parcimonie peut être définie, pour une matrice ou un vecteur, comme le fait d'être « souvent nul » [6] : plus un vecteur tend vers le fait de ne posséder qu'un seul coefficient non-nul, plus il est considéré comme parcimonieux. Nous souhaitons utiliser la contrainte de parcimonie pour assurer, à chaque trame n, l'activation d'un seul filtre dans $\boldsymbol{H}_n^{\Phi}$ et d'une seule excitation dans $\boldsymbol{H}_n^{\text{ex}}$.
Joder et *al.* [28] proposent plusieurs mesures de parcimonie pour une matrice. Nous avons décidé d'utiliser la suivante :

$$\mathcal{P}_p(\boldsymbol{H}) = \sum_{n=1}^{N} \frac{\sum_{k=1}^{K} \boldsymbol{H}_{kn}}{\sqrt{\sum_{k=1}^{K} \boldsymbol{H}_{kn}^2}} \tag{2.4}$$

Ce choix repose sur plusieurs critères :

- Ce coût mesure la parcimonie des colonnes de $\boldsymbol{H}$. Or nous cherchons à imposer, dans une trame, une seule activation (des excitations ou des filtres) ; cela revient donc à vouloir chaque colonne $\boldsymbol{H}_n$ le plus parcimonieuse possible.
- Comme la β-divergence matricielle, cette mesure est dépendante de la taille de la matrice.
- De plus, cette mesure ne dépend que de la valeur relative des coefficients entre eux (grâce à la division par la norme ℓ_2 de $\boldsymbol{H}_n$).

La FIGURE 2.4 montre un exemple de l'utilisation de cette parcimonie sur la matrice d'activations des filtres vocaux $\boldsymbol{H}^{\Phi}$. On remarque bien une forte tendance à une activation unique (ou du moins très supérieur aux autres en terme d'ordre de grandeur) à chaque instant ; cela se traduit par un spectrogramme estimé des filtres $\boldsymbol{\Lambda}^{\Phi}$ moins large bande et, on l'espère, plus spécifique à la voix.



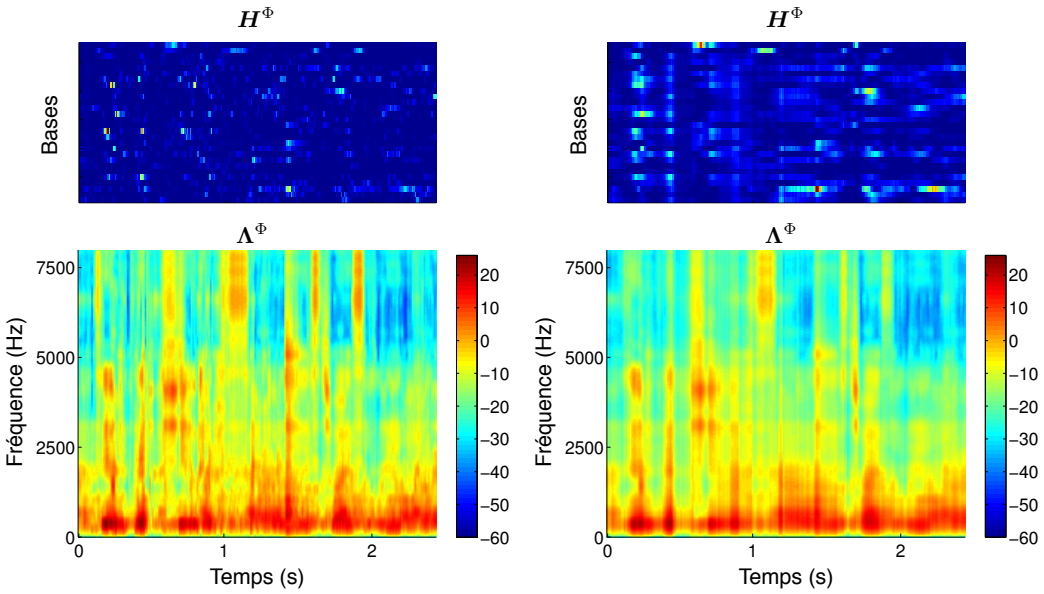
(a) Sans parcimonie (b) Avec parcimonie

FIGURE 2.4 – Comparaison des matrices d'activations $\boldsymbol{H}^{\Phi}$ et des spectrogrammes des filtres vocaux $\boldsymbol{\Lambda}^{\Phi}$ obtenus par S-IMM sur un signal de parole bruitée : (a) sans et (b) avec contrainte de parcimonie sur $\boldsymbol{H}^{\Phi}$. Le coefficient μ_p pour la parcimonie vaut 1.

Contrainte de continuité temporelle La NMF, dans sa formulation classique, ne prend pas en compte l'ordonnancement des colonnes $\boldsymbol{V}_n$ de la matrice observée, i.e. ne considère pas la temporalité des données. Dans les applications audios de la NMF, on trouve donc souvent l'utilisation d'une contrainte qui permet d'assurer que H_k, l'activation de la base k, présente une certaine continuité temporelle. Pour la séparation de la parole, cette contrainte nous permet d'empêcher l'apparition et/ou disparition trop brusque d'un phonème ; il faut toutefois faire attention à ne pas vouloir imposer une continuité trop forte qui forcerait les phonèmes naturellement courts (comme les consonnes plosives) à perdurer dans le temps.
Nous avons choisi d'utiliser la mesure proposée par Virtanen [29] :

$$\mathcal{P}_c(\boldsymbol{H}) = \sum_{k=1}^{K} \frac{1}{\sigma_k} \sum_{n=2}^{N} \left(\boldsymbol{H}_{kn} - \boldsymbol{H}_{kn-1}\right)^2 \quad \text{avec } \sigma_k = \sqrt{\frac{1}{N}\sum_{n=1}^{N} \boldsymbol{H}_{kn}^2} \tag{2.5}$$

La FIGURE 2.5 montre l'effet de la continuité temporelle sur la matrice d'activations des filtres vocaux $\boldsymbol{H}^{\Phi}$. On observe un étalement temporel des activations, qui se transmet à $\boldsymbol{\Lambda}^{\Phi}$ et rend son évolution plus régulière.



(a) Sans continuité temporelle (b) Avec continuité temporelle

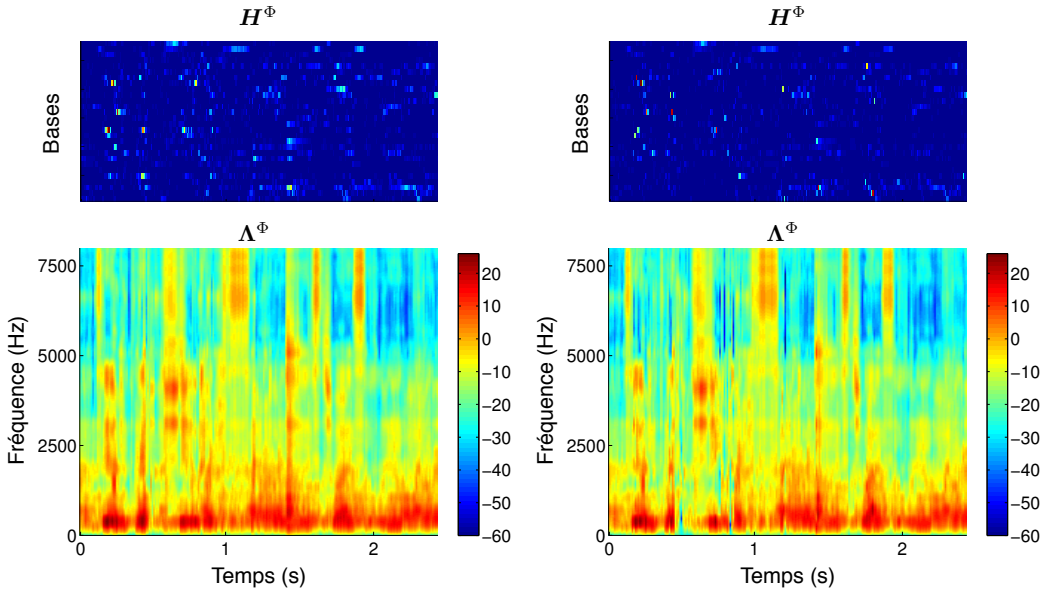
FIGURE 2.5 – Comparaison des matrices d'activations $\boldsymbol{H}^{\Phi}$ et des spectrogrammes des filtres vocaux $\boldsymbol{\Lambda}^{\Phi}$ obtenus par S-IMM sur un signal de parole bruitée : (a) sans et (b) avec contrainte de continuité temporelle sur $\boldsymbol{H}^{\Phi}$. Le coefficient μ_c pour la continuité vaut 0.5.

Contrainte de décorrélation Pour s'assurer que les bases apprises par la NMF ne correspondent pas à deux parties distinctes d'un même phénomène physique, il est possible d'imposer une contrainte de décorrélation entre les lignes de $\boldsymbol{H}$. Bertin [6] propose une mesure basée sur la matrice de corrélation $\boldsymbol{H}\boldsymbol{H}^T$; à partir de cela, et en s'inspirant de (2.4), nous proposons une mesure de décorrélation normalisée (i.e. dont la valeur ne change pas si les rapports d'amplitude

entre les lignes H_k changent) :

$$\mathcal{P}_d(\boldsymbol{H}) = \sum_{k=1}^{K} \sum_{\substack{l=1 \\ l \neq k}}^{K} \frac{\sum_{n=1}^{N} \boldsymbol{H}_{kn} \boldsymbol{H}_{ln}}{\sqrt{\sum_{n=1}^{N} \boldsymbol{H}_{kn}^2} \sqrt{\sum_{n=1}^{N} \boldsymbol{H}_{ln}^2}} \tag{2.6}$$

Les termes du gradient à ajouter aux formules de mises à jour sont donnés en ANNEXE A.3.3. On peut voir dans la FIGURE 2.6 que la contrainte de déccorélation proposé n'a pas un effet aussi important sur $\boldsymbol{H}^{\Phi}$ (et donc $\boldsymbol{\Lambda}^{\Phi}$) que les contraintes de parcimonie ou de continuité. En effet, la contrainte de décorrélation est utilisé pour assurer un bon apprentissage des bases de la NMF ; or, lors de la séparation, nos bases sont déjà connues et fixes. Nous utilisons ici cette contrainte pour assurer la suppression d'activations ayant des évolutions temporelles similaires, même avec des échelles d'amplitudes différentes.



(a) Sans décorrélation (b) Avec décorrélation

FIGURE 2.6 – Comparaison des matrices d'activations $\boldsymbol{H}^{\Phi}$ et des spectrogrammes des filtres vocaux $\boldsymbol{\Lambda}^{\Phi}$ obtenus par S-IMM sur un signal de parole bruitée : (a) sans et (b) avec contrainte de décorrélation sur $\boldsymbol{H}^{\Phi}$. Le coefficient μ_d pour la parcimonie vaut 0.5.

2.2.3 Contrainte de cohérence source/filtre

En linguistique, on distingue deux grands type de phonèmes :

- les phonèmes voisés, qui correspondent à une excitation harmonique venant des plis vocaux ; ce sont les voyelles, les semi-voyelles (/j/, /w/), les nasales (/m/, /n/), la liquide /l/, certaines occlusives (/b/, /d/ et /g/) et certaines fricatives /v/, /z/ et /ʒ/).
- les phonèmes non-voisés, qui correspondent à une excitation bruitée ; ce sont certaines occlusives (/p/, /t/ et /k/) et certaines fricatives (/f/, /s/, /ʃ/ et /r/).

Au cours d'expériences préliminaires, nous avons pu observer que des filtres vocaux associés à des phonèmes voisés étaient activées en même temps que des excitations bruitées, et de même avec des filtres vocaux de phonèmes non-voisés et des excitations périodiques. Cela résultait en la présence d'artefacts très audibles, et baissait la qualité de la reconstruction.
Puisque nous connaissons, pour chaque base de $\boldsymbol{W}^{\Phi}$, le phonème correspondant, nous avons pensé à créer une contrainte de cohérence source/filtre pour empêcher l'apparition de ce phénomène.

Le principe de cette contrainte est d'empêcher les activations simultanées de certaines lignes de $\boldsymbol{H}^{\text{ex}}$ et $\boldsymbol{H}^{\Phi}$; nous avons donc pensé à utiliser une formule basée sur celle de la contrainte de décorrélation (2.6). Notons $\mathcal{V}$ l'ensemble des indices des bases représentant des phonèmes voisés et $\mathcal{H}$ l'ensemble des indices des bases d'excitations harmoniques. Notre contrainte s'écrit alors :

$$\mathcal{P}_{\phi}(\boldsymbol{H}^{\text{ex}}, \boldsymbol{H}^{\Phi}) = \sum_{\substack{k=1 \\ l\in\mathcal{H}}}^{K_e} \sum_{\substack{l=1 \\ l\notin\mathcal{V}}}^{K_f} \frac{\sum_{n=1}^{N} \boldsymbol{H}^{\text{ex}}_{kn} \boldsymbol{H}^{\Phi}_{ln}}{\sqrt{\sum_{n=1}^{N} {\boldsymbol{H}^{\text{ex}}_{kn}}^2} \sqrt{\sum_{n=1}^{N} {\boldsymbol{H}^{\Phi}_{ln}}^2}} + \sum_{\substack{k=1 \\ l\notin\mathcal{H}}}^{K_e} \sum_{\substack{l=1 \\ l\in\mathcal{V}}}^{K_f} \frac{\sum_{n=1}^{N} \boldsymbol{H}^{\text{ex}}_{kn} \boldsymbol{H}^{\Phi}_{ln}}{\sqrt{\sum_{n=1}^{N} {\boldsymbol{H}^{\text{ex}}_{kn}}^2} \sqrt{\sum_{n=1}^{N} {\boldsymbol{H}^{\Phi}_{ln}}^2}} \tag{2.7}$$

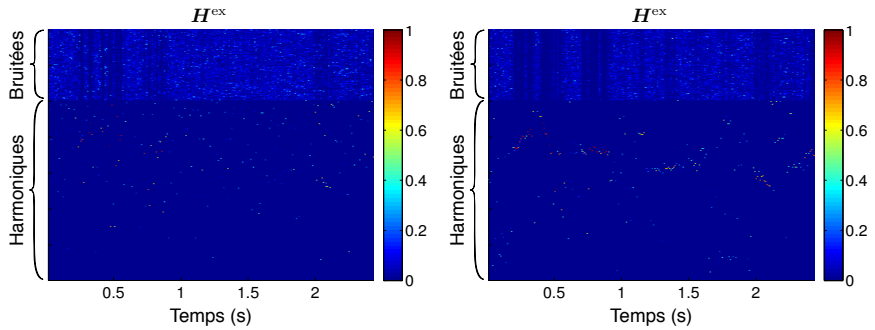
Les termes du gradient à ajouter aux formules de mises à jour sont donnés en Annexe A.3.4.

2.2.4 Contraintes adaptatives

L'effet des contraintes dépend d'un poids μ, comme introduit dans (2.2) : un facteur trop petit rend celle-ci sans effet, tandis qu'un facteur trop grand enlève l'importance donnée à la reconstruction de $\boldsymbol{V}$ et contraint trop la mise à jour dès la première itération. Cela donne lieu à des reconstructions très sensibles à l'initialisation aléatoire, ce qui peut pousser la mise à jour dans une mauvaise direction.
Nous avons donc pensé à développer une méthode de poids adaptatif qui privilégierait la β-divergence, et ne permettrait un effet des contraintes que lorsque cela ne puisse pas nuire à l'estimation. L'idée en est simple : on décide d'une valeur maximale μ_M pour le coefficient, et à chaque itération, le coefficient pour la prochaine itération est calculé selon l'évolution de la β-divergence par rapport à l'itération précédente :

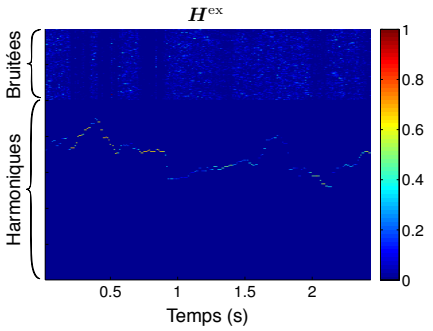
$$\mu^{(i)} = \begin{cases} 0 & \text{pour } i = \{1, 2\} \\ \mu_M \dfrac{D_\beta(\boldsymbol{V}|\boldsymbol{\Lambda}^{(i-2)}) - D_\beta(\boldsymbol{V}|\boldsymbol{\Lambda}^{(i-1)})}{D_\beta(\boldsymbol{V}|\boldsymbol{\Lambda}^{(i-2)})} & \text{pour } i \geq 3 \end{cases} \tag{2.8}$$

Avec cette méthode, plus l'estimation se précise (et donc l'évolution de D_β diminue), plus les contraintes prennent du poids et influent sur la mise à jour pour amener à une reconstruction ayant un sens physique.
La Figure 2.7 montre un exemple de l'effet de cette méthode d'adaptation dans le cas d'une contrainte de parcimonie sur la matrice des excitations $\boldsymbol{H}^{\text{ex}}$. On peut y voir que sans cette méthode, le « ligne mélodique » de la voix (visible sur l'exemple utilisant Viterbi, donné comme comparaison) est très mal estimé dans $\boldsymbol{H}^{\text{ex}}$, tandis qu'on la retrouve assez clairement lorsque le poids est adaptatif.



(a) Avec parcimonie et μ_p fixe

(b) Avec parcimonie et μ_p adaptatif



(c) Avec Viterbi

FIGURE 2.7 – Comparaison des $\boldsymbol{H}^{\text{ex}}$ obtenues par S-IMM sur un signal de parole bruitée : (a) avec parcimonie dont le poids μ_p est fixe, (b) avec parcimonie dont le poids μ_p est adaptatif et (c) avec un Viterbi postérieur à la NMF. La parcimonie et l'algorithme de Viterbi sont appliqués seulement sur les excitations harmoniques. Le coefficient μ_p pour la parcimonie vaut 5.

2.3 Structure de l'algorithme

La structure de notre algorithme est très similaire à celle usuelle pour la séparation de source, décrite dans la section 1.1.3 :

1. Apprentissage des filtres vocaux du locuteur par NMF (cf. section 2.1) ;
2. Séparation par NMF semi-supervisée, avec possibilité d'utilisation de contraintes (cf. section 2.2) ;
3. Extraction de la voix par filtrage de Wiener.

Nous avons de plus une variante utilisant un algorithme Viterbi dans le but de comparer son utilisation avec notre méthode basée sur les contraintes pour la séparation de la parole. Cet algorithme de Viterbi est applicable aux excitations harmoniques et/ou aux filtres. Pour les excitations, la probabilité de transition est le logarithme de la distance en fréquence entre les notes représentées bases ; pour les filtres, nous avons utilisé le log de la valeur de la β-divergence. La structure globale de notre algorithme S-IMM est décrite dans la FIGURE 2.8. On se référera à la variante utilisant les contraintes par SC-IMM et à la variante utilisant Viterbi par S-IMM-V.

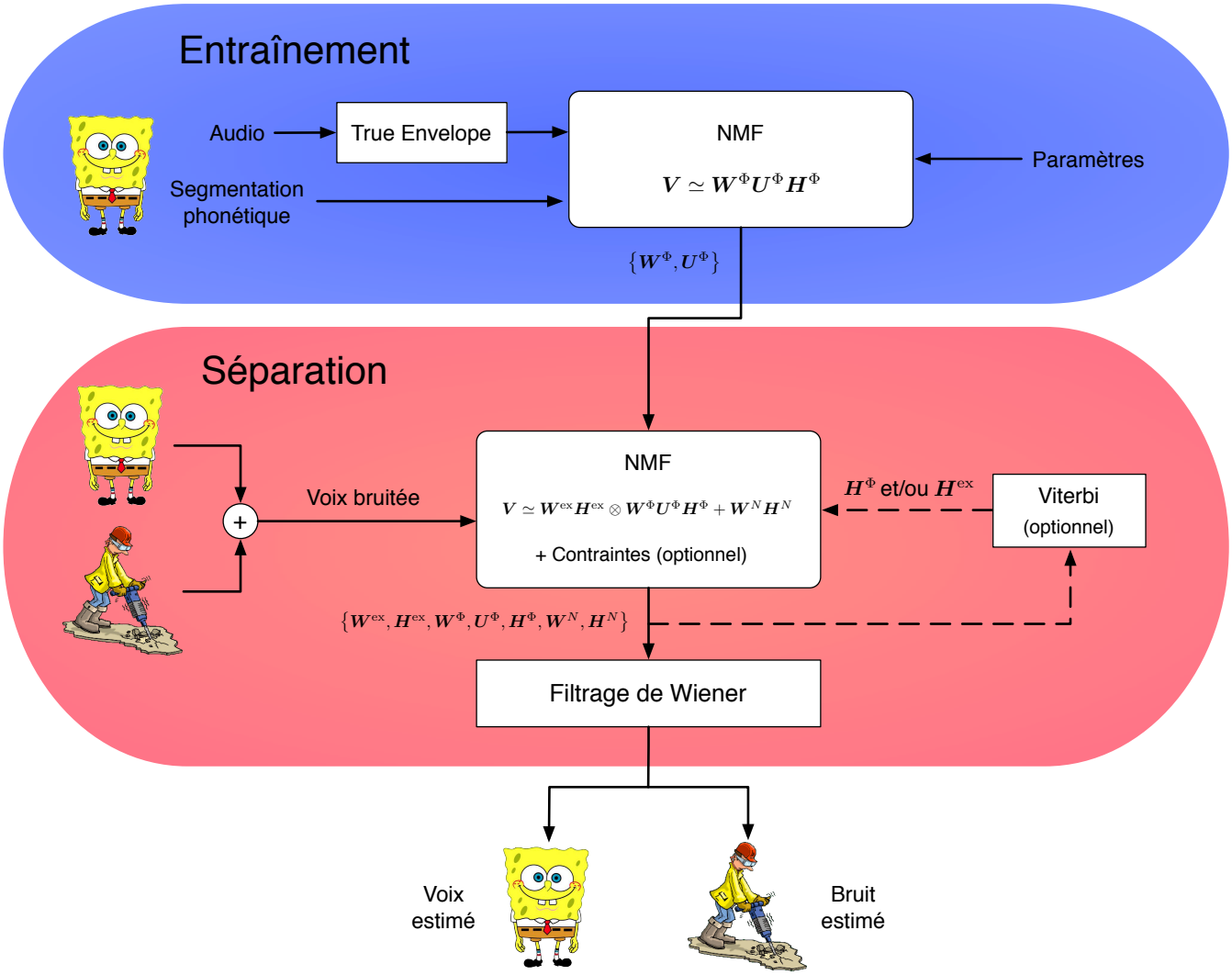


FIGURE 2.8 – Structure générale de l'algorithme proposé.

Chapitre 3

Évaluation expérimentale

Dans ce chapitre, nous évaluons l'algorithme proposé sur une tâche de séparation de la parole dans un environnement bruité.

3.1 Bases de données

Afin d'évaluer la tâche proposé, nous avons construit une base de données de signaux de paroles bruitées. Pour cela, nous avons utilisé comme base de parole TIMIT [30], et comme base de bruit QUT-NOISE [31].

TIMIT est une base de données comprenant 3600 phrases en anglais-américain, divisées en 360 locuteurs différents. Les enregistrements sont à 16kHz et sont tous pourvus d'une segmentation en phonèmes de référence.
Chaque locuteur a enregistré 10 phrases, dont 2 sont identiques pour tous ; ces 2 phrases exposent toutes les variantes de prononciations des locuteurs, et nous serviront de base d'apprentissage. Les 8 phrases restantes nous serviront de base de test.

QUT-NOISE contient 20 enregistrements de 30mn de différents environnements sonores (cuisine, rue, café, intérieur d'une voiture, etc). Ces enregistrements ont été faits à 48kHz ; nous les avons sous-échantillonnés à 16kHz pour rendre possible la création de signaux de parole bruités. L'algorithme décrit dans [32], que nous utilisons comme référence, nécessite un apprentissage du bruit. Nous avons pour cela utilisé les 5 premières minutes des exemples sonores ; les extraits nécessaires à la création de notre base de test ont été pris aléatoirement dans les 25 minutes restantes.

Notre base de test est composé de :

- 20 locuteurs, 10 hommes et 10 femmes (leurs identifiants TIMIT sont `FBCG1`, `FSDJ0`, `FCJF0`, `FDAW0`, `FDML0`, `FECD0`, `FETB0`, `FMBG0`, `FHEW0`, `FJXM0`, `MKLS0`, `MJEB1`, `MGRL0`, `MMRP0`, `MPGH0`, `MPDF0`, `MRAI0`, `MBWP0`, `MMGK0` et `MTRR0`).
- 5 types de bruits : 4 venants de QUT-NOISE-TIMIT (nommés `STREET-CITY-1`, `CAFE-CAFE-1`, `HOME-KITCHEN-1` et `CAR-WINDOWNB-1`) et du bruit blanc ;
- 3 ratios signal-sur-bruit (SNR, pour *Signal-to-Noise Ratio*) différents : -6dB, $+0$dB et $+3$dB.

En utilisant les 8 phrases de test par locuteur, nous obtenons un total de 2400 fichiers de tests.

3.2 Algorithmes et références

Afin de pouvoir correctement estimer le fonctionnement de notre algorithme pour la tâche définie, nous comparons les résultats obtenus avec ceux de deux algorithmes de référence, que nous avons testés sur la même base de données :

- l'algorithme V-IMM de Durrieu et *al.* [14] [7], qui est une méthode non supervisée destinée à la base à la séparation de la partie principale d'un morceau (voix ou instrument).
- l'algorithme ASNA de Virtanen et *al.* [32] [8], qui utilise une version modifiée de l'algorithme de Newton pour obtenir une représentation non-négative parcimonieuse du signal audio. Cet algorithme supervisé nécessite une phase d'apprentissage de la voix et du bruit. Nous avons choisi cette méthode puisque son principe est proche de celui de la NMF et car elle est citée comme une méthode de référence pour la séparation de la parole dans [2] et [3].

De plus, nous avons testé plusieurs combinaisons des paramètres de notre algorithme ; l'ensemble des méthodes au banc d'essai est décrit par le Tableau 3.1.

	Modèle	Apprentissage		Contraintes		Viterbi	
	source/filtre	Parole	Bruit	Littérature	Cohérence	$\boldsymbol{H}^{\text{ex}}$	$\boldsymbol{H}^{\Phi}$
ASNA		✓	✓				
V-IMM	✓				✓		
S-IMM	✓	✓					
SC-IMM1	✓	✓		✓			
SC-IMM2	✓	✓			✓		
SC-IMM3	✓	✓		✓	✓		
S-IMM-V1	✓	✓				✓	
S-IMM-V2	✓	✓					✓
S-IMM-V3	✓	✓				✓	✓

Tableau 3.1 – Algorithmes au banc d'essai. Les contraintes de la littérature désignent les contraintes de parcimonie, de continuité temporelle et de décorrélation.

3.3 Protocoles expérimentaux

Nous mesurons l'efficacité des algorithmes à l'aide des mesures suivantes :

- le SDR, le SIR et le SAR, définies dans [33] [9] :
  - le SIR (*Signal-to-Interference Ratio*) quantifie, dans le signal reconstitué, le rapport en décibels entre le signal utile et les interférences venues des autres sources.
  - le SAR (*Signal-to-Artifact Ratio*) quantifie, dans le signal reconstitué, le rapport en décibels entre le signal utile et les artefacts créés par l'algorithme.

7. Disponible en Python sur `http://durrieu.ch/phd/software.html`
8. Disponible en MatLab® sur `http://www.cs.tut.fi/ tuomasv/software.html`
9. Une implémentation par les auteurs est disponible en MatLab® sur `http://bass-db.gforge.inria.fr/bss_eval/`

  - le SDR (*Signal-to-Distortion Ratio*) quantifie, dans le signal reconstitué, le rapport en décibels entre le signal utile et la somme des artefacts et des interférences ; il englobe le SIR et le SAR et caractérise la qualité de la séparation.
- le PESQ (*Perceptual Evaluation of Speech Quality*) proposé dans [34] [10] ; le PESQ est un critère MOS (*Mean Opinion Score*, ou Note d'opinion moyenne) qui caractérise la qualité de la restitution sonore du signal de voix reconstruit ; il varie entre 1 (très mauvais) et 5 (excellent, comparable à la version d'origine). Le PESQ a été développée originellement pour l'évaluation des réseaux téléphoniques et des codecs audio.

Les algorithmes de référence, ASNA et V-IMM, ont été testées avec les paramètres par défaut donnés par les auteurs, supposés optimaux, sauf le nombre de bases de bruit pour V-IMM et le nombre de bases de parole et de bases de bruit pour ASNA, dont nous avons pris les valeurs maximisant le SDR.
L'algorithme V-IMM a donc été utilisé avec 40 bases de bruit, 50 bases pour les filtres vocaux, une taille de fenêtre de 46.4ms et un décalage de 5.8ms, et utilise pour coût la divergence IS.
L'algorithme ASNA a lui été utilisé avec les paramètres suivants : 200 bases de parole, 1000 bases de bruit [11], une taille de fenêtre de 60ms et un décalage de 15ms.

Pour l'algorithme proposé, nous avons choisi une taille de fenêtre de 64ms, un décalage de 32ms et le nombre maximum d'itérations est fixé à 100. Nous utilisons la divergence IS, avec un palier de bruit à -60dB (cf. section 1.1.1 et [8]), et fixons le paramètre de puissance des spectres à 1 (cf. (1.2)). Le banc de filtre $\boldsymbol{W}^{\Phi}$ est composé de 50 fenêtres de Hann espacées linéairement de 0 à 8000 Hz (la fréquence d'échantillonnage étant 16kHz), et le dictionnaire d'excitations $\boldsymbol{W}^{\text{ex}}$ comprend 350 bases, divisées entre :

- 250 bases périodiques (raies harmoniques plates) espacées tout les dixièmes de demi-tons entre 80 et 350Hz.
- 100 bases bruitées, construites à partir de bruit blanc.

	$\boldsymbol{H}^{\text{ex}}$	$\boldsymbol{H}^{\Phi}$
Parcimonie	1	1
Continuité	0	0.5
Décorrélation	0	0.5
Cohérence	0.5	

TABLEAU 3.2 – Tableau de valeurs des coefficients de poids μ des contraintes

Pour les contraintes issues de la littérature, nous appliquons seulement la parcimonie sur la matrice d'activation des excitations $\boldsymbol{H}^{\text{ex}}$, tandis que nous appliquons les 3 contraintes (parcimonie, continuité et décorrélation) sur $\boldsymbol{H}^{\Phi}$. Les valeurs à donner aux poids μ des contraintes ont été testées entre $\left[10^{-3}\ 10^{3}\right]$; les coefficients optimaux utilisés (en terme de SDR) sont donnés dans le TABLEAU 3.2.
Le nombre de bases de bruit K_n et l'utilisation ou non de la méthode d'adaptation du poids des contraintes (cf. section 2.2.4) sont des paramètres laissés libres testés avec les différentes méthodes décrites dans le TABLEAU 3.1.

10. Nous avons utilisé l'implémentation MatLab® disponible sur
`http://www.mathworks.com/matlabcentral/fileexchange/30815-phase-spectrum-compensation/content/psc/pesq.m`
11. Le nombre de bases peut sembler important par rapport aux autres algorithmes, mais cette différence est normal puisque l'algorithme ASNA n'utilise pas de modèle source/filtre pour la représentation de la voix et que les bases, de parole et de bruit, sont fixes une fois apprises.

3.4 Résultats

Dans un premier temps, nous comparons les résultats de l'algorithme proposé de référence (S-IMM) pour différentes valeurs K_n du nombre de bases de bruit. On voit Tableau 3.3 que le K_n optimal est très dépendant du SNR : un mélange favorisant la voix nécessitera moins de bases de bruit qu'un mélange où la parole est noyé dans le fond sonore. On peut tenter d'expliquer ce phénomène par le fait qu'augmenter K_n permet au bruit de représenter plus de phénomènes, et donc possiblement de la voix, ce qui sera d'autant enclin à arriver si le signal contient de base peu de bruit.
Par la suite, les algorithmes proposés utiliseront 25 bases de bruit, valeur qui apparaît être globalement optimal.

SNR	Mesure	K_n						
		5	10	15	20	25	50	100
−6dB	SDR (dB)	2.0	2.7	3.2	3.7	4.0	5.0	**5.1**
	PESQ	1.77	1.82	1.86	1.89	1.91	**1.94**	1.86
+0dB	SDR (dB)	7.9	8.4	8.8	9.0	**9.1**	**9.1**	7.9
	PESQ	2.20	2.25	2.27	2.29	**2.30**	2.26	2.11
+6dB	SDR (dB)	**13.4**	**13.4**	**13.4**	13.2	13.0	11.6	9.1
	PESQ	2.61	2.64	**2.65**	2.64	2.62	2.49	2.26
Moyenne	SDR (dB)	7.7	8.1	8.4	8.6	**8.7**	8.6	7.4
	PESQ	2.19	2.24	2.26	2.27	**2.28**	2.23	2.08

Tableau 3.3 – Résultats pour l'algorithme S-IMM avec différentes valeurs de K_n. Les termes en gras représentent le meilleure score pour un SNR donné.

SNR	Mesure	Avec adaptation			Sans adaptation		
		SC-IMM1	SC-IMM2	SC-IMM3	SC-IMM1	SC-IMM2	SC-IMM3
−6dB	SDR (dB)	4.1	**5.2**	**5.4**	4.1	5.0	5.2
	PESQ	1.91	**2.01**	**2.01**	1.91	1.94	1.92
+0dB	SDR (dB)	9.2	**9.8**	**9.8**	9.2	9.0	8.9
	PESQ	2.30	**2.34**	**2.35**	2.30	2.24	2.23
+6dB	SDR (dB)	**13.0**	**12.8**	**12.9**	12.8	11.1	10.9
	PESQ	**2.62**	**2.59**	**2.62**	2.61	2.46	2.44
Moyenne	SDR (dB)	8.7	**9.3**	**9.4**	8.7	8.4	8.3
	PESQ	**2.28**	**2.31**	**2.33**	2.27	2.21	2.20

Tableau 3.4 – Résultats pour les algorithmes SC-IMM1, SC-IMM2 et SC-IMM3, avec et sans adaptation du poids μ des contraintes. Les termes en gras marquent les scores où la méthode d'adaptation apporte une amélioration.

Le Tableau 3.4 montre les apports de l'utilisation de la méthode d'adaptation du poids μ des contraintes. On remarque immédiatement que cette méthode contribue à l'amélioration des scores dans presque tous les cas, mais aussi qu'elle ne les fait en aucun cas baisser. On peut voir

que ce dispositif est très peu efficace lorsque la contrainte de cohérence n'est pas utilisée (pour SC-IMM1), et que l'amélioration apportée augmente avec le SNR.
Par la suite, la méthode d'adaptation du poids des contraintes sera toujours utilisée (lorsque des contraintes sont intégrées au modèle).

Enfin, le Tableau 3.5 montre l'ensemble des résultats pour les algorithmes du banc d'essai décrit dans le Tableau 3.1.
On constate que les résultats de l'algorithme de référence S-IMM se situent entre ceux de ASNA et V-IMM, les deux algorithmes de la littérature ; notre méthode étant semi-supervisée (nous n'apprenons qu'un modèle de voix), il est naturel qu'elle soit plus efficace que la méthode V-IMM non supervisée et obtiennent de moins bons résultats que la méthode ASNA, qui a de plus appris un modèle de bruit. On remarque de plus un phénomène de vases communicants entre le SIR et le SAR : quand l'un augmente, le second diminue et inversement. Cela peut s'expliquer par le fait qu'un algorithme très bon pour éliminer le bruit va généralement le faire en dénaturant la voix, et donc créer des artefacts, tandis qu'un modèle qui essaye de respecter au mieux les signaux présents aura plus de mal à les séparer.
Ces résultats permettent d'observer que l'intégration des contraintes issues de la littérature (SC-IMM1) ont un apport négligeable ; la contrainte de cohérence (SC-IMM2) apportent quant à elle une nette amélioration des scores pour de faibles SNR, et sa contribution est négligeable, voire néfaste, à +6dB. La combinaison des deux types de contraintes (SC-IMM3) accentue l'effet bénéfique visible pour la cohérence, et de tous les algorithmes proposés, offre les meilleurs scores en terme de SDR et de PESQ.
L'utilisation de l'algorithme de Viterbi dans notre système ne semble pas être efficace. À part à -6dB, les SDR et PESQ des différents algorithmes S-IMM-V ne dépassent jamais la référence S-IMM. Il est intéressant de noter que l'application de Viterbi sur les filtres vocaux (S-IMM-V2) est plus performante que sur les excitations (S-IMM-V1).

Le Tableau 3.6 montre les résultats spécifiques à chaque type de bruit pour le meilleur algorithme proposé et ceux de la littérature. On remarque que les résultats sont très dépendants du type de bruit ; le cas de bruits stationnaires et quasi-stationnaires (le bruit blanc et ceux de `CAR-WINDOWNB-1`, qui sont les bruits d'intérieur d'une voiture et évoluant peu dans le temps) sont ceux où la séparation est la plus efficace. De plus, on peut voir que l'algorithme SC-IMM3 arrive à dépasser l'algorithme ASNA dans le cas du bruit blanc.

SNR	Mesure	Algorithmes								
		ASNA	V-IMM	S-IMM	SC-IMM1	SC-IMM2	SC-IMM3	S-IMM-V1	S-IMM-V2	S-IMM-V3
−6dB	SDR (dB)	5.8	4.4	4.0	4.1	5.2	**5.4**	3.8	5.1	4.2
	SIR (dB)	8.9	13.1	6.2	6.4	9.0	9.4	7.4	9.8	**10.8**
	SAR (dB)	11.2	6.6	**10.5**	10.4	9.6	9.4	8.7	8.5	6.8
	PESQ	2.00	1.22	1.91	1.91	**2.01**	**2.01**	1.95	1.87	1.79
+0dB	SDR (dB)	10.7	7.8	9.1	9.2	**9.8**	**9.8**	7.8	9.0	7.2
	SIR (dB)	14.2	19.2	12.1	12.3	14.6	15.0	13.2	15.2	**16.0**
	SAR (dB)	14.7	8.7	**13.5**	13.4	12.6	12.4	10.5	10.9	8.4
	PESQ	2.44	1.54	2.30	2.30	2.34	**2.35**	2.32	2.20	2.11
+6dB	SDR (dB)	15.0	9.7	**13.0**	**13.0**	12.8	12.9	9.8	11.2	8.3
	SIR (dB)	18.8	23.9	17.5	17.7	19.4	**19.7**	18.0	20.2	20.1
	SAR (dB)	18.2	10.0	**15.8**	15.7	14.5	14.4	11.2	12.2	8.8
	PESQ	2.85	1.82	**2.62**	**2.62**	2.59	**2.62**	2.60	2.47	2.33
Moyenne	SDR (dB)	10.5	7.3	8.7	8.7	9.3	**9.4**	7.1	8.4	6.6
	SIR (dB)	13.9	18.7	11.9	12.2	14.4	14.7	12.9	15.1	**15.6**
	SAR (dB)	14.7	8.5	**13.3**	13.2	12.2	12.1	10.1	10.5	8.0
	PESQ	2.43	1.52	2.28	2.28	2.31	**2.33**	2.29	2.18	2.08

TABLEAU 3.5 – Résultats pour tous les algorithmes du banc d'essai. Les termes en gras sont les meilleurs scores parmi les méthodes proposées pour un SNR donné.

SNR	Mesure	ASNA	V-IMM	SC-IMM3
`STREET-CITY-1`	SDR (dB)	9.9	7.1	7.7
	PESQ	2.38	1.53	2.22
`CAFE-CAFE-1`	SDR (dB)	5.9	4.4	4.9
	PESQ	2.03	1.35	1.97
`HOME-KITCHEN-1`	SDR (dB)	8.0	6.4	7.5
	PESQ	2.35	1.52	2.22
`CAR-WINDOWNB-1`	SDR (dB)	13.9	8.9	11.1
	PESQ	2.77	1.69	2.50
`Bruit blanc`	SDR (dB)	14.8	9.8	15.7
	PESQ	2.62	1.53	2.73

TABLEAU 3.6 – Résultats des algorithmes de la littérature (ASNA et V-IMM) et du meilleur algorithme proposé (SC-IMM3) en fonction du type de bruit.

Chapitre 4

Conclusion

L'objectif de ce stage était la réalisation d'une méthode semi-supervisée de séparation de la parole, basée sur un modèle source/filtre et l'intégration de contraintes pour informer la factorisation en matrices non-négatives sur les spécificités du signal de parole.

Un apprentissage a tout d'abord été ajouté à un modèle source/filtre de la littérature pour la factorisation en matrices non-négatives ; des contraintes, dont une qui été développée spécialement pour la séparation de la parole, ont ensuite été intégrées dans la NMF. L'ensemble des algorithmes a ensuite été testé et comparé à deux algorithmes de la littérature sur une base de données de paroles bruitées.
La semi-supervision ajoutée au modèle de Durrieu et *al.* [14] apporte à elle seule une amélioration non négligeable. De plus, l'imposition d'un sens physique dans la NMF par l'ajout de contraintes s'est révélé efficace, surtout dans le cas de signaux où le bruit est prépondérant. L'algorithme semi-supervisé proposé le plus performant offre des résultats très satisfaisant, se rapprochant de ceux obtenus par une méthode supervisée de la littérature.

Ainsi, il peut être bénéfique de s'intéresser à une méthode d'estimation du bruit. Une perspective est d'utiliser une méthode temps réel, comme l'ont déjà fait Germain et Mysore [19], ce qui permettrait une estimation plus précise du bruit ; une seconde possibilité est l'utilisation d'une méthode se basant sur une représentation parfaite du signal, comme l'ont proposé Roebel et *al.* [8].
La méthode proposée nécessite des exemples du locuteur que l'on veut séparer, ce qui est très contraignant. Une amélioration possible serait l'intégration d'un modèle de locuteur adaptatif ou l'utilisation du Modèle Universel de Parole proposé par Sun et Mysore [18].

L'intégration de connaissances a priori sur le signal de parole, comme le modèle source/filtre, dans une factorisation en matrices non-négatives semi-supervisée pour une tâche de séparation de la parole dans un environnement bruitée paraît donc une piste intéressante à explorer plus en profondeur.

Annexe A

Formules de mises à jour multiplicatives

A.1 Cas général de la β-divergence

On cherche à minimiser le coût suivant $\mathcal{C}$:

$$\begin{aligned} \mathcal{C} &= D_\beta(\boldsymbol{V}|\boldsymbol{\Lambda}) \\ &= \frac{1}{\beta(\beta-1)} \sum_{f=1}^{F} \sum_{n=1}^{N} \left(\boldsymbol{V}_{fn}^{\beta} + (\beta-1)\boldsymbol{\Lambda}_{fn}^{\beta} - \beta \boldsymbol{V}_{fn} \boldsymbol{\Lambda}_{fn}^{\beta-1} \right) \end{aligned} \tag{A.1}$$

où $\boldsymbol{V}$ est la matrice observée et $\boldsymbol{\Lambda}$ la matrice estimé. Soit $\boldsymbol{\Theta}$ une matrice utilisée dans la décomposition de $\boldsymbol{\Lambda}$, alors :

$$\begin{aligned} \frac{\partial \mathcal{C}}{\partial \boldsymbol{\Theta}_{pq}} &= \frac{1}{\beta(\beta-1)} \sum_{f=1}^{F} \sum_{n=1}^{N} \left(\beta(\beta-1)\boldsymbol{\Lambda}_{fn}^{\beta-1} \frac{\partial \boldsymbol{\Lambda}_{fn}}{\partial \boldsymbol{\Theta}_{pq}} - \beta \boldsymbol{V}_{fn}(\beta-1)\boldsymbol{\Lambda}_{fn}^{\beta-2} \frac{\partial \boldsymbol{\Lambda}_{fn}}{\partial \boldsymbol{\Theta}_{pq}} \right) \\ &= \sum_{f=1}^{F} \sum_{n=1}^{N} \left(\boldsymbol{\Lambda}_{fn}^{\beta-1} \frac{\partial \boldsymbol{\Lambda}_{fn}}{\partial \boldsymbol{\Theta}_{pq}} \right) - \sum_{f=1}^{F} \sum_{n=1}^{N} \left(\boldsymbol{V}_{fn} \boldsymbol{\Lambda}_{fn}^{\beta-2} \frac{\partial \boldsymbol{\Lambda}_{fn}}{\partial \boldsymbol{\Theta}_{pq}} \right) \end{aligned} \tag{A.2}$$

On en déduit donc les formules de mises à jours multiplicatives générales pour la β-divergence :

$$\boldsymbol{\Theta}_{pq}^{(i+1)} \longleftarrow \boldsymbol{\Theta}_{pq}^{(i)} \otimes \frac{\sum_{f=1}^{F} \sum_{n=1}^{N} \boldsymbol{V}_{fn} \boldsymbol{\Lambda}_{fn}^{\beta-2} \frac{\partial \boldsymbol{\Lambda}_{fn}}{\partial \boldsymbol{\Theta}_{pq}}}{\sum_{f=1}^{F} \sum_{n=1}^{N} \boldsymbol{\Lambda}_{fn}^{\beta-1} \frac{\partial \boldsymbol{\Lambda}_{fn}}{\partial \boldsymbol{\Theta}_{pq}}} \tag{A.3}$$

Lorsque que l'on veut obtenir les mises à jours multiplicatives d'un nouveau modèle, il suffit donc de calculer la dérivée de $\boldsymbol{\Lambda}$ par chaque matrice utilisée dans la décomposition de $\boldsymbol{\Lambda}$ que l'on veut mettre à jour.

A.2 Modèle source/filtre

A.2.1 Modèle de Durrieu

À partir de l'équation (A.3), on peut déduire des règles de mises à jour scalaires pour le modèle source/filtre proposé par Durrieu et *al.* [14], *scalaire* signifiant ici des règles de mises à

jour pour chaque terme $\boldsymbol{\Theta}_{pq}$. En partant de ces règles scalaires, on peut déterminer les règles de mises à jour matricielles, rappelées dans le Tableau A.1.

$$\boldsymbol{\Lambda} = \boldsymbol{W}^{\text{ex}}\boldsymbol{H}^{\text{ex}} \otimes \boldsymbol{W}^{\Phi}\boldsymbol{U}^{\Phi}\boldsymbol{H}^{\Phi} + \boldsymbol{W}^{N}\boldsymbol{H}^{N}$$

$$\boldsymbol{U}^{\Phi(i+1)} \longleftarrow \boldsymbol{U}^{\Phi(i)} \otimes \frac{\left(\boldsymbol{W}^{\Phi}\right)^{T}\left(\boldsymbol{V} \otimes \boldsymbol{\Lambda}^{\beta-2} \otimes \boldsymbol{\Lambda}^{\text{ex}}\right)\left(\boldsymbol{H}^{\Phi}\right)^{T}}{\left(\boldsymbol{W}^{\Phi}\right)^{T}\left(\boldsymbol{\Lambda}^{\beta-1} \otimes \boldsymbol{\Lambda}^{\text{ex}}\right)\left(\boldsymbol{H}^{\Phi}\right)^{T}}$$

$$\boldsymbol{H}^{\Phi(i+1)} \longleftarrow \boldsymbol{H}^{\Phi(i)} \otimes \frac{\left(\boldsymbol{W}^{\Phi}\boldsymbol{H}^{\Phi}\right)^{T}\left(\boldsymbol{V} \otimes \boldsymbol{\Lambda}^{\beta-2} \otimes \boldsymbol{\Lambda}^{\text{ex}}\right)}{\left(\boldsymbol{W}^{\Phi}\boldsymbol{H}^{\Phi}\right)^{T}\left(\boldsymbol{\Lambda}^{\beta-1} \otimes \boldsymbol{\Lambda}^{\text{ex}}\right)}$$

$$\boldsymbol{W}^{\text{ex}(i+1)} \longleftarrow \boldsymbol{W}^{\text{ex}(i)} \otimes \frac{\left(\boldsymbol{V} \otimes \boldsymbol{\Lambda}^{\beta-2} \otimes \boldsymbol{\Lambda}^{\Phi}\right)\left(\boldsymbol{H}^{\text{ex}}\right)^{T}}{\left(\boldsymbol{\Lambda}^{\beta-1} \otimes \boldsymbol{\Lambda}^{\Phi}\right)\left(\boldsymbol{H}^{\text{ex}}\right)^{T}}$$

$$\boldsymbol{H}^{\text{ex}(i+1)} \longleftarrow \boldsymbol{H}^{\text{ex}(i)} \otimes \frac{\left(\boldsymbol{W}^{\text{ex}}\right)^{T}\left(\boldsymbol{V} \otimes \boldsymbol{\Lambda}^{\beta-2} \otimes \boldsymbol{\Lambda}^{\Phi}\right)}{\left(\boldsymbol{W}^{\text{ex}}\right)^{T}\left(\boldsymbol{\Lambda}^{\beta-1} \otimes \boldsymbol{\Lambda}^{\Phi}\right)}$$

$$\boldsymbol{W}^{N(i+1)} \longleftarrow \boldsymbol{W}^{N(i)} \otimes \frac{\left(\boldsymbol{V} \otimes \boldsymbol{\Lambda}^{\beta-2}\right)\left(\boldsymbol{H}^{N}\right)^{T}}{\left(\boldsymbol{\Lambda}^{\beta-1}\right)\left(\boldsymbol{H}^{N}\right)^{T}}$$

$$\boldsymbol{H}^{N(i+1)} \longleftarrow \boldsymbol{H}^{N(i)} \otimes \frac{\left(\boldsymbol{W}^{N}\right)^{T}\left(\boldsymbol{V} \otimes \boldsymbol{\Lambda}^{\beta-2}\right)}{\left(\boldsymbol{W}^{N}\right)^{T}\left(\boldsymbol{\Lambda}^{\beta-1}\right)}$$

Tableau A.1 – Règles de mises à jours multiplicatives du modèle IMM de Durrieu et *al.* [14]

A.2.2 Modèle proposé

Le modèle que nous utilisons est très similaire [14], mais il y a quelques différences. Tout d'abord, lors de l'apprentissage, nous suivons la factorisation décrite en (2.1) ; donc, toujours en partant de (A.3), on obtient les règles de mises à jour suivantes :

$$\begin{cases} \boldsymbol{U}^{\Phi(i+1)} \longleftarrow \boldsymbol{U}^{\Phi(i)} \otimes \dfrac{\left(\boldsymbol{W}^{\Phi}\right)^{T}\left(\boldsymbol{V} \otimes \boldsymbol{\Lambda}^{\beta-2}\right)\left(\boldsymbol{H}^{\Phi}\right)^{T}}{\left(\boldsymbol{W}^{\Phi}\right)^{T}\boldsymbol{\Lambda}^{\beta-1}\left(\boldsymbol{H}^{\Phi}\right)^{T}} \\ \boldsymbol{H}^{\Phi(i+1)} \longleftarrow \boldsymbol{H}^{\Phi(i)} \otimes \dfrac{\left(\boldsymbol{W}^{\Phi}\boldsymbol{H}^{\Phi}\right)^{T}\left(\boldsymbol{V} \otimes \boldsymbol{\Lambda}^{\beta-2}\right)}{\left(\boldsymbol{W}^{\Phi}\boldsymbol{H}^{\Phi}\right)^{T}\boldsymbol{\Lambda}^{\beta-1}} \end{cases} \tag{A.4}$$

où $\boldsymbol{\Lambda} = \boldsymbol{W}^{\Phi}\boldsymbol{U}^{\Phi}\boldsymbol{H}^{\Phi}$ et $\boldsymbol{W}^{\Phi}$ n'a pas de règles de mises à jour puisqu'il est gardé fixe.
Lors de la séparation, nous suivons le même modèle que [14], mais nous conservons $\boldsymbol{U}^{\Phi}$ fixe ; il suffit donc d'utiliser les mises à jour données dans le Tableau A.1 sans appliquer celle de $\boldsymbol{U}^{\Phi}$.

A.3 Gradient des contraintes

Comme nous l'avons expliqué en 2.2.1, l'ajout d'une contrainte à la NMF ne nécessite que de connaître les parties positives et négatives du gradient de la contrainte souhaitée. Dans cette partie, nous donnons donc ces termes pour les contraintes utilisées lors de ce stage, en détaillant leur calcul pour les deux contraintes qui ont été développées (la contrainte de décorrélation normalisée et la contrainte de cohérence source/filtre).

A.3.1 Contrainte de parcimonie

Le critère de parcimonie choisie [28] a pour mesure :

$$\mathcal{P}_p(\boldsymbol{H}) = \sum_{n=1}^{N} \frac{\sum_{k=1}^{K} \boldsymbol{H}_{kn}}{\sqrt{\sum_{k=1}^{K} \boldsymbol{H}_{kn}^2}}$$

Son gradient se décompose donc de la manière suivante :

$$\left\{ \begin{array}{rcl} \left(\nabla^+_{\boldsymbol{H}}\right)_{kn} \mathcal{P}_p & = & \dfrac{1}{\sqrt{\sum_{k=1}^{K} \boldsymbol{H}_{kn}^2}} \\ \left(\nabla^-_{\boldsymbol{H}}\right)_{kn} \mathcal{P}_p & = & \boldsymbol{H}_{kn} \dfrac{\sum_{k=1}^{K} \boldsymbol{H}_{kn}}{\sum_{k=1}^{K} \boldsymbol{H}_{kn}^2} \end{array} \right. \tag{A.5}$$

A.3.2 Contrainte de continuité temporelle

La critère de continuité temporelle choisie [29] a pour mesure :

$$\mathcal{P}_c(\boldsymbol{H}) = \sum_{k=1}^{K} \frac{1}{\sigma_k} \sum_{n=2}^{N} \left(\boldsymbol{H}_{kn} - \boldsymbol{H}_{kn-1}\right)^2 \text{ avec } \sigma_k = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \boldsymbol{H}_{kn}^2}$$

Son gradient se décompose donc :

$$\left\{ \begin{array}{rcl} \left(\nabla^+_{\boldsymbol{H}}\right)_{kn} \mathcal{P}_c & = & \dfrac{4\boldsymbol{H}_{kn}}{\sigma_k^2} \\ \left(\nabla^-_{\boldsymbol{H}}\right)_{kn} \mathcal{P}_c & = & 2\dfrac{\boldsymbol{H}_{kn-1} + \boldsymbol{H}_{kn+1}}{\sigma_k^2} + 2\dfrac{\boldsymbol{H}_{kn}}{N\sigma_k^4} \sum_{n=2}^{N} \left(\boldsymbol{H}_{kn} - \boldsymbol{H}_{kn-1}\right)^2 \end{array} \right. \tag{A.6}$$

A.3.3 Contrainte de décorrélation

Le critère de décorrélation normalisé proposé a pour mesure :

$$\mathcal{P}_d(\boldsymbol{H}) = \sum_{l=1}^{K} \sum_{\substack{p=1 \\ p \neq l}}^{K} \frac{\left(\boldsymbol{H}\boldsymbol{H}^T\right)_{lp}}{\sigma_l \sigma_p} \text{ avec } \sigma_k = \sqrt{\sum_{n=1}^{N} \boldsymbol{H}_{kn}^2}$$

Son gradient s'écrit donc :

$$\begin{aligned}
(\nabla_{\boldsymbol{H}})_{kn}\mathcal{P}_d &= \sum_{l=1}^{K}\sum_{\substack{p=1\\p\neq l}}^{K}\frac{\partial}{\partial \boldsymbol{H}_{kn}}\left(\frac{\left(\boldsymbol{H}\boldsymbol{H}^T\right)_{lp}}{\sigma_l\sigma_p}\right)\\
&= \sum_{l=1}^{K}\sum_{\substack{p=1\\p\neq l}}^{K}\frac{\frac{\partial\left(\boldsymbol{H}\boldsymbol{H}^T\right)_{lp}}{\partial \boldsymbol{H}_{kn}}\sigma_l\sigma_p-\left(\boldsymbol{H}\boldsymbol{H}^T\right)_{lp}\frac{\partial\left(\sigma_l\sigma_p\right)}{\partial \boldsymbol{H}_{kn}}}{\sigma_l^2\sigma_p^2}\\
&= \sum_{l=1}^{K}\sum_{\substack{p=1\\p\neq l}}^{K}\frac{\left(\boldsymbol{H}_{ln}\delta_{kp}+\boldsymbol{H}_{pn}\delta_{kl}\right)\sigma_l\sigma_p-\left(\boldsymbol{H}\boldsymbol{H}^T\right)_{lp}\left(\boldsymbol{H}_{pn}\delta_{kp}\frac{\sigma_l}{\sigma_p}+\boldsymbol{H}_{ln}\delta_{kl}\frac{\sigma_p}{\sigma_l}\right)}{\sigma_l^2\sigma_p^2}\\
&= \sum_{\substack{l=1\\l\neq k}}^{K}\frac{\boldsymbol{H}_{ln}\sigma_k-\left(\boldsymbol{H}\boldsymbol{H}^T\right)_{lk}\boldsymbol{H}_{kn}\sigma_k^{-1}}{\sigma_l\sigma_k^2}+\sum_{\substack{p=1\\p\neq k}}^{K}\frac{\boldsymbol{H}_{kn}\sigma_k-\left(\boldsymbol{H}\boldsymbol{H}^T\right)_{kp}\boldsymbol{H}_{kn}\sigma_k^{-1}}{\sigma_k^2\sigma_p}
\end{aligned}$$

Or $\left(\boldsymbol{H}\boldsymbol{H}^T\right)_{kp}=\left(\boldsymbol{H}\boldsymbol{H}^T\right)_{pk}$, donc les deux sommes sont identiques. Alors :

$$(\nabla_{\boldsymbol{H}})_{kn}\mathcal{P}_d = 2\sum_{\substack{l=1\\l\neq k}}^{K}\frac{\boldsymbol{H}_{ln}\sigma_k-\left(\boldsymbol{H}\boldsymbol{H}^T\right)_{lk}\boldsymbol{H}_{kn}\sigma_k^{-1}}{\sigma_l\sigma_k^2} \tag{A.7}$$

Le matrices étant à coefficients positifs ou nuls, le gradient de la contrainte de parcimonie se décompose donc comme suit :

$$\begin{cases}
\left(\nabla^{+}_{\boldsymbol{H}}\right)_{kn}\mathcal{P}_d &= \dfrac{2}{\sigma_k}\displaystyle\sum_{\substack{l=1\\l\neq k}}^{K}\frac{\boldsymbol{H}_{ln}}{\sigma_l}\\
\left(\nabla^{-}_{\boldsymbol{H}}\right)_{kn}\mathcal{P}_d &= \dfrac{2\boldsymbol{H}_{kn}}{\sigma_k^3}\displaystyle\sum_{\substack{l=1\\l\neq k}}^{K}\frac{\left(\boldsymbol{H}\boldsymbol{H}^T\right)_{kl}}{\sigma_l}
\end{cases} \tag{A.8}$$

A.3.4 Contrainte de cohérence source/filtre

Le critère de cohérence proposé a pour mesure :

$$\mathcal{P}_{\phi}(\boldsymbol{H}^{\text{ex}},\boldsymbol{H}^{\Phi}) = \sum_{\substack{l=1\\l\in\mathcal{H}}}^{K_e}\sum_{\substack{p=1\\p\notin\mathcal{V}}}^{K_f}\frac{\left(\boldsymbol{H}^{\text{ex}}\boldsymbol{H}^{\Phi^T}\right)_{lp}}{\sigma_l^{\text{ex}}\sigma_p^{\Phi}}+\sum_{\substack{l=1\\l\notin\mathcal{H}}}^{K_e}\sum_{\substack{p=1\\p\in\mathcal{V}}}^{K_f}\frac{\left(\boldsymbol{H}^{\text{ex}}\boldsymbol{H}^{\Phi^T}\right)_{lp}}{\sigma_l^{\text{ex}}\sigma_p^{\Phi}}$$

$$\text{avec } \sigma_l^{\text{ex}}=\sqrt{\sum_{n=1}^{N}\boldsymbol{H}_{ln}^{\text{ex}\,2}} \text{ et } \sigma_p^{\Phi}=\sqrt{\sum_{n=1}^{N}\boldsymbol{H}_{pn}^{\Phi\,2}}$$

Son gradient (selon $\boldsymbol{H}^{\text{ex}}$) s'écrit donc :

$$
\begin{aligned}
(\nabla_{\boldsymbol{H}^{\text{ex}}})_{kn}\mathcal{P}_\phi &= \sum_{\substack{l=1\\ l\in\mathcal{H}}}^{K_e}\sum_{\substack{p=1\\ p\notin\mathcal{V}}}^{K_f} \frac{\partial}{\partial \boldsymbol{H}^{\text{ex}}_{kn}}\left(\frac{\left(\boldsymbol{H}^{\text{ex}}{\boldsymbol{H}^{\Phi}}^T\right)_{lp}}{\sigma^{\text{ex}}_l\sigma^{\Phi}_p}\right) + \sum_{\substack{l=1\\ l\notin\mathcal{H}}}^{K_e}\sum_{\substack{p=1\\ p\in\mathcal{V}}}^{K_f} \frac{\partial}{\partial \boldsymbol{H}^{\text{ex}}_{kn}}\left(\frac{\left(\boldsymbol{H}^{\text{ex}}{\boldsymbol{H}^{\Phi}}^T\right)_{lp}}{\sigma^{\text{ex}}_l\sigma^{\Phi}_p}\right)\\
&= \begin{cases}
\displaystyle\sum_{\substack{l=1\\ l\in\mathcal{H}}}^{K_e}\sum_{\substack{p=1\\ p\notin\mathcal{V}}}^{K_f} \frac{\frac{\partial\left(\boldsymbol{H}^{\text{ex}}{\boldsymbol{H}^{\Phi}}^T\right)_{lp}}{\partial \boldsymbol{H}^{\text{ex}}_{kn}}\sigma^{\text{ex}}_l\sigma^{\Phi}_p - \left(\boldsymbol{H}^{\text{ex}}{\boldsymbol{H}^{\Phi}}^T\right)_{lp}\frac{\partial\left(\sigma^{\text{ex}}_l\sigma^{\Phi}_p\right)}{\partial \boldsymbol{H}^{\text{ex}}_{kn}}}{\left(\sigma^{\text{ex}}_l\sigma^{\Phi}_p\right)^2} & \text{si } k\in\mathcal{H}\\
\displaystyle\sum_{\substack{l=1\\ l\notin\mathcal{H}}}^{K_e}\sum_{\substack{p=1\\ p\in\mathcal{V}}}^{K_f} \frac{\frac{\partial\left(\boldsymbol{H}^{\text{ex}}{\boldsymbol{H}^{\Phi}}^T\right)_{lp}}{\partial \boldsymbol{H}^{\text{ex}}_{kn}}\sigma^{\text{ex}}_l\sigma^{\Phi}_p - \left(\boldsymbol{H}^{\text{ex}}{\boldsymbol{H}^{\Phi}}^T\right)_{lp}\frac{\partial\left(\sigma^{\text{ex}}_l\sigma^{\Phi}_p\right)}{\partial \boldsymbol{H}^{\text{ex}}_{kn}}}{\left(\sigma^{\text{ex}}_l\sigma^{\Phi}_p\right)^2} & \text{sinon}
\end{cases}
\end{aligned}
$$

On retrouve une expression similaire que celle obtenue lors de la dérivation de la contrainte de décorrélation ; mais ici, les termes ${\boldsymbol{H}^{\Phi}}^T$ et σ^{Φ}_p sont indépendants de $\boldsymbol{H}^{\text{ex}}_{kn}$. En adaptant (A.7), on obtient donc :

$$
(\nabla_{\boldsymbol{H}^{\text{ex}}})_{kn}\mathcal{P}_\phi = \begin{cases}
\displaystyle\sum_{\substack{p=1\\ p\notin\mathcal{V}}}^{K_f} \frac{\boldsymbol{H}^{\Phi}_{pn}\sigma^{\text{ex}}_k - \left(\boldsymbol{H}^{\text{ex}}{\boldsymbol{H}^{\Phi}}^T\right)_{kp}\boldsymbol{H}^{\text{ex}}_{kn}\sigma^{\text{ex}-1}_k}{{\sigma^{\text{ex}}_k}^2\sigma^{\Phi}_p} & \text{si } k\in\mathcal{H}\\
\displaystyle\sum_{\substack{p=1\\ p\in\mathcal{V}}}^{K_f} \frac{\boldsymbol{H}^{\Phi}_{pn}\sigma^{\text{ex}}_k - \left(\boldsymbol{H}^{\text{ex}}{\boldsymbol{H}^{\Phi}}^T\right)_{kp}\boldsymbol{H}^{\text{ex}}_{kn}\sigma^{\text{ex}-1}_k}{{\sigma^{\text{ex}}_k}^2\sigma^{\Phi}_p} & \text{sinon}
\end{cases}
$$

En faisant de même avec $\boldsymbol{H}^{\Phi}$, on montre que le gradient de $\mathcal{P}_\phi$ est composé des termes

suivants :

$$\begin{cases} \left(\nabla^{+}_{\boldsymbol{H}^{\text{ex}}}\right)_{kn}\mathcal{P}_{\phi} &= \begin{cases} \dfrac{1}{\sigma_k^{\text{ex}}}\displaystyle\sum_{\substack{p=1\\p\notin\mathcal{V}}}^{K_f}\dfrac{\boldsymbol{H}^{\Phi}_{pn}}{\sigma_p^{\Phi}} & \text{si } k\in\mathcal{H} \\ \dfrac{1}{\sigma_k^{\text{ex}}}\displaystyle\sum_{\substack{p=1\\p\in\mathcal{V}}}^{K_f}\dfrac{\boldsymbol{H}^{\Phi}_{pn}}{\sigma_p^{\Phi}} & \text{sinon} \end{cases} \\ \left(\nabla^{-}_{\boldsymbol{H}^{\text{ex}}}\right)_{kn}\mathcal{P}_{\phi} &= \begin{cases} \dfrac{\boldsymbol{H}^{\text{ex}}_{kn}}{\sigma_k^{\text{ex}3}}\displaystyle\sum_{\substack{p=1\\p\notin\mathcal{V}}}^{K_f}\dfrac{\left(\boldsymbol{H}^{\text{ex}}\boldsymbol{H}^{\Phi T}\right)_{kp}}{\sigma_p^{\Phi}} & \text{si } k\in\mathcal{H} \\ \dfrac{\boldsymbol{H}^{\text{ex}}_{kn}}{\sigma_k^{\text{ex}3}}\displaystyle\sum_{\substack{p=1\\p\in\mathcal{V}}}^{K_f}\dfrac{\left(\boldsymbol{H}^{\text{ex}}\boldsymbol{H}^{\Phi T}\right)_{kp}}{\sigma_p^{\Phi}} & \text{sinon} \end{cases} \\ \left(\nabla^{+}_{\boldsymbol{H}^{\Phi}}\right)_{kn}\mathcal{P}_{\phi} &= \begin{cases} \dfrac{1}{\sigma_k^{\Phi}}\displaystyle\sum_{\substack{l=1\\l\notin\mathcal{V}}}^{K_e}\dfrac{\boldsymbol{H}^{\text{ex}}_{ln}}{\sigma_l^{\text{ex}}} & \text{si } k\in\mathcal{V} \\ \dfrac{1}{\sigma_k^{\Phi}}\displaystyle\sum_{\substack{l=1\\l\in\mathcal{V}}}^{K_e}\dfrac{\boldsymbol{H}^{\text{ex}}_{ln}}{\sigma_l^{\text{ex}}} & \text{sinon} \end{cases} \\ \left(\nabla^{-}_{\boldsymbol{H}^{\Phi}}\right)_{kn}\mathcal{P}_{\phi} &= \begin{cases} \dfrac{\boldsymbol{H}^{\Phi}_{kn}}{\sigma_k^{\Phi 3}}\displaystyle\sum_{\substack{l=1\\l\notin\mathcal{H}}}^{K_e}\dfrac{\left(\boldsymbol{H}^{\text{ex}}\boldsymbol{H}^{\Phi T}\right)_{lk}}{\sigma_l^{\text{ex}}} & \text{si } k\in\mathcal{V} \\ \dfrac{\boldsymbol{H}^{\Phi}_{kn}}{\sigma_k^{\Phi 3}}\displaystyle\sum_{\substack{k=1\\k\in\mathcal{H}}}^{K_e}\dfrac{\left(\boldsymbol{H}^{\text{ex}}\boldsymbol{H}^{\Phi T}\right)_{lk}}{\sigma_l^{\text{ex}}} & \text{sinon} \end{cases} \end{cases} \tag{A.9}$$

Bibliographie

[1] Steven F. Boll , Suppression of acoustic noise in speech using spectral subtraction. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 27(2),113–120, 1979.

[2] Yuxuan Wang et DeLiang Wang , A neural network for time-domain signal reconstruction : Towards improving the perceptual quality of supervised speech separation. Rapport technique, Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210-1277, 2014.

[3] Yuxuan Wang, Arun Narayanan et DeLiang Wang , On training targets for supervised speech separation. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 22(12),1849–1858, 2014.

[4] Pentti Paatero et Unto Tapper , Positive matrix factorization : A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2),111–126, 1994.

[5] Daniel D. Lee et H. Sebastian Seung , Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755),788–791, 1999.

[6] Nancy Bertin , *Les factorisations en matrices non-négatives. Approches contraintes et probabilistes, application à la transcription automatique de musique polyphonique.* Thèse de doctorat, Télécom ParisTech, 2009.

[7] Cédric Févotte, Nancy Bertin et Jean-Louis Durrieu , Nonnegative matrix factorization with the itakura-saito divergence : With application to music analysis. *Neural computation*, 21(3),793–830, 2009.

[8] Axel Roebel, Jordi Pons, Marco Liuni et Mathieu Lagrange , On automatic drum transcription using non-negative matrix deconvolution and itakura-saito divergence. *In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 414–418. IEEE, 2015.

[9] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan et Shun-ichi Amari , *Nonnegative matrix and tensor factorizations : applications to exploratory multi-way data analysis and blind source separation.* John Wiley & Sons, 2009.

[10] Daniel D. Lee et H. Sebastian Seung , Algorithms for non-negative matrix factorization. *In Advances in neural information processing systems*, pages 556–562, 2001.

[11] Masahiro Nakano, Hirokazu Kameoka, Jonathan Le Roux, Yu Kitano, Nobutaka Ono et Shigeki Sagayama , Convergence-guaranteed multiplicative algorithms for nonnegative matrix factorization with β-divergence. *International Workshop on Machine Learning for Signal Processing, In Proc. IEEE*, 10,283–288, 2010.

[12] Jonathan Le Roux et Emmanuel Vincent , Consistent wiener filtering for audio source separation. *Signal Processing Letters, IEEE*, 20(3),217–220, 2013.

[13] Paul Magron, Roland Badeau et David Bertrand , Phase recovery in NMF for audio source separation : an insightful benchmark. *In 40th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 81–85. IEEE, 2015.

[14] Jean-Louis Durrieu, Alexey Ozerov, Cédric Févotte, Gaël Richard et Bertrand David , Main instrument separation from stereophonic audio signals using a source/filter model. *In EUSIPCO*, numéro EPFL-CONF-163310, pages 15–19, 2009. URL : `http://www.durrieu.ch/phd/eusipco09/`.

[15] Tuomas Virtanen et Anssi Klapuri , Analysis of polyphonic audio using source-filter model and non-negative matrix factorization. *In Advances in models for acoustic processing, neural information processing systems workshop*. Citeseer, 2006.

[16] Gautham J. Mysore et Paris Smaragdis , A non-negative approach to semi-supervised separation of speech from noise with the use of temporal dynamics. *In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 17–20. IEEE, 2011.

[17] Gautham J. Mysore et Paris Smaragdis , A non-negative approach to language informed speech separation. *In Latent Variable Analysis and Signal Separation*, pages 356–363. Springer, 2012. URL : `https://ccrma.stanford.edu/~gautham/Site/lva_ica_2012.html`.

[18] Dennis L. Sun et Gautham J. Mysore , Universal speech models for speaker independent single channel source separation. *In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 141–145. IEEE, 2013.

[19] François G. Germain et Gautham J. Mysore , Speaker and noise independent online single-channel speech enhancement. *In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 71–75. IEEE, 2015.

[20] Alexey Ozerov, Pierrick Philippe, Frédéric Bimbot et Rémi Gribonval , Adaptation of bayesian models for single-channel source separation and its application to voice/music separation in popular songs. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(5),1564–1578, 2007.

[21] Luc Le Magoarou, Alexey Ozerov et Ngoc QK Duong , Text-informed audio source separation using nonnegative matrix partial co-factorization. *In Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on*, pages 1–6. IEEE, 2013.

[22] Luc Le Magoarou, Alexey Ozerov et Ngoc QK Duong , Text-informed audio source separation. example-based approach using non-negative matrix partial co-factorization. *Journal of Signal Processing Systems*, pages 1–15, 2014.

[23] Donald S. Williamson, Yuxuan Wang et DeLiang Wang , Deep neural networks for estimating speech model activations. *In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5113–5117. IEEE, 2015.

[24] Jonathan Le Roux, John R Hershey et Felix Weninger , Deep NMF for speech separation. 2015.

[25] Fernando Villavicencio, Axel Röbel et Xavier Rodet , Improving lpc spectral envelope extraction of voiced speech by true-envelope estimation. *In Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pages 869–872. IEEE, 2006.

[26] Mikkel N. Schmidt et Rasmus Kongsgaard Olsson , Single-channel speech separation using sparse non-negative matrix factorization. *In Spoken Language Proceesing, ISCA International Conference on (INTERSPEECH)*, 2006.

[27] Julian Eggert et Edgar Körner , Sparse coding and NMF. *In Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 4, pages 2529–2533. IEEE, 2004.

[28] Cyril Joder, Felix Weninger, David Virette et Bjorn Schuller , A comparative study on sparsity penalties for NMF-based speech separation : Beyond Lp-norms. *In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 858–862. IEEE, 2013.

[29] Tuomas Virtanen , Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(3),1066–1074, 2007.

[30] Victor Zue, Stephanie Seneff et James Glass , Speech database development at MIT : TIMIT and beyond. *Speech Communication*, 9(4),351–356, 1990.

[31] David B. Dean, Sridha Sridharan, Robert J. Vogt et Michael W. Mason , The QUT-NOISE-TIMIT corpus for the evaluation of voice activity detection algorithms. *Proceedings of Interspeech 2010*, pages 3110–3113, 2010.

[32] Tuomas Virtanen, Jort F Gemmeke et Bhiksha Raj , Active-set newton algorithm for overcomplete non-negative representations of audio. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(11),2277–2289, 2013.

[33] Emmanuel Vincent, Rémi Gribonval et Cédric Févotte , Performance measurement in blind audio source separation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(4),1462–1469, 2006.

[34] Antony W. Rix, John G. Beerends, Michael P. Hollier et Andries P. Hekstra , Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs. *In Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 2, pages 749–752. IEEE, 2001.

Liste des figures

1.1 Exemple d'une décomposition par NMF 4
1.2 Exemple du produit spectral du modèle source/filtre 8
1.3 Décomposition source/filtre du spectrogramme d'une voix 8
1.4 Décomposition source/filtre dans le modèle IMM 10

2.1 Estimation d'une enveloppe spectrale avec l'algorithme *True Envelope* 13
2.2 Comparaison des $\boldsymbol{\Lambda}^{\Phi}$ estimées avec et sans *True Envelope* 14
2.3 Exemples de phonèmes appris sur un locuteur 14
2.4 Comparaison des $\boldsymbol{H}^{\Phi}$ et $\boldsymbol{\Lambda}^{\Phi}$ estimées avec et sans parcimonie 17
2.5 Comparaison des $\boldsymbol{H}^{\Phi}$ et $\boldsymbol{\Lambda}^{\Phi}$ estimées avec et sans continuité temporelle 18
2.6 Comparaison des $\boldsymbol{H}^{\Phi}$ et $\boldsymbol{\Lambda}^{\Phi}$ estimées avec et sans décorrélation 19
2.7 Comparaison des $\boldsymbol{H}^{\text{ex}}$ estimées avec parcimonie, avec et sans adaptation du poids 21
2.8 Structure générale de l'algorithme proposé 22

Liste des tableaux

1.1 Formules de la β-divergence et de ses cas particuliers : IS, KL et EUC 5
1.2 Rôles des matrices dans le modèle IMM 9

3.1 Algorithmes au banc d'essai 24
3.2 Tableau de valeurs des coefficients de poids μ des contraintes 25
3.3 Résultats pour l'algorithme S-IMM, avec différents nombre de bases de bruit . . 26
3.4 Résultats pour l'algorithme SC-IMM, avec et sans adaptation des poids μ 26
3.5 Résultats du banc d'essai 28
3.6 Résultats des meilleurs algorithmes en fonction du type de bruit 28

A.1 Règles de mises à jours multiplicatives du modèle IMM 31