Dictionary learning methods for single-channel source separation

Augustin Lefèvre

▶ **To cite this version:**

Augustin Lefèvre. Dictionary learning methods for single-channel source separation. General Mathematics [math.GM]. École normale supérieure de Cachan - ENS Cachan, 2012. English. NNT : 2012DENS0051 . tel-00764546v1

HAL Id: tel-00764546

https://theses.hal.science/tel-00764546v1

Submitted on 13 Dec 2012 (v1), last revised 20 Feb 2018 (v2)

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

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

THÈSE DE DOCTORAT DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

présentée par **Augustin Lefèvre**

pour obtenir le grade de

Docteur de l'École Normale Supérieure de Cachan

Domaine: **Mathématiques appliquées**

Sujet de la thèse:

Méthodes d'apprentissage de dictionnaire pour la séparation de sources audio avec un seul capteur

—

Dictionary learning methods for single-channel audio source separation

---

Thèse présentée et soutenue à Cachan le 4 Octobre 2012

devant le jury composé de:

Francis BACH	ENS/INRIA Paris	Directeur de thèse
Cédric FÉVOTTE	LTCI/Telecom ParisTech	Directeur de thèse
Laurent DAUDET	Université Paris Diderot - Paris 7	Rapporteur
Guillermo SAPIRO	Duke University	Rapporteur
Arshia CONT	IRCAM/CNRS/INRIA	Examinateur
Olivier CAPPÉ	LTCI/Telecom Paristech	Examinateur
Pierre-Antoine ABSIL	UCL	Invité

Thèse préparée au sein de l'équipe SIERRA

au département d'informatique de l'ENS Ulm

(INRIA/ENS/CNRS UMR 8548)

Résumé

Étant donné un mélange de plusieurs signaux sources, par exemple un morceau et plusieurs instruments, ou un entretien radiophonique et plusieurs interlocuteurs, la séparation de source mono-canal consiste à estimer chacun des signaux sources à partir d'un enregistrement avec un seul microphone. Puisqu'il y a moins de capteurs que de sources, il y a a priori une infinité de solutions sans rapport avec les sources originales. Il faut alors trouver quelle information supplémentaire permet de rendre le problème bien posé.

Au cours des dix dernières années, la factorisation en matrices positives (NMF) est devenue un composant majeurs des systèmes de séparation de sources. En langage profane, la NMF permet de décrire un ensemble de signaux audio á partir de combinaisons d'éléments sonores simples (les *atomes*), formant un *dictionnaire*. Les systèmes de séparation de sources reposent alors sur la capacité à trouver des atomes qui puissent être assignés de façon univoque à chaque source sonore. En d'autres termes, ils doivent être interprétables.

Nous proposons dans cette thèse trois contributions principales aux méthodes d'apprentissage de dictionnaire. La première est un critère de parcimonie par groupes adapté à la NMF lorsque la mesure de distortion choisie est la divergence d'Itakura-Saito. Dans la plupart des signaux de musique on peut trouver de longs intervalles où seulement une source est active (des soli). Le critère de parcimonie par groupe que nous proposons permet de trouver automatiquement de tels segments et d'apprendre un dictionnaire adapté à chaque source. Ces dictionnaires permettent ensuite d'effectuer la tâche de séparation dans les intervalles où les sources sont mélangées. Ces deux tâches d'identification et de séparation sont effectuées simultanément en une seule passe de l'algorithme que nous proposons.

Notre deuxième contribution est un algorithme en ligne pour apprendre le dictionnaire à grande échelle, sur des signaux de plusieurs heures, ce qui était impossible auparavant. L'espace mémoire requis par une NMF estimée en ligne est constant alors qu'il croit linéairement avec la taille des signaux fournis dans la version standard, ce qui est impraticable pour des signaux de plus d'une heure.

Notre troisième contribution touche à l'interaction avec l'utilisateur. Pour des signaux courts, l'apprentissage aveugle est particulièrement dificile, et l'apport d'information spécifique au signal traité est indispensable. Notre contribution est similaire à l'inpainting et permet de prendre en compte des annotations temps-fréquence. Elle repose sur l'observation que la quasi-totalité du spectrogramme peut etre divisé en régions spécifiquement assignées à chaque source. Nous décrivons une extension de NMF pour prendre en compte cette information et discutons la possibilité d'inférer cette information automatiquement avec des outils d'apprentissage statistique simples.

Abstract

Given an audio signal that is a mixture of several sources, such as a music piece with several instruments, or a radio interview with several speakers, single-channel audio source separation aims at recovering each of the source signals when the mixture signal is recorded with only one microphone. Since there are less sensors (one microphone) than sources (several sources), there is a priori an infinite number of solutions to this problem that are not related to the original source signals. They key ingredient in single-channel audio source separation is to decide what kind of additional information must be provided to disambiguate the problem.

In the last decade, nonnegative matrix factorization (NMF) has become a major building block in source separation. In lay man's term, NMF consists in learning to describe a collection of audio signals as linear combinations of typical *atoms* forming a *dictionary*. Source separation algorithms are then built on the idea that each atom can be assigned unambiguously to a source. However, since the dictionary is learnt on a mixture of several sources, there is no guarantee that each atom corresponds to one source rather than of a mixture of them : put in other words, the dictionary atoms are not interpretable *a priori*, they must be made so, using additional information in the learning process.

In this thesis we provide three main contributions to blind source separation methods based on NMF. Our first contribution is a group-sparsity inducing penalty specifically tailored for Itakura-Saito NMF : in many music tracks, there are whole intervals where at least one source is inactive. The group-sparsity penalty we propose allows identifying these intervals blindly and learn source specific dictionaries. As a consequence, those learned dictionaries can be used to do source separation in other parts of the track were several sources are active. These two tasks of identification and separation are performed simultaneously in one run of group-sparsity Itakura-Saito NMF.

Our second contribution is an online algorithm for Itakura-Saito NMF that allows learning dictionaries on very large audio tracks. Indeed, the memory complexity of a batch implementation NMF grows linearly with the length of the recordings and becomes prohibitive for signals longer than an hour. In contrast, our online algorithm is able to learn NMF on arbitrarily long signals with limited memory usage.

Our third contribution deals with user informed NMF. In short mixed signals, blind learning becomes very hard and sparsity do not retrieve interpretable dictionaries. Our contribution is very similar in spirit to inpainting. It relies on the empirical fact that, when observing the spectrogram of a mixture signal, an overwhelming proportion of it consists in regions where only one source is active. We describe an extension of NMF to take into account time-frequency localized

information on the absence/presence of each source. We also investigate inferring this information with tools from machine learning.

Remerciements

Si j'ai tant appris et tant grandi ces trois dernières années, je le dois à de nombreuses personnes, et c'est pourquoi j'adresse un grand salut et tous mes remerciements

À Cédric Févotte et Francis Bach, mes directeurs de thèse, qui m'ont toujours poussé à faire le pari des idées originales même lorsque leur réussite était loin d'être garantie. En particulier à Cédric pour son sens de la mise en perspective, bibliographique et scientifique, et son souci de la rédaction, qui ont permis d'éclaircir mille fois des épreuves d'article, dans les derniers moments avant qu'ils soient soumis. En particulier à Francis qui a toujours trouvé des suggestions constructives à faire sur des des notes de travail rédigées parfois à la hâte, et à anticiper les écueils avant même que j'aie eu le temps de les atteindre.

To the reviewers of this manuscript, Laurent Daudet and Guillermo Sapiro, with whow I have enjoyed great discussions during conferences, for the great job they have done and also for helping me improve this manuscript as much it could be.

À Olivier Cappé, Arshia Cont, et Pierre-Antoine Absil qui ont accepté de faire partie du jury de thèse.

À Marine Meyer, Lindsay Poliénor, Patricia Friedrich, Joëlle Isnard, Christine Rose et Géralidine Carbonel, qui ont assuré les conditions matérielles et le bon déroulement de mon doctorat, ainsi que des missions en France ou à l'étranger.

Aux encadrants des équipes WILLOW et SIERRA qui font vivre le laboratoire au rythme des deadlines et des séminaires, et entretiennent une atmosphère d'émulation et de créativité impressionantes : Jean Ponce, Josef Sivic, Ivan Laptev, Guilllaume Obozinski, et Sylvain Arlot.

À tous mes collègues, avec qui j'ai partagé nombre de repas, discussions scientifiques (ou pas), de code informatique, et de pâtisseries : Armand Joulin, Rodolphe Jenatton, Florent Couzinié, Olivier Duchenne, Louise Benoit, Julien Mairal, Vincent Delaitre, Edouard Grave, Mathieu Solnon, Toby Dylan Hocking, Petr Gronat, Guillaume Seguin, Y-Lan Boureau, Oliver Whyte, Nicolas Le Roux, Karteek Alahari, Mark Schmidt, Sesh Kumar, Simon Lacoste-Julien, José Lezama,

Aux collègues et confrères avec qui j'ai fait mon apprentissage des premiers posters, des premiers oraux, avec qui j'ai eu des discussions intéressantes au détour d'un séminaire ou d'une pause café : Onur Dikmen, Romain Hennequin, Manuel Moussallam, Benoit Fuentes, Cyril Joder, Gilles Chardon, Antoine Liutkus, Rémi Foucard, Félicien Vallet, Thomas Schatz, Alexis Bénichoux, Mathieu Kowalski, Alexandre Gramfort, Rémi Flamary, Pablo Sprechmann, Ignacio Ramirez, mais aussi Gaël Richard, Slim Essid, Laurent Daudet, Rémi Gribonval,

Paris Smaragdis, John Hershey, Jonathan Le Roux, Nobutaka Ono, Emmanuel Vincent, Alexei Ozerov.

Aux organismes qui ont financé ma thèse, le ministère de la Recherche et de l'Enseignement Supérieur ainsi que l'European Research Council,

À Marine et Gabriel, Iris, Paul, Raphaël, Armand, Samuel, David, Youssef, et à toute ma famille sur qui je peux compter en toutes circonstances et qui m'apportent tant de bonheur,

À Louise mon amour.

Contents

Contents **ix**

1 Introduction **5**
1.1 Mixing assumptions and related tasks 5
1.1.1 Assumptions on the mixing process 5
1.1.2 Standard metrics for single-channel source separation . . . 6
1.1.3 Related tasks 7
1.2 Time-frequency representations 9
1.2.1 Fourier Transform 9
1.2.2 Short time Fourier transform 10
1.2.3 Recovery of source estimates via time-frequency masking . 12
1.3 Models for audio source separation 13
1.3.1 Mixture models : hard sparsity constraints 14
1.3.1.1 Inference 14
1.3.1.2 Learning: trained models or blind learning ? . . . 17
1.3.1.3 Extension to hidden Markov models 17
1.3.2 Nonnegative matrix factorization with sparsity constraints 19
1.3.2.1 Trained learning with sparsity 19
1.3.2.2 Partially blind learning or model calibration . . . 21
1.3.2.3 Smoothness penalties 21
1.3.2.4 Parameterized atoms 22
1.3.3 Other latent variable models : Complex Matrix Factorization 22
1.3.4 Other approaches : Computational Audio Scene Analysis . 24
1.3.4.1 Basic complexity issues 24
1.3.4.2 A clustering approach to audio source separation 24
1.4 Conclusion 26

2 Structured NMF with group-sparsity penalties **29**
2.1 The family of NMF problems 31
2.1.1 The family of beta-divergences 32
2.1.2 Identification problems in NMF 33
2.1.3 Determining which divergence to choose 34
2.2 Optimization algorithms for the β-divergences 36
2.2.1 Non-convexity of NMD with beta-divergences 36
2.2.2 MM algorithms and multiplicative updates 37
2.2.3 Expectation-Maximization algorithms 39
2.2.3.1 Itakura-Saito divergence 39
2.2.3.2 Kullback-Leibler divergence 41

2.3 Itakura-Saito NMF . . . 42
2.3.1 Generative model . . . 43
2.3.2 Recovery of source estimates . . . 43
2.3.3 Consistent source estimates . . . 44
2.3.4 Derivation of a descent algorithm . . . 47
2.3.5 Discussion of convergence properties . . . 48
2.3.6 Discussion of convergence properties: empirical results . . . 51
2.3.7 Overview of the algorithm . . . 53
2.3.8 Related Work . . . 53
2.4 Group-sparsity enforcing penalty in NMF . . . 54
2.4.1 Presentation . . . 55
2.4.2 Interpretation of the penalty term . . . 56
2.4.3 Extension to block-structured penalties . . . 57
2.4.4 Algorithm for group Itakura Saito NMF . . . 58
2.5 Model selection in sparse NMF . . . 59
2.5.1 Kolmogorov-Smirnov statistic . . . 59
2.5.2 Bayesian approaches . . . 61
2.6 Experiments with group-sparse NMF . . . 61
2.6.1 Validation on synthetic data . . . 61
2.6.2 Results in single channel source separation . . . 61
2.6.3 Block-structured penalty . . . 64
2.7 Related work . . . 64
2.8 Conclusion . . . 66

3 Online NMF **69**
3.1 Algorithm for online IS-NMF . . . 70
3.1.1 Itakura-Saito NMF . . . 70
3.1.2 Recursive computation of auxiliary function . . . 71
3.1.3 Practical implementation . . . 72
3.2 Experimental study . . . 74
3.3 Related Work . . . 75
3.4 Conclusion . . . 77

4 User informed source separation **79**
4.1 A GUI for time-frequency annotations . . . 81
4.2 Annotated NMF . . . 83
4.3 Towards automatic annotations . . . 84
4.3.1 Learning algorithms . . . 85
4.3.2 Features . . . 86
4.3.3 Raw Patches . . . 86
4.3.4 Oriented filters . . . 87
4.3.5 Averaging training labels . . . 88
4.4 Experimental results . . . 89
4.4.1 Description of music databases . . . 89
4.4.2 Ideal performance and robustness . . . 89

4.4.3 Evaluation of automatic annotations 90
4.4.4 Overall results 92
4.5 Miscellaneous 94
4.5.1 Detailed comparison of detectors 94
4.5.2 Extension of annotated NMF with more than one dominant source 96
4.5.2.1 Mathematical formulation 96
4.5.2.2 Source separation of three instrumental tracks . . 96
4.5.3 Handling uncertainty in automatic annotations 97
4.5.4 Predicting source specific time intervals 98
4.6 Related work 99
4.7 Conclusion 100

5 Conclusion **103**

A Projected gradient descent **107**

B Description of databases **109**

C Detailed results **111**

Bibliography **115**

Structure of this thesis

Audio source separation systems are a blend of signal processing techniques and machine learning tools. Signal processing techniques are used to extract features of interest from the signal, and outline strategies to recover source estimates, and machine learning tools are used to make decisions based on available training data and possibly additional information (specific models, user information, etc.).

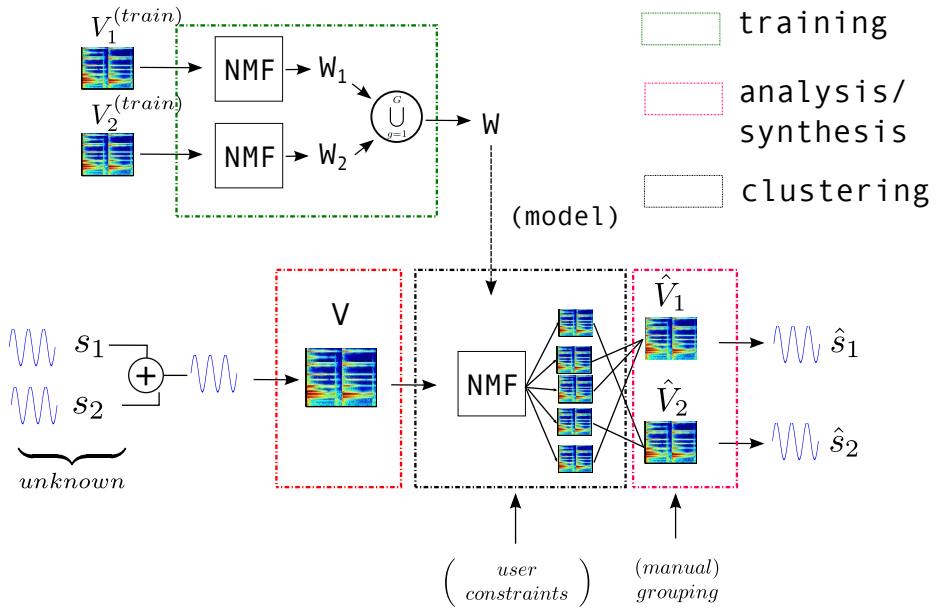


Figure 0.1: Workflow of a typical source separation system

The main building blocks of a typical single-channel source separation system are illustrated in Figure 0.1. In the training block (in *green*), offline databases specific to various types of sound signals are first exploited to build interpretable models. At test time, a mixture of unknown sound signals is presented, which is first transformed in the *analysis* step into a feature matrix. This feature matrix is then clustered, or decomposed into elementary matrices, and those are grouped to yield estimates of the sources. These estimates must then be transformed back into time domain signals in the *synthesis* step. The clustering step is the main block of the source separation step : we use here the term clustering in a generic way, since this unit accommodates matrix factorization algorithms as well as segmentation/clustering algorithms. The clustering block may perform *blind* source separation, in which case a model of the data is learnt at the same time

as source estimates are computed. It may also be fed with a pre-learnt model computed by the training block, but also with user specific constraints.

In this thesis, we study every aspect of this workflow In Chapter 1, we will give an overview of each step of the system with a particular emphasis on latent variable models used to build interpretable representations of spectrograms. In Chapter 2, we will discuss algorithms for nonnegative matrix factorization as well as model selection issues and in particular present our contribution to that point with group nonnegative matrix factorization [Lefèvre et al., 2011a]. In Chapter 3 we will present a modification of the multiplicative updates algorithm to large scale settings where only a few passes over the entire data set are allowed [Lefèvre et al., 2011b] Finally, in Chapter 4 we will present recent contributions in user informed source separation, among which our recent work on time-frequency annotated NMF [Lefèvre et al., 2012].

Contributions

Our contributions in this thesis are the following :

- ⋆ We propose in Chapter 2 a group-sparsity penalty for Itakura-Saito NMF. This penalty is designed for audio signals where each source "switches" on and off at least once in the recording. Our group-sparsity penalty allows identifying segments where sources are missing, learn an appropriate dictionary each source, and un-mix sources elsewhere. Simple temporal dependencies may be enforced in the form of a block-sparsity penalty, which favors contiguous zeroes in the decomposition coefficients. Moreover, we propose a criterion to select the penalty parameter based on tools from statistical theory.

- ⋆ Our second contribution in Chapter 3 is an online algorithm to learn dictionaries adapted to the Itakura-Saito divergence. We show that it allows a ten times speedup for signals longer than three minutes, in the small dictionary setting. It also allows running NMF on signals longer than an hour which was previously impossible.

- ⋆ Our third contribution, presented in Chapter 4, goes back to short signals and blind separation : we introduce in NMF additional constraints on the estimates of the source spectrograms, in the form of time-frequency annotations. While time annotations have been proposed before, time-frequency annotations allow retrieving perfect source estimates provided only 20% of the spectrogram is annotated, and annotations are correct. Our formulation is robust to small errors in the annotations. We provide a graphical interface for user annotations, and investigate algorithms to automatize the annotation process.

These contributions have led to the following publications:

A. Lefèvre and F. Bach and C. Févotte, "Itakura-Saito Nonnegative Matrix Factorization with group sparsity", *in Proceedings of the International Conference on Acoustique Speech and Signal Processing (ICASSP)*,2011.

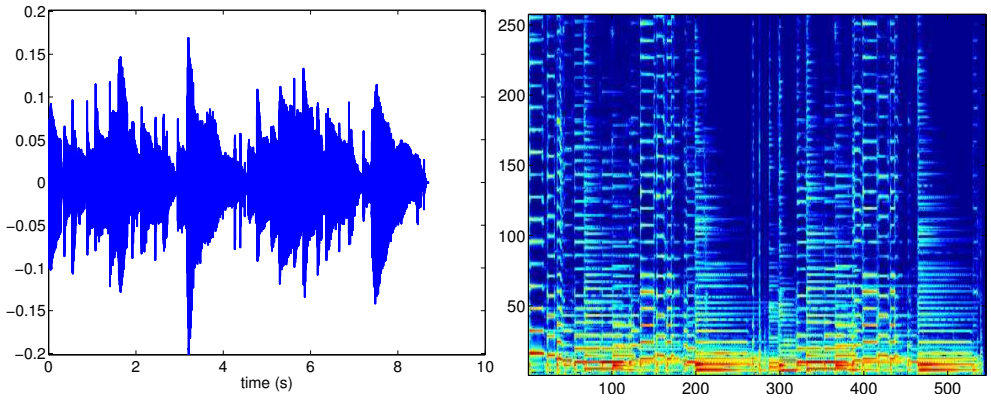
A. Lefèvre and F. Bach and C. Févotte, "Factorisation de matrices structurée en groupes avec la divergence d'Itakura-Saito", *in Proceedings of 23e colloque GRETSI sur le Traitement du Signal et des Images*, 2011.

A. Lefèvre and F. Bach and C. Févotte, "Online algorithms for nonnegative matrix factorization with the Itakura-Saito divergence", *in Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2011.

A. Lefèvre and F. Bach and C. Févotte, "Semi-supervised NMF with time-frequency annotations for single-channel source separation", *in Proceedings of the International Conference on Music Information Retrival (ISMIR)*, 2012.

Notations

- $x \in \mathbb{R}^T$ a mixture signal, $s^{(g)} \in \mathbb{R}^T$, for $g = 1 \dots G$ source signals. T is the length of the recording acquired at sampling rate f_s (usually $44,100Hz$, i.e., $44,100$ samples per seconds, or equivalently one sample every 22 every millisecond (ms).
- Superscript (g) is used to index source numbers, and should not be confused with the power operator $x^g = \underbrace{x \times \cdots \times x}_{g \text{ times}}$.



(a) Time signal (b) Power spectrogram

Figure 0.2: Various representations of an audio signal.

- Linear instantaneous mixing assumption, $x_t = \sum_{g=1}^{G} s_t^{(g)}$.

- $X \in \mathbb{C}^{F \times N}$ is a short time Fourier transform (STFT) of x.
- Likewise $S^{(g)}$ will denote the STFT of $s^{(g)}$ for $g = 1 \dots G$.
- F is the analysis window length, thus f coordinates span the *frequency* axis. N is the number of time frames, so n spans the *time* axis. We will often refer to f as a *frequency bin*, n as a time bin (or sometimes a time frame[1]), and (f, n) as a *time-frequency bin*. In every case, context will make clear which meaning is used.
- $W^{(g)} \in \mathbb{R}_+^{F \times K_g}$ is a *dictionary* associated with each source g.
- $H^{(g)} \in \mathbb{R}_+^{K_g \times N}$ is a matrix of *activation* or *amplitude* coefficients associated with each source g.
- The dictionary $W = (W^{(1)}, \dots, W^{(G)}) \in \mathbb{R}_+^{F \times K}$ where $K = \sum_g K_g$ is formed by concatenating the $W^{(g)}$ column by column. Throughout this thesis we will assume that all K_g are equal to some constant Q for simplicity.
- Likewise $H \in \mathbb{R}_+^{K \times N}$ is formed by concatenating matrices $H^{(g)}$ along the row dimension.
- If g is a subset (the cardinality of which is denoted by $|g|$), then x_g is a vector in $\mathbb{R}^{|g|}$ formed of the coefficients of x indexed by g (in the same order, i.e., if $g = \{2, 3\}$ then $x_g = (x_2, x_3)^\top$.
- Subscript $\cdot$ stands for all coordinates along one dimension, so

$$W_{\cdot 1} = (W_{11}, \dots, W_{F1})^\top . \tag{1}$$

- Whenever notations become too heavy we bundle all relevant parameters into a single Θ, e.g., $\Theta = (W, H)$.

[1] There is a difference between *time* frames, which are time intervals in which local Fourier transforms are computed, and frame *operators* which are linear operators used in time-frequency analysis. In any case, which meaning is used will be clear from context.

Chapter 1

Introduction

Structure of this chapter in Section 1.1, we will introduce elementary assumptions on the signals we deal with, evaluation criteria and describe tasks related to audio source separation. In Section 1.2, the analysis/synthesis block of our source separation system will be described : it extracts a spectrogram from the mixed signal, and conversely, maps spectrograms of the estimated sources back into time domain signals. We introduce the Fourier transform and short time Fourier transform (STFT), which allow analyzing the properties of long audio signals in the time domain and the frequency domain simultaneously. Moreover, we explain why nonnegativity is important to build translation invariant representations of the recorded signals.

At the heart of the source separation system lie nonnegative sparse coding and dictionary learning. Source-specific dictionaries are learnt either beforehand on a set of isolated source signals, or directly on the mixed signal (blind source separation). Dictionary learning is a vast topic with applications in neurosciences [Olshausen and Field, 1997], image denoising [Mairal et al., 2010], texture classification [Ramirez et al., 2010], among other topics. In Section 1.3, we will review the main dictionary learning models used in audio source separation, and then describe in more details how and why sparse representations evolved from mixture models to nonnegative matrix factorization, and highlight the need for additional prior knowledge for blind source separation. In order to grasp the computational advantages of NMF, we will also present alternative methods for single-channel source separation relying on clustering methods and tools from computational audio scene analysis, so that the reader can compare the advantages and drawbacks of each approach.

1.1 Mixing assumptions and related tasks

1.1.1 Assumptions on the mixing process

Given G source signals $s^{(g)} \in \mathbb{R}^T$, and one microphone, we assume the acquisition process is well modelled by a *linear instantaneous* mixing model :

$$x_t = \sum_{g=1}^{G} s_t^{(g)} \,. \tag{1.1}$$

It is standard to assume that microphones are linear as long as the recorded signals are not too loud. If signals are too loud, they are usually *clipped*. The mixing process is modelled as *instantaneous* as opposed to *convolutive*. Indeed, when multiple microphones are used($I > 1$), the output x_{it} at each microphone can be modelled by : $x_{it} = \sum_{g=1}^{G} \sum_{s=0}^{+\infty} h_s^{(g,i)} s_{t-s}^{(g)}$, where $h^{(g,i)}$ are impulse responses depending on the relative position of source g to microphone i and the configuration of the room in which signals are recorded. While crucial when using multiple microphones, taking into account convolutive mixing is not as important in the case $I = 1$: in this case, we would recover a linear transformation of each source which can then be processed independently.

In the multiple microphone setting, source separation, also known as the "cocktail party problem", or the un-mixing problem, has given birth to the technique of independent component analysis (ICA)[Comon, 1994, Cardoso, 1998, Hyvärinen, 1999].

1.1.2 Standard metrics for single-channel source separation

Listening tests provide the most valuable insights on the quality of source estimates $\hat{s}^{(g)}$. Purely quantitative criteria, on the other hand, are much less time consuming, and provide a good check of source separation quality before listening tests. The simplest performance criterion is the signal to noise ratio:

$$SNR = 10 \log_{10} \frac{\|\hat{s}\|_2^2}{\|s - \hat{s}\|_2^2} . \tag{1.2}$$

Current standard metrics were proposed in [Vincent et al., 2006] and consist in decomposing a given estimate $\hat{s}^{(g)}$ as a sum :

$$\hat{s}_t^{(g)} = s_{target} + e_{interf} + e_{artif} . \tag{1.3}$$

where s_{target} is an *allowed* deformation of the target source $s^{(g)}$, e_{interf} is an *allowed* deformation of the sources which accounts for the interferences of the unwanted sources, e_{artif} is an "artifact" term that accounts for deformations induced by the separation algorithm that are not allowed. Given such a decomposition, one can compute the following criteria :

$$\text{SDR} = 10 \log 10 \frac{\|s_{target}\|_2^2}{\|e_{interf} + e_{artif}\|_2^2} \qquad \text{(Signal to Distortion Ratio)}$$

$$\text{SIR} = 10 \log 10 \frac{\|s_{target}\|_2^2}{\|e_{interf}\|_2^2} \qquad \text{(Signal to Interference Ratio)}$$

$$\text{SAR} = 10 \log 10 \frac{\|s_{target}\|_2^2}{\|e_{artif}\|_2^2} \qquad \text{(Signal to Artifact Ratio)}$$

	auto	**user**	**baseline**	**self**	**oracle**
SDR1	1.83	8.92	9.41	7.35	18.43
SDR2	1.17	-0.39	-0.61	-7.34	10.82

Table 1.1: Example of source separation results on one audio track

Higher values of the metrics are sought for ($+\infty$ means the true source signal is recovered exactly, $-\infty$ that anything *but* the source signal can be heard in $\hat{s}$). Values of the SDR are hard to interpret, they depend highly on the structure of the signal. The SDR of a proposed method must always be compared to that obtained when using the mixed signal as a source estimate, in order to measure the improvement accomplished rather than an absolute value that is not always meaningful. [1]. In Table 1.1 this is displayed in the column **self**. As we can see, the SDR values obtained by the proposed methods are above that threshold, which means that an improvement was obtained in extracting the source from the mixture. Incidentally, note that the SAR obtained by **self** is always ∞, since the mixed signal x is a linear combination of source signals with no additional noise.

Allowing simple deformations of the target signal is important : for instance, if the estimate $\hat{s}^{(g)}$ was simply a scaled version of the true source signal $\lambda s^{(g)}$ then perceptually the result would be perfect, but the SNR would be $10 \log_{10} \frac{\lambda^2}{(1-\lambda)^2}$ which can be arbitrarily low. However, the SDR and SAR (of *this* source) would still be $+\infty$ because scaling is one of the allowed deformations[2]. Other deformations can be allowed in the `bss_eval` toolbox released by Vincent et al. [2006], which are especially useful in a multichannel setting.

1.1.3 Related tasks

Single-channel source separation may be useful in many different scenarios :

- Fine level decomposition : for each instrument, each note must be extracted as an individual source. In contrast with polyphonic music transcription, recovering source estimates may be useful in this case to perform modifications such as modifying notes or changing the length of a note without changing that of the others.

- Instrument per instrument separation : this is the kind of task proposed in separation campaigns such as MIR-1K or SISEC. While there may be several instruments, in popular songs those are highly synchronized. Due to its particular role in Western popular music, it is of particular interest

[1] For readers with a machine learning background, it would seem more natural to use a random method to evaluate the significance of a method, but in the case of audio source separation, randomly sampled signals yield $-\infty$ SDR so using the mixed signal more relevant

[2] however, since $\sum_g \hat{s}^{(g)} = \sum_g s^{(g)}$ the interference ratios will be low for other sources.

to extract voice from the accompaniment. It might be also interesting to extract solo instruments (e.g., electric guitar in rock and roll recordings).

- "Nonstationary" denoising : while denoising has been a successful application of signal processing since the 1970's, it is limited to stationary noise e.g., ambient noise in room recordings, the spectral properties of which are considered as constant throughout time. In field recordings, on the contrary, or in real-life situations, what we commonly experience as noise may have time-varying spectral properties : wind, traffic, water trickling, etc. In this case, denoising is no longer an appropriate term and source separation methods should be used.

- Source separation can also be used as a pre-processing step for other tasks, such as automatic speech recognition (ASR). When there are sources of interference (other voices for instance), source separation could be used to clean the target speech signal from interferences and feed it to a standard recognizer. This approach to speech enhancement is used by many participants in the Pascal Chime Challenge and allows achieving recognition rates that are close to that of human hearing.

Source separation databases

- **SISEC**[3] : held in 2008,2010, 2011. Several tasks proposed, including Professionally Produced Music Recordings. Short 10 seconds' excerpts from songs are provided as well as original sources as train data. For test data, other songs were released in full. This data set is particularly challenging because instruments change entirely from train to test. Moreover, training samples are very short (10 seconds), so that sophisticated methods are likely to overfit.

- **QUASI**[4] : released in 2012, is very similar to the Professionally Produced Music Recordings task of SISEC, but provides more training data.

- **Chime challenge**[5] : held in 2011, it aims at evaluating the performance of automatic speech recognizers (ASR) in real-world environments. Sentences from the grid corpus are mixed with environmental noise. There are several hours of training data for environmental noise and clean speech. In this setting, source separation algorithms were key in providing reliable estimates of clean speech on test data.

- **NOIZEUS dataset**[6] : this corpus has thirty short English sentences (each about three seconds long) spoken by three female and three male speakers.

[3]http://SISEC.wiki.irisa.fr/tiki-index.php
[4]http://www.tsi.telecom-paristech.fr/aao/en/2012/03/12/quasi/
[5]spandh.dcs.shef.ac.uk/projects/chime/PCC/introduction.html
[6]http://www.utdallas.edu/ loizou/speech/noizeus/

1.2 Time-frequency representations of sound signals

In this short section we present as much mathematical content as necessary for the reader to understand their essential properties, in particular the short time Fourier Transform. A complete presentation of time-frequency representations can be found in books such as [Kovačevič et al., 2012, Mallat, 2008].

1.2.1 Fourier Transform

Given a time signal $x \in \mathbb{R}^T$, the Fourier transform of x is defined as :

$$\hat{x}_k = \sum_{t=0}^{T-1} x_t \exp(-i\frac{2\pi kt}{T}) \qquad k = 0 \dots T-1 \,. \tag{1.4}$$

$\hat{x}$ has Hermitian symmetry around $T/2$:

$$\forall k, \hat{x}_{T-k} = \hat{x}_k^\star \quad k = 0 \dots T-1 \,. \tag{1.5}$$

Hermitian symmetry compensates the fact that $\hat{x}$ lives in $\mathbb{C}^T$ which has twice as many dimensions as $\mathbb{R}^T$.

Coefficient k of the Fourier transform describes the contribution of a sinusoid at frequency $f_s * k/T$, from $0Hz$ to the Shannon-Nyquist rate $f_s/2Hz$. In sound signals, coefficient $k = 0$ is always zero, and Fourier coefficients decay fast with k, provided there are no discontinuities in the signal (which happens at notes onset and offset in music signals, at the beginning and end of utterances in speech, and so on).

The Fourier transform is invertible and conserves energy, i.e.,

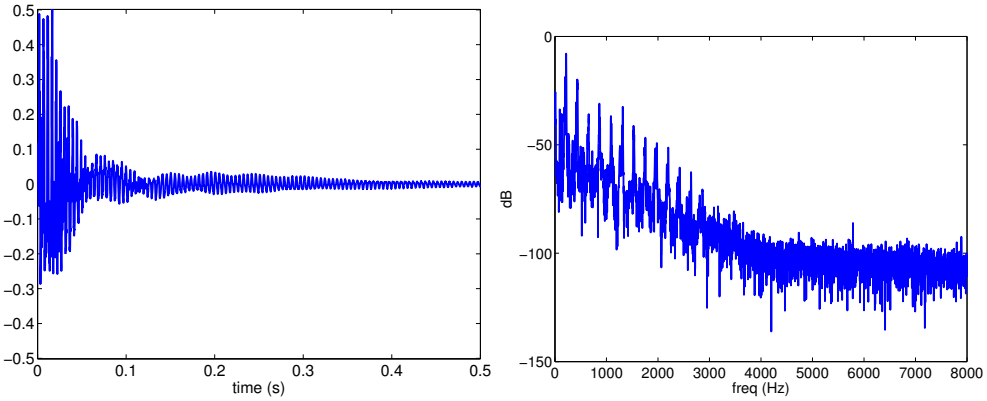
$$\forall x, \|x\|^2 = \|\hat{x}\|^2 \tag{1.6}$$

so the contribution of all sinusoids is sufficient to describe the whole signal.

Note that a circular translation of x does not change the modulus of its Fourier coefficients : thus if we keep only magnitude coefficients $|\hat{x}|$ we obtain a translation invariant representation of x.

It is particularly relevant for sound signals at time scales of the order of a few tens of milliseconds, who typically have few nonzero coefficients in the Fourier domain, but are very dense in the time domain. This sparsity property is exploited in lossy compression schemes such as AAC.

On the other hand, at larger time scales, localized structures such as sequences of vowels in speech or of notes in music cannot be described by the Fourier transform, while they are easily discriminable in the time domain. Trading off time localization versus frequency localization is at the heart of time-frequency representations of signal, as we shall now see.



(a) Time domain (b) Frequency domain[

Figure 1.1: Time vs frequency representation of sounds. The magnitude of Fourier coefficients is displayed on the right plot.

1.2.2 Short time Fourier transform

Time-frequency representations are a compromise between time and frequency localized properties of sound signals. In this thesis, we use a well-known time-frequency representation, called the short time Fourier Transform (STFT).

The STFT is computed in two steps : first the signal is cut into smaller pieces and multiplied by a smoothing window. The Fourier transform of each $x^{(n)}$ is then computed and those are stacked column-by-column into $X \in \mathbb{R}^{F \times N}$. These operations may be summed up in the following formula :

$$\mathcal{S}(x)_{f,n} = \sum_{t=0}^{F-1} w_t x_{nL+t} \exp(-i\frac{2\pi ft}{F}) \,. \tag{1.7}$$

where $w \in \mathbb{R}^F$ is a smoothing window of size F, L is a shift parameter (also called hop size), or equivalently $H = F - L$ is the overlap parameter.

Thus each column of X gives a complete description of the frequency content of x locally in an interval of the form $[nL, nL + F]$.

The STFT operator $\mathcal{S}$ has several important properties :

- it is linear, i.e. : $\forall x, y \in \mathbb{R}^T, \forall \lambda, \mu \in \mathbb{R}, \mathcal{S}(\lambda x + \mu y) = \lambda \mathcal{S}x + \mu \mathcal{S}y$.
- Suppose that

$$\forall t, \sum_{n=-\infty}^{\infty} w_{t-nL}^2 = 1 \,. \tag{1.8}$$

  where $w_t = 0$ if $t < 0$ or $t \geq T$, by convention. Then, the following reconstruction formula holds :

$$x_t = \frac{1}{F} \sum_{n=0}^{N-1} \sum_{f=0}^{F-1} (\mathcal{S}x)_{fn} w_{t-nL} \exp(i\frac{2\pi ft}{F}) \,. \tag{1.9}$$

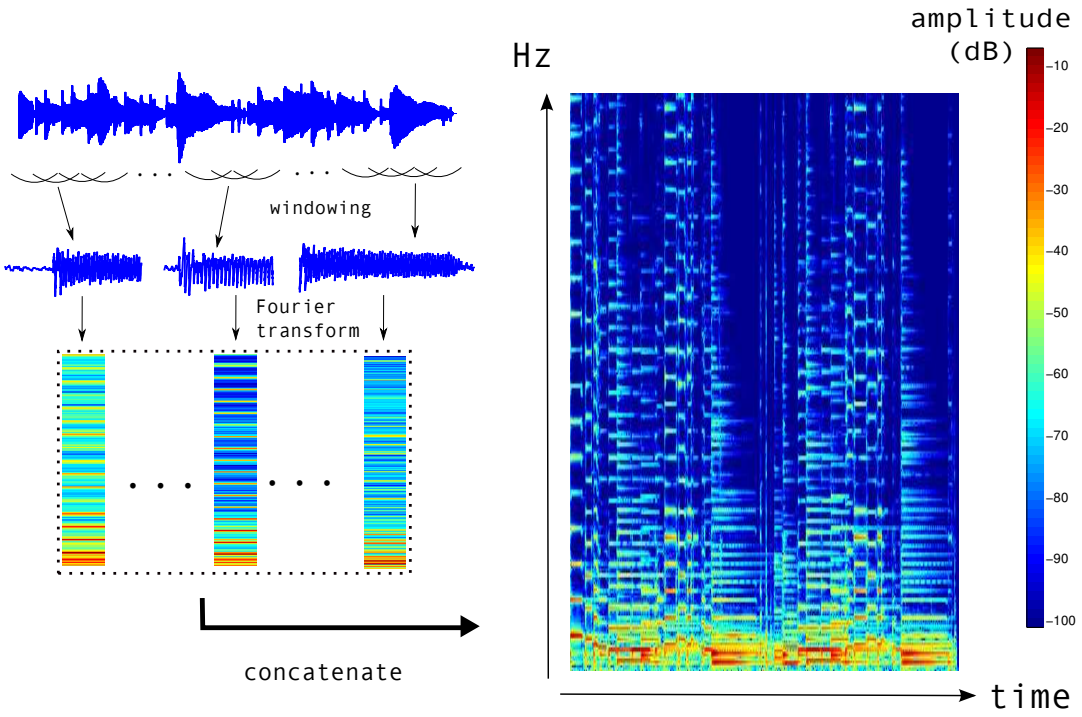


Figure 1.2: Illustration of the short time Fourier Transform.

More concisely, let $\mathcal{S}^\top$ be the conjugate transpose of $\mathcal{S}$. We then have $\mathcal{S}^\top\mathcal{S} = FI_d$, where $I_d \in R^{T\times T}$ is the identity matrix. A generalization of condition 1.8 exists if the analysis window in 1.7 and the synthesis window in 1.9 are not equal. In the rest of this thesis, $\mathcal{S}^\dagger$ will stand for the inverse short time Fourier transform. Note that the term inverse is not meant in a mathematical sense, we will come back to this point later.

Remark 1. *In this thesis, we use sinebell windows for which 1.8 holds as long as* $L \leq \frac{F}{2}$ *:*

$$w_t = \begin{cases} \sin(\frac{\pi}{2}\frac{t-1/2}{H}) & if \quad 0 \leq t \leq H-1 \\ 1 & if \quad H \leq t \leq F-H-1 \\ \sin(\frac{\pi}{2}\frac{F-1-t-1/2}{H}) & if \quad F-H \leq t \leq F-1 \end{cases} \tag{1.10}$$

Once the STFT is computed, phase coefficients are discarded and either the magnitude coefficients $|X_{fn}|$ or the squared power $|X_{fn}|^2$ is kept. As noticed in the last Section, the magnitude of the Fourier transform $|\hat{x}|$ is invariant by circular translation. As for the magnitude spectrogram, only translations by a multiple of the shift parameter preserve it strictly. For smaller shifts, the discrepancy between the magnitude spectrogram of the shifted signal and that of the original is small if the signal is stationary inside each analysis window : it is close to 0 where the signal is sinusoidal, and the largest discrepancies are found at transients.

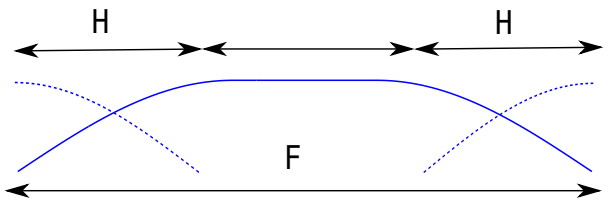


Figure 1.3: The sinebell window fulfills 1.8 and allows up to 50% overlap.

On the other hand, enforcing translation invariance in linear models is so costly in terms of computational resources that, on the whole, the magnitude spectrogram is widely accepted as a convenient tool to provide approximate translation invariance at minimal cost in terms of accuracy.

1.2.3 Recovery of source estimates via time-frequency masking

Given estimates of the source power spectrograms $\hat{V}^{(g)}$, a naive reconstruction procedure would consist in keeping the same phase as the input mixture for each source

$$\hat{S}_{fn}^{(g)} = \sqrt{\hat{V}_{fn}^{(g)}} \exp(i\phi_{fn}) \,. \tag{1.11}$$

where ϕ_{fn} is the phase of spectrogram X (modulo $[0, 2\pi]$). However, source spectrograms estimates are often noisy due to over-simplifying assumptions, suboptimal solutions, etc. Using the reconstruction formula 1.11 would imply in particular that source estimates do not add to the observed spectrogram $\sum_g \hat{S}_{fn}^{(g)} \neq X$.

Instead, it is preferable to compute source estimates by filtering the input :

$$S_{fn}^{(g)} = M_{fn}^{(g)} X \qquad \text{where } M_{fn}^{(g)} = \frac{\hat{V}_{fn}^{(g)}}{\sum_g \hat{V}_{fn}^{(g)}} \,. \tag{1.12}$$

We will show in Section 2.3.2 that if a Gaussian model is assumed for the source spectrograms $S^{(g)}$, then this formula corresponds to computing Minimum Mean Square Estimates of the sources. These particular coefficients $M_{fn}^{(g)}$ will be referred to as Wiener masking coefficients, or oracle coefficients : indeed, for each time frame n and each source g, $M_{fn}^{(g)}$ may be interpreted as the f-th Fourier coefficient of a linear filter. Linear filters given determined by Formula 1.12 were derived by Wiener to estimate clean signals corrupted by Gaussian white noise.

Other probabilistic models imply different recovery formulae, see [Benaroya and Bimbot, 2003] for a discussion.

Ideal binary masks also work surprisingly well :

$$M_{fn}^{(g)} = \begin{cases} 1 & \text{if } V_{fn}^{(g)} > \max_{g' \neq g} V_{fn}^{(g')} \\ 0 & \text{otherwise} \end{cases} \tag{1.13}$$

1.3 Models for audio source separation

Once a time-frequency representation has been computed, latent variable models are used to estimate the contribution of putative sources to the observed mixed signal. In this thesis we assume that the number of sources is known as well as the source types (voice, instrument, environmental noise), although the source signals are not. Latent variable models capture typical sounds emitted by each source in a compact model called a dictionary. Given a mixed signal, the most plausible combination of dictionary atoms of each source are searched for, and used to estimate source spectrograms.

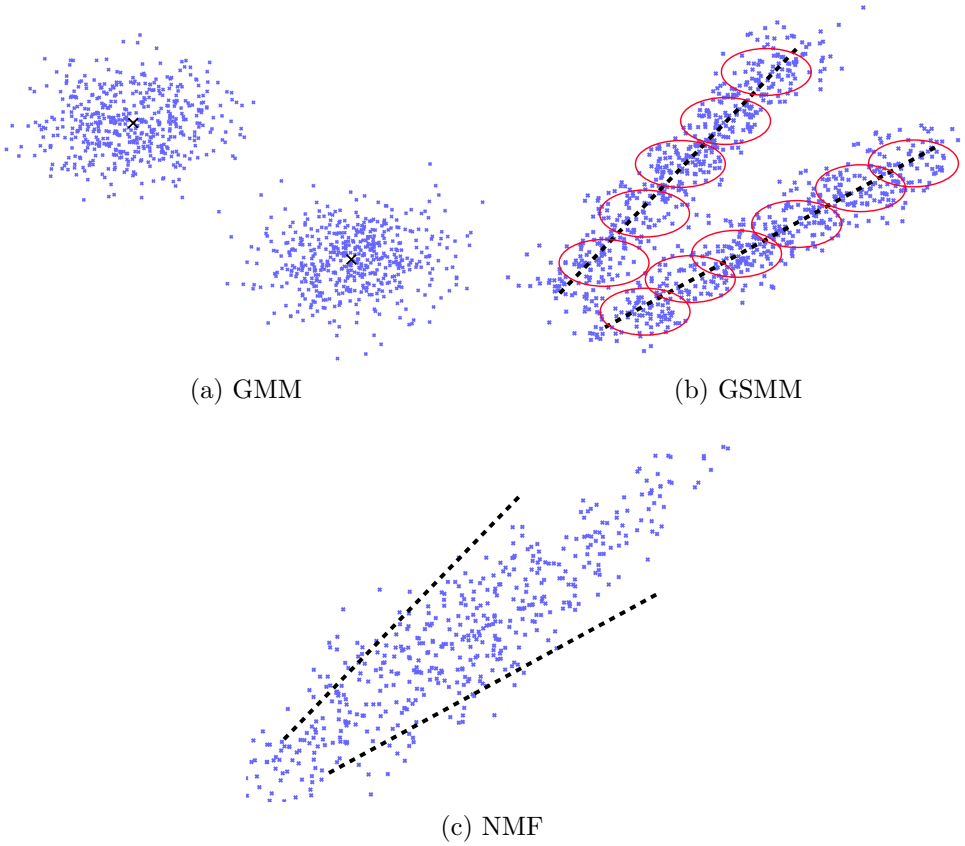
The first latent variable models for single-channel source separation where based on independent component analysis [Casey and Wetsner, 2000, Jang et al., 2003] (ICA). Note however that those works differ from classical approaches of ICA (see [Cardoso, 1998, Comon, 1994, Hyvärinen, 1999]), which require more channels than sources. In [Casey and Wetsner, 2000], each frequency in the STFT operator is considered as a channel. In this case, ICA can be viewed as an instance of a matrix factorization problem, sharing similarities with NMF but requiring different assumptions on the source signals. In [Jang et al., 2003], a single-channel recording is broken into a collection of 25 milliseconds' long segments, without aplication of the Fourier transform. Those are passed as input to an ICA algorithm that enforces a translation-invariant representation of the signals.

At the same time, mixture models where proposed by [Roweis, 2001] to model the nonstationarity of source signals. While ICA may be seen as a matrix factorization technique similar in spirit to PCA, and is widely used for multichannel source separation, it relies on the assumption that source spectrograms are independent. In many audio signals such as music signals, this assumption is incorrect, since several instruments may play notes which are very similar at similar times. NMF was then introduced as a natural way to circumvent this problem while keeping the idea of a low-rank approximation of the observed spectrogram. It was first presented as a tool for polyphonic transcription [Smaragdis and Brown, 2003], and was intensively studied in the following years.

In this Section, we will show how NMF may be seen as an extension of mixture models and outline the main challenges in learning appropriate models for source separation. In early contributions, one model was trained for each source in isolation and then models were combined at test time to infer the state of each source. These models proved successful in controlled experimental conditions, but in real-world source separation benchmarks, learning models directly on mixed data became primordial as training data is scarce and sometimes missing.

We begin this section by presenting mixture models with a special emphasis on the Gaussian (scaled) mixture model (GSMM) proposed by [Benaroya and Bimbot, 2003, Benaroya et al., 2006] for audio source separation and an application of hidden Markov models (HMM) proposed by [Roweis, 2001]. We then show that nonnegative matrix factorization may be seen as a relaxation of GSMM

where the sparsity of each source is no longer fixed to one.



(a) GMM (b) GSMM

(c) NMF

Figure 1.4: Data points generated from each model with the same basis elements.

1.3.1 Mixture models : hard sparsity constraints

1.3.1.1 Inference

Latent variables $H^{(g)} \in \{0,1\}^{K_g \times N}$ represent the state of each source at a given time bin. A global matrix $H = ((H^{(1)})^\top, \ldots, (H^{(G)})^\top)^\top$ is created by concatenating $H^{(g)}$ row by row.

To each source is associated a dictionary of template spectra $W^{(g)} \in \mathbb{R}_+^{F \times K_g}$. Each column of $W^{(g)}$ is interpreted as a typical spectrum observed in source g. Since the class of sources encountered in this thesis are quite general (voice, guitar, piano, etc.), it is reasonable to assume that each source emits several typical spectra, the collection of which corresponds to $W^{(g)}$.

These dictionaries are concatenated column by column to yield

$$W = (W^{(1)}, \ldots, W^{(g)}) \in \mathbb{R}_+^{F \times K} \tag{1.14}$$

where $K = \sum_g K_g$. Throughout this thesis we will assume that all K_g are equal to some constant K without loss of generality.

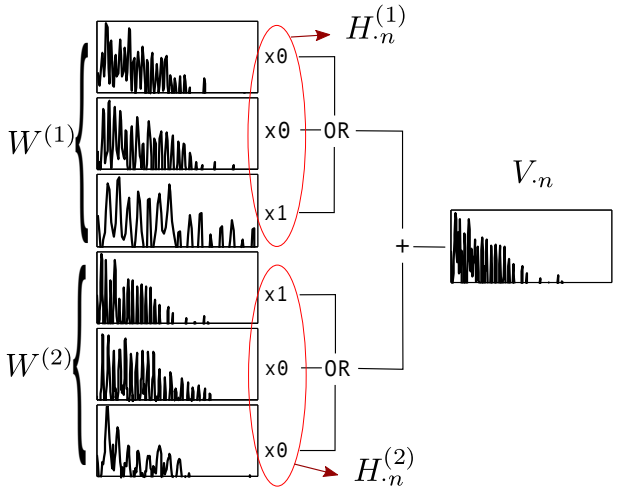


Figure 1.5: Graphical representation of the mixing process in a mixture model : the observed output is modelled as a combination of one and only one atom per source

Each column of the spectrogram is then modelled as a linear combination of columns of W :

$$\hat{V}_{fn} = \sum_{g=1}^{G} \sum_{k=1}^{K} W_{fk}^{(g)} H_{kn}^{(g)} = \sum_{k=1}^{K} W_{fk} H_{kn} \,. \tag{1.15}$$

One and only one column of each $W^{(g)}$ contributes to the output. Assuming an i.i.d. generative model of the output , maximum-likelihood inference of H reads :

$$\begin{array}{cc} \min & -\sum_{fn} \log p(V_{fn}|\hat{V}_{fn}) \,. \\ \text{subject to} & \|H_{\cdot n}^{(g)}\|_0 = 1 \\ & H \in \{0,1\}^{K \times N} \end{array} \tag{1.16}$$

Example 1. *If* $\hat{V}_{fn} \sim \mathcal{N}(\sum_{g=1}^{G} W_{f z_n^{(g)}}^{(g)}, \sigma^2)$, *then* $-\log p(V_{fn}|\hat{V}_{fn}) = \frac{1}{2\sigma^2}\|V_{fn} - \hat{V}_{fn}\|^2$.

Example 2. *If* $\hat{V}_{fn} \sim Exp(\sum_{g=1}^{G} W_{f z_n^{(g)}}^{(g)})$, *then* $-\log p(V_{fn}|\hat{V}_{fn}) = \frac{V_{vn}}{\hat{V}_{fn}} + \log \hat{V}_{fn}$. *This model was used by Benaroya and Bimbot [2003] in the context of audio source separation, and the connexion with exponential models and multiplicative noise was observed in [Févotte et al., 2009].*

Prior knowledge Additional knowledge about the latent variables may be used by assuming a *prior* distribution of the latent variables $p(H)$. In this case,

since they are binary variables, if we assume that sources are independent and i.i.d., $p(H) = \prod_n \prod_g \prod_k (p_k^{(g)})^{H_{kn}^{(g)}}$.

Maximum-likelihood[7] estimates of H are then computed by solving :

$$\begin{array}{cc} \min & -\log p(V_{\cdot n}|\hat{V}_{fn}) - \log p(H_{\cdot n})\,. \\ \text{subject to} & \|H_{\cdot n}^{(g)}\|_0 = 1 \\ & H \in \{0,1\}^{K\times N} \end{array} \tag{1.17}$$

With or without prior knowledge, solving for H is a combinatorial problem, because it involves evaluating the objective function for all G-uplets of states $(k_1, \ldots, k_G)$ and keeping that with lowest values. The cost is of order $O(FK^GN)$ in time and $O(FNGK)$ in space.

Scaling factors As noticed in [Benaroya et al., 2006], the number of components be may reduced by introducing scaling factors so that still only one component H_{kn} is active at a time, but it is allowed to take values in $\mathbb{R}_+$ to compensate for amplitude modulations in the spectrogram (typically a 5 seconds' second piano notes with very slow damping would require a linear number of components to quantize the variations of intensity whereas only one or two components suffice if scaling is allowed). We sketch this idea in Figure 1.6, where a scaled mixture model captures the whole data set with two components (dashed black lines), whereas mixture models need six components (red circles).

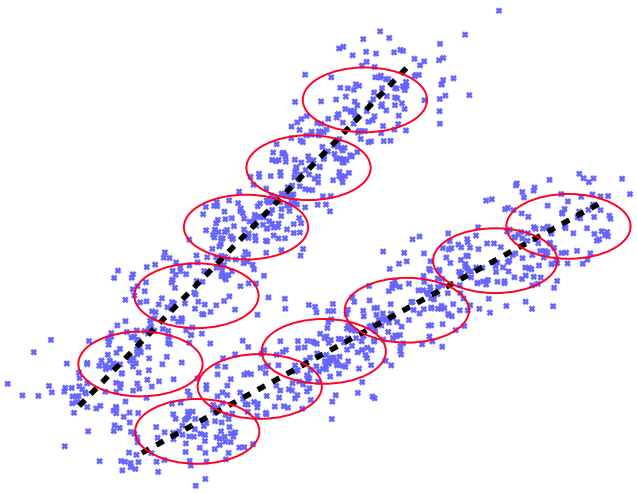


Figure 1.6: Adding a scaling factor allows reducing the number of components dramatically.

[7] Maximum A Posteriori when prior knowledge is added.

This amounts to dropping the binary assumption so the inference problem becomes

$$\begin{array}{cc} \min & -\log p(V_{\cdot n}|\hat{V}_{fn}) - \log p(H_{\cdot n})\,. \\ \text{subject to} & \|H_{\cdot n}^{(g)}\|_0 = 1 \\ & H \geq 0\,. \end{array} \tag{1.18}$$

Solving for H in 1.18 is still of order $O(FQ^GN)$, but with a much higher multiplicative constant since for each G-uplet of states $(k_1, \ldots, k_G)$, a nonnegative matrix division problem must be solved. There is a tradeoff to make between the decrease in the number of components Q and that multiplicative constant.

1.3.1.2 Learning: trained models or blind learning ?

The key to successful inference is that columns of $W^{(g)}$ should provide good estimates of spectrograms generated by source g and bad estimates of other sources. As proposed in [Benaroya and Bimbot, 2003, Roweis, 2001], the first part of this statement is achieved by optimizing the likelihood of isolated samples from source g, with respect to $W^{(g)}$. However, when benchmarks were introduced (SASSEC,SISEC, Chime, RWC), participants submitted mostly source separation systems where models where learnt partially or completely on mixed data, because training samples are sometimes missing or inadequate to represent mixed signals (variability in between instrument classes, in between singers for instance, usage of linear/nonlinear effects on instruments in some mixed signals, etc.). Blind learning of $W$is a non-convex problem involving continuous variables W and discrete variables H.

$$\begin{array}{cc} \min & -\log p(V_{\cdot n}|\hat{V}_{fn}) - \log p(H_{\cdot n})\,, \\ \text{subject to} & \|H_{\cdot n}^{(g)}\|_0 = 1\,, \\ & W \geq 0, H \geq 0\,. \end{array} \tag{1.19}$$

It is solved by an EM algorithm, which has the attractive property of being a descent algorithm [Dempster et al., 1977]. However, while for many choices of $p(H)$ and $p(V|\hat{V})$, inference in H is a convex problem, it is no longer the case when (W, H) are estimated jointly. This entails that there are many stationary points of the problem, and that the solution found by the EM algorithm will highly depend on the chosen initial point. In practice, several initial points are tried and that with the lowest objective cost function is kept.

1.3.1.3 Extension to hidden Markov models

In the early days of speech recognition, dynamic time warping (DTW) emerged as an essential tool for accurate recognition of phonemes [Sakoe and Chiba, 1978]. It was then superseded by hidden Markov models (HMM). Training HMMs for single-channel source separation was proposed in [Roweis, 2001]. In principle, a (factorial) HMM consists in introducing a probabilistic model for H with the

following minus log-likelihood :

$$P_{HMM}(H) = -\sum_{g}\sum_{k=1}^{K} H_{k1}^{(g)} \log p_k^{(g)} - \sum_{n=1}^{N-1}\sum_{g}\sum_{k,k'} H_{kn}^{(g)} \log P_{kk'}^{(g)} H_{k'n+1}^{(g)} \,. \quad (1.20)$$

where

$$p_k \geq 0\,, \quad \sum_k p_k^{(g)} = 1\,, \quad P_{kk'}^{(g)} \geq 0 \quad \sum_{k'} P_{kk'} = 1\,. \quad (1.21)$$

This term replaces static prior information $-\log p(H)$ in 1.17. The stochastic matrix $P^{(g)}$ describes the most likely pairwise sequences of states $(H_{kn}^{(g)}, H_{kn+1}^{(g)})$. $p^{(g)}$, the a priori distribution of initial states, should be adapted to provide the best fit to data. Figure 1.7 provides a graphical representation of the conditional independencies induced by this optimization problem. Inference in hidden Markov models is conducted with a forward-backward algorithm of complexity $O(FK^{2G}N)$, where in each pass, the objective function value must be evaluated for every combination of pairwise states $((k_{1n}, k_{1n+1}), \ldots, (k_{Gn}, k_{Gn+1}))$.

Additional scaling factors may be added without changing the order of complexity, as shown in [Ozerov et al., 2009], at the cost however of a much larger multiplicative constant (typically 10 to 100 depending on the number of sources considered and the correlation of the design matrix W).

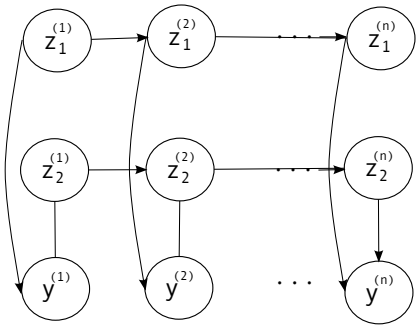


Figure 1.7: Graphical representation of a factorial hidden Markov model.

Note that Markov models of higher order can be introduced to tie together sequences of three, four, states or more, instead of two. However the complexity grows exponentially as $O(FK^{(p+1)G}N)$, where p is the order of the Markov model.

Alternative mixture models The ℓ_2 norm is a poor measure of distortion for audio signals, because the human ear is sensitive to variations in log scale (in dB). Multiplying the amplitude of sound by 10 leads to an increase of 10dB whereas the ℓ_2 norm is sensitive to linear variations. The Itakura-Saito divergence

is introduced in [Févotte et al., 2009] to measure distortion as a function of $\frac{V}{\hat{V}}$ rather than $V - \hat{V}$. We will discuss this choice in more details in Chapter 2.

Roweis [2001] take a different point of view and use the ℓ_2 norm as a measure of distortion while transforming the input data into $\log(V)$.

$$\log V = \log \sum_g V^{(g)} = \log \sum_g \exp(\log V^{(g)}) \,. \tag{1.22}$$

Since the function $\log(\exp(x_1) + \exp(x_2))$ is roughly equivalent to $\max(x_1, x_2)$, [Roweis, 2001] propose replacing + by max in the mixture model, i.e.,

$$\log \hat{V}_{fn} = \max_g \log(\sum_{k \in g} W_{fk}^{(g)} H_{kn}^{(g)}) \,. \tag{1.23}$$

1.3.2 Nonnegative matrix factorization with sparsity constraints

In this Section, we introduce nonnegative matrix factorization. Similarly to the case of mixture models, there are several settings : either one learns a dictionary $W^{(g)}$ on training data for each source, and then uses the concatenated dictionary W to infer decomposition coefficients H on mixed data, or one learns blindly (W, H) directly on mixed data.

The key to successful inference is that $W^{(g)}$ should be good at reconstructing source g (*interpretability*) and bad at reconstructing the others (*incoherence*). We argue in Section 1.3.2.1 that sparsity penalties are needed to learn incoherent dictionaries.

In the case of blind learning, sparsity in itself may not be sufficient to learn interpretable dictionaries and additional prior knowledge is necessary. The easiest case is when a learnt dictionary is provided, and must be re-calibrated on mixed data, rather than be learnt from scratch (Section 1.3.2.2). Taking into account temporal dependencies is important when dealing with speech signals. We present in Section 1.3.2.3 counterparts of the Hidden Markov Model that have been proposed for NMF. Another way of enforcing prior knowledge is through re-parameterization as we will see in Section 1.3.2.4.

1.3.2.1 Trained learning with sparsity

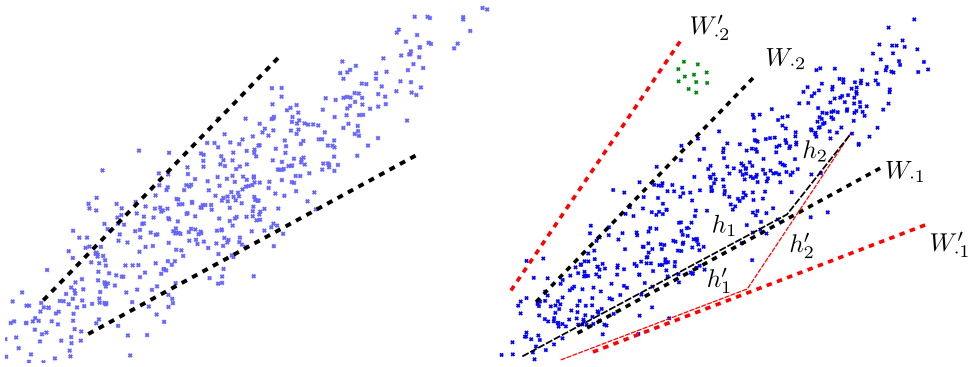
Placing a hard constraint on the number of nonzero coefficients in H entails a complexity of order $O(FK^GN)$. By relaxing this constraint, instead of fitting a mixture of K one-dimensional half-lines, nonnegative matrix factorization consists in fitting a K dimensional cone to the data :

$$\begin{array}{cc} \min & \sum_{fn} - \log p(V_{fn}|\hat{V}_{fn}) \,. \\ \text{subject to} & W \geq 0, H \geq 0 \,. \end{array} \tag{1.24}$$

where $\hat{V}_{fn} = \sum_k W_{fk} H_{kn}$. When only H is optimized for fixed W, this problem is referred to as nonnegative regression or nonnegative matrix division (NMD). We will arbitrarily use the latter name.

With the same number of dictionary elements, a much larger portion of the space is spanned in NMF than in mixture models. However, this advantage comes at a cost. Consider the cloud of points in Figure 1.8b, for instance. Two pairs of dictionary elements $(W_{\cdot 1}, W_{\cdot 2})$ and $(W'_{\cdot 1}, W_{\cdot 2})$ fit the data equally well. However, the cone generated by W' is too big so learning W' on training data might fit well but at test time, it might contain data points from other sources : they would be mistakenly represented as points from source 2.

Selecting model W rather than W' is hard problem. However, in this simple case we can see that, given a data point, the latent coefficients satisfy $h_1 + h_2 < h'_1 + h'_2$.



(a) NMF (b) Need for sparsity constraint

Thus, a reasonable model selection procedure would consist in learning W for good reconstruction while enforcing an upper-bound on the value of $\sum_k H_{kn}$ for every n.

$$\begin{array}{cc} \min & -\log p(V_{\cdot n}|\hat{V}_{fn})\,. \\ \text{subject to} & \sum_k H_{kn} \leq C \\ & H \geq 0\,. \end{array} \tag{1.25}$$

or equivalently learn W to minimize the reconstruction cost penalized by the sum of coefficients :

$$\begin{array}{cc} \min & -\log p(V_{\cdot n}|\hat{V}_{fn}) + \lambda \sum_{k,n} H_{kn} \\ \text{subject to} & H \geq 0\,. \end{array} \tag{1.26}$$

This is a simple illustration of the benefits of sparsity in the particular case of NMF. In the wider scope of dictionary learning (with or without nonnegativity constraints), there are other ways of learning source specific dictionaries, see e.g., [Ramirez et al., 2010].

1.3.2.2 Partially blind learning or model calibration

Blind learning may be too hard in the sense that many local minima exist that do not yield satisfactory source estimates. However, given rough estimates of $(\tilde{W}^{(g)}, \tilde{H}^{(g)})$ from prior training data, one may enforce the additional constraints that estimates computed on new mixed data should be close to $(\tilde{W}, \tilde{H})$. This type of solution can be straightforwardly addressed in a penalized likelihood setting, where prior distributions $p(W|\tilde{W})$ and $p(H|\tilde{H})$ are chosen to be concentrated around $\tilde{W}$ and $\tilde{H}$. For instance, when V is modelled as a multinomial, a Dirichlet prior on W with scale parameter $\tilde{W}$ was used in [Smaragdis, 2009]. If V is modelled as gamma random variable, an inverse gamma prior with scale $\tilde{W}$ could also be used.

$$\min \quad -\log p(V_{\cdot n}|\hat{V}_{fn}) - \log p(W|\tilde{W})\,. \tag{1.27}$$

Additionally, if the generative model is such that prior distributions $p(W|\tilde{W})$ and $p(H|\tilde{H})$ are conjugate with the probability distribution of the data given W, H, then inference in H and W can be addressed straightforwardly.

Following this rule, [Smaragdis, 2009] propose a user guided source separation system where a rough estimate of the voice is provided by the user. This side signal is used to train a model which is then re-calibrated on test data.

1.3.2.3 Additional prior knowledge for blind learning : taking into account temporal dependencies and basis priors

One drawback of NMF is that Markov models can no longer be used to model temporal dependencies. A "brute force" solution consists in learning small sequences of atoms coupled together by a unique gain. Instead of one dictionary W, one then learns L dictionaries ${}^{\rightarrow l}W$ such that the spectrogram model is :

$$\hat{V}_{fn} = \sum_{k} \sum_{l=0}^{L-1} {}^{\rightarrow l}W_{fk} H_{kn-l}\,, \tag{1.28}$$

where we define by convention $H_{kn} = 0$ if $n \leq 0$. This approach, called convolutive NMF, is used in speech separation and speech recognition since it allows learning time-varying spectra which correspond to phonemes.

Another line of work consists in enforcing smoothness in the decomposition coefficients, such as in [Virtanen et al., 2008, Cemgil et al., 2007, Févotte and Cemgil, 2009, Févotte, 2011a]. For instance, [Févotte, 2011a] propose penalty terms of the form :

$$P(H) = \sum_{n=1}^{N} d_{IS}(H_{kn-1}, H_{kn})\,. \tag{1.29}$$

where divergence term $d_{IS}(x, y) \geq 0$ is such that $d_{IS}(x, y) = 0$ if and only if $x = y$.

Of course, since W and H play a symmetric role, enforcing smoothness in the spectral shape of atoms and/or in decomposition coefficients is also possible, see [Dikmen and Cemgil, 2009].

1.3.2.4 Additional prior knowledge for blind learning : model re-parameterization

When the model is learnt blindly on mixed signals, sparsity in itself is not sufficient to yield interpretable dictionaries. Indeed, dictionary learning algorithms are prone to local minima, so either they should be provided with a good enough initialization near the global optimum, or additional constraints must be added to prune irrelevant local minima.

Source/filter models Among several contributions, we discuss here the case of source/filter models, proposed by [Durrieu et al., 2010]. Voice signals can be well approximated at the scale of a few hundreds of milliseconds as a periodic signal, corresponding to the glottal flow, convolved by a filter representing the vocal tract, whose impulse response is short. In the frequency domain, this convolution turns into multiplication of a sparse spectrum with a smooth spectrum. Allowing several periodicity patterns (several pitches) and several vocal tract transfer functions leads to the following representation :

$$V_{fn}^{(voice)} \simeq (W^{(source)}H^{(source)})_{fn}(W^{(filter)}H^{(filter)})_{fn}\,. \tag{1.30}$$

where $W^{(source)}$ and $W^{(filter)}$ have respectively K_1 and K_2 columns. $W^{(source)}$ is given by the KLGLOTT model (see [Durrieu, 2010] for details), while $W^{(filter)}$ is estimated on the training data. Durrieu et al. [2010] consider either inference of $H_{kn}^{(source)}$ with only one atom active at a time, or let all vary.

Note that this model is formally equivalent to estimating a dictionary for the voice $\tilde{W}$ of $K_1 \times K_2$ components with additional constraints on the shape of the dictionary elements :

$$\tilde{W}_{f,(k_1-1)K_2+k_2} = W_{fk_1}^{(source)}W_{fk_2}^{(filter)} \qquad \forall k_1 = 1,\ldots,K_1 \quad k_2 = 1,\ldots,K_2\,. \tag{1.31}$$

By restricting the space of possible values of the dictionary $\tilde{W}$, this parameterization acts as a form of regularization.

1.3.3 Other latent variable models : Complex Matrix Factorization

One problem with NMF is that if two sources are active in the same time-frequency bin, they can never be recovered by NMF, because phase information has been lost. This illustrated in Figure 1.8 : because of phase differences, the magnitude of the observed signal at a given time-frequency bin may be smaller than that of the sources. In this case, even with a perfect fit, NMF would assign smaller magnitude to each source and the same phase.

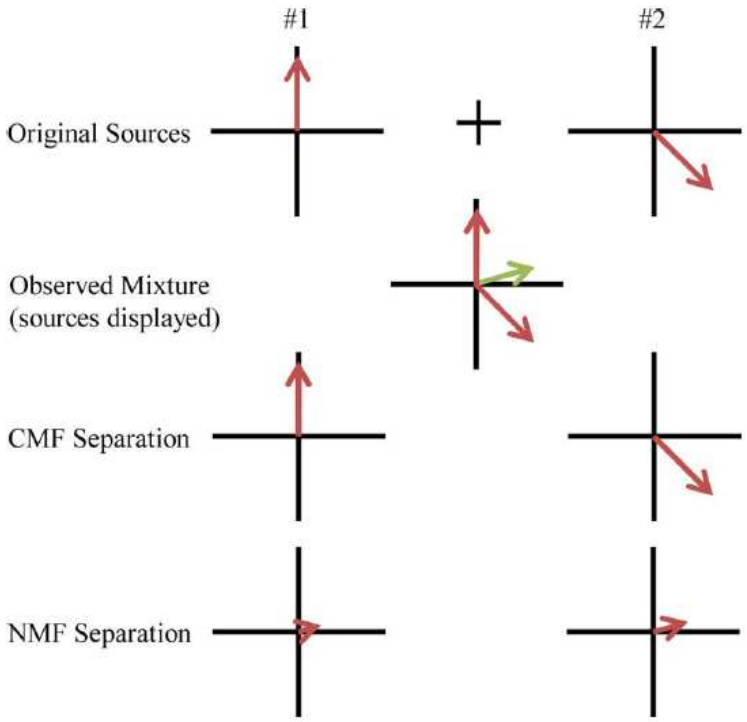


Figure 1.8: How CNMF and NMF differ in estimating individual sources $S_{fn}^{(g)}$ from an observed mixture X_{fn}.*(Figure borrowed from [King and Atlas, 2011])*

This is because additivity of the spectra is not a correct hypothesis in source separation : only additivity of the complex spectrograms holds.

In order to restore this property, [Kameoka et al., 2009] have proposed a new representation of audio signals called complex matrix factorization (CMF).

$$\hat{X}_{fn} = \sum_k W_{fk} H_{kn} \exp^{i\phi_{fkn}} \tag{1.32}$$

and propose to fit this representation to the observed complex spectrogram X_{fn} using the ℓ_2 norm:

$$\min_{W \geq 0, H \geq 0, \phi_{fnk} \in [0, 2\pi]} \sum_{fn} \|X_{fn} - \hat{X}_{fn}\|^2 \tag{1.33}$$

Because KFN phase parameters ϕ_{fnk} are introduced, for a given pair (W, H) there might still be multiple global minima to this problem. Descent algorithms have been proposed, with convergence of the objective cost function. CMF was compared to NMF (with squared loss) in recent work [King and Atlas, 2010, 2011], on an automatic speech recognition task, in which a gain of 10% was observed in word recognition accuracy (in absolute terms).

As we will see in chapter 2, Itakura-Saito NMF is another model that assumes only additivity of the complex spectrograms and not of the power spectrograms. On the other hand, there is no need to introduce phase parameters in Itakura-Saito NMF, which saves a lot of the computational efforts needed in CMF.

1.3.4 Other approaches : Computational Audio Scene Analysis

We discuss here alternative approaches to source separation based on ideas from computational audio scene analysis.

1.3.4.1 Basic complexity issues

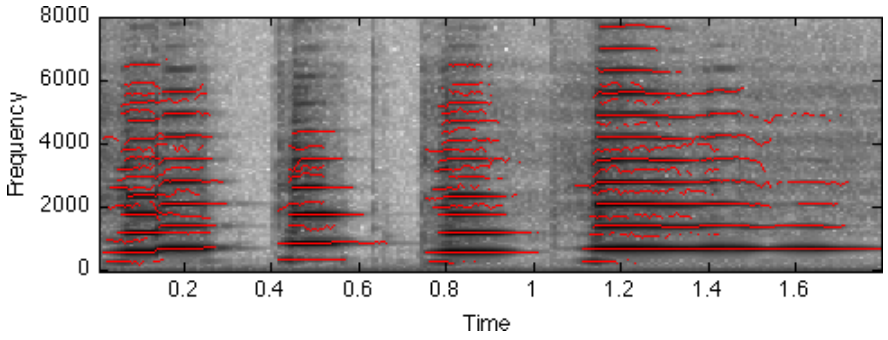


Figure 1.9: Example of partial tracking (from [Ellis, 2003])

Early studies in source separation typically involved mixtures of two speakers [Parsons, 1976, Quatieri and Danisewic, 1990], and relied on sinusoidal modelling.

At each time frame, the g-th source signal is modelled as :

$$x^{(g)}(t) = \sum_{k=1}^{K_g} a_k^{(g)} \cos(\omega_k^{(g)} t + \phi_k^{(g)}) \,, \tag{1.34}$$

where $(\omega_k^{(g)})$ are a set of spectral peaks (also called partials), and for each spectral peak, $a_k^{(g)}$ and $\phi_k^{(g)}$ are the associated phase and amplitude parameters. Only the sum $x = \sum_g x^{(g)}$ is observed, which makes the problem ill-posed. Indeed, while estimating the whole set of instantaneous frequencies is a well-studied problem, deciding which frequency belongs to which speaker is the key issue. Without further modelling, if there are G sources and each source has K spectral peaks per each time frame, then there are G^{KN} possible models of the form 1.34 that yield the same observed signal x.

1.3.4.2 A clustering approach to audio source separation

At the same time as we introduce basic CASA concepts, we will also present here another point of view on blind source separation, based on the principle of

clustering. This approach has been taken in [Bach and Jordan, 2004] to deal with separation of speech signals. We saw earlier in this chapter that source spectrograms were estimated by masking the spectrogram of the mix. Among all, binary masks are particularly simple and surprisingly, ideal binary masks provide excellent source separation results, both subjectively and quantitatively. Thus, "tagging" together time-frequency bins belonging to the same auditory object is sufficient to recover good source estimates.

"Fortunately, in audition (as in vision), natural signals exhibit a lot of regularity in the way energy is distributed across the time-frequency plane. Grouping cues based on these regularities have been studied for many years by psychophysicists and are hand built into many CASA systems. Cues are based on the idea of suspicious coincidences. Upward/downward sweeps are more likely to be grouped into the same stream. Also, many real world sounds have harmonic spectra so frequencies which lie exactly on a harmonic "stack" are often perceptually grouped together. "[8] Actually, beyond harmonicity multi-pitch detectors are essential for CASA methods to succeed, as multiple harmonic stacks may be present if two speakers speak simultaneously or if several instruments play a note at the same time.

Source separation can thus be formulated as a problem of segmentation in the time-frequency plane. This problem has been a field of intense study in vision, with now mature procedures such as graph cuts, and normalized cuts [Shi and Malik, 2000], which we now present briefly.

For simplicity, time frequency bins will be indexed by i where $i - 1 \dots I$ and $I = FN$. Given pairwise similarity measures M_{ij} between time-frequency points i and j, the similarity matrix $M \in \mathbb{R}^{I \times I}$ is normalized and its first two eigenvalues and eigenvectors are computed. Then, forming a $I \times 2$ with these eigenvectors as columns, we cluster the I rows of this matrix as points in $\mathbb{R}^2$ using K-means. These clusters define the final partition.

There are two main difficulties with this approach : the first is to build a relevant affinity matrix, the second is to deal with the size of the matrix which is huge ($I = 10^6$ for a ten seconds' signal). Given J affinity matrices $M_1, \dots, M_J$ built each on different cues, [Bach and Jordan, 2004] proposes using combinations of cues of the form :

$$M = \sum_{k=1}^{K} \lambda_k M_1^{\alpha_{k_1}} \times \cdots \times M_J^{\alpha_{k_J}} \tag{1.35}$$

where products are taken component-wise. Intuitively, if the entries of M_j are thought of as (soft) boolean variables, then taking products amounts to an AND operation, while sums amount to an OR operation, so if K is taken large enough, M can represent any logical procedure to build an affinity measure from $M_1, \dots, M_J$. Additionally, the combination may be optimized given training examples of binary masks computed on mixed signals for which sources are known.

[8] [Roweis, 2001]

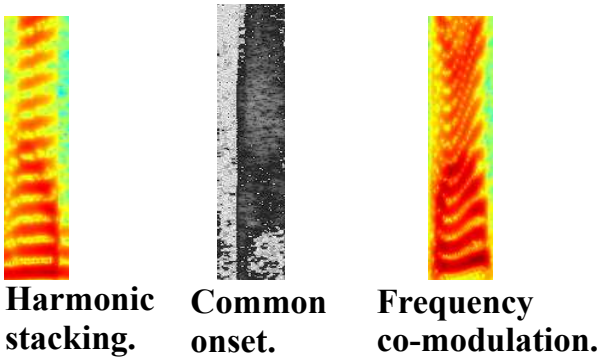


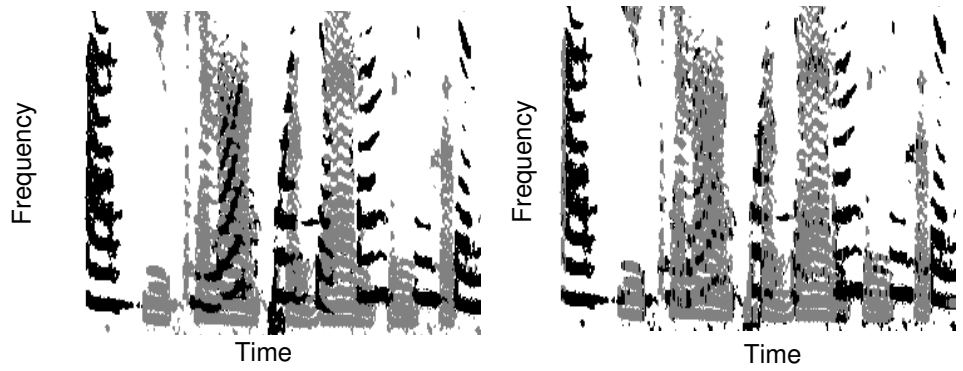
Figure 1.10: Example of auditory grouping cues in computational auditory scene analysis. (Figure from [Roweis, 2001])

For fixed affinity matrix M, the complexity of normalized cuts is dominated by that of computing two principal eigenvectors, which involves matrix-vector operations of order $O(I^2)$ where I is roughly equal to twice the number of samples of the test signal. A remedy is to compute affinity measures such that points too far apart in the time-frequency plane always have zero similarity : then the number of non-zeros per row of M is never more than a fixed amount L and matrix-vector multiplications $O(LI)$. Following this idea, [Lagrange et al., 2008] propose to subdivide spectrograms into short segments of a few seconds, cluster each separately, and then ensure coherence between labels in each segment by hand. Additionally, they compute affinity matrices only for time-frequency points located at spectral peaks.

Clustering methods yield promising results for unsupervised audio source separation, however they cannot yet identify binary masks correctly when two sources are active in the same time-frequency region (see Fig 1.11).

1.4 Conclusion

We have outlined in this chapter the main building blocks of a source separation system. At the core of the system lie latent variable models which may be either optimized on a training set or blindly on mixed signals. While mixture models were initially proposed for that purpose, nonnegative matrix factorization penalized (or constrained) by sparsity penalties naturally extend them while providing more efficient algorithms in $O(FGQN)$ instead of $O(FQ^GN)$. For trained models, sparsity is crucial in estimating source-specific dictionaries that correlate most with their target source and least with potential interfering sources. We refer to this property as *interpretability*. [Ramirez et al., 2010], in the context of classification, propose learning dictionaries that are specific enough for each class and at the same time share features. In addition to sparsity, the authors propose



(a) (b)

Figure 1.11: (Left) Optimal segmentation for the spectrogram in Figure 1 (right), where the two speakers are "black" and "grey;" this segmentation is obtained from the known separated signals. (Right) The blind segmentation obtained by [Bach and Jordan, 2004] *(Figure reproduced from there).*

a penalty that promotes incoherent dictionaries and show that the reconstruction cost may be used as a heuristic for multi-class classification.

In recent benchmarks, re-calibration or even blind learning of models appears necessary, owing to high intra-class variability. In this, we have emphasized the need for additional constraints, either in the form of sparsity penalties on the decomposition coefficients or parameterization of the dictionary.

Intensive research efforts have then been spent on extending sparsity penalties to smoothness penalties. These mimic the ability of factorial scaled hidden Markov models to take into account temporal dependencies between latent variables, which are essential in modelling speech and music. Hidden Markov Models were introduced for NMF in [Mysore et al., 2010] to learn models of speech with state persistence.

In the wider scope of sparse representations (with or without nonnegative constraints), efficient algorithms for structured sparsity-inducing penalties have been recently proposed by [Jenatton et al., 2011c, Mairal et al., 2011] to model general patterns of dependencies between latent variables. Structured decompositions tailored to audio applications have also been proposed in [Daudet, 2006] for the task of coding. Sparse representations yield state-of-the art performance in audio restoration tasks such as inpainting [Adler et al., 2012]. A complete survey of the applications of sparse audio representations can be found in [Plumbley et al., 2010].

Dictionary learning with sparsity is now an established research topic, with several branching paths : study of the stability of local minima for square dictionaries [Gribonval and Schnass, 2010], or overcomplete dictionaries in the presence of noise[Geng et al., 2011, Jenatton et al., 2011b], applications to image restoration [Mairal et al., 2010]. In the field of audio, dictionary learning and especially

nonnegative matrix factorization obtain state-of-the art results in polyphonic transcription. Denoising of audio signals with learnt dictionaries has been the subject of recent work [Jafari and Plumbley, 2009]. In the context of music/voice separation, Sprechmann et al. [2012] propose a nonnegative version of robust principal component analysis to learn a decomposition of the spectrogram into a low-rank component for the musical accompaniment, and a sparse component for the vocal part.

As we have seen in Section 1.3.4, there are alternative approaches to dictionary learning for audio source separation. Among those are contributions coming from computer vision that have been successfully transposed to audio signals. In particular, in the context of music/voice separation, Rafii and Pardo [2011] use a background/foreground segmentation technique.

Chapter 2

Structured NMF with group-sparsity penalties

In this chapter, we first study algorithms for NMF in more details. We first outline algorithms for NMF with a general family of loss functions called beta-divergences and emphasize the importance of multiplicative updates when using other divergences than the Euclidean loss.
Our main contribution is a group-sparsity penalty which is adapted to Itakura-Saito NMF. Unlike mixed norms used with the Euclidean loss, we advocate concave penalty terms. Concavity is important because it allows keeping a multiplicative updates algorithm. Simple temporal dependencies may be enforced in the form of a block-sparsity penalty, which favors contiguous zeroes in the decomposition coefficients. We also contribute to model selection in matrix factorization problems, by proposing a criterion to select the number of components and the penalty parameter. Our criterion is a competitive alternative to cross-validation and may be used out of the box for as soon as a probabilistic model of the data is provided.
This work has led to the following publication(s):

> A. Lefèvre and F. Bach and C. Févotte, "Itakura-Saito Nonnegative Matrix Factorization with group sparsity", *in Proceedings of the International Conference on Acoustique Speech and Signal Processing (ICASSP)*,2011.
>
> A. Lefèvre and F. Bach and C. Févotte, "Factorisation de matrices structurée en groupes avec la divergence d'Itakura-Saito", *in Proceedings of 23e colloque GRETSI sur le Traitement du Signal et des Images*, 2011.

Audio demonstrations are available online[a].

[a] www.di.ens.fr/ lefevrea/demo-group.html

Nonnegative matrix factorization is an instance of the more general problem of dictionary learning. This problem has practical applications in neurosciences, image processing and audio processing, but also in text analysis. The most basic instance of dictionary learning is Principal Component Analysis: computing the SVD of an input matrix $X = USV^\top$ and extracting the K principal eigenvectors may be interpreted as cleaning X from noise (under a Gaussian generative model [Tipping and Bishop, 1999]) and interpreting the K principal eigenvectors as latent factors. In image processing, dictionary learning has showed state-of-the-art results in denoising experiments [Mairal, 2011, Aharon et al., 2005]. In the field of audio signal processing, subspace tracking methods were introduced for high-resolution tracking of partials in harmonic signals [Badeau et al., 2004].

The analogy between dictionary learning may be extended further than the Euclidean loss. As observed in [Buntine, 2002], topic modelling may be cast as multinomial PCA: the term document-document matrix X is factored into WH where W is a matrix of latent topics and H describes each document as a mixture of relevant topics.

Thus, finding factorizations of matrices $X \simeq WH$ is a common trait of many methods in machine learning. The key difference is the choice of the loss function used: in topic modelling, the Kullback-Leibler divergence is used to compare distributions of words. As discussed in Chapter 1, NMF for audio signals was originally introduced in the context of polyphonic music transcription [Smaragdis and Brown, 2003]. It is more sensitive than the ℓ_2 loss to relative errors in the frequency counts $\frac{X_{fn}}{(WH)_{fn}}$. As we will argue in this chapter, the Itakura-Saito divergence is an interesting measure of distortion for sounds, among others. Using non-Euclidean measures of distortion implies new optimization problems that are not always as well-behaved as the ℓ_2 norm. We present in Section 2.1 a general class of losses for NMF which includes classical Euclidean loss, Kullback-Leibler divergence and Itakura-Saito divergence. Among this family, we have chosen the Itakura-Saito divergence because it captures the sensitivity observed in the human auditory system (sensitivity in log scale).

In Section 2.2, we outline the main challenges in optimizing NMF with beta divergences. Multiplicative updates algorithms are then presented to estimate NMF on a given set of training data. The algorithm covers the optimization of both H and W and may be specialized to optimization of either one when the other is fixed. Thus blind learning and inference of H for fixed W are covered. We compare the multiplicative updates algorithm with a standard projected gradient descent method and discuss the use of expectation-maximization algorithms in special cases where a probabilistic generative model of the data is available.

In Section 2.3, we provide an in-depth study of multiplicative updates for Itakura-Saito NMF, with a discussion of convergence properties and an overview of the complete multiplicative updates algorithm.

Sparsity inducing penalties were introduced in the last chapter as a way to learn interpretable dictionaries while relaxing the hard sparsity constraints imposed by mixture models. Sparse NMF with the Euclidean loss was proposed in [Laurberg et al., 2008b],and for the Kullback-Leibler divergence in [Smaragdis et al., 2007, Virtanen, 2007]. In the case of the Itakura-Saito divergence, a general framework is proposed in [Févotte et al., 2009] to include probabilistic models of the dictionary and/or decomposition coefficients : a cascade of gamma prior distributions mimicking a hidden Markov model is studied, although static priors could also be chosen.

We propose in Section 2.4 an extension of multiplicative updates to account for a group-sparsity inducing penalty. This penalty is designed for audio signals containing intervals where one of the sources is missing. The simplest case is that of a two instrument track divided in three parts: one part with instrument A, one part with instrument B, and a third part where both instruments are mixed. This setting may be generalized to the case of several sources. Provided that for each source, there is at least one interval where this and only this source, it was shown in [Laurberg et al., 2008b] that this setting is equivalent to learning a dictionary on isolated source signals and UN-mixing them when they are mixed. Our group-sparsity penalty allows identifying segments where sources are missing, learn an appropriate dictionary each source, and un-mix sources elsewhere.

In Section 2.5, we discuss model selection in NMF. Indeed, choosing the appropriate number of components and additional parameters such as the strength of the penalty is a hard problem in statistics. In Section 2.5.1, we show that the generative model in Itakura-Saito NMF may be used to derive goodness-of-fit statistics that are sensitive to the parameters used in NMF. In Section 2.6, we validate the user of the Kolmogorov-Smirnov statistics presented in Section 2.5 and the effect the proposed group-sparsity penalty on real music signals.

2.1 The family of NMF problems

Noisy vs exact NMF Given an observed matrix $V \in \mathbb{R}_+^{F\times N}$, nonnegative matrix factorization (NMF) is the problem of finding $W \in \mathbb{R}_+^{F\times K}$,$H \in \mathbb{R}_+^{K\times N}$ such that

$$V = WH\,. \tag{2.1}$$

Either K is fixed or the smallest possible K is sought for.

For real life signals, the NMF model cannot be expected to be exact, given its simplistic formulation, and it is unlikely to find anything other than trivial solutions (e.g., choose $K = F$ and $W = I$ and $H = V$, or vice versa with $K = N$), so one may instead look for approximate solutions:

$$\min_{W\geq 0, H\geq 0} \sum_{fn} d\Big(V_{fn}, (WH)_{fn}\Big)\,. \tag{2.2}$$

where $d : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ is an appropriate measure of similarity[1]. In this thesis we are interested in noisy NMF problems, although we may use exact NMF cases for illustration.

2.1.1 The family of beta-divergences

Given an observed matrix $V \in \mathbb{R}_+^{F\times N}$, nonnegative matrix factorization may be defined for a wide range of divergence measures, as the following optimization problem:

$$\begin{array}{ll} \min & \sum_{fn} d\Big(V_{fn}, (WH)_{fn}\Big)\,. \\ \text{subject to} & W \geq 0, H \geq 0\,. \end{array} \tag{2.3}$$

and the optimization is over $W \in \mathbb{R}_+^{F\times K}$ and $H \in \mathbb{R}_+^{K\times N}$. When W is fixed and only H is optimized we refer to Eq. (2.3) as nonnegative matrix division (NMD). Three choices of divergences are frequently used:

$$\begin{aligned} d(x,y) =& \frac{1}{2}(x-y)^2 \quad \text{the Euclidean (or } \ell_2\text{) norm} \\ d(x,y) =& x\log\frac{x}{y} + y - x \quad \text{the Kullback-Leibler (KL) divergence} \\ d(x,y) =& \frac{x}{y} - \log\frac{x}{y} - 1 \quad \text{the Itakura-Saito divergence} \end{aligned}$$

In any case, divergence d must be chosen such that: $\forall (x,y)$, $d(x,y) \geq 0$ and $d(x,y) = 0 \Rightarrow x = y$. These three commonly used divergences are part of the larger family of β-divergences, defined by:

$$d_\beta(x,y) = \begin{cases} \frac{1}{\beta(\beta-1)}\left(x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}\right) & \text{if } \beta \in \mathbb{R}\setminus\{0,1\}\,, \\ x\log\frac{x}{y} + y - x & \text{if } \beta = 1\,, \\ \frac{x}{y} - \log\frac{x}{y} - 1 & \text{if } \beta = 0\,. \end{cases} \tag{2.4}$$

Note that $d_\beta(x,y)$ is continuous in β at 0 and 1, so that by changing β we can move continuously from the Euclidean norm ($\beta = 2$) to the Itakura-Saito divergence.

The essential features of β-divergences are discussed thoroughly in [Févotte and Idier, 2011]. $d_\beta(x,y)$ is homogeneous of degree β: $d_\beta(\lambda x, \lambda y) = \lambda^\beta d_\beta(x,y)$. It implies that factorizations obtained with $\beta > 0$ (such as with the Euclidean distance or the KL divergence) will rely more heavily on the largest data values and less precision is to be expected in the estimation of the low-power components, and conversely factorizations obtained with $\beta < 0$ will rely more heavily on smallest data values. The IS divergence ($\beta = 0$) is scale-invariant, i.e., $d_{IS}(\lambda x, \lambda y) = d_{IS}(x,y)$, and is the only one in the family of β-divergences to possess this property.

[1]Note that similarity measures between matrices might also be used, however none will be encountered in this thesis

2.1.2 Identification problems in NMF

For every pair (W, H), the cost function value is unchanged under any of the following transformations:

- Permutation: choose a permutation matrix P and set $W' = WP$, $H' = PH$.
- Scaling: choose a diagonal matrix $\Lambda = \mathrm{diag}(\lambda)$ where $\lambda \in (\mathbb{R}_+^\star)^K$ and set $W' = W\Lambda$, $H' = \Lambda H$.
- Dilation: see below.

Scaling indeterminacy is often solved by constraining the columns of W to sum to 1. In practice, these indeterminacies must be taken into account when comparing two pairs (W, H) and (W', H'). They can be resolved by constraining the columns of W to sum to 1, and sorting them in lexicographic order (sorting the first row, then for all elements equal, sorting based on the second row, etc.).

Dilations of a cone. Scaling and permutations are not the only transformations that leave the product invariant. For any matrix W the cone spanned by W is defined as $\mathcal{C} = \{Wh, h \in \mathbb{R}_+^K\}$. $\mathcal{C}$ contains all nonnegative linear combinations of columns of W. The set $\mathcal{V} = \{V_{\cdot 1}, \ldots, V_{\cdot n}\}$ is entirely contained in $\mathcal{C}$ *if and only if* there exists an exact NMF of V. Figure 2.1 shows an example in 2 dimensions of two cones $\mathcal{C}_1$ and $\mathcal{C}_2$ spanned by two matrices W_1 and W_2 that contain $\mathcal{V}$. A numerical example can be found in [Hennequin, 2010].

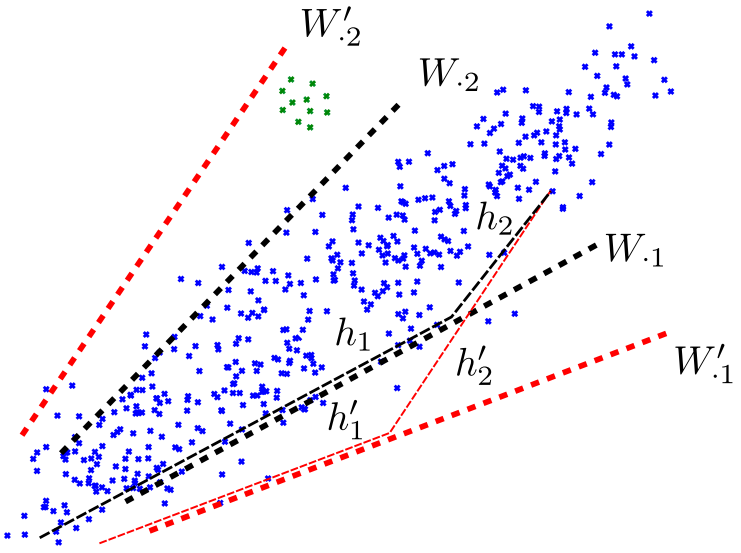


Figure 2.1: Two cones $\mathcal{C}_1$ and $\mathcal{C}_2$ containing the whole set of data points $\mathcal{V}$.

Several lines of research have been taken to tackle this problem: in [Ding et al., 2010], the columns of W are constrained to be linear combinations of data points: we can see on Figure 2.1 that the exact NMF is then unique and $\mathcal{C}$ is

exactly the cone spanned by the data points. In [Zhou et al., 2011], the authors add a constraint of the form $\log|\det(W)| \leq C$ for square matrices W: in the two-dimensional case this amounts to enforce small angles between columns of W.

2.1.3 Determining which divergence to choose

Factorizations with small positive values of β are relevant to decomposition of audio spectra. Indeed those typically exhibit exponential power decrease along frequency f, and it is important to keep track of low-power amplitudes because the ear is sensitive to differences in log amplitudes (measured in dB).

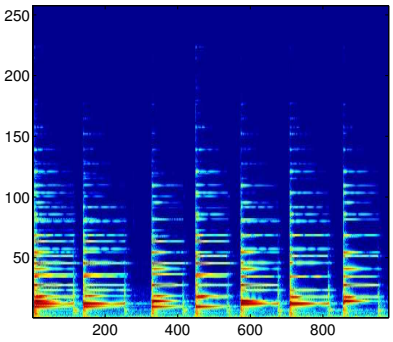
Choosing β for the decomposition of spectrograms is a delicate question. There are at least three aspects to this question : for some values of β, the NMF problem corresponds to a probabilistic model, which may be useful to obtain a certain number of mathematical properties such as consistency of the estimates of (W, H). In Subsection 2.3.2 for instance, this probabilistic framework helps design optimal estimates of the source signals which are otherwise difficult to justify.

The secon aspect aspect is that β might depend on the task to which NMF is aimed : for instance, the value $\beta = 0.5$ is advocated by [FitzGerald et al., 2009, Dessein et al., 2010] and has been shown to give optimal results in music transcription based on NMF of the magnitude spectrogram by [Vincent et al., 2010a].

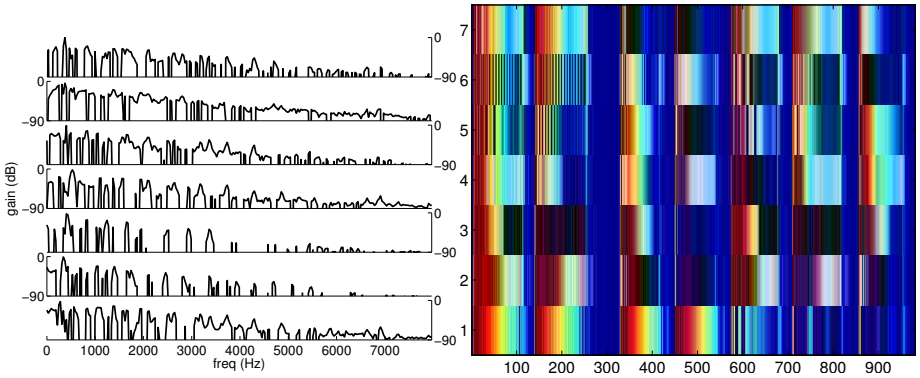
In the case of audio source separation, a case study is proposed in [Févotte et al., 2009]. The authors present the results of NMF on a very specific piano recording : four notes are present (D_4^b, F_4, A_4^b, and C_5 in increasing pitch order), and the recording consists of all pairs of notes played simultaneously, plus a combination of all four notes at the beginning of the recording [2]. Figure 2.2 reproduces these results for the three most commonly used divergences in audio source separation : the Euclidean norm, the Kullback-Leibler divergence and the Itakura-Saito divergence. Components may be interpreted either as transients or pitched spectras, upon listening to the reconstructed waveforms. [Févotte et al., 2009] then perform pitch estimation to assign each component to a note. We performed the assignment again by listening to the reconstructed waveforms. In Figure 2.2, components 1 to 4 correspond for all three divergences to D_4^b, F_4, A_4^b, and C_5 respectively, and component 5 to 7 to transients. In the case of Euclidean NMF and Kullback-Leibler NMF, the remaining transients contain interferences pitched components. In Euclidean NMF, components 1 and 4 contain interference from each other. In IS-NMF. component 5 corresponds to a hammer hit and components 6-7 to broadband noise.

Although listening tests are strikingly in favour of IS-NMF, a comprehensive comparison of all three divergences is yet to be found for more complex tasks. An interesting benchmark would be matrix completion or annotated NMF (which

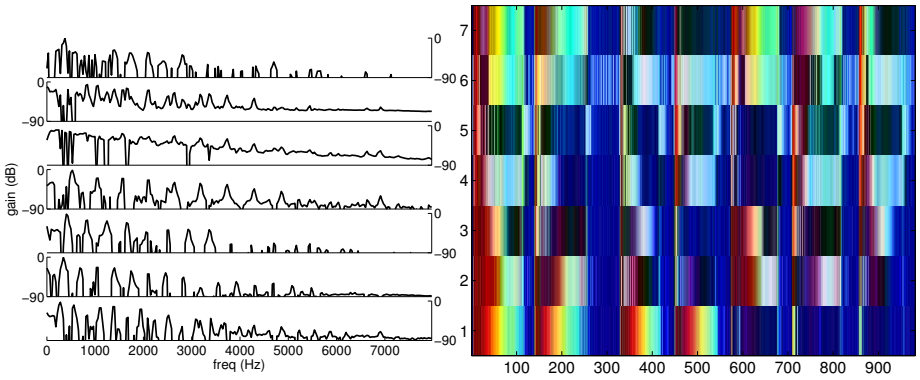
[2] audio samples available at `http://perso.telecom-paristech.fr/~fevotte/Samples/is-nmf/`



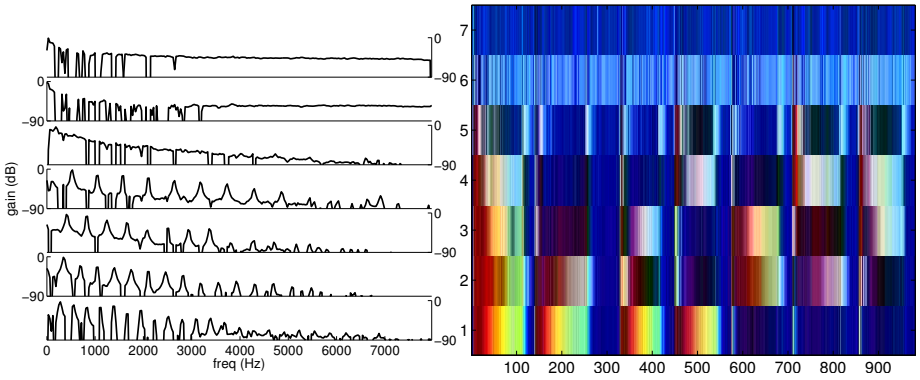
(a) input spectrogram



(b) $\beta = 2$ (Euclidean NMF)



(c) $\beta = 1$ (KL NMF)



(d) $\beta = 0$ (IS NMF)

Figure 2.2: Comparison of NMF decompositions using either the IS divergence or the ℓ_2 norm. Columns of W are displayed on the left in deciBels/Hertz scale. rows of H are displayed on the right in deciBels/time scale.

will be presented in 4), evaluated on a standard source separation database such as SiSEC or QUASI.

Finally, the third aspect to take into account when deciding which divergence to choose, but certainly not the least, is optimization issues. For that matter, let us go into more details in the next subsection.

2.2 Optimization algorithms for the β-divergences

In this Section we assume that the reader is familiar with basic optimization techniques such as convex functions, and projected gradient descent techniques (see Appendix A for the particular version we use here, or e.g., [Bertsekas, 1999] for a complete presentation). We will discuss algorithms for the problem of nonnegative matrix division (NMD). Note that all algorithms for nonnegative matrix factorization presented in this section rely on alternate optimization of W and H in 2.3. Thus, once an algorithm is proposed for NMD, an algorithm for NMF is deduced.

Algorithms for Euclidean NMD/NMF have been extensively studied, and several competitive alternatives have been proposed : the standard projected gradient descent with guaranteed convergence rates[Lin, 2007a], multiplicative updates, the easiest to implement but not the most efficient [Daube-Witherspoon and Muehllehner, 1986], active set methods which are particularly useful when K is small [Kim and Park, 2008], and finally block-coordinate descent methods which seem to be the most efficient method in many datasets [Gillis and Glineur, 2012, Mairal et al., 2010] (block-coordinate descent for NMF is also referred to as HALS, for complete bibliographic references see [Gillis and Glineur, 2012]).

A crucial point in NMF with β-divergences is the difficulty of the optimization problem : indeed, as soon as $\beta < 1$, optimizing H for fixed W is already a nonconvex problem. Moreover, as soon as $\beta < 2$ (recall that Euclidean NMF corresponds to $\beta = 2$), second order derivatives of the problem are unbounded. Consequently, NMF with β-divergences have received a very special treatment, which is the subject of debate in the optimization community : namely the use of multiplicative updates, the efficiency of which is difficult to analyze, in contrast with gradient descent and related methods which have been extensively studied for a long time, but are difficult to apply in NMF.

We give here theoretical as well as practical insights into this debate.

2.2.1 Non-convexity of NMD with beta-divergences

In this Section, we show that NMD is not convex even though W is fixed, for certain values of β.

Let us first consider first-order derivatives of the objective function :

$$\min_{W\geq 0, H\geq 0} \underbrace{\sum_{fn} d_\beta(V_{fn}, (WH)_{fn})}_{G(W,H)} \ . \tag{2.5}$$

$$\frac{\partial}{\partial W_{fk}} G(W,H) = \sum_n H_{kn} (WH)_{fn}^{\beta-2} \left((WH)_{fn} - V_{fn}\right) \ , \tag{2.6}$$

$$\frac{\partial}{\partial H_{kn}} G(W,H) = \sum_f W_{fk} (WH)_{fn}^{\beta-2} \left((WH)_{fn} - V_{fn}\right) \ , \tag{2.7}$$

For the sake of this discussion let us also give second order derivatives with respect to W, for fixed H. Notice that the objective function is separable in f so that $\frac{\partial^2}{\partial W_{fk} W_{f'k'}} = 0$ if $f \neq f'$.

The second order derivative of d_β with respect to y is :

$$\frac{\partial^2}{\partial y^2} d_\beta(x,y) = \begin{cases} y^{\beta-3}\left((\beta-1)y - (\beta-2)x\right) & \text{if } \beta \neq 2 \\ 1 & \text{if } \beta = 2 \end{cases} \tag{2.8}$$

Therefore, second order derivatives of G are equal to :

$$\sum_n H_{kn} H_{k'n} \frac{\partial^2}{\partial y^2} d_\beta(V_{fn}, (WH)_{fn}) \tag{2.9}$$

or, in expanded form :

$$\frac{\partial^2}{\partial W_{fk} W_{fk'}} G(W,H) = \begin{cases} \sum_n H_{kn} H_{k'n} (WH)_{fn}^{\beta-3} \left((\beta-1)(WH)_{fn} - (\beta-2)V_{fn}\right) & \text{if } \beta \neq 2 \\ \sum_n H_{kn} H_{k'n} & \text{if } \beta = 2 \end{cases} \tag{2.10}$$

As a matter of fact, the objective function is only convex with respect to W (or H) if $\beta \geq 1$. Otherwise, g is convex in some places and concave in others. This implies that, for $\beta < 1$, even the problem of inferring H with W fixed (Nonnegative Matrix Division) is non-convex ! Moreover, as soon as $\beta < 2$, second-order derivatives are unbounded near $(WH)_{fn} = 0$.

2.2.2 Majorization-minimization algorithms and multiplicative updates

Although multiplicative updates algorithms for NMF were originally proposed by [Lee and Seung, 1999], similar updates in the case where the dictionary is fixed may be found in [Richardson, 1972, Lucy, 1974] in the case of the KL divergence. In [Lee and Seung, 2001], a justification of multiplicative update algorithms is given based on majorization-minimization algorithms. We outline in this section the essential properties of majorization-minimization algorithms. Based on empirical comparison, we then show that, while projected gradient

descent is preferable to optimize the Euclidean loss, it fails for other values of β such as the Itakura-Saito divergence ($\beta = 0$).

A derivation of multiplicative updates for the Itakura-Saito divergence will be presented later in Section 2.3.4.

Theorem 1. *Let $g(h, \underline{h})$ satisfy the following properties :*

- $\forall (h, \underline{h}), g(h) \leq g(h, \underline{h})$.
- $\forall h, g(h, \underline{h}) = g(h) \Rightarrow \underline{h} = h$.

$g(h, \underline{h})$ is said to be an auxiliary function of g. Then, any sequence $(h^{(t)})_{t \geq 0}$ defined in $\mathbb{R}_+^K$, satisfying

$$\forall t, g(h^{(t)}, h^{(t-1)}) \leq g(h^{(t-1)}, h^{(t-1)}) \tag{2.11}$$

also satisfies

$$\forall t, g(h^{(t)}) \leq g(h^{(t-1)}) \,. \tag{2.12}$$

In particular, if $h^{(t)} = \arg\min_h g(h, h^{(t-1)})$, we obtain a majorization-minimization algorithm. The Expectation-Maximization is an example of majorization-minimization algorithm that efficiently solve otherwise intractable maximum-likelihood optimization problems.

Auxiliary functions for the Euclidean loss and the Kullback-Leibler divergence were proposed in [Lee and Seung, 2001]. An extension to the whole family of beta divergences may be found in [Févotte and Idier, 2011].

An experimental comparison In the following experiment we compare projected gradient descent and multiplicative updates when W is fixed and $N = 1$ (inference in H only). We construct $W \in \mathbb{R}_+^{F \times K}$ with $F = 257$ and $K = 12$. The columns of W correspond to the twelve semitones of the second octave of a piano from the Iowa database of recorded instruments[3]. A true $H_0 \in \mathbb{R}_+^{K \times 1}$ is chosen at random, and V is drawn at random according to $V \sim \text{Exp}(Wh_0)$.

For projected gradient descent, we use a diminishing step size $\mu_t = \frac{\mu}{t}$ where μ is tuned at the first iteration to yield descent of the cost function.

As we can see on Figure 2.3, in the case of the ℓ_2 norm ($\beta = 2$), the time taken by gradient descent to reach convergence is order of magnitudes shorter than that of multiplicative updates. On the other hand, when $\beta = 0$ the situation is reversed, and gradient descent is stuck at high function values compared to multiplicative updates.

The reason for this is that while the ℓ_2 norm has bounded second-order derivatives, the Itakura-Saito divergence is unbounded near zero.

[3] http://theremin.music.uiowa.edu/MIS.html

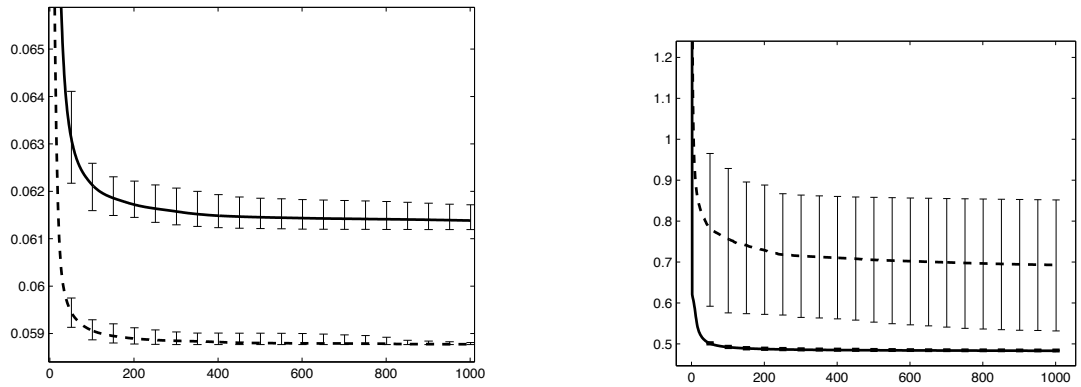


Figure 2.3: (left) Euclidean loss (right) Itakura-Saito divergence. Dashed curve is gradient descent, solid curve is multiplicative updates. Error bars measure the variability across initializations.

2.2.3 Expectation-Maximization algorithms

The expectation-maximization[Dempster et al., 1977, Hathaway, 1986] algorithm is a particular case of majorization-minimization. It is adapted to inference or estimation in probabilistic models and provides for those a generic recipe. It is a competitive alternative to gradient descent in a number of situations where the objective function does not satisfy minimal assumptions (bounded second order derivatives, etc.). For a quick introduction we refer the reader to [Hastie et al., 2009]. We present here two cases of expectation-maximization algorithms used to solve NMF, that are different from multiplicative updates.

2.2.3.1 Itakura-Saito divergence

Itakura-Saito NMF is equivalent to maximum-likelihood in a probabilistic model with latent variables $S_{fn}^{(k)}$ for each component k. [Févotte et al., 2009] take advantage of this to derive an auxiliary function of the form :

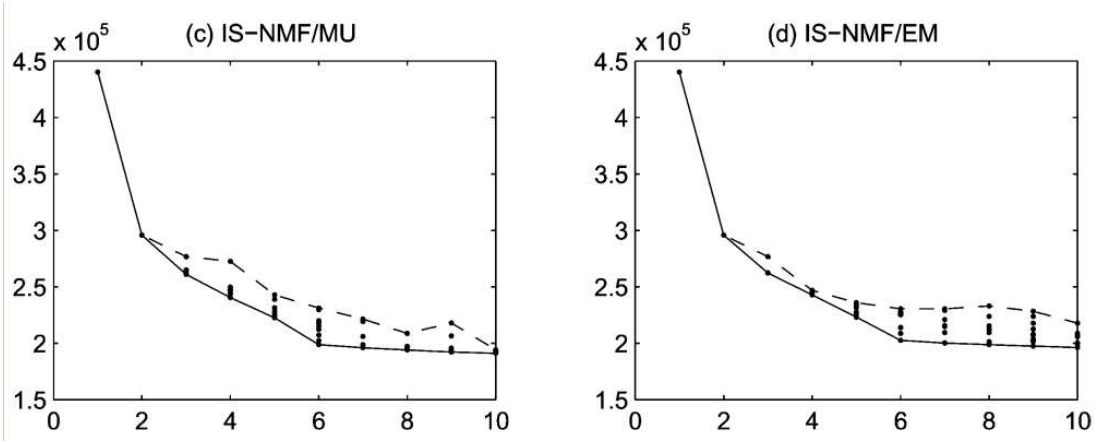
$$G(W,H,\underline{W},\underline{H}) = \sum_k \sum_{fn} \frac{T_{fnk}}{W_{fk}H_{kn}} + \log\left(W_{fk}H_{kn}\right) , \tag{2.13}$$

where

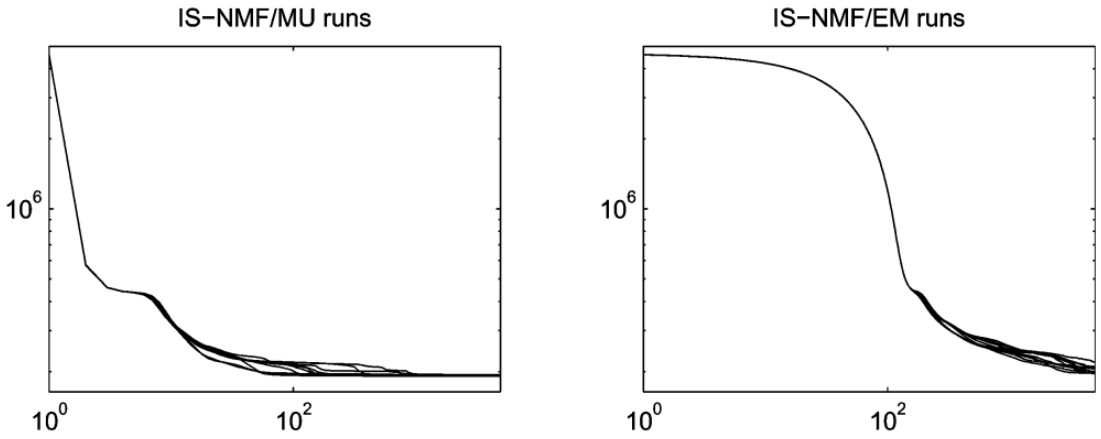
$$T_{fnk} = \left(\frac{\underline{W}_{fk}\underline{H}_{kn}}{\hat{V}_{fn}}\right)^2 V_{fn} + \underline{W}_{fk}\underline{H}_{kn}\left(1 - \frac{\underline{W}_{fk}\underline{H}_{kn}}{\hat{V}_{fn}}\right) . \tag{2.14}$$

$G(W,H,\underline{W},\underline{H})$ is then optimized alternately in W and H. At each step the minimum is obtained in closed form.

Altogether, the following steps are repeated until convergence :



(a) K vary



(b) Number of iterations vary

Figure 2.4: Comparison of multiplicative updates and the EM algorithm for IS-NMF, as the number of components and the number of iterations vary. For each experiment, 10 different initializations are tried. (a) 5000 iterations (b) 6 components.*(Figure reproduced from [Févotte et al., 2009])*

$$\begin{aligned} &\text{Compute } T_{fnk} \text{ as in 2.14,} \\ W_{fk} &= \frac{1}{N} \sum_n \frac{T_{fnk}}{\underline{H_{kn}}}, \\ H_{kn} &= \frac{1}{F} \sum_f \frac{T_{fnk}}{\underline{W_{fk}}}. \end{aligned} \tag{2.15}$$

The cost of each iteration of EM (all steps together) is $O(FNK)$, which is comparable to multiplicative updates (MU). Another common feature with MU is sensitivity to the initial point. Both algorithms give comparable objective function values for a fixed number of iterations.

2.2.3.2 Kullback-Leibler divergence

[Smaragdis et al., 2007] introduce a different parameterization of NMF with the Kullback-Leibler divergence with an additional constraint and obtain a different algorithm by taking advantage of a probabilistic interpretation. This approach was called probabilistic latent component analysis (PLCA).

$$\begin{array}{c} \min \sum_{fn} -V_{fn} \log \left(\sum_k s_k W_{fk} H_{kn}\right) \\ \sum_f W_{fk} = 1 \\ \sum_n H_{kn} = 1 \\ \sum_k s_k = 1 \end{array} \tag{2.16}$$

where $W \in \mathbb{R}_+^{F \times K}, s \in \mathbb{R}_+^K, H \in \mathbb{R}_+^{K \times N}$, and V is rescaled so that $\sum_{fn} V_{fn} = 1$.

Introduce $\hat{V}_{fn} = \sum_k s_k W_{fk} H_{kn}$. Then for any (W, H, s) satisfying the constraints, the generalized Kullback-Leibler ($\beta = 1$) evaluates to :

$$\sum_{fn} -V_{fn} \log \hat{V}_{fn} + \underbrace{\sum_{fn} \hat{V}_{fn} - \sum_{fn} V_{fn}}_{=0} . \tag{2.17}$$

In this sense, PLCA is equivalent to re-parameterizing KL-NMF and adding the constraint that $\sum_{fn} \hat{V}_{fn} = 1$.

Auxiliary function and EM steps An auxiliary function of the objective function is obtained by introducing auxiliary variables Q_{fnk} such that $\sum_k Q_{fnk} = 1$:

$$\begin{aligned} \underbrace{\sum_{fn} -V_{fn} \log \left(\sum_k s_k W_{fk} H_{kn}\right)}_{G(W,H)} &= \sum_{fn} -V_{fn} \log \left(\sum_k \tfrac{s_k}{Q_{fnk}} W_{fk} H_{kn} Q_{fnk}\right) \\ &\leq \underbrace{\sum_k \sum_{fn} -Q_{fnk} V_{fn} \log(W_{fk} H_{kn} \frac{s_k}{Q_{fnk}})}_{G_u(W,H,Q)} . \end{aligned}$$

Moreover, one can show that

$$\min_{\sum_k Q_{fnk}=1} G_u(W, H, Q) = G(W, H) . \tag{2.18}$$

and the minimum is reached at

$$Q_{fnk} = \frac{s_k W_{fk} H_{kn}}{\sum_{k'} s_{k'} W_{fk'} H_{k'n}} . \tag{2.19}$$

For fixed Q, minimizing $G_u(W, H, Q)$ with respect to W, H and s amounts to the following subproblems independently :

$$\begin{array}{ll} \min_H & \sum_n - \sum_f Q_{fnk} V_{fn} \log H_{kn} \,, \\ \text{subject to} & \sum_n H_{kn} = 1 \,. \end{array}$$

$$\begin{array}{ll} \min_W & \sum_f - (\sum_n Q_{fnk} V_{fn}) \log W_{fk} \,, \\ \text{subject to} & \sum_f W_{fk} = 1 \,. \end{array}$$

$$\begin{array}{ll} \min_s & \sum_k - \left(\sum_{fn} V_{fn} Q_{fnk}\right) \log s_k \\ \text{subject to} & \sum_k s_k = 1 \end{array}$$

For each of these problems, closed form updates are obtained.

$$\begin{aligned} W_{fk} &= \frac{\sum_n Q_{fnk} V_{fn}}{\sum_{f'n} Q_{f'nk} V_{f'n}} \\ H_{kn} &= \frac{\sum_f Q_{fnk} V_{fn}}{\sum_{fn'} Q_{fn'k} V_{fn'}} \\ s_k &= \frac{\sum_{fn} Q_{fnk} V_{fn}}{\sum_{fnk'} Q_{fnk'} V_{fn}} \,. \end{aligned} \tag{2.20}$$

The obtained updates are different from multiplicative updates, however a comparison of both algorithms either on generic data or based on theoretical study has not been provided up to date.

2.3 Itakura-Saito NMF

We introduced Itakura-Saito NMF in previous Sections as an acoustically-motivated measure of distortion. In this section, we show that Itakura-Saito NMF derives from a generative model of the source spectrograms. In addition to the EM algorithm presented in Section 2.2.3.1, maximum-likelihood estimates of the sources given estimates of $\hat{V}^{(g)} = W^{(g)} H^{(g)}$ are explicitly computed in Section 2.3.2. In Section 2.3.4 we derive an auxiliary function for IS-NMF, from which multiplicative updates follow.

A useful property brought by majorization-minimization is that the cost function decreases with each iteration. Convergence of W and H however, cannot be guaranteed, but we can still study the properties of the limit points generated by multiplicative updates. This will be the subject of Sections 2.3.5 and 2.3.6.

Itakura-Saito NMF is not the only one which derives from a probabilistic model, see for instance [Abdallah and Plumbley, 2004, Plumbley et al., 2006], who introduce a Gamma distribution for the spectrogram. However, it is the only model for which strict additivity of the source spectrograms holds.

2.3.1 Generative model

Given a short time Fourier transform $X \in \mathbb{C}^{F\times N}$ of an audio track, we make the assumption that X is a linear instantaneous mixture of i.i.d. Gaussian signals :

$$X_{fn} = \sum_g S_{fn}^{(g)} \quad \text{where} \quad S_{fn}^{(g)} \sim \mathcal{N}(0, \sum_{k=1}^{Q} W_{fk}^{(g)} H_{kn}^{(g)}) \,. \tag{2.21}$$

The power spectral density of each source g is thus $\mathbb{E}(|S^{(g)}|^2) = W^{(g)}H^{(g)} \in \mathbb{R}_+^{F\times N}$. As a consequence, we have $\mathbb{E}(V) = WH$ where $V = |X|^2$ is the observed power spectrogram. Furthermore, V has the following distribution :

$$p(V|\tilde{V}) = \prod_{f,n} \frac{1}{\tilde{V}_{fn}} \exp\left(-\frac{V_{fn}}{\tilde{V}_{fn}}\right). \tag{2.22}$$

As pointed out in [Févotte et al., 2009], maximum-likelihood estimation of (W, H) is equivalent to minimizing the Itakura-Saito divergence between V and WH.

The Itakura-Saito loss is defined on strictly positive scalars by :

$$d_{IS}(x, y) = \frac{x}{y} - \log\frac{x}{y} - 1 \,. \tag{2.23}$$

2.3.2 Recovery of source estimates

The advantage of having a generative model is that maximum-likelihood estimates provide a grounded and non-intuitive formula for source estimates.

The likelihood of complex spectrograms must be computed conditional to the observation of $X = \sum_g S^{(g)}$:

$$\max \prod_{fn} p(S_{fn}^{(1)}, \ldots, S_{fn}^{(G)} | X_{fn}, \hat{V}_{fn}^{(1)}, \ldots, \hat{V}_{fn}^{(g)}) \,. \tag{2.24}$$

where $\hat{V}_{fn}^{(g)} = \sum_k W_{fk}^{(g)} H_{kn}^{(g)}$. We express the constraint that $X = \sum_g S^{(g)}$ by setting $S^{(G)} = X - \sum_{g=1}^{G-1} S^{(g)}$ so that we need only compute $p(S^{(1)}, \ldots, S^{(G-1)}|X)$.

We use the following property to compute this conditional distribution :

Property 1. *Let* $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ *be a Gaussian random variable in* $\mathbb{R}^{n_1+n_2}$ *with mean* $\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$ *and covariance matrix* $\begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$. *Then the distribution of* x_1 *conditional to* x_2 *is also Gaussian with mean and covariance given by :*

$$\mu = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \qquad \Sigma = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \,. \tag{2.25}$$

As a consequence, conditionally to the observation of X, spectrograms satisfy an independence property across time-frequency entries :

$$p(S^{(1)}, \ldots, S^{(G-1)}|X) = \prod_{fn} p(S_{fn}^{(1)}, \ldots, S_{fn}^{(G-1)}|X_{fn}) \tag{2.26}$$

The distribution of the complex spectrograms $(S_{fn}^{(g)})_{g=1}^{G}$ conditional to X_{fn} is Gaussian with mean and covariance given by :

$$\mu_g = \frac{\hat{V}_{fn}^{(g)}}{\hat{V}_{fn}} X_{fn} \qquad \Sigma_{gg'} = \begin{cases} \dfrac{\hat{V}_{fn}^{(g)} \sum_{g' \neq l} \hat{V}_{fn}^{(g')}}{\hat{V}_{fn}} & \text{if } g = g' \\ -\dfrac{\hat{V}_{fn}^{(g)} \hat{V}_{fn}^{(g')}}{\sum_{g'} \hat{V}^{(g')_{fn}}} & \text{otherwise} \end{cases} . \tag{2.27}$$

and we have dropped (f, n) indices from μ and Σ for simplicity. Since $X = \sum_g S^{(g)}$ is fixed, there are only $G - 1$ free sources in the above distribution, and $S^{(G)} = X - \sum_{g=1}^{G-1} S^{(g)}$.

MMSE estimates of the sources' STFT are thus given by:

$$\hat{S}_{fn}^{(g)} = \frac{\hat{V}_{fn}^{(g)}}{\hat{V}_{fn}} X_{fn} \,. \tag{2.28}$$

Note that the sources' STFT share the same phase as the mixture signal. This is because we have supposed that phases are distributed uniformly, so without additional information, the best estimate is to keep the mixture's phase. Time signals are then obtained by taking $\hat{s}^{(g)} = \text{iSTFT}(\hat{S}^{(g)})$.

2.3.3 Consistent source estimates

The material in this section is related to previous work by [Le Roux et al., 2010]. The source estimates in the previous section are projections of the maximum-likelihood solutions back into the subspace of time-domain signals. However, signals $s^{(1)}, \ldots, s^{(G)}$ such that $S^{(g)} = \text{STFT}(s^{(g)})$ do not necessarily exist. In particular, $S^{(g)} \neq \mathcal{S}(\mathcal{S}^{\dagger}(S^{(g)}))$. This is because the $\mathcal{S}$ is an overcomplete operator, so $\mathcal{S}^{\dagger}$ has a nontrivial kernel.

We may instead solve directly for time-domain signals, as we shall now explain.

Since we have at hand the distribution of $(S_{fn}^{(g)})_{f,n,g}$ we need only solve a maximum-likelihood problem where we parameterize $S^{(g)} = \mathcal{S}(s^{(g)})$ and solve directly for $s^{(g)}$ in the time domain. For the sake of simplicity we will restrict ourselves to the case where $G = 2$. Since $s^{(1)}$ and $s^{(2)}$ are tied, we need only solve for $s^{(1)}$. In the remaining of this section we drop indices (g) for clarity.

The maximum likelihood problem then writes:

$$\min_{s} \sum_{fn} \alpha_{fn} ((\mathcal{S}s)_{fn} - \hat{S}_{fn})^2 \,, \tag{2.29}$$

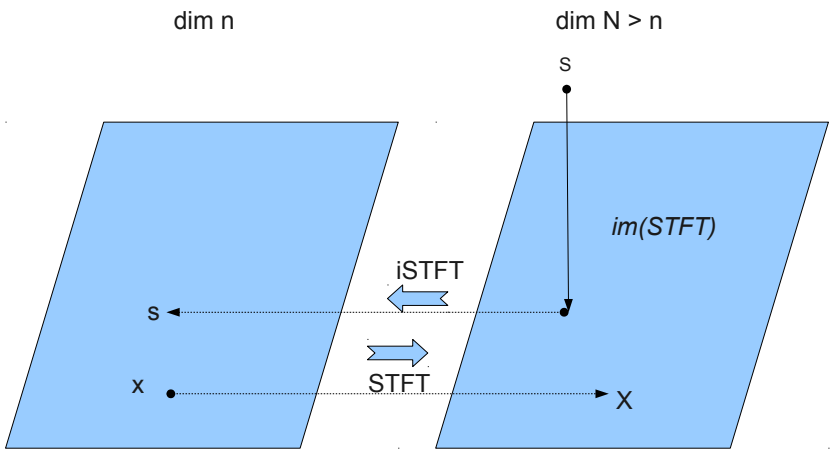


Figure 2.5: While the observed spectrogram X is the image of signal x by operator $\mathcal{S}$, the estimate $\hat{S}$ that we compute may not be the image of any source signal s so in this case the $\mathcal{S}^\dagger$ operator projects S on $im(\mathcal{S})$ before inverting it to a time domain signal.

where $\alpha_{fn}^{-1} = \frac{1}{\hat{V}_{fn}^{(1)}} + \frac{1}{\hat{V}_{fn}^{(2)}}$. Since $\mathcal{S}$ is a linear operator, this is an unconstrained quadratic optimization problem, with Hessian $\mathcal{S}^\top \mathrm{diag}(\alpha)\mathcal{S}$. If all α_{fn} were equal to one, then the solution of Problem 2.29 would be $s = \mathcal{S}^\dagger \hat{S} = (\mathcal{S}^\top \mathcal{S})^{-1}\mathcal{S}^\top \hat{S}$. Otherwise, we can still compute a close form solution but it requires inverting a $n \times n$ matrix (where n is the size of the time-domain signal), which is computationally very expensive. Instead, we exploit the fact that the operation $\mathcal{S}s$ can be computed efficiently via Fast Fourier Transforms, by using gradient descent to solve for s. We choose $s^{(0)} = \mathcal{S}^\dagger(\hat{S})$ as initial point, and compute the t-th step of gradient descent:

$$s^{(t)} = s^{(t-1)} - \mu_t \mathcal{S}^\top \mathrm{diag}(\alpha)(\mathcal{S}s^{(t-1)} - \hat{S})\,. \tag{2.30}$$

The cost of computing this step is dominated by the computation of one STFT and one inverse STFT (indeed $\mathcal{S}^\dagger$ and $\mathcal{S}^\top$ are equal up to a multiplicative constant since we use the same window for analysis and synthesis).

We study the effect of these estimates in an ideal setting where the optimal $\hat{S}^{(g)}$ are known, so that we can compare the benefits of consistent source signal estimates independently of the quality of model estimates. Table 1 compares the quality of source separation with Wiener vs consistent estimates on a 10 seconds' excerpt from the SISEC database (Tamy - Que pena tanto faz). While the decrease in cost function value between Wiener estimates and consistent estimates is of several orders of magnitude, the difference in terms of standard metrics is very small. [Le Roux et al., 2010] obtain a 2dB improvement on average.

There might be two reasons why our findings disagree : suboptimality of our estimates, while [Le Roux et al., 2010] compute exact solutions (method referred to as "time domain"). Moreover, we have tested our method only on one mixture signal, so that more extensive results should be collected before we can make any firm claim.

	SDR	SIR	SAR	CPU time
Wiener	11.9018	23.7229	12.2155	0.0785
Consistent	11.9057	23.7310	12.2191	27.6253

Table 2.1: Comparison of Wiener estimates and consistent estimates on a 10 seconds' audio excerpt.

In gradient descent, we choose μ_t such that $\mu_t^{-1} I$ upper-bounds the Hessian of Problem 2.29, so we observe descent of the cost function at each step in Figure 2.6. As we can see, after a few iterations, the objective function decreases very slowly, so maybe exploiting the structure of the Hessian to compute a closed form solution efficiently would improve the quality of consistent estimates in terms of SDR.

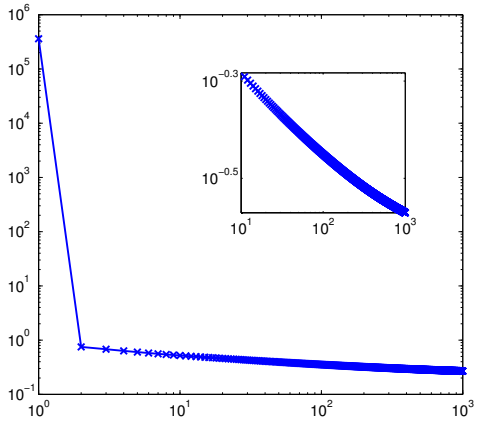


Figure 2.6: Descent of the cost function is observed with an appropriate choice of μ_t.

We have shown how to compute time domain source estimates directly via a maximum-likelihood approach. The approach taken in [Le Roux et al., 2010] is slightly different, and consists in finding a source signal x such that the modulus $|\mathcal{S}(x)|$ is as close to $|S^{(0)}|$ as possible. The reader may point out, with reason, that a proper approach would consist in formulating a generative model directly in terms of time signals, since postulating independence of the time-frequency entries of a complex spectrogram is unrealistic.

Finally, estimates of the source spectrograms are mapped back to time-domain signal. Since the mapping is linear, the conservation property $X = \sum_g \hat{S}^{(g)}$ translates to $x = \sum_g s^{(g)}$.

2.3.4 Derivation of a descent algorithm

In this Section we derive an efficient algorithm for solving Equation Eq. (2.3) with the Itakura-Saito divergence, which is inspired by [Cao et al., 1999]. Define $G(W, H) = d_{IS}(V, WH)$ the objective function of the NMF problem. G is not convex so we cannot hope for a unique global minimum. We optimize alternately in W and H. Descent at each step yields a descent algorithm.

$$G(W^{(t+1)}, H^{(t+1)}) \leq G(W^{(t)}, H^{(t+1)}) \leq G(W^{(t)}, H^{(t)})\,. \tag{2.31}$$

The essential building block in multiplicative updates, as explained in Section 2.2.2 is to find an auxiliary function $g(h, \underline{h})$ for g that is easy to minimize. The objective function is separable in the columns of H, so we need only consider the following subproblem:

$$\begin{array}{ll} \min_h & g(h)\,, \\ h \geq 0 & \end{array} \tag{2.32}$$

where

$$g(h) = \sum_f \frac{v_f}{\sum_k W_{fk} h_k} + \log(\sum_k W_{fk} h_k)\,, \tag{2.33}$$

and minimization is over $h \in \mathbb{R}_+^K$.

g is the sum of convex terms of the form $\frac{1}{x}$ and concave terms of the form $\log(x)$. In the following, we deal with each separately. Let $\underline{h} \geq 0$ be the current estimate for h, and assume $\forall k, \underline{h}_k > 0$. In particular, $\forall f, \sum_k W_{fk}\underline{h}_k > 0$, and $\sum_k \underline{h}_k > 0$. Introduce $\rho_{fk} = \frac{W_{fk}\underline{h}_k}{\sum_l W_{fl}\underline{h}_l}$, and $x_k = \frac{h_k}{\underline{h}_k}$. Since the function $x \to \frac{1}{x}$ is convex, we may apply Jensen's inequality, thereby obtaining:

$$\begin{aligned} \frac{v_f}{\sum_k W_{fk} h_k} &= \frac{v_f}{\sum_k W_{fk}\underline{h}_k} (\sum_k \rho_{fk} x_k)^{-1} \\ &\leq \frac{v_f}{\sum_k W_{fk}\underline{h}_k} \sum_k \rho_{fk} x_k^{-1} \\ &= \frac{v_f}{(\sum_k W_{fk}\underline{h}_k)^2} \sum_k \frac{\underline{h}_k^2}{h_k} \end{aligned}$$

On the other hand, since the function $x \to log(x)$ is concave, we may apply the tangent inequality :

$$\log(\sum_k W_{fk} h_k) \leq \log(W_{fk}\underline{h}_k) + \frac{1}{\sum_k W_{fk} h_k} \sum_k (h_k - \underline{h}_k)\,.$$

Moreover, by strict concavity, the inequality is an equality if and only if $h = \underline{h}$. Summing both inequalities over f, we obtain an inequality of the form $g(h) \leq g(h, \underline{h})$, where

$$g(h, \underline{h}) = \sum_k p_k \frac{h_k^2}{\underline{h}_k} + q_k(h_k - \underline{h}_k) + c\,,$$

and

$$p_k = \sum_f W_{fk} \frac{v_f}{(\sum_l W_{fl}\underline{h}_l)^2} \quad q_k = \sum_f \frac{1}{\sum_l W_{fl}\underline{h}_l} \quad c = \sum_f \log(\sum_l W_{fl}\underline{h}_l)\,. \quad (2.34)$$

Moreover, the inequality is an equality if and only if $h = \underline{h}$. Thus, $\forall \underline{h} \in (\mathbb{R}_+^\star)^K, g(h, \underline{h})$ is an auxiliary function for $g(h)$.

As shown in [Cao et al., 1999], we may rewrite $g(h, \underline{h})$ as follows:

$$g(h, \underline{h}) - g(\underline{h}) = \sum_k \left(-p_k \frac{\underline{h}_k}{h_k} + q_k \right) (h_k - \underline{h}_k)\,. \quad (2.35)$$

g is separable in h_k, so we can deal with each term independently of the others. We may either (a) minimize the right hand side with respect to h_k, or (b) set each term $-p_k \frac{\underline{h}_k}{h_k} + q_k$ to 0, yielding the multiplicative updates found in [Févotte et al., 2009]. Both cases can be summarized as:

$$\forall k,\ h_k = \underline{h}_k \left(\frac{\sum_f w_{fk} \frac{v_f}{(Wh)_f^2}}{\sum_f w_{fk} \frac{1}{(Wh)_f}} \right)^\delta, \quad (2.36)$$

where $\delta = 0.5$ (a) or $\delta = 1$ (b).

Similar updates can be found for W with the same arguments. Note that multiplicative updates are only valid for $\underline{h} \in (\mathbb{R}_+^\star)^K$. This has an important consequence: if at any iterate a coefficient $h_k^{(t)} = 0$, then it stays at zero for all $s > t$. It is possible to guarantee that $h_k^{(t)} > 0$ at every iteration by appropriately choosing $h^{(0)}$,as we will see in Section 2.3.5, but this does not really change the problem in practice: as $h_k^{(t)}$ approaches zero, it is likely to become stuck. This statement can be made more precise by interpreting multiplicative updates differently, as we will do in the next Section. This problematic effect of *absorbing zeroes* makes the choice of initial points $h^{(0)}$ all the more important: not only is NMD sensitive to the choice of the initial point because it is non-convex, but also because absorbing zeroes must be avoided.

As we we will see in the next section, solutions to this problem were found by studying the convergence properties of multiplicative updates.

2.3.5 Discussion of convergence properties

The material in this section is directly inspired by [Lin, 2007b]. The main property of multiplicative updates is that they enforce descent of the cost function at

each step, and hence convergence of the cost function.

$$\forall t, g(h^{(t+1)}) \leq g(h^{(t)})$$
$$\exists l, \lim_{t \to +\infty} g(h^{(t)}) = l\,.$$

A desirable property would be that limit points of the sequence of estimates $(W^{(t)}, H^{(t)})_{t \geq 1}$ be stationary points of the NMF problem.

$h^\star$ is a stationary point of NMD if and only if is satisfies the following first-order optimality conditions:

$$h_k^\star \geq 0\,, \tag{2.37}$$

$$\frac{\partial}{\partial h_k} g(h^\star) = 0 \qquad \text{if } h_k^\star > 0\,, \tag{2.38}$$

$$\frac{\partial}{\partial h_k} g(h^\star) \geq 0 \qquad \text{if } h_k^\star = 0\,. \tag{2.39}$$

First-order optimality conditions for the NMF are found by writing first-order optimality conditions for H and W. For the sake of clarity we will not write them here.

Multiplicative updates produce bounded sequences and thus have limit points. These limit points may not necessarily be stationary points of NMF, because of the problem of absorbing zeroes. Modifications of multiplicative updates were proposed to circumvent this problem [Lin, 2007b]. We show here how to adapt those modifications for NMD (i.e. updating H for fixed W) with the Itakura-Saito divergence, and show that any limit points found by modified multiplicative updates are stationary points of the NMD algorithm. The proof may be extended to NMF straightforwardly. Whether modified multiplicative updates still yield a descent algorithm in the case of the Itakura-Saito divergence is an open question.

Let $h^\star$ be a limit point of a sequence of multiplicative updates $(h^{(t)})_{t \geq 0}$, so that:

$$h_k^\star = h_k^\star \frac{p_k}{q_k}\,. \tag{2.40}$$

where p_k and q_k are defined in Eq. (2.34). As a consequence

$$h_k^\star = 0 \quad \text{OR} \quad \frac{\partial}{\partial h_k} g(h^\star) = 0\,. \tag{2.41}$$

Thus, condition 2.38 is satisfied but not 2.39. In [Lin, 2007b], a modification of NMF is proposed to overcome this limitation and ensure that all limit points are stationary points of NMF, in the case where the divergence used is the Euclidean norm. We introduce here an adaptation to the case of the Itakura-Saito divergence and discuss whether their proof can also be "adapted". Updates of h can be rewritten as:

$$h_k^{(t+1)} = h_k^{(t)} - \frac{h_k^{(t)}}{q_k^{(t)}}(p_k^{(t)} - q_k^{(t)}) = h_k^{(t)} - \frac{h_k^{(t)}}{q_k^{(t)}} \frac{\partial}{\partial h_{kn}} g(h^{(t)})\,. \tag{2.42}$$

KKT conditions can be violated if $h_{kn}^{(t)} = 0$ while $\frac{\partial}{\partial h_{kn}} g(h^{(t)}) < 0$. In this case, $H_{kn}^{(t)}$ is stuck at zero without guarantee on the sign of the gradient.

A natural modification of multiplicative updates is thus:

$$h_k^{(t+1)} = h_k^{(t)} - \frac{\bar{h}_k^{(t)}}{q_k^{(t)}} \frac{\partial}{\partial h_{kn}} g(h^{(t)}) \,, \tag{2.43}$$

where

$$\bar{h}_k^{(t)} = \begin{cases} h_k^{(t)} & \text{if} \quad \frac{\partial}{\partial h_{kn}} g(h^{(t)}) \geq 0 \\ \max(h_k^{(t)}, \sigma) & \text{if} \quad \frac{\partial}{\partial h_{kn}} g(h^{(t)}) < 0 \end{cases} \,. \tag{2.44}$$

and σ is a small pre-defined constant. Similar modifications should be made to updates in W.

These updates are of the same order of complexity in time and memory as multiplicative updates. They guarantee that if $h_k^{(t)} = 0$ but $\frac{\partial}{\partial h_{kn}} g(h^{(t)}) < 0$, then $h_k^{(t+1)} > 0$. The following Theorem asserts that this is sufficient to ensure convergence to stationary points of the objective function.

Theorem 2. *Let $h^\star$ be a limit point of the modified updates in Equation 2.44. Then $h^\star$ is a stationary point of the NMD problem.*

In order to prove Theorem 2, we will need the following property:

Property 2. *If $\forall k, h_k^{(0)} > 0$, then $\forall k, h_k^{(t)} > 0$*

Proof. If $h_k^{(t)} > 0 \forall k$, then $\forall f, \frac{\epsilon + V_f}{\epsilon + (Wh)_f} > 0$, and $\forall f, \frac{1}{\epsilon + (Wh)_f} > 0$. Since each column of W has at least one nonzero coefficient, it follows that $p_k^{(t)} > 0$ and $q_k(t) > 0$, so $h_k^{(t)} > 0$. □

Proof. (*of theorem 2*) Let $h^\star \in \mathbb{R}_+^K$ be a limit point of Eq. (2.44), i.e., $\forall k, \lim_{t \to +\infty} h_k^{(t)} = h_k^\star$. Let k be fixed.

Assume $h_k^\star > 0$. For t large enough: $0 < C_1 = \min(\sigma, h_k^\star/2) \leq \bar{h}_k^{(t)} \geq C_2 = 2h_k^\star$. Thus,

$$h_k^{(t)} - h_k^{(t+1)} = \bar{h}_k^{(t)} \frac{p_k^{(t)} - q_k^{(t)}}{q_k^{(t)}} \,,$$

and,

$$C_1(h_k^{(t)} - h_k^{(t+1)}) < \frac{p_k^{(t)}}{q_k^{(t)}} - 1 < C_2(h_k^{(t)} - h_k^{(t+1)}) \,,$$

By continuity of $p_k^{(t)}$ and $q_k^{(t)}$, and since $\frac{\partial}{\partial h_{kn}} g(h^{(t)}) = p_k - q_k$, we may conclude

$$h_k^\star > 0 \Rightarrow \frac{\partial}{\partial h_{kn}} g(h^{(t)}) = 0 \,.$$

Suppose $h_k^\star = 0$ **and** $\frac{\partial}{\partial h_k} g(h^\star) < 0$. By continuity, there exists $c < 0$ such that for t large enough: $p_k^{(t)} - q_k^{(t)} = \frac{\partial}{\partial h_k} g(h^{(t)}) \leq c$, and so $\bar{h}_k^{(t)} = \sigma$. Since $h^\star \neq 0$ and every column of W has at least one nonzero value, $\forall k, q_k^\star > 0$. By continuity there exist $q_1, q_2 > 0$ such that $q_1 \leq q_k^{(t)} \leq q_2$ for t large enough. Therefore, there exists $C < 0$ such that for t large enough:

$$\frac{p_k(t) - q_k^{(t)}}{q_k^{(t)}} \bar{h}_k^{(t)} = h_k^{(t)} - h_k^{(t+1)} \leq C < 0$$

which contradicts the fact that $h_k^{(t)}$ converges.

□

The essential point in [Lin, 2007b] is that the authors prove their modified updates still yield a descent algorithm, whatever the value of σ. Whether this is still true in the case of the Itakura-Saito divergence is an open question. We make the following conjecture: there exists $\sigma^\star$, and a sequence $(\sigma^{(t)})_{t \geq 0}$ such that:

$$\forall t, g(h^{(t+1)}) \leq g(h^{(t)}) \tag{2.45}$$

$$\sigma^{(t)} > \sigma^\star \,. \tag{2.46}$$

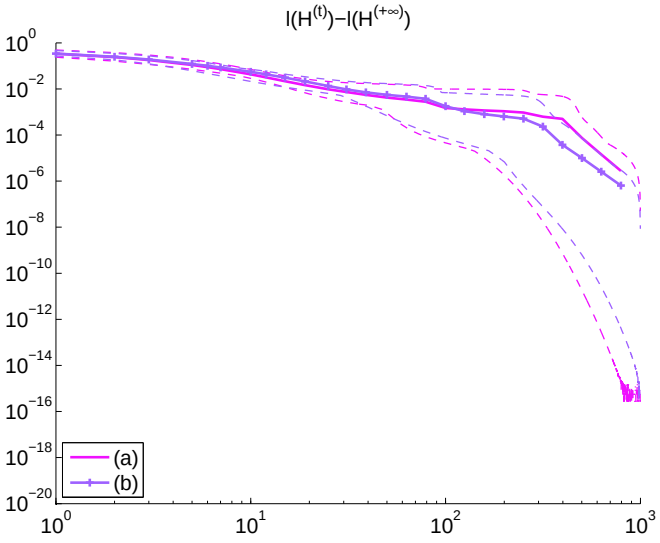
If this is true, then the proof of theorem 2 still holds.

In this section, we have highlighted two important facts : descent of the cost function is important to ensure boundedness of sequences generated by multiplicative updates. Dealing with absorbing zeroes is important to obtain limit points with good theoretical properties. Modified multiplicative updates suppress the problem of absorbing zeroes but descent of the cost function still needs to be fixed.

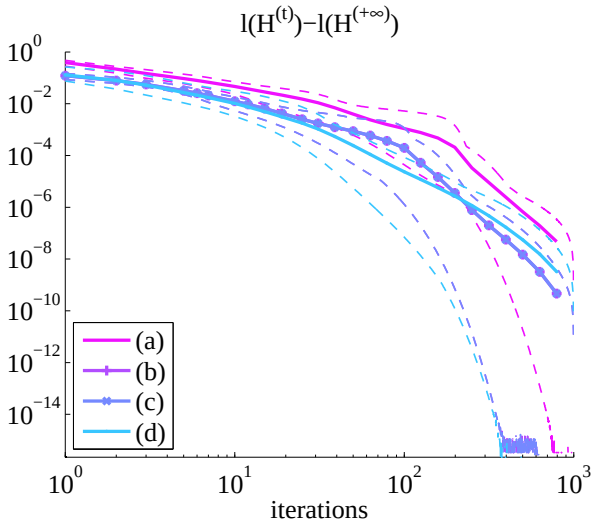
2.3.6 Discussion of convergence properties: empirical results

In this subsection we focus on the case $N = 1$ and W is fixed to examine the behavior of multiplicative updates of h. In Figure (2.7a) we compare the rate of convergence of the algorithm for $\delta = 1$ and $\delta = 0.5$, obtained on synthetic data sets. A surprising property of multiplicative updates with $\delta = 1$ is that while the auxiliary function is not fully minimized, the quantity $g(h^{(t)})$ decreases much faster than with $\delta = 0.5$. This is another missing elements in the analysis of multiplicative updates.

In order to monitor the rate of change of the objective function, we display $f(h^{(t)}) - f(h^{(\infty)})$ where $h^{(\infty)}$ is the last iterate of h we obtain. Dotted lines indicate variation of each curve across 10 different data sets. The differences in $l^\infty = f(h^\infty)$ between $\delta = 1$ and 0.5 are negligible with respect to the variations across data sets. Empirically, the rate of convergence of multiplicative updates is a power law (linear in log-log scale), so that the number of iterations before stopping



(a) Compare updates with (a) $\delta = 1$,(b) $\delta = 0.5$



(b) Different initializations schemes, cf. text for legend. Curve (b) is masked by curve (c)

Figure 2.7: Convergence of the loss function value for different initializations. $N = 1$,$F = 10^3$, $K = 30$. Data is sampled from a known model W. Cost is averaged over 10 data sets (plain curves), and standard deviations are plotted (dashed lines).

the algorithm can vary dramatically depending on the precision threshold (10 iterations for a precision of 10^{-2}, 100 iterations for a precision of 10^{-3}).

MM algorithms are prone to local minima and thus very sensitive to initialization, so we try different strategies :

(a) Choose $h^{(0)}$ at random

(b) $h^{(0)} = \mathbf{1}$

(c) $h^{(0)} = c^\star \mathbf{1}$, the optimal $c^\star$ being given by ($c = \frac{1}{FK} \sum_f v_f (W\mathbf{1})_f^{-1}$).

(d) $h_k^{(0)} = 1/K \arg\min_h d_{IS}(v, W_{\cdot k} h)$.

(e) (Not shown) $h^{(0)} = \arg\min_{h \geq 0} \sum_f \frac{(v_f - (Wh)_f)^2}{(v_f + \sigma)^2}$. The objective function is a second-order approximation of $d_{IS}(v, \tilde{v})$ at $\tilde{v} = v$. The σ makes the problem less ill-conditioned. That method yielded worse local minima.

In Figure (2.7b), we compare those 4 initialization methods. How good the initial estimates are really makes the difference in the ten first iterations. After that, the four different initialization schemes do not make much difference except that (a) and (d) tend to be more robust across data sets.

2.3.7 Overview of the algorithm

Algorithm 1 summarizes the descent algorithm that we have laid out in the previous subsections. We will denote by $(W^{(t)}, H^{(t)})_{t \geq 1}$ any sequence of updates given by this Algorithm. Note that non-negativity of W and H is naturally handled by multiplicative updates since only multiplications, divisions and additions by nonnegative numbers are used. The use of a small $\epsilon > 0$ constant guarantees that multiplicative updates are well defined even if a whole row of H or a whole column of W goes to zero.

It has a time complexity of $O(FKN)$ and a memory complexity of $O(FK + KN)$. In audio applications, F may vary between 128 and 2048 while N grows proportionally to the length of the analyzed signal, for 10 seconds' signals N is typically of the order of 10^4. K depends on the number of sources and their complexity.

Note that if $W_{fk}^{(t)} = 0$ (resp. $H_{kn}^{(t)}$) for some t then $\forall s > t, W_{fk}^{(s)} = 0$. Special care must be taken in Algorithm 1 if any column of $W^{(t)}$ becomes equal to zero, or any row of $H^{(t)}$: in that case, at any rate, subsequent updates of H_{kn} will be equal to zero for that particular k, so one should stop updating W_{fk} and H_{kn} for all (f, n) and consider the sequences $H_{kn}^{(t)}$ and $W_{fk}^{(t)}$ as stationary.

2.3.8 Related Work

Finding efficient algorithms for NMD when the ℓ_2 loss is used (also called nonnegative least squares) has been an active topic of research. In this case, there are

Input V, (W, H), δ,T
For T iterations
$\hat{V} \leftarrow WH$
$$H \leftarrow H \odot \left(\frac{W^\top (\epsilon + V) \odot (\epsilon + \hat{V})^{\cdot -2})}{W^\top ((\epsilon + \hat{V})^{\cdot -1})} \right)^{\cdot \delta} \quad \hat{V} \leftarrow WH \,,$$
$$W \leftarrow W \odot \left(\frac{H^\top ((\epsilon + V) \odot (\epsilon + \hat{V})^{\cdot -2})}{H^\top ((\epsilon + \hat{V})^{\cdot -1})} \right)^{\cdot \delta} ,$$
$\Lambda = \mathrm{diag}(\|W_{\cdot 1}\|_1, \ldots, \|W_{\cdot K}\|_1)$
$W \leftarrow W \Lambda^{-1} \quad H \leftarrow \Lambda H.$
End

Algorithm 1 Multiplicative updates algorithm for IS-NMF

several alternatives to multiplicative updates that are more efficient : active set methods [Kim and Park, 2008], block-coordinate descent algorithms [Gillis and Glineur, 2012, Mairal et al., 2010], and projected gradient descent [Lin, 2007a]. A comparison of these algorithms on several types of datasets (image, text, sound) may be found in [Gillis and Glineur, 2012]. Interestingly, the authors show that the complexity of updating H in block-coordinate descent is essentially equal to that in multiplicative updates.

2.4 Group-sparsity enforcing penalty in NMF

For simple signals, individual components of NMF were found to retrieve meaningful signals such as notes or events [Smaragdis et al., 2007, Févotte et al., 2009]. However, when applied to more complex signals, such as music instruments, it is more reasonable to suppose that each sound source corresponds to a subset of components. Grouping is usually done either by the user, but as the number of components grows large, this task becomes time-consuming and also very subjective. Automatic grouping criteria were proposed in [Murao et al., 2010], based on the correlation of atoms $W_{\cdot k}$ or the decorrelation of their decomposition coefficients. In this case the complexity of optimizing such criteria is combinatorial as it requires searching all permutations of components.

In this Section, we argue that grouping may be incorporated in the inference of the dictionary $\mathbf{W}$ as a maximum-likelihood problem penalized by a group-sparsity inducing term.

The penalty that we propose is designed for audio signals containing intervals where one of the sources is missing. The simplest case is that of a two instrument track divided in three parts: one part with instrument A, one part with instrument B, and a third part where both instruments are mixed. This setting may be generalized to the case of several sources. Provided that for each source, there is at least one interval with this and only this source missing, it was shown

in [Laurberg et al., 2008b] that this setting is equivalent to learning a dictionary on isolated source signals and un-mixing them when they are mixed. Our group-sparsity penalty allows identifying segments where sources are missing, learn an appropriate dictionary for each source, and un-mix sources elsewhere.

Sparsity-inducing functions have been a subject of intensive research. According to the loss function used, either sparsity-inducing norms [Bengio et al., 2010, Jenatton et al., 2009] or divergences [Smaragdis et al., 2007, Virtanen, 2007] are preferred. The penalty term we introduce is designed to deal with a specific choice of loss function, the Itakura-Saito divergence.

After presenting our model for group structure, we derive it from a simple graphical model, and give an intuitive interpretation in terms of amplitude of the sources.

Warning By "Group structure", we mean here that *components* of NMF are grouped (see Figure 2.8). This is different from grouping *features* as is done in feature selection or grouping *observations* as might be done in clustering. This has practical implications since in this case groups tie columns of W AND rows of H, but the NMF problem is still separable in n and f.

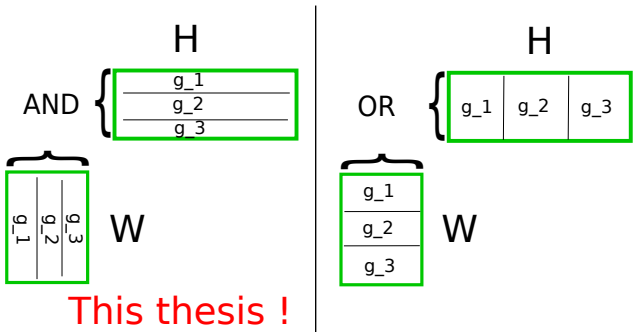


Figure 2.8: Different meanings of group-sparsity in matrix factorization

2.4.1 Presentation

We wish to partition the K components into G non-overlapping groups. In the following a source will be uniquely identified by the subset g to which it corresponds. In the framework of statistical inference, many priors have been proposed to identify components, either on W or H or both (see e.g., [Smaragdis et al., 2007, Virtanen, 2007]). We focus here on a simple grouping principle : if a source is inactive at a given frame n of the spectrogram, then all the corresponding gains H_{gn} should be set to zero.

An additional benefit of our group-sparsity penalty is that it automatically solves the practical problem of assigning components to sources once NMF is

computed from mixed signals.

$$\begin{array}{rl} \min & D_{IS}(V, WH) + \lambda\Psi(H), \\ W \geq 0, H \geq 0 & \\ \forall k,\ \|W_{\cdot k}\|_1 = 1 & \end{array} \tag{2.47}$$

with $\Psi(H) = \sum_{g,n} \psi(\|H_{gn}\|_1)$ and $\psi(x) = \log(a+x)$. We refer to Eq. (2.47) as the GIS-NMF problem (group Itakura-Saito NMF), and call $\mathcal{L}(W, H)$ the objective function. Eq. (2.47) generalizes IS-NMF in the sense that when $\lambda = 0$ we recover the standard IS-NMF problem. In GIS-NMF a tradeoff is made between the fit to data as measured by the loss term, and the grouping criterion defined by Ψ. Although we impose a particular choice of ψ, note that for optimization purposes we only require that ψ be a differentiable, concave, increasing function.

2.4.2 Interpretation of the penalty term

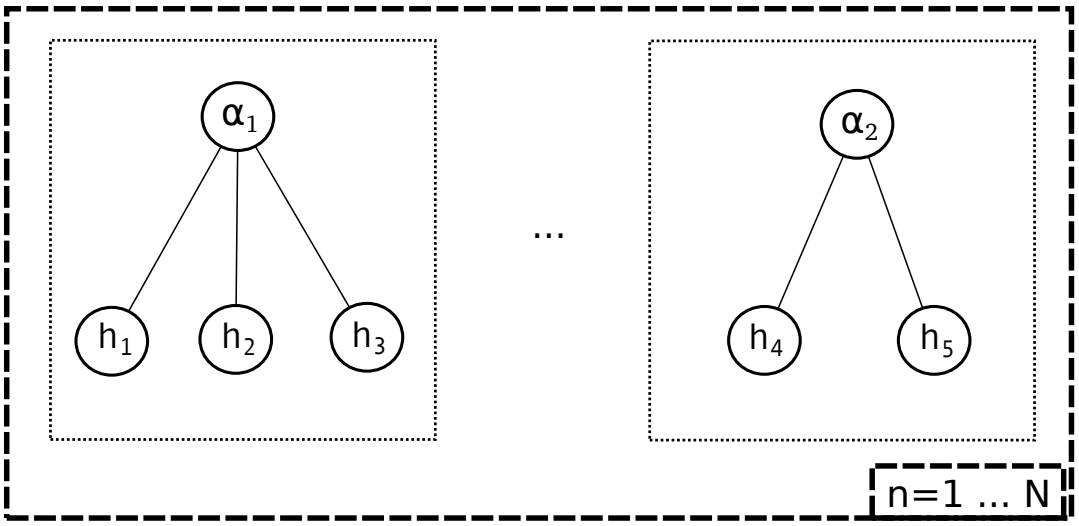


Figure 2.9: A graphical model for grouping components in NMF

The penalty term $\psi(H)$ may be interpreted in terms of a probabilistic model with latent variables. Assume $\forall k, \sum_f W_{fk} = 1$. For $g = 1 \dots G$, and $n = 1 \dots N$, define random variables $\alpha_n^{(g)}$ and suppose they are mutually independent and identically drawn from an inverse Gamma distribution with shape parameter b and scale a:

$$p(\alpha) = \frac{a^b}{\Gamma(b)} \alpha^{b-1} \exp(-\frac{\alpha}{a}) \,. \tag{2.48}$$

Furthermore we suppose that the conditional distribution of the gains H_{kn} factorizes in groups, i.e.,

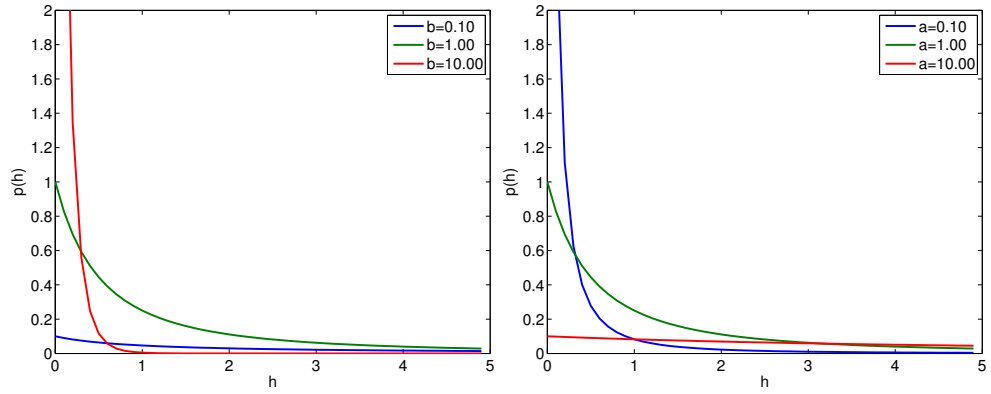
$$p(H_{\cdot n} | (\alpha_n^{(g)})_{g \in \mathcal{G}}) = \prod_g \prod_{k \in g} p(h_{kn} | \alpha_n^{(g)}) \tag{2.49}$$

and that h_{kn} are exponentially distributed conditionally on $\alpha_n^{(g)}$, with mean $\alpha_n^{(g)}$. The conditional independence structure of this model is captured in Figure 2.9

The marginal distribution of $H_{\cdot n}$ is then given by:

$$p(H_{\cdot n}) = \prod_g \frac{\Gamma(K_g + b)}{\Gamma(b)} \frac{a^b}{(a + \|H_{gn}\|_1)^{b+K_g}} \,. \tag{2.50}$$

By taking the minus logarithm of this expression, one obtains the penalty term $\psi(H)$ in Eq. (2.47), with $\lambda = b + K_g \geq 0$.



(a) $a = 1$ (b) $b = .1$

Figure 2.10: Probability density function of a Gamma random variable, for various values of the shape b and scale a

Now we can give another interpretation of our model in physical terms: since, $\sum_f W_{fk} = 1$, then $\sum_{k \in g} H_{kn} = \sum_f \hat{V}_{fn}^{(g)} = \mathbb{E}(\sum_f |S_{fn}^{(g)}|^2)$. $\sum_{k \in g} H_{kn}$ is a local measure of the energy of source g in time bin number n. Group sparsity consists thus in assuming that the energies of each source are independently distributed.

As we can see on Figure 2.10, there is little gain in varying both parameters b and a, since the effect on the penalty term is essentially the same: large values of b (resp. small values of a) favor small values of h. In our experiments, we thus fix a to a small value while varying b to control sparsity. Since $\lambda = K + b$, λ is the hyperparameter we will tune in our experiments.

2.4.3 Extension to block-structured penalties

One limitation of the sparsity penalty proposed in Eq. (2.47) is that temporal dependency between adjacent observations is not taken into account: indeed, musical notes or phonemes typically last from ten to hundreds of STFT frames, which does not correspond to an i.i.d. model. We thus propose to extend the group-sparsity penalty to a block-sparsity penalty:

$$\Psi(H) = \sum_{n=0}^{N-1} \sum_{g \in \mathcal{G}} \psi(\|H_{g\,n} + H_{g\,n+1}\|_1) \,. \tag{2.51}$$

Ψ is structured in overlapping blocks of size 2 in the observation dimension. This kind of smoothing differs from that proposed for instance in [Févotte, 2011b, Virtanen, 2007], who propose pairwise penalties encouraging continuity in the activation coefficients, of the form $\psi(H_{kn+1} - H_{kn})$. Penalizing $\|H_{g\,n} + H_{g\,n+1}\|_1$ favors zeroes in adjacent frames. Since each coefficient H_{kn} is linked to H_{kn-1} and H_{kn-1}, this induces a chain effect in support recovery : hopefully zeroes will tend to cluster in large blocks, instead of being scattered.

2.4.4 Algorithm for group Itakura Saito NMF

We derive an algorithm for group Itakura-Saito NMF based on the same considerations as in Section 2.3.4. Since the objective function is separable in the columns of H, we need only consider the following subproblem:

$$\begin{array}{ll} \min_h & g(h)\,, \\ h \geq 0 & \end{array} \tag{2.52}$$

where

$$g(h) = D_{IS}(v, Wh) + \lambda\psi(h)\,. \tag{2.53}$$

and minimization is over $h \in \mathbb{R}_+^K$. Since the additional penalty term is concave, we may apply the same arguments as before to obtain an auxiliary function and multiplicative updates.

To optimize with respect to W, we notice that the minimizers of Eq. (2.47) are also minimizers of:

$$\begin{array}{ll} \min & D_{IS}(V, WH) + \lambda\Phi(W, H), \\ W \geq 0, H \geq 0 & \end{array} \tag{2.54}$$

where $\Phi(W, H) = \sum_g \sum_n \psi(\sum_{k\in g} h_{kn}\|W_{\cdot k}\|_1)$. Thus updates for W may be derived in the same way as for H. Since the objective function in (2.54) is unchanged under the transformation $W \leftarrow W\Lambda^{-1}$, $H \leftarrow \Lambda H$, where Λ is a diagonal matrix, we may rescale matrices W and H at each step to return to the feasible set of (2.47).

Thus, we derived a descent algorithm to solve Eq. (2.47), that is summed up in Algorithm 2.

Algorithm 2 provides an overview of the alternate descent algorithm. It is very similar to the standard multiplicative updates algorithm and differs only in the presence of an additional term in the denominator of the updates. This additional term in Equation (2.36) favors low values of H_{kn} : since $\psi'(x)$ decreases with x (ψ is concave), low values of $\|H_g n\|_1$ are more penalized than high values. Moreover the quantity $\|H_{gn}\|_1$ is the same for all k in group g. Thus, if at a given frame n the volume of source g is small with respect to that of source g', the updates in (2.47) tend to mute source g. We thus get the same grouping effect than the traditional penalization by the ℓ_2-norms $\|H_{gn}\|_2$ [Bengio et al., 2010], but with the added benefit of natural multiplicative updates. The choice of a is not important, as argued previously, in practice we choose a relatively small

Input V, $(\mathbf{W}, \mathbf{H})$, $\mathcal{G}$, (λ, a), δ,t
For t iterations
$\hat{V} \leftarrow WH$
For $n = 1 \dots N, g \in \mathcal{G}, k \in g$
$p_{kn} \leftarrow \psi'(\|H_{gn}\|_1)$
End

$$H \leftarrow H \odot \left(\frac{W^\top (V \odot \hat{V}^{\cdot -2})}{W^\top (\hat{V}^{\cdot -1}) + \lambda P} \right)^{\cdot \delta}, \quad \hat{V} \leftarrow WH\,,$$

For $f = 1 \dots F, g \in \mathcal{G}, k \in g$
$r_{fk} = \sum_n h_{kn} \psi'(\|H_{gn}\|_1)$
End

$$W \leftarrow W \odot \left(\frac{H^\top (V \odot \hat{V}^{\cdot -2})}{H^\top (\hat{V}^{\cdot -1}) + \lambda R} \right)^{\cdot \delta},$$

$\Lambda = \mathrm{diag}(\|W_{\cdot 1}\|_1, \dots, \|W_{\cdot K}\|_1)$
$W \leftarrow W\,\Lambda^{-1} \quad H \leftarrow \Lambda\, H.$
End

Algorithm 2 Algorithm for GIS-NMF

value $a = 0.01$. In practice the choice of regularization parameter λ is crucial and will be discussed in Section 2.5.

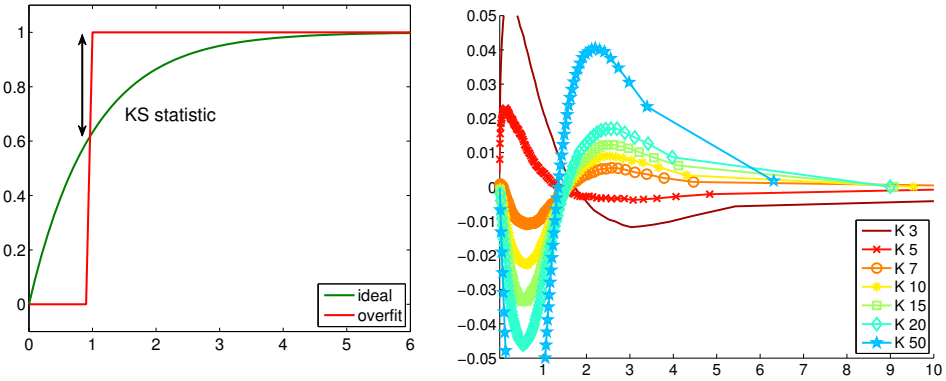
We run the algorithm with several different initializations and keep the result that yields the lowest cost value, in order to avoid local minima. The objective function decreases at each step, and convergence of the parameters is observed in practice.

2.5 Model selection in sparse NMF

Selecting the right number of components in NMF is an active line of research. When penalties or prior knowledge is introduced, selecting hyperparameters is also an important problem. Cross-validation could be used to measure the fit of learnt dictionaries to the data, but this involves computing as many NMF as there are folds in the cross-validation procedure times the size of the (K, λ) parameter grid. Instead, we propose to select models based on statistics of goodness-of-fit. Other approaches are based on probabilistic extensions of NMF [Tan and Févotte, 2009, Hoffmann et al., 2010b, Févotte and Cemgil, 2009]. We briefly present the most recent and discuss its benefits compared to ours.

2.5.1 Kolmogorov-Smirnov statistic

Define standardized observations $\varepsilon_{fn} = \frac{V_{fn}}{\hat{V}_{fn}}$. Then if the observed data follow the model in Eq.(2.22), the empirical distribution function of E converges towards



(a) sketch (b) experiment

Figure 2.11: (a) When K is too large, the empirical cdf (red) deviates from the assumed cdf $F(x) = 1 - \exp(-x)$ (green). (b) The right plot displays the discrepancy between empirical cdfs obtained for various values of K and the expected cdf.

that of an exponential random variable, in the following sense:

Theorem 3. *Let $x_1, \ldots, x_n$ be i.i.d random variables with distribution function F. Let $x_{\lfloor 1 \rfloor} \leq \ldots x_{\lfloor n \rfloor}$ define a re-ordering of $x_1, \ldots, x_n$ in increasing order. Define the empirical distribution function $\hat{F}_n$:*

$$\hat{F}_n(y) = \begin{cases} 0 & \text{if } y \leq x_{\lfloor 1 \rfloor}\,, \\ \frac{k}{n} & \text{if } x_{\lfloor k \rfloor} \leq y \leq x_{\lfloor k+1 \rfloor}\,, \\ 1 & \text{if } x_{\lfloor n \rfloor} \leq y\,. \end{cases} \tag{2.55}$$

Then,

$$\lim_{n \to +\infty} \|\hat{F}_n - F\|_\infty \xrightarrow{a.s.} 0\,. \tag{2.56}$$

The reader is referred to [Lehmann and Romano, 2005] for a detailed proof and other goodness-of-fit statistics. The quantity $\|\hat{F}_n - F\|$ is referred to as the Kolmogorov-Smirnov (KS) statistic. We propose to select the parameters of our model λ that yield the minimum KS statistic.

Intuitively, if we let $K = F$ then we obtain a perfect fit since $V = WH$ exactly. However, the empirical distribution function will be far away from that of an exponential random variable. Indeed, since all $\varepsilon_{fn} = 1$, the empirical distribution function $\hat{F}(x)$ is a step function where the step is located at $x = 1$. This is illustrated in Figure 2.11a: if $V = WH$, the KS statistic is exactly equal to $\exp^{-1} = 0.3679\ldots$. Figure 2.11b displays emprical cdfs obtained after running NMF on the piano excerpt presented in Section 2.1.3, for various values of K. As one can see, values around $K = 7$ yield the best KS statistic, in surprisingly good agreement with comments in Section 2.1.3.

The advantage of selecting with a statistic is that the number of NMF to be computed is equal to the size of the grid, unlike in cross-validation. We study the impact of this statistic on model selection in Subsection 2.6.1.

2.5.2 Bayesian approaches

In [Hoffmann et al., 2010b], the authors propose a Bayesian nonparametric approach to selecting the number of components in NMF. Instead of selecting a known number of components K, they pick a large K and introduce hidden variables θ_k for the global scale of component k. More precisely, they define the following model:

$$\begin{aligned} W_{fk} &\sim \Gamma(a, a) \\ H_{kn} &\sim \Gamma(b, b) \\ \theta_k &\sim \Gamma(\alpha/K, \alpha c) \\ V_{fn} &\sim Exp(\sum_{k=1}^{K} \theta_k W_{fk} H_{kn}) \end{aligned} \tag{2.57}$$

This model differs from Itakura-Saito NMF only in the appearance of scaling terms θ_k. As K grows, the number of components k such that $\theta_k > \epsilon$ for some $\epsilon > 0$ is finite almost surely, and is expected to be small. Exact inference of (W, H, θ) given V is intractable so the authors appeal to variational Bayesian inference, a faster alternative to MCMC sampling methods.

Bayesian nonparametric procedures are attractive because the number of components K is estimated at the same time as the model (W, H).

2.6 Experiments with group-sparse NMF

2.6.1 Validation on synthetic data

We designed an optimization procedure to enforce structured sparsity on the columns of H. In order to validate our algorithm, we picked $W^{(\star)} \in \mathbb{R}_+^{100\times 20}$ at random and $H^{(\star)}$ with two groups of 10 components each and disjoint supports. 10 synthetic data sets of various sizes were generated according to model (2.22). Define the support recovery error as the proportion of frames where the active sources are incorrectly identified. Figure 2.2 displays, for various data set sizes N, how the test statistic and the support recovery error vary with λ. For fixed N, the KS statistic reaches a minimum in the interval $[10^0, 10^2]$. As N grows large, the support recovery error decreases towards zero, and the minimizer of the KS statistic (which does not require to know the ground truth) matches the one of the recovery error.

2.6.2 Results in single channel source separation

We experiment our algorithm on two audio tracks found on the Internet Archive (`www.archive.org`): the individual sources $\mathbf{x}^{(g)}$, $g = 1 \dots 2$. were available, from which we took 20-30 seconds excerpts [2]. For each track, we propose the following

[2] Complete results on all mixtures, including .wav files, are available online (`www.di.ens.fr/~lefevrea/demos.html`)

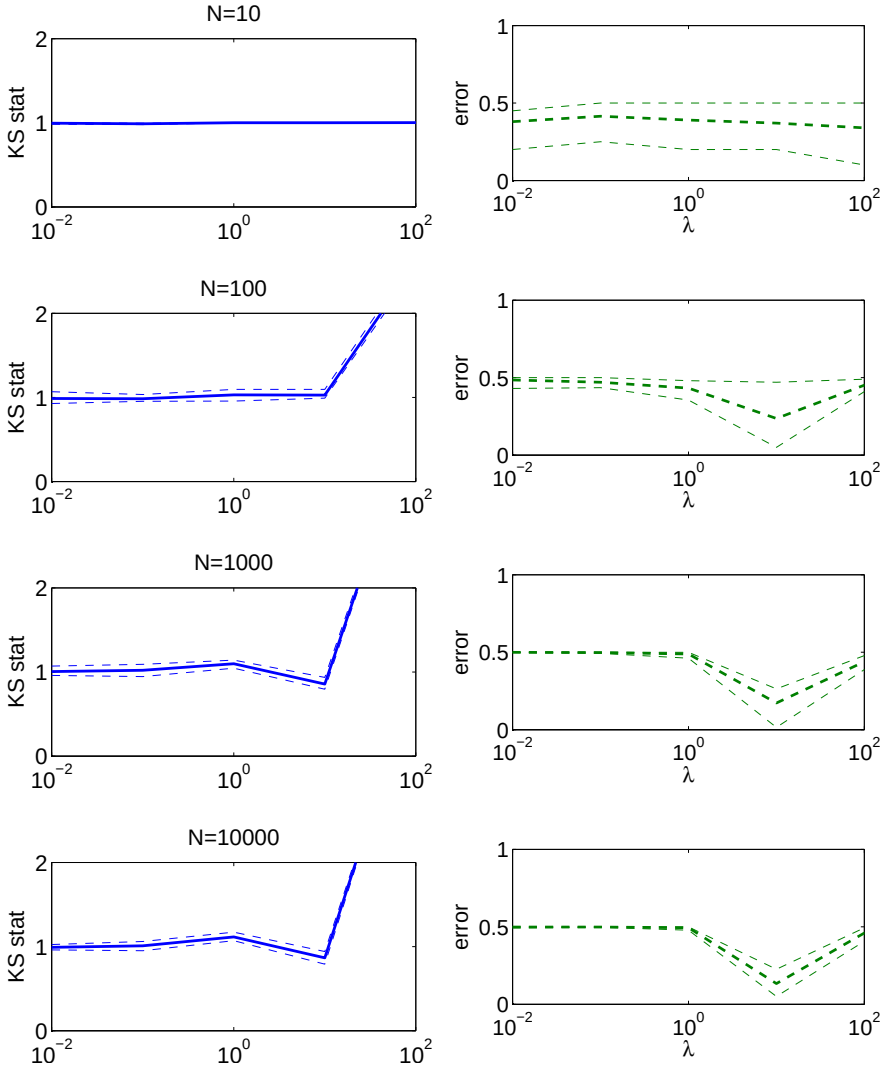


Table 2.2: Relationship between support recovery error and KS statistic as the size N of the data set increases. x-axis: regularization parameter λ. y-axis: KS statistic (solid line) and the support recovery error (dashed line). Thin dashed lines are error bars

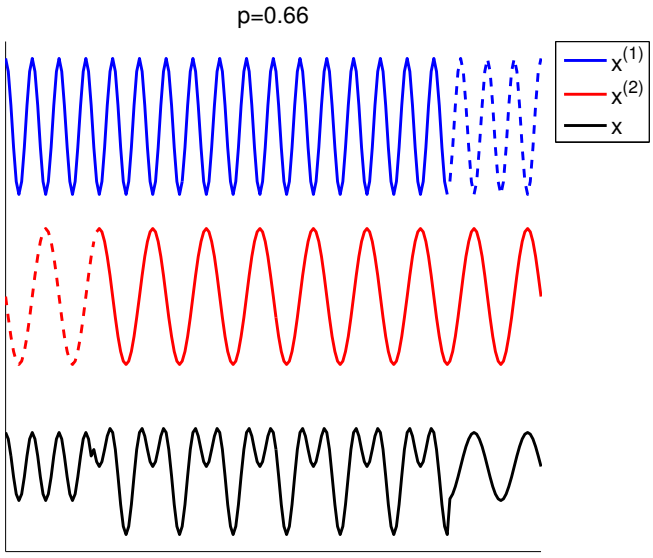


Figure 2.12: Sketch of the experimental setup. True source signals (in blue and red), are partly muted (dashed parts) in order to control that they overlap over no more than a fraction p of the total length, here $p = 66\%$. The mixed signal is displayed in black.

mixture:

$$x_n = \begin{cases} x_n^{(1)} & \text{if } n \leq \frac{1-p}{2} T \\ x_n^{(2)} & \text{if } n \geq \frac{1+p}{2} T \\ x_n^{(1)} + x_n^{(2)} & \text{otherwise} \end{cases} . \tag{2.58}$$

where T is the total length of the track: thus if $p = 0.33$, we make sure that

track	source	GIS-NMF	base	random	ideal
love	bass	**8.88**	-67.53	-8.55	8.86
0 %	guitar	**13.60**	3.77	-2.19	13.94[1]
love	bass	**4.33**	-4.60	-8.74	4.56
33 %	guitar	**9.77**	-7.40	-2.02	9.90
love	bass	**1.47**	-5.29	-9.08	3.12
66 %	guitar	**7.72**	-8.11	-1.94	8.68
love	bass	-5.13	**-4.16**	-9.02	2.54
100 %	guitar	**-0.21**	-2.68	-2.02	8.09

Table 2.3: Source to distortion ratios (SDR) for the track "We are in love"[2] . $x\%$ is the overlap between sources.

sources overlap over no more than 33% of the track. The goal is to analyze how

[1] "ideal NMF" serves for comparison, but is not an upper bound for the performance of our algorithm, see text.

important sparsity is to estimate the mixtures correctly by varying p. Table 2.3 compares our algorithm (GIS-NMF) with several other strategies:

- The *baseline* consists in estimating Itakura-Saito NMF and then group components so as to minimize $\Psi(H)$, so that $\Psi(H)$ plays the role of a heuristic criterion to group components.
- *Ideal NMF* consists in running NMF and choose groups that yield optimal SDR (by selecting from all possible of $K!$ permutations): ideally we should perform at least as well. However, note that it is not an oracle performance (not the same objective function).
- *random*: the average SDR of 10 random binary masks.

In Table 2.3 we display our results on one audio track. In GIS-NMF, parameters (a, λ) were chosen to minimize the test statistic, then we tuned the number of components per group as to maximize SDR. In most cases, we perform better than a random binary mask, unlike the baseline. For overlap p up to 66%, we obtain SDR values close to that of the ideal i.e., we find the best assignment for source separation. Thus group-sparsity in the columns of H plays a key role in identifying sources. Our algorithm meets his limits when there is too much overlap, then we fail to identify the sources correctly, and more knowledge about the sources is needed.

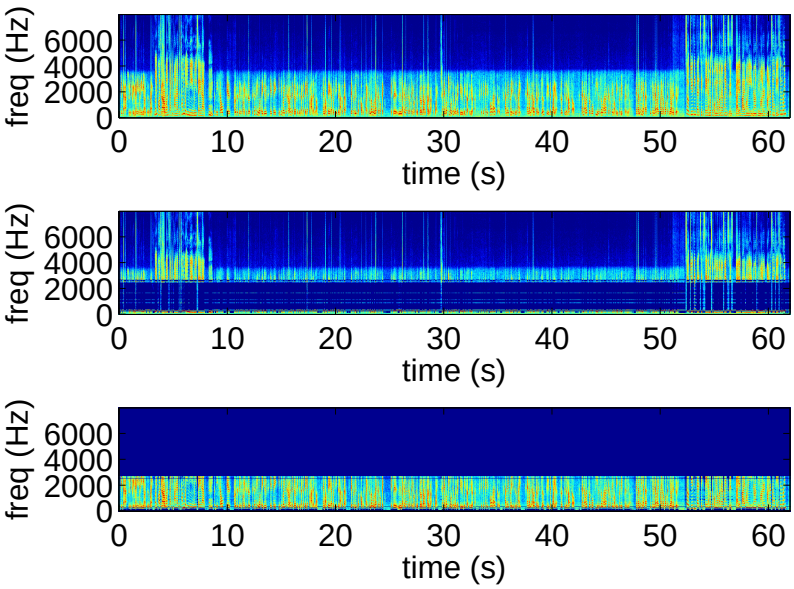
2.6.3 Block-structured penalty

We experiment the block-structured penalty presented in Section 2.4.3 on a segmentation task. The recorded signal is a radio interview with a female interviewer and a male interviewee. Figure 2.13 compares the result of NMF with and without block-structured penalty. Given the length of the recording, we chose only a few values of (K, λ), and kept $K = 15$, $\lambda = 1$. Recovered supports of each group are almost disjoint with our block-structured penalty, which is not the case with the baseline NMF. This is an interesting case where NMF fails to recover an appropriate model for each source whereas intuitively the task is particularly simple since source signals are never mixed. In particular it illustrates why sparsity is important to learn interpretable dictionaries.

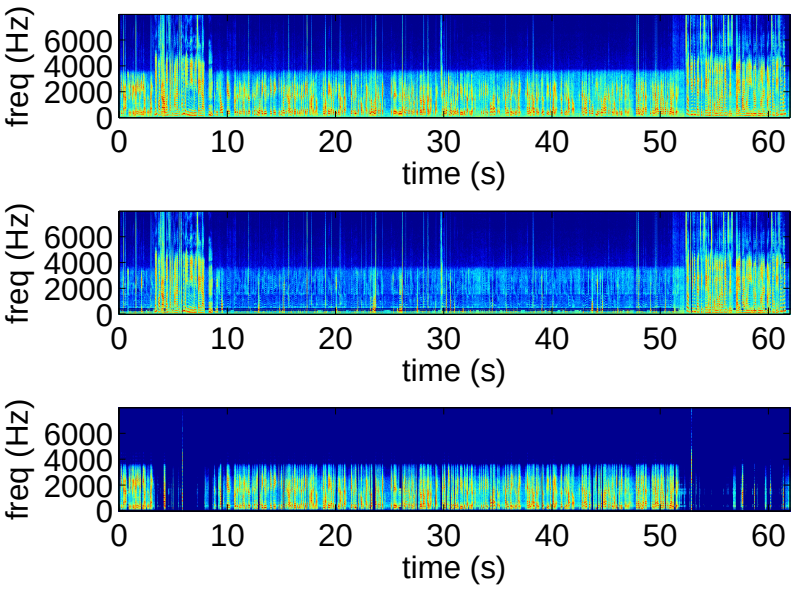
2.7 Related work

Exploiting structure in regression problems is a very active topic in statistics and optimization. In this paragraph, we make no distinction between nonnegativity-constrained problems and unconstrained ones. For clarity we will consider the following general formulation of penalized regression problems :

$$\min \|x - Da\|_2^2 + \lambda\Omega(a)\,. \tag{2.59}$$



(a) without penalty



(b) block-sparse penalty

Figure 2.13: Application of the block-sparse penalty term to a segmentation task. (Top) Mixture (Middle) Female voice estimate (Down) Male voice estimate

Regression with an ℓ_1 norm penalty was shown to promote sparsity in regression coefficients [Tibshirani, 1996]. Sparsity may be important either for prediction (e.g., denoising) or for support identification : variables which have nonzero coefficients may be interpreted as causal factors of the observed variable. Provided sufficient conditions on the design matrix, and signals are sufficiently sparse, the support may be correctly identified. As the conditioning of the design matrix worsens, more elaborate structure must be exploited to identify the support. Group-structured penalties were proposed as a way to enforce zeroes in several coefficients simultaneously [Yuan and Lin, 2006], where groups form a partition of the regression coefficient indices.

Tree-structured penalties were proposed to handle the case when variables are assumed to be embedded in a tree, and the ancestor relationship is interpreted as :

$$\alpha_k = 0 \Rightarrow \alpha_j = 0 \; \forall j \in \text{descendants}(k)\,. \tag{2.60}$$

In this case, the right penalty also has group structure but now groups have a specific overlapping structure. Efficient algorithms were proposed to compute regression coefficients in a finite number of iterations [Jenatton et al., 2011c]. Extensions to general loss functions (other than the ℓ_2 norm) are proposed based on these algorithms. Recovery in the general case of overlapping groups was studied in [Jenatton et al., 2011a].

When a collection of signals are presented simultaneously, as is the case in source separation, there are various settings.

$$\min \|X - DA\|_F^2 + \lambda\Omega(A) \tag{2.61}$$

In multi-task regression, each dimension of the observed signals is interpreted as a different task, and group-sparsity is imposed in such a way that multiple tasks share the same nonzero regression coefficients, for all observations, see e.g., [Sprechmann et al., 2011]. In the other setting, various sets of observations share the same regression coefficients. As argued in [Bengio et al., 2010], "this approach can also be used to encourage using the same dictionary words for all the images in a class, providing a discriminative method in the construction of image representations".

Note that using mixed norms is not the only way of exploiting structure in regression problems. Coming back to the "hard" sparse coding problem where the pseudo-norm $\Omega(a) = \|a\|_0$ is used, structured decompositions specifically tailored for audio signals were proposed in [Daudet, 2006]. In this case, structure is exploited by adding simultaneously groups of coefficients at each step of a matching pursuit algorithm.

2.8 Conclusion

In this Section, we have provided an overview of NMF with the family of β divergences. Among this family, the Itakura-Saito divergence was singled out

because it provides a generative model which accounts for the additivity of the mixing process. It is the only one which does not rely on approximate additivity of the power spectrograms.

Optimizing NMF with β divergences implies a careful comparison between algorithms : for $\beta = 2$, projected gradient descent algorithms are preferable to multiplicative updates, but the latter are faster for $\beta = 0$. Moreover, the story of multiplicative updates is still not closed: indeed, we have seen that in this case that partial optimization of the auxiliary function yields faster convergence rates than full optimization. Multiplicative updates suffer from the problem of absorbing zeroes, which is especially problematic when using sparsity penalties: in order to be able to select components accurately it is necessary that all subsets of components be searched throughout the algorithm, which is not possible with absorbing zeroes. Adapting propositions of [Lin, 2007b] to the general case of β divergences is an interesting direction, and will be the subject of future work.

We have introduced a penalized maximum-likelihood principle to find groups in NMF with the Itakura-Saito divergence. Instead of finding groups after running NMF, which is computationally expensive, we introduce a grouping criterion based on the natural structure of music signals.

Our algorithm keeps the attractive features of multiplicative updates algorithm (low complexity, descent property), and allows performing blind source separation on complex signals, with no assumption on the frequency profiles of the sources. Moreover, we show how to incorporate temporal dependencies between coefficients of H by replacing group-sparsity by block-sparsity. Other penalties to capture such dependencies were proposed e.g., [Févotte, 2011a], which enforce smoothness rather than sparsity. Finally, introducing temporal dependencies between components of NMF is still an open subject.

Indeed, the Euclidean loss accommodates very-well Markov terms of the form $P(H) = \sum_n \sum_{k,k'} H_{k,n} L H_{k',n+1}$, where L may be learnt on training data. It would be interesting to investigate possible adaptations to beta-divergences.

Chapter 3

Online algorithms for large-scale nonnegative matrix factorization

Our contribution in this Chapter is an online algorithm to learn dictionaries adapted to the Itakura-Saito divergence. We show that it allows a ten times speedup for signals longer than three minutes, in the small dictionary setting. It also allows running NMF on signals longer than an hour which was previously impossible. This work has led to the following publication(s):

> A. Lefèvre and F. Bach and C. Févotte, "Online algorithms for nonnegative matrix factorization with the Itakura-Saito divergence", *in Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2011.

Code is available online[a]

[a] www.di.ens.fr/ lefevrea/xnmf.zip

Estimating the dictionary can be quite slow for long audio signals, and indeed intractable for training sets of more than a few hours. We propose an algorithm to estimate Itakura-Saito NMF (IS-NMF) on audio signals of possibly infinite duration with tractable memory and time complexity. This chapter is organized as follows : in Section 3.1, we review the essential structure of the auxiliary function used to derive multiplicative updates for Itakura-Saito NMF, then propose a recursive computation of the auxiliary function, which is the essential ingredient of our online algorithm, and provide implementation details. In Section 3.2, we experiment our algorithms on real audio signals of short, medium and long durations. We show that our approach outperforms regular batch NMF in terms of computer time.

3.1 An online algorithm for Itakura-Saito NMF

Various methods were recently proposed for online dictionary learning [Mairal et al., 2010, Hoffmann et al., 2010a, Bucak and Gunsel, 2009]. However, to the best of our knowledge, no algorithm exists for online dictionary learning with the Itakura-Saito divergence. In this section we summarize IS-NMF, then introduce our algorithm for online NMF and explain briefly the mathematical framework.

3.1.1 Itakura-Saito NMF

Define the Itakura-Saito divergence as $d_{IS}(y,x) = \sum_i (\frac{y_i}{x_i} - \log \frac{y_i}{x_i} - 1)$. Given a data set $V = (v_1, \ldots, v_N) \in \mathbb{R}_+^{F \times N}$, Itakura-Saito NMF consists in finding $W \in \mathbb{R}_+^{F \times K}$, $H = (h_1, \ldots, h_N) \in \mathbb{R}_+^{K \times N}$ that minimize the following objective function :

$$\mathcal{L}_H(W) = \frac{1}{N} \sum_{n=1}^{N} d_{IS}(v_n, W h_n)\,, \tag{3.1}$$

The standard approach to solving IS-NMF is to optimize alternately in W and H and use majorization-minimization [Févotte and Idier, 2011]. At each step, the objective function is replaced by an auxiliary function of the form $\mathcal{L}_H(W, \underline{W})$ such that $\mathcal{L}_H(W) \leq \mathcal{L}_H(W, \underline{W})$ with equality if $W = \underline{W}$:

$$\mathcal{L}_H(W, \underline{W}) = \sum_{fk} A_{fk} \frac{1}{W_{fk}} + B_{fk} W_{fk} + c\,. \tag{3.2}$$

where $A, B \in \mathbb{R}_+^{F \times K}$ and $c \in \mathbb{R}$ are given by:

$$\begin{aligned} A_{fk} &= \textstyle\sum_{n=1}^{N} H_{kn} V_{fn} (\underline{W} H)_{fn}^{-2} \underline{W}_{fk}^2\,, \\ B_{fk} &= \textstyle\sum_{n=1}^{N} H_{kn} (\underline{W} H)_{fn}^{-1}\,, \\ c &= \textstyle\sum_{f=1}^{F} \sum_{n=1}^{N} \log \frac{V_{fn}}{(\underline{W} H)_{fn}} - F\,. \end{aligned} \tag{3.3}$$

Thus, updating W by $W_{fk} = \sqrt{A_{fk}/B_{fk}}$ yields a descent algorithm. Similar updates can be found for h_n so that the whole process defines a descent algorithm in (W, H) (for more details see, e.g., [Févotte and Idier, 2011]). In a nutshell, batch IS-NMF works in cycles: at each cycle, all sample points are visited, the whole matrix H is updated, the auxiliary function in Eq. (3.2) is re-computed, and W is then updated. We now turn to the description of online NMF.

3.1.2 Recursive computation of auxiliary function

When N is large, multiplicative updates algorithms for IS-NMF become expensive because at the dictionary update step, they involve large matrix multiplications with time complexity in $O(FKN)$ (computation of matrices A and B). We present here an online version of the classical multiplicative updates algorithm, in the sense that only a subset of the training data is used at each step of the algorithm.

Suppose that at each iteration of the algorithm we are provided a new data point v_t, and we are able to find h_t that minimizes $d_{IS}(v_t, W^{(t)}h_t)$. Let us rewrite the updates in Eq. (3.3). Initialize $A^{(0)}, B^{(0)}, W^{(0)}$ and at each step compute :

$$\begin{aligned} A^{(t)} &= A^{(t-1)} + (\tfrac{v_t}{(W^{(t-1)}h_t)^2} h_t^\top) \cdot (W^{(t-1)})^2 \,, \\ B^{(t)} &= B^{(t-1)} + \tfrac{1}{W^{(t-1)}h_t} h_t^\top \,, \\ W^{(t)} &= \sqrt{\tfrac{A^{(t)}}{B^{(t)}}} \,. \end{aligned} \tag{3.4}$$

Now we may update W each time a new data point v_t is visited, instead of visiting the whole data set. This differs from batch NMF in the following sense : suppose we replace the objective function in Eq. (3.1) by

$$L_T(W) = \frac{1}{T} \sum_{t=1}^{T} d_{IS}(v_t, Wh_t) \,, \tag{3.5}$$

where $(v_1, v_2, \dots, v_t, \dots)$ is an infinite sequence of data points, and the sequence $(h_1, \dots, h_t, \dots)$ is such that h_t minimizes $d_{IS}(v_t, W^{(t)}h)$. Then we may show that the modified sequence of updates corresponds to minimizing the following auxiliary function :

$$\hat{L}_T(W) = \sum_k \sum_f \left(A_{fk}^{(T)} \frac{1}{W_{fk}} + B_{fk}^{(T)} W_{fk} \right) + c \,. \tag{3.6}$$

If T is fixed, this problem is exactly equivalent to IS-NMF on a finite training set. Whereas in the batch algorithm described in Section 3.1.1, all H is updated once and then all W, in online NMF, each new h_t is estimated exactly and then W is updated once. Another way to see it is that in standard NMF, the auxiliary function is updated at each pass through the whole dataset from the most recent updates in H, whereas in online NMF, the auxiliary function takes into account all updates starting from the first one.

Extensions Prior information on H or W is often useful for imposing structure in the factorization [Lefèvre et al., 2011a, Virtanen, 2007, Smaragdis et al., 2007]. Our framework for online NMF easily accommodates penalties such as :

- Penalties depending on the dictionary W only.

- Penalties on H that are decomposable and expressed in terms of a concave increasing function ψ, such as those presented in chapter 2 : $\Psi(H) = \sum_{n=1}^{N} \psi(\sum_k H_{kn})$.

3.1.3 Practical implementation

Algorithm 3 Online Algorithm for IS-NMF

Input training set, $W^{(0)}$, $A^{(0)}$, $B^{(0)}$, ρ, β, η, ε.
$t \leftarrow 0$
repeat
  $t \leftarrow t + 1$
  draw v_t from the training set.
  $h_t \leftarrow \arg\min_h d_{IS}(\varepsilon + v_t, \varepsilon + Wh)$
  $a^{(t)} \leftarrow (\frac{\varepsilon + v_t}{(\varepsilon + Wh_t)^2} h_t^\top) \cdot W^2$
  $b^{(t)} \leftarrow \frac{1}{\varepsilon + Wh_t} h_t^\top$
  if $t \equiv 0\ [\beta]$
    $A^{(t)} \leftarrow A^{(t-\beta)} + \rho \sum_{s=t-\beta+1}^{t} a^{(s)}$
    $B^{(t)} \leftarrow B^{(t-\beta)} + \rho \sum_{s=t-\beta+1}^{t} b^{(s)}$
    $W^{(t)} \leftarrow \sqrt{\frac{A^{(t)}}{B^{(t)}}}$
    for $k = 1 \dots K$
      $s \leftarrow \sum_f W_{fk}\,, \quad W_{fk} \leftarrow W_{fk}/s$
      $A_{fk} \leftarrow A_{fk}/s\,, \quad B_{fk} \leftarrow B_{fk} \times s$
    end for
  end if
until $\|W^{(t)} - W^{(t-1)}\|_F < \eta$

We provided a description of a pure version of online NMF, we now discuss several extensions that are commonly used in online algorithms and allow considerable gains in speed.

Finite data sets. When working on finite training sets, we cycle over the training set several times, and randomly permute the samples at each cycle.

Sampling method for infinite data sets. When dealing with large (or infinite) training sets, samples may be drawn in batches and then permuted at random to avoid local correlations of the input.

Fresh or warm restarts. Minimizing $d_{IS}(v_t, Wh_t)$ is an inner loop in our algorithm. Finding an exact solution h_t for each new sample may be costly (a rule of thumb is 100 iterations from a random point). A shortcut is to stop the inner loop before convergence. This amounts to compute only an upper-bound of

$d_{IS}(v_t, Wh_t)$. Another shortcut is to warm restart the inner loop, at the cost of keeping all the most recent regression weights $H = (h_1, \ldots, h_N)$ in memory. For small data sets, this allows running online NMF very similarly to batch NMF : each time a sample is visited h_t is updated only once, and then W is updated. When using warm restarts, the time complexity of the algorithm is not changed, but the memory requirements become $O((F + N)K)$.

Mini-batch. Updating W every time a sample is drawn costs $O(FK)$: as shown in simulations, we may save some time by updating W only every β samples i.e., draw samples in batches and then update W. This is also meant to stabilize the updates.

Scaling past data. In order to speed up the online algorithm it is possible to scale past information so that newer information is given more importance :

$$\begin{aligned} A^{(t+\beta)} &= A^{(t)} + \rho \textstyle\sum_{s=t+1}^{t+\beta} a^{(s)}\,, \\ B^{(t+\beta)} &= B(t) + \rho \textstyle\sum_{s=t+1}^{t+\beta} b^{(s)}\,, \end{aligned} \tag{3.7}$$

where we choose $\rho = r^{\beta/N}$. We choose this particular form so that when $N \to +\infty$, $\rho = 1$. Moreover, ρ is taken to the power β so that we can compare performance for several batch sizes and the same parameter r. In principle this rescaling of past information amounts to discount each new sample at rate ρ, thus replacing the objective function in Eq. (3.5) by :

$$\frac{1}{\sum_{t=1}^{T} r^t} \sum_{t=1}^{T} r^{T+1-t} l(v_t, W)\,, \tag{3.8}$$

Rescaling W. In order to avoid the scaling ambiguity, each time W is updated, we rescale $W^{(t)}$ so that its columns have unit norm. $A^{(t)}$, $B^{(t)}$ must be rescaled accordingly (as well as H when using warm restarts). This does not change the result and avoids numerical instabilities when computing the product WH.

Dealing with small amplitude values. The Itakura-Saito divergence $d_{IS}(y, x)$ is badly behaved when either $y = 0$ or $x = 0$. As a remedy we replace it in our algorithm by $d_{IS}(\varepsilon + y, \varepsilon + x)$. The updates were modified consequently in Algorithm 3.

Overview. Algorithm 3 summarizes our procedure. The two parameters of interest are the mini-batch size β and the forgetting factor r. Note that when $\beta = N$, and $r = 0$, the online algorithm is equivalent to the batch algorithm.

3.2 Experimental study

In this section we validate the online algorithm and compare it with its batch counterpart. A natural criterion is to train both on the same data with the same initial parameters $W^{(0)}$ (and $H^{(0)}$ when applicable) and compare their respective fit to a held-out test set, as a function of the computer time available for learning. The input data are power spectrogram extracted from single-channel audio tracks sampled at $16000Hz$, with analysis windows of 512 samples and 256 samples overlap. All silent frames were discarded.

We make the comparison for small, medium, and large audio tracks (resp. 10^3, $10^4, 10^5$ time windows). W is initialized with random samples from the train set. For each process, several seeds were tried, the best seed (in terms of objective function value) is shown for each process. Finally, we use $\varepsilon = 10^{-10}$ which is well below the hearing threshold.

Small data set (30 seconds). We ran online NMF with warm restarts and one update of h every sample. From Figure 3.1, we can see that there is a restriction on the values of (β, r) that we can use : if $r < 1$ then β should be chosen larger than 1. On the other hand, as long as $r > 0.5$, the stability of the algorithm is not affected by the value of β. In terms of speed, clearly setting $r < 1$ is crucial for the online algorithm to compete with its batch counterpart. Then there is a tradeoff to make in β : it should be picked larger than 1 to avoid instabilities, and smaller than the size of the train set for faster learning (this was also shown in [Mairal et al., 2010] for the square loss).

Medium data set (4 minutes). We ran online NMF with warm restarts and one update of h every sample. The same remarks apply as before, moreover we can see on Figure 3.2 that the online algorithm outperforms its batch counterpart by several orders of magnitude in terms of computer time for a wide range of parameter values.

Large data set (1 hour 20 minutes). For the large data set, we use fresh restarts and 100 updates of h for every sample. Since batch NMF does not fit into memory any more, we compare online NMF with batch NMF learnt on a subset of the training set. In Figure 3.3, we see that running online NMF on the whole training set yields a more accurate dictionary in a fraction of the time that batch NMF takes to run on a subset of the training set. We stress the fact that we used fresh restarts so that there is no need to store H offline.

The online algorithm we proposed is stable provided minimal restrictions on the values of the parameters (r, β) : if $r = 1$, then any value of β is stable. If $r < 1$ then β should be chosen large enough. Clearly there is a tradeoff in choosing the mini-batch size β, which is explained by the way it works : when β is small, frequent updates of W are an additional cost as compared with batch NMF. On the other hand, when β is small enough we take advantage of the redundancy

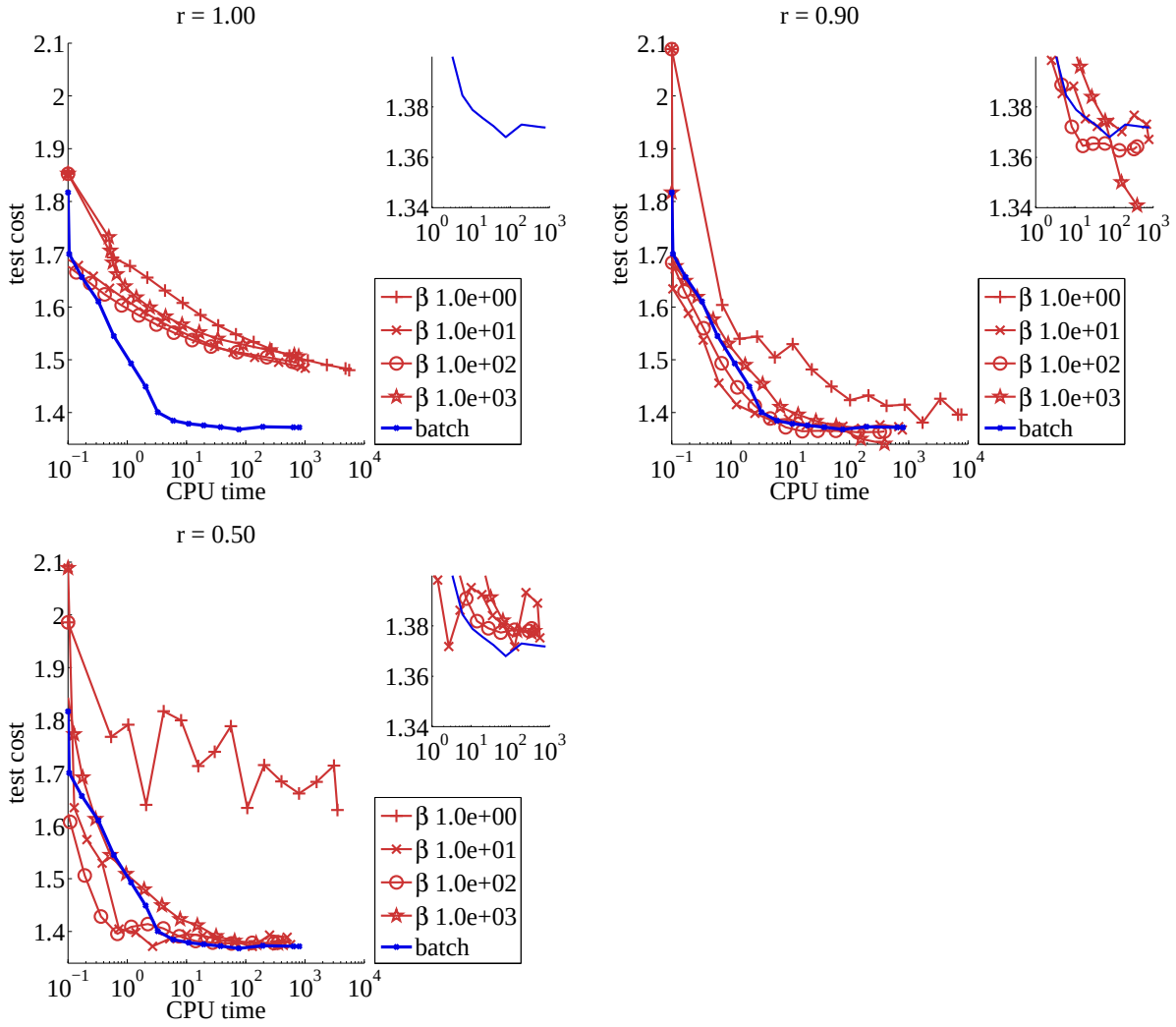


Figure 3.1: Comparison of online and batch algorithm on a thirty-seconds long audio track.

in the training set. From our experiments we find that choosing $r = 0.7$ and $\beta = 10^3$ yields satisfactory performance.

3.3 Related Work

Since the publication of our work [Lefèvre et al., 2011b], there have been a number of publications on online algorithms for NMF, which fall into two main categories : algorithms based on stochastic gradient descent, and algorithms based on the method of cumulating auxiliary functions. The latter category, to which our contribution belongs, may be traced back to [Zhang and Scordilis, 2008]. It includes : matrix factorizations with the ℓ_2 loss [Mairal et al., 2010] and generic constraint sets (nonnegativity, sparsity), matrix factorizations for the Kullback-

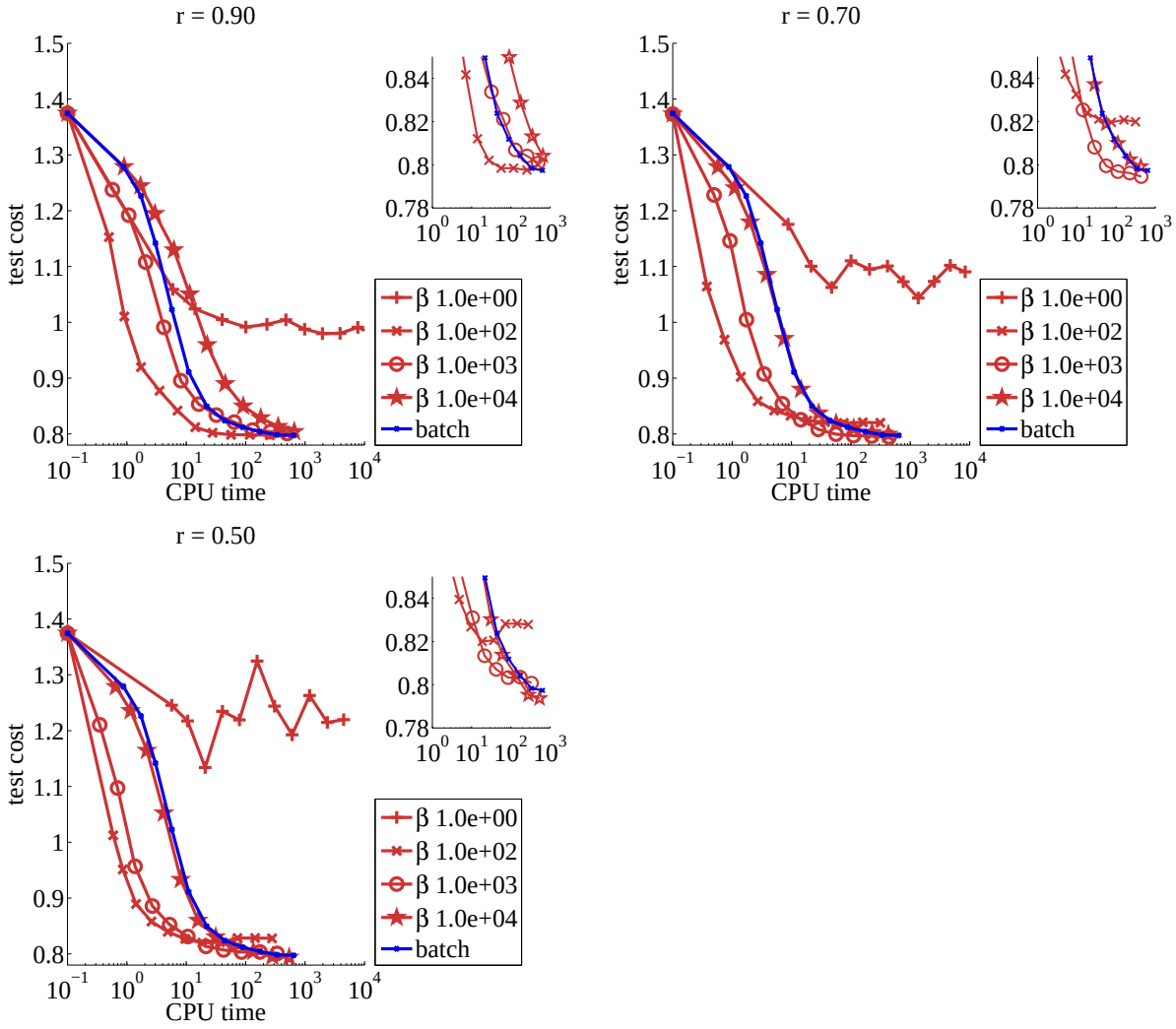


Figure 3.2: Comparison of online and batch algorithm on a three-minutes long audio track.

Leibler divergence [Duan et al., 2012], online multiplicative updates tailored for NMF with the ℓ_2 loss [Bucak and Gunsel, 2009], extensions to convolutive non-negative sparse coding [Wang et al., 2011]. Bayesian extensions have also been proposed for online dictionary learning, either in the framework of variational inference [Hoffmann et al., 2010a] or using sampling strategies [Cappé et al., 2011].

There is also an alternative approach which was presented in the context of text mining[Cao et al., 2007]. It relies on the following principle : once a dictionary $W^{(1)}$ is learnt on a first batch of data $V^{(1)}$, then this data may as well be replaced by $W^{(1)}$. To be more precise, once the next batch of data $V^{(2)}$ is available, $W^{(2)}$ is optimized to reconstruct the modified input matrix $(W^{(1)}V^{(2)})$, instead of $(V^{(1)}V^{(2)})$. Like ours, this procedure requires constant memory instead of linear. In the noiseless case, the batch and online algorithms yield the same

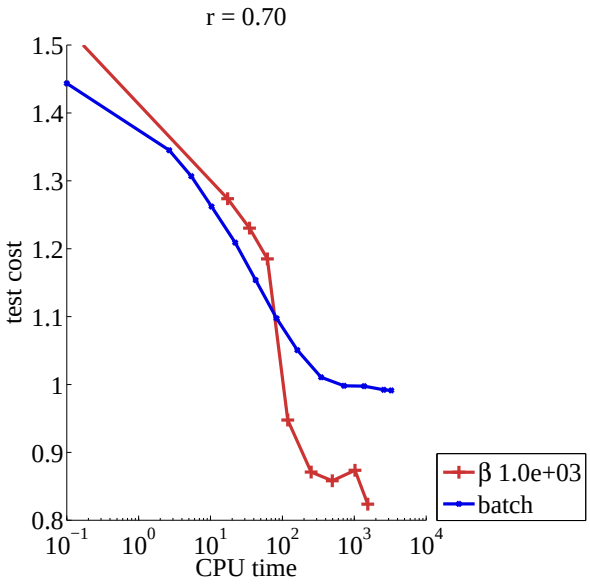


Figure 3.3: Comparison of online and batch algorithm on an album of Django Reinhardt (1 hour 20 minutes).

solution. The behavior of this algorithm in the noisy setting is yet to be studied.

Finally, a divide-and-conquer matrix factorization method was recently proposed in [Mackey et al., 2011], where base matrix factorization algorithms are used to factorize subsets of the data, and solutions of each subproblem are combined using techniques from randomized matrix approximation.

3.4 Conclusion

In this Chapter, we have provided an algorithm for online IS-NMF with a complexity of $O(FK)$ in time and memory for updates in the dictionary. We have also proposed several extensions to stabilisze online NMF and summarize them in a concise algorithm[1]. We show that online NMF competes with its batch counterpart on small data sets, while on large data sets it outperforms it by several orders of magnitude. In a pure online setting, data samples are processed only once, with constant time and memory cost. Thus, online NMF algorithms may be run on data sets of potentially infinite size which opens up many possibilities for audio source separation.

[1]Code is available online at `http://www.di.ens.fr/~lefevrea/xnmf.zip`

Chapter 4

Informed source separation : how user annotations disambiguate the source separation problem

Our third contribution, presented in this Chapter, is an extension of NMF to incorporate additional constraints on the estimates of the source spectrograms, in the form of time-frequency annotations. While time annotations have been proposed before, almost perfect source estimates may be obtained with as little as 20% of annotations, provided those are correct, whereas this is not guaranteed with time annotations, even when the whole recording is annotated. Our formulation is robust to small errors in the annotations. We provide a graphical interface for user annotations, and investigate algorithms to automatize the annotation process. This work has led to the following publication(s) :

> A. Lefèvre and F. Bach and C. Févotte, "Semi-supervised NMF with time-frequency annotations for single-channel source separation", *in Proceedings of the International Conference on Music Information Retrival (ISMIR)*, 2012.

Audio demonstrations are available online[a], as well as a GUI for annotating spectrograms[b]

[a] http://www.di.ens.fr/~lefevrea/annot.html
[b] http://www.di.ens.fr/~lefevrea/annot_gui.zip

In this chapter we present a recent contribution to informed source separation [Lefèvre et al., 2012]. In Chapter 1, we have discussed how sparsity penalties and/or prior information could be added to nonnegative matrix factorization $V = WH$. In parallel, recent contributions propose to incorporate user information that are tailored to the specific mixed signal at hand. Indeed, such information as time activation of the various sources is easy to produce even for non-trained listeners, whereas it is very hard to estimate it computationally. [Ozerov et al., 2011] have shown that time annotations, together with multichannel modelling of

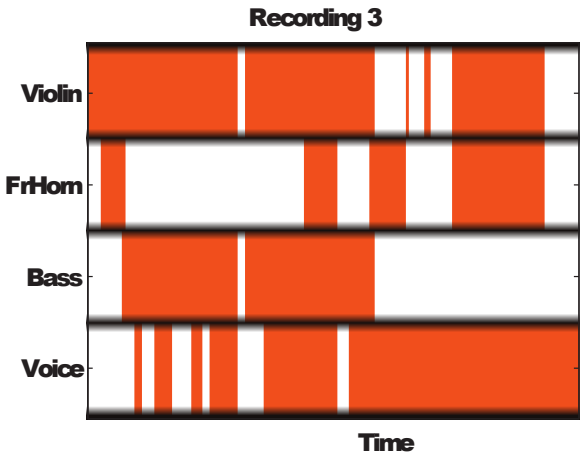


Figure 4.1: Example of time activations on a track from the SiSEC database. (*Figure reproduced from [Ozerov et al., 2011]*)

the mixing process, produce excellent separation results. Pitch information has also helped improve separation quality sharply, as shown in [Durrieu and Thiran, 2012]. Finally, score-guided source separation has been the subject of several contributions [Raphael and Han, 2008, Hennequin et al., 2011, Ganseman et al., 2012].

A common trait of these methods is that are all based on a simple extension of NMF : either annotations are used to identify zeroes in the matrix of activation coefficients in NMF; in this case, using some form of multiplicative update algorithm, it is straightforward to constrain solutions of NMF to have zeroes at required time frames. When activation coefficients are fixed, annotations help learn a source specific dictionary $W^{(g)}$ on segments of the recording where only source g is active. [Ganseman et al., 2012] propose instead to synthetize tracks using score information and sophisticated synthetizers, learn prior distributions for the dictionary and activation coefficients, and use those priors in the main source separation step.

In this chapter, we advocate direct annotations of time-frequency regions in the spectrogram. It is an empirical fact that a large fraction of time-frequency bins may be assigned unambiguously to one dominant source : that is why optimal binary masks are often considered as an upper-bound of source separation performance. As illustrated in Figure 4.2, some patches in the spectrogram are cues for source-specific activity, which may be exploited as information on the optimal binary mask : techniques from computational audio scene analysis (CASA) make extensive use of such cues to build interpretations of "acoustic scenes". If the spectrogram was entirely annotated, then we could obtain almost perfect source separation results. Off course, such annotations are hard to find, and in practice we can only hope for a fraction of the time-frequency plane. The main contribution of this chapter is to show how NMF can exploit a partially annotated spectrogram to learn an accurate model of the sources. With no annotations, we

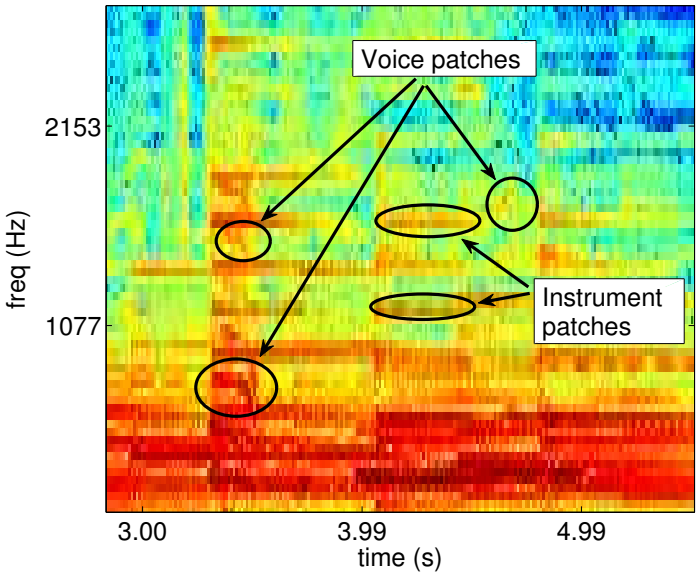


Figure 4.2: Some patches in the spectrogram are easy to read for the user

recover standard NMF, and with all annotations, the spectrogram is already separated. Therefore, there must be a small enough fraction of annotations such that NMF can complete the rest of the spectrogram.

The rest of this chapter is organized as follows : in Section 4.1, we introduce a graphical user interface to retrieve such time-frequency annotations, and examine how hard it is to produce such annotations. In Section, 4.2, we propose a modification of NMF to take into account time-frequency annotations of the spectrogram, that is robust to errors in the annotations. Since our modified NMF algorithm does not depend on the way annotations were obtained, we study in Section 4.3 how the process of annotating may be automatized with supervised learning algorithms. Finally, we illustrate our contributions on publicly available source separation databases, and incidentally provide early results on the recently released QUASI database.

Section 4.5 is devoted to specific aspects of this contribution and may be skipped at first reading.

4.1 A graphical user interface for time-frequency annotation of spectrograms

In this section, we investigate manual annotation of the spectrogram. A GUI was designed in Matlab to annotate spectrograms (see Figure 4.3), with some extra sound functionalities to help the user. It takes sound files (in the .wav format) as input, applies some basic preprocessing (re-sampling at user-specified rate, down-mixing to mono), computes a time-frequency representation via user-

specified parameters, and displays the spectrogram. Zooming and slide-rule navigation are enabled for better visualization. Annotation of sources is done with a simple rectangle drawing utility : one color for each source. Annotations are stored in an annotation mask, short for a 3-dimensional array of size $F \times N \times G$ (where (F, N) is the size of the spectrogram and G the number of sources). They are displayed in transparency as in Figure 4.3. Several annotation masks may be loaded into memory and displayed alternatively, via a list-box, so the user can compare, for instance, manual annotations with the output of a blind source separation algorithm. Annotation masks may be exported to .mat format for further processing. Finally, we implemented playback functionalities to help the user read the spectrogram : play, pause, and navigate by clicking. Before playback, the track can be filtered according to any of the annotation masks loaded by the user.

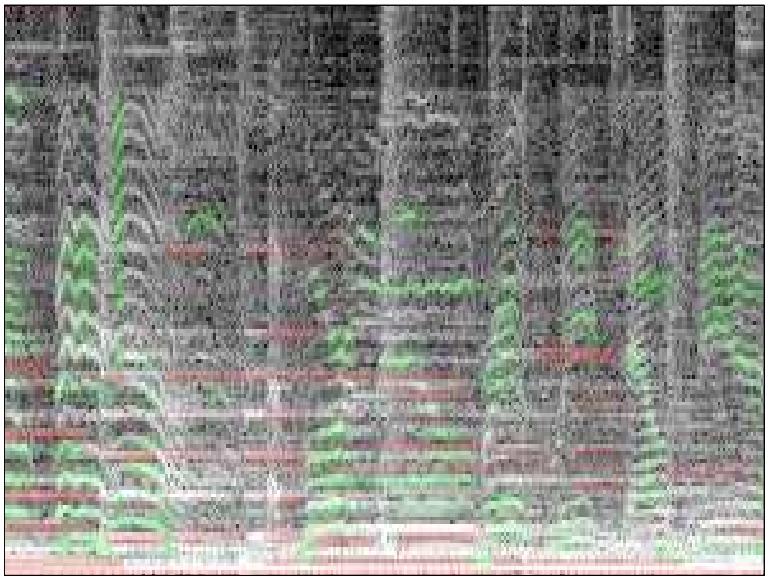


Figure 4.3: Example of user annotations in a ten seconds' audio track: green regions are assigned to voice, and red regions to accompaniment.

We designed the GUI to make the annotation process easier and faster : indeed, in our experience, while time annotations are easy and require only listening once or twice to the mix, time-frequency annotation is a hard task, and it may take up to one hour to annotate a twenty seconds track. While 100% of the track can be covered in the case of time annotations or pitch tracks, time-frequency annotations are partial, and prone to errors (see the experimental section for more details). Moreover, time-frequency annotations are difficult to correct since the contribution of one time-frequency bin to the signal is difficult to discriminate by listening. In the next section, we explain how NMF can exploit such incomplete and partly incorrect annotations to provide a complete source separation mask.

4.2 Annotated NMF

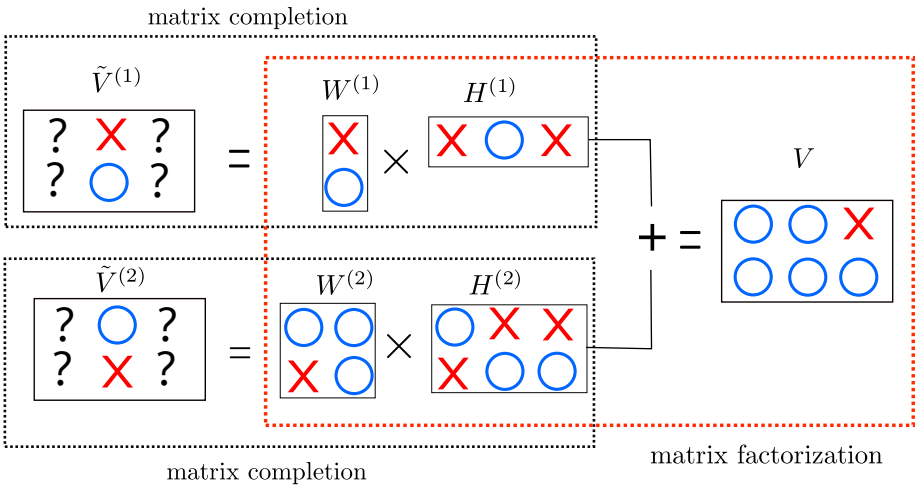


Figure 4.4: Semi-supervised NMF may be understood as solving G matrix completion problems, coupled by a matrix factorization problem.

In this Section we show how NMF may be extended to take into account time-frequency annotations. These come in the form of incomplete masks $M_{fn}^{(g)}$, whose values are defined on a subset $\mathcal{L}$ of the time-frequency plane. An example of user annotations is displayed in Figure 4.3, where time-frequency masks take values in $\{0, 1\}$. Each color corresponds to one source, and we assume that only one source is active at each time-frequency bin. Ideally, coefficients $M_{fn}^{(g)}$ should be equal to the Wiener coefficients in $[0, 1]$, so the general admissible form of time-frequency masks will be :

$$M_{fn}^{(g)} \in [0,1] \qquad \sum_g M_{fn}^{(g)} = 1 \quad \forall (f,n) \in \mathcal{L}\,. \tag{4.1}$$

In a first step, we could consider recovering source spectrograms by matrix completion : for each source g, define target values $\tilde{V}_{fn}^{(g)} = M_{fn}^{(g)} V_{fn}$. In the noiseless case, this would amount to solving G inpainting problems :

$$\begin{array}{rc} \text{Find} & W^{(g)} \geq 0, H^{(g)} \geq 0\,. \\ \text{subject to} & (W^{(g)} H^{(g)})_{fn} = \tilde{V}_{fn}\,. \end{array} \tag{4.2}$$

We should also introduce the additional constraint of reconstructing the mixed spectrogram :

$$\begin{array}{rc} \text{Solve} & V = WH\,. \\ \text{subject to} & (W^{(g)} H^{(g)})_{fn} = \tilde{V}_{fn}^{(g)}\,, \\ & W \geq 0, H \geq 0\,. \end{array} \tag{4.3}$$

where

$$\tilde{V}_{fn}^{(g)} = M_{fn}^{(g)} V_{fn} \tag{4.4}$$

The noiseless formulation gives an idea of how manually labelled spectrograms are handled by our algorithm. As illustrated in Figure 4.4, it may be interpreted as G inpainting problems, one for each source, coupled together by an NMF problem. We now extend this formulation to the noisy case.

Let $\mathcal{L}$ be the set of annotated bins, and $M_{fn}^{(g)}$a set of time-frequency masks such that : $M_{fn}^{(g)} \in [0,1]$, and $\sum_g M_{fn}^{(g)} = 1$ if $(f,n) \in \mathcal{L}$, $\sum_g M_{fn}^{(g)} = 0$ otherwise. For annotated time-frequency bins, we use $M_{fn}^{(g)} V_{fn}$ as a target for $V_{fn}^{(g)}$. The remaining entries of (W,H) are then computed so as to fit the observed spectrogram. This idea translates into the following optimization problem :

$$\begin{aligned} \min \quad & \textstyle\sum_{(f,n)} d_{IS}(V_{fn}, \hat{V}_{fn}) \dots \\ & + \lambda \textstyle\sum_{(f,n)\in\mathcal{L}} \sum_{g\in\mathcal{G}} d_{IS}(\tilde{V}_{fn}^{(g)}, \hat{V}_{fn}^{(g)}), \\ W \geq 0, H \geq 0 & \end{aligned} \tag{4.5}$$

where

$$\tilde{V}_{fn}^{(g)} = M_{fn}^{(g)} V_{fn} \qquad \hat{V}_{fn}^{(g)} = \sum_k W_{fk}^{(g)} H_{kn}^{(g)} . \tag{4.6}$$

The second and third sums in Eq. 4.5 act as soft versions of the constraint that $\hat{V}_{fn}^{(g)}$ be equal to its target value $M_{fn}^{(g)} V_{fn}$. We may tune the relative importance of annotation by varying parameter λ, from $\lambda = 0$ (standard NMF), to $\lambda \to +\infty$ (in which case $(WH)_{fn} = V_{fn}^{g}$ is enforced exactly if there are any feasible solutions). Thus, robustness to uncertainty in the annotations is introduced by replacing hard constraints by penalty terms in the NMF optimization problem. Note that since annotations dictate the assignment of components to sources, there is no need to group components by hand.

Due to the constraint, that there is only one dominant source per time-frequency bin, our model cannot handle time annotations such as those found in [Ozerov et al., 2011], unless there are only two sources. However, when there are more than two sources, a simple extension of 4.5 allows dealing with the case where more than one source is allowed to be active (see Section 4.5.2).

Remark 2. *Given that some values are set to zero, we replace the IS-divergence $d_{IS}(x,y)$ by $d_IS(\epsilon + x, \epsilon + y)$ (where $\epsilon = 10^{-10}$) in our optimization problem, in order to deal with numerical instabilities.*

4.3 Towards a supervised algorithm for annotation

Research in computational audio scene analysis (CASA) has emphasized the role of frequency tracks in source identification : indeed by looking at a spectrogram, it is easy to assign a significant number of frequency tracks either to a voiced

source or a musical source (see Figure 4.2). In previous works, such cues have been used to compute a similarity matrix that would then be used to perform clustering, see [Lagrange et al., 2008, Bach and Jordan, 2004]. We propose here a supervised learning procedure to predict annotations automatically.

This is a radical change of perspective since in this case, we will use training data to supervise these "detectors". We argued in previous chapters that some benchmark datasets in source separation such as the "Professionally Produced Music Recordings" task in SISEC where too small to train NMF. Across tracks variations are so important that training NMF on some tracks would not allow to generalize well to the other tracks. Indeed, state-of-the art algorithms are obtained by blind source separation systems using either additional modelling of the sources [Durrieu and Thiran, 2012], or exploiting stereo tracks to learn spatial models [Ozerov et al., 2012].

The detectors we present in this Section, on the other hand, are built on typical cues from CASA : frequency co-modulation and harmonic stacking will be captured by oriented filters proposed in Section 4.3.4. As we will see in Section 4.3.2, the detection task we propose to solve is very noisy : very similar time-frequency patches may have very different labels. However, we count on the fact that some cues such as frequency co-modulation, clearly identify time-frequency bins where voice is dominant. Also we expect such cues to be used in a supervised setting, i.e., they should generalize well from train to test set, even with the wide inter-track variations seen in databases such as SiSEC.

4.3.1 Learning algorithms

For convenience we use index $i \in \{1, \dots, I\}$ to denote time frequency bins (f, n), so that $I = FN$. For each time-frequency bin i, a set of features is x_i extracted to describe local information in the neighborhood. The training data thus consists of pairs (x_i, y_i) where $y_i = (M_i^{(1)}, \dots, M_i^{(G)})$ are the optimal masking coefficients computed using ground truth source signals.

We thus face a classical machine learning task which consists in fitting a smooth decision surface $f : \mathcal{X} \to [0, 1]^G$ to the observed data points (x_i, y_i). Originally we were considering classification, i.e., predicting binary values of masking coefficients. However, since masking coefficients are allowed in $[0, 1]$, we finally opted for regression algorithms.

In order to alleviate the computational burden, we make two restrictions on the learning procedure : vectors y_i are predicted independently, and based only on local information x_i (contrary to [Lagrange et al., 2008, Bach and Jordan, 2004] who use global cues such as multi-pitch detectors). Since there is scarce literature related to our approach, we experimented with several features and several algorithms in order to get an idea of how hard the problem is.

The interference patterns observed on the training set are interpolated with standard regression algorithms [Hastie et al., 2009].

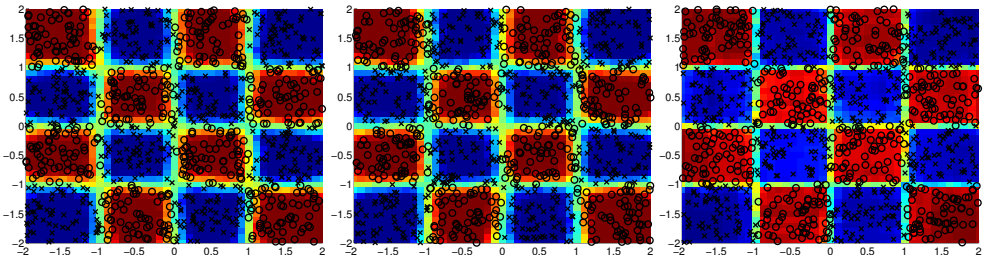
K-Nearest neighbors : for each test point $x_i^{(test)}$, the Q nearest points $x_j^{(train)}$, $j \in \{1, \ldots, Q\}$ from the train set are used to predict $M_i^{(g)} = 1/Q \sum_j M_j^{(g)}$.

K-means : We learn Q clusters from the train set; for each cluster, we compute average prediction coefficients $M_q^{(g)}$. For each test point, we predict $M_q^{(g)}$ from the nearest cluster q.

Random Forests: We learn Q regression trees of depth d from the train set and average over the Q predictions for each test point.

Figure 4.5 displays the accuracy of our detectors trained on a toy dataset. As we can see, random forests interpolate a sharper decision boundary thanks to bagging. Moreover, recursive partitioning of the input space allows keeping a low test time complexity as the training data size grows.

SVMs were considered in the first place, but abandoned because of the dimensionality of the problem : in our experiments the train set consisted of 10^4 points in dimensions up to 200, while the test sets were of size 10^5 with the same dimensions.



(a) KNN (b) Quantized KNN (c) Random Forest

Figure 4.5: Comparison of textbook detectors on the checkers dataset. Test points ('x' and 'o' markers) are plotted on top of the decision surface. For all three experiments, $Q = 100$. $d = 10$ for random forest.

4.3.2 Features

Several features and transformations of these were considered according to the data extraction chart in Figure 4.6.

4.3.3 Raw Patches

The features we use for regression are simple time-frequency blocks extracted from the SiSEC 2010 database. (train set : 5 tracks 10 to 30 seconds' long; test set : 5 tracks of several minutes). Blocks of a given size are extracted those with lowest energy are discarded. They are then normalized to have unit norm so that

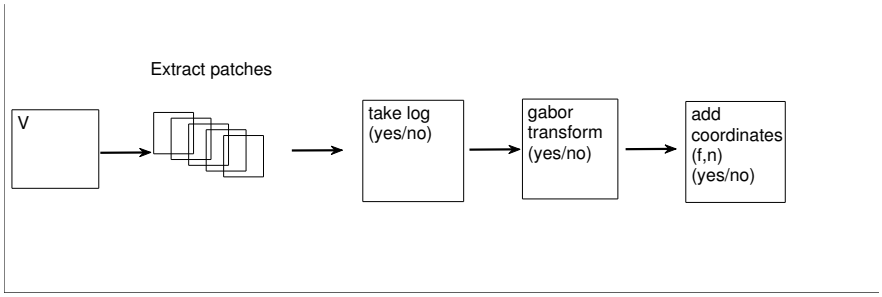


Figure 4.6: Data extraction flow chart

patches extracted at high frequencies are comparable to patches extracted at low frequencies. We also considered taking the log of patches, and adding coordinates of the patch as additional information.

Examples of raw patches extracted from the SiSEC 2010 database are shown in Figure 4.8. Patches are colored according to the value of their Wiener coefficients : red if $M_{fn}^{(1)} = 1$ (accompaniment), green if $M_{fn}^{(1)} = 0$ (voice). As we can see, regressing Wiener coefficients onto local time-frequency patches is a hard task : very similar patches may have widely different labels.

4.3.4 Oriented filters

Gabor wavelets are widely used in texture discrimination because they are sensitive to specific orientations in the signal, as illustrated in Figure 4.7 :

$$g_{\omega,\theta}(x,y) = \exp\left(-\frac{x^2+y^2}{2\sigma^2} + i\omega(x\cos\theta + y\sin\theta)\right). \tag{4.7}$$

the Fourier transform of g is highly concentrated around $(\omega\cos\theta, \omega\sin\theta)$, so θ controls the spatial orientation of the wavelet, and ω its scale.

For every patch $f \in \mathbb{R}^{p\times p}$ computed in the previous subsection, we compute a new feature vector : $\left(|\langle f, g_{\omega,\theta}\rangle|^2\right)_{\omega\in\Omega,\theta\in\Theta}$, where $\Theta = \frac{2\pi}{K}\{0,\dots,K-1\}$, $\Omega = \frac{2\pi}{p}\{0,\dots,p-1\}$.

The number of orientations K is chosen arbitrarily (it will be cross-validated eventually). The bandwidth is not a critical parameter so we set it to $\sigma = p$ for all experiments. The new feature vector allows classifying patches according to their orientation and scale, which makes more sense than classifying pixel values. Another asset is that the size of the feature vector grows as p instead of p^2 for raw patches.

They are then normalized to have unit ℓ_1 norm so the features are scale invariant. We also considered taking the log of patches, adding coordinates of the patch as additional information, and taking a Gabor transform of the patches. The Gabor transform in particular was introduced so that correlations between pixels in each time-frequency blocks is taken into account. Such oriented filters were also used [Yu and Slotine, 2009, Bach and Jordan, 2004].

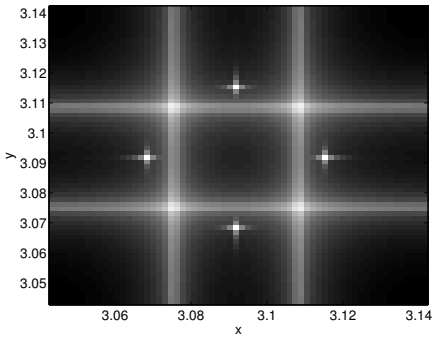


Figure 4.7: Repartition of the energy of gabor wavelets at a given scale and orientations $k\pi/4$ for $k = 0, \ldots, 7$, in the 2-D Fourier plane.

As noted in [Bach and Jordan, 2004] they may be related to harmonic stacking and frequency co-modulation in CASA terminology.

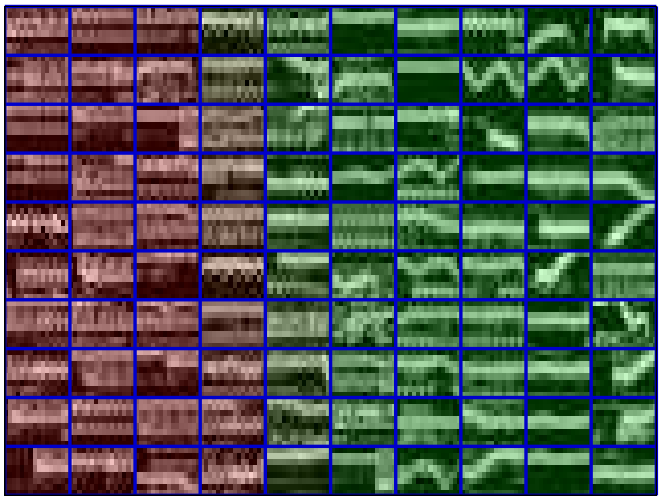


Figure 4.8: Samples of patches extracted from the SISEC database. Intensity reflects amplitude, patches which are labeled as accompaniment are in red, while patches which are labeled as voice are in green. Patches in brown have mixed Wiener coefficients.

4.3.5 Averaging training labels

A desirable feature of our detector is that it should be robust to the chosen discretization of the time-frequency plane. To achieve this property we introduce a small amount of averaging of y_i in the neighborhood of the time-frequency bin i. As we will see in 4.5.1, averaging reduces the prediction error or our detector at the cost of biasing predictions towards 0.5.

4.4 Experimental results

4.4.1 Description of music databases

We used two publicly available databases in our experiments: the QUASI database[1] and the SISEC database for Professionally produced music recordings[2] (PPMR). All source tracks were down-sampled from 44100 Hz to 16000 Hz, and down-mixed to mono by taking the average of left and right channels. A voice track and accompaniment track are then created by aggregating the various source files, and then a final mix is created by summing the two tracks. Sine-bell windows of size 1024 with 512 overlap were used to compute short time Fourier transforms. The QUASI database contains longer tracks that are amenable to time annotations. The SISEC database contains short tracks where only time-frequency annotations can be used. Although detailed instrumental tracks are provided for most of the mixtures, we work only on single-channel signals. Since we are dealing with under-determined mixtures, we restrict ourselves to separating voice from accompaniment in each track, in order to alleviate the difficulty of the problem.

4.4.2 Ideal performance and robustness

	SDR1	SDR2	SIR1	SIR2	SAR1	SAR2
0.1 %	-0.02	-0.60	5.15	5.16	3.62	2.33
1 %	0.70	0.24	4.59	6.25	4.39	2.85
10 %	6.71	6.68	13.57	16.53	7.95	7.40
100 %	10.40	10.41	19.88	20.88	11.00	10.88

Table 4.1: Mean results on the SISEC database, as the proportion of annotation increases.

Table 4.1 displays source separation results achieved by semi-supervised NMF on the SISEC database when fed with the actual Wiener coefficients computed from the ground truth sources. Source separation performance is measured by Source to Distortion Ratio (SDR), Source to Interference Ratio (SIR), and Source to Artefact Ratio (SAR). Higher values indicate better performance. As we can see, satisfactory results are obtained with as little as 10% of annotations. When 100% of annotations are given, NMF does nothing and the computed masks are simply the ideal Wiener coefficients computed from the sources.

We study the robustness of our NMF routine by replacing part of the ideal annotations by noise to simulate human errors. Table 4.2 displays average SDRs obtained when fixing the annotation rate to 10% and varying either the rate of wrong annotations p or the optimization parameter λ. As expected, for fixed

[1] `www.tsi.telecom-paristech.fr/aao/`

[2] `sisec.wiki.irisa.fr`

λ the average SDR drops as p increases. When p is fixed, there is an optimal value of λ that trades off the benefits and drawbacks of annotations. Fixing the target annotation rate to 10%, satisfactory results are obtained with up to 10% of wrong annotations (i.e., 1% of the spectrogram). Note that wrong annotations were simulated by choosing at random p% of the annotations and replacing them with uniform draws in $[0, 1]$.

λ	$p = 0$	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.5$
10^{-1}	0.11	-0.08	-1.76	-1.47	-1.47
10^{0}	5.59	4.10	3.50	2.29	1.20
10^{1}	7.59	6.53	5.32	3.43	0.59
10^{2}	7.07	5.66	4.54	3.15	0.77

Table 4.2: Mean SDR value as λ and the proportion of wrong annotations vary. The proportion of annotations is set to 0.1

4.4.3 Evaluation of automatic annotations

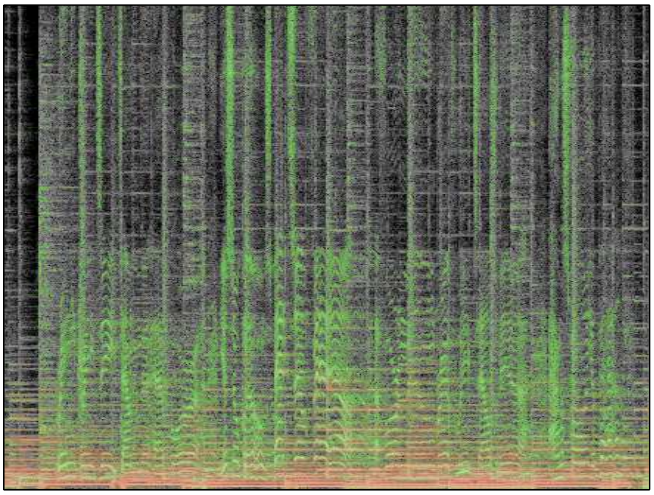
method	**mean error (% improvement)**
4 8 loggabor km avg	0.141 ±0.018 (0.1493)
4 16 wcoords knn avg	0.140 ±0.015 (0.1589)
4 8 wcoords knn avg	0.138 ±0.015 (0.1682)
4 32 loggabor rf avg	0.137 ±0.013 (0.1736)
4 32 loggabor knn avg	0.137±0.010 (0.1739)

Table 4.3: Mean error on Wiener coefficient predictions on the SISEC database (% improvement over random prediction), for various learning strategies .

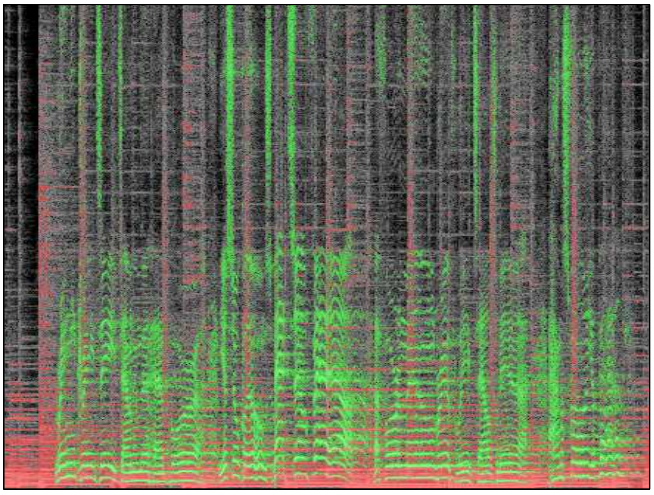
Learning algorithms were trained by dividing the SISEC database in two sets of tracks. For each set, we train detectors and test them on the other set. Thus we may compute annotations and run semi-supervised NMF for all tracks without the risk of overfitting. We emphasize the fact that each track is annotated with a detector that has never seen the spectrogram before : our method is purely supervised with no adaptation to test data. Parameters of the learning algorithms were selected at train stage by cross-validation. Time-frequency patches of size in $\{4, 8\} \times \{8, 16, 32\}$ were extracted. Out of each track we extract 5×10^3 patches at train time, and 10^5 patches a test time, so approximately 10% of the track is annotated at test time when semi-supervised NMF is called.

We display in Table 4.3 the results of the best 5 detectors, in terms of mean prediction error (first column) and in terms of relative improvement over a purely random predictor (see Section 4.5.1 for a precise definition).

Detectors are named after the following rule : {patch size} {feature} {learning method} {averaging or identical}. For instance, the tag **loggabor** corresponds



(a) Automatic



(b) Correct

Figure 4.9: Comparison of automatic annotations and correct annotations (at the same time-frequency bins). Gray-scale time-frequency bins are not annotated, red bins are annotated as accompaniment, green bins as voice.

to taking log then Gabor transform of patches, and **wcoords** adding frequency coordinates of the patches as side information. Note that we used exact Wiener coefficients to compute errors, so that all detectors can be compared even when averaging was used at train stage. The improvement over a random predictor is consistent across the features and the algorithms that were used. Figure 4.9 compares annotations provided by the best detectors from Table 4.3 with ideal annotations at the same points were automatic annotations were made. Red time-frequency bins correspond to accompaniment, and green to voice.

The most striking observation is that, while ideal annotations are in very bright colors (few Wiener coefficients are different from 0 or 1), automatic annotations, on the other hand, are generally biased towards 0.5. This is to be expected since predicting 0.5 incurs a risk of losing at most 0.25 (since we use a regression loss), while predicting 0 or 1 incurs a maximum loss of 1. The main asset of automatic annotations is that pitch tracks with varying frequency are successfully predicted as voice. Automatic annotations are biased towards predicting voice in the higher frequencies : however the learning algorithm in this example did not have the information of frequency. This might be because transients "look" a lot like patches of unvoiced speech. Finally, one may spot inconsistencies in the predictions in the sense that points belonging to the same pitch tracks are sometimes classified incoherently, which is not surprising since the learning algorithms we have proposed predict time-frequency bins independently.

To sum up, predictions of Wiener coefficients from local patches are not perfect but provide a good starting point for further modelling of the spectrogram. We expect that better performance could be obtained by using more advanced cues from CASA, such as pre-clustering the spectrogram into pitch tracks and transient tracks, before learning[3].

	% annotated	% correct
track 1	0.23	0.91
track 2	0.10	0.89
track 3	0.29	0.91
track 4	0.17	0.81
track 5	0.22	0.95

Table 4.4: Evaluation of user annotations on the SISEC database.

4.4.4 Overall results

We now turn to results obtained by semi-supervised NMF combined with various annotation methods. On the SISEC database, manual time-frequency annotations were done with the GUI presented in Section 4.1. On the QUASI database,

[3]This is very similar to what is done in vision, where super-pixels help deal with consistency in prediction and alleviate the computational burden of predicting all pixel values.

	auto	user (t-f)	baseline	self	oracle
SDR1	0.97	6.21	6.16	3.09	14.79
SDR2	0.51	2.58	1.61	-3.18	11.53
SIR1	3.17	18.64	9.91	3.09	24.00
SIR2	4.57	11.35	5.09	-3.18	23.90
SAR1	6.74	6.91	9.26	279.17	15.41
SAR2	4.18	3.91	5.58	279.17	11.84
% ann.	8.69	19.81	0.00	0.00	100.00

(a) SISEC

	auto	user (t)	baseline	self	oracle
SDR1	6.76	7.59	6.29	6.21	16.88
SDR2	-4.33	-4.57	-1.71	-6.22	10.37
SIR1	6.97	15.05	13.81	6.21	25.62
SIR2	-3.75	4.09	1.88	-6.22	24.83
SAR1	21.91	9.00	7.71	268.45	17.66
SAR2	10.28	0.21	4.29	268.45	10.60
% ann.	6.91	100.00	0.00	0.00	100.00

(b) QUASI

Table 4.5: Results on the evaluated databases: (a) time-frequency annotations, (b) time annotations.

tracks were amenable to significant time annotations, so by comparing results on both databases we can compare the respective benefits of time-frequency annotations VS time annotations.

In both scenarios, we compare five methods :

auto : Automatic annotations and semi-supervised NMF. The best detector from Table 4.3 was chosen.

user : User annotations and semi-supervised NMF (time-frequency annotations for SISEC, manual annotations for QUASI). We tried $K \in \{5, 10, 20\}$ for the SISEC database and $\{10, 20, 50\}$ for the QUASI database, as well as $\lambda \in \{1, 10, 100\}$, and selected parameters yielding highest SDR for fair comparison with the baseline.

baseline : Run NMF and permute factors to obtain optimal SDR. We set $K = 8$ because it already takes a 10 times as long to evaluate SDRs for all permutation on a single track as it takes to run semi-supervised NMF.

self : set $s^{(g)} = \frac{1}{G}x$ as estimates for the sources, it serves to estimate the difficulty of the source separation problem for a given database.

oracle : results obtained with Wiener coefficients computed from the ground truth. In addition we display track by track annotation accuracy for user annotations, for comparison with Table 4.2. For each method, we ran NMF three times for 1000 iterations to avoid local minima, and kept the run with the lowest objective cost value.

Tables 4.5a and 4.5b display average evaluation metrics for each source (source 1 is always the accompaniment, and source 2 is always the voice), on two different databases : on the SISEC database, we experimented with time-frequency annotations since the tracks were too short for time annotations. Overall, results on the SISEC database are better than those on QUASI. Our interpretation is that since most of the time the accompaniment is active, the dictionaries tend to overfit the accompaniment and underfit the voice. Time-frequency annotations on SISEC yield SDRs that are a few points below that predicted by our benchmark from Table 4.2 : indeed human errors are not distributed randomly as was the case in our benchmark. Time-frequency annotations outperform the baseline by 1 point in SDR, which is important because in semi-supervised NMF there is no manual grouping of the components, whereas the baseline required knowing the ground truth to find the optimal permutation of components. Time annotations loose to the baseline by -1 in SDR, but they are still significantly correlated with the true sources when compared with the baseline.

On the SISEC database, automatic annotations are also below the baseline, however they are also significantly correlated with the true sources, when compared with the "self" column. Signal to Interference Ratios are even comparable with those of the baseline on the SISEC database. Automatic annotations do not perform as well on the QUASI database since we trained detectors only on tracks from SISEC, so that more supervision would significantly improve those figures.

For the sake of completeness, we display in Appendix B track by track detailed results. Audio samples are available online[4].

4.5 Miscellaneous

4.5.1 Detailed comparison of detectors

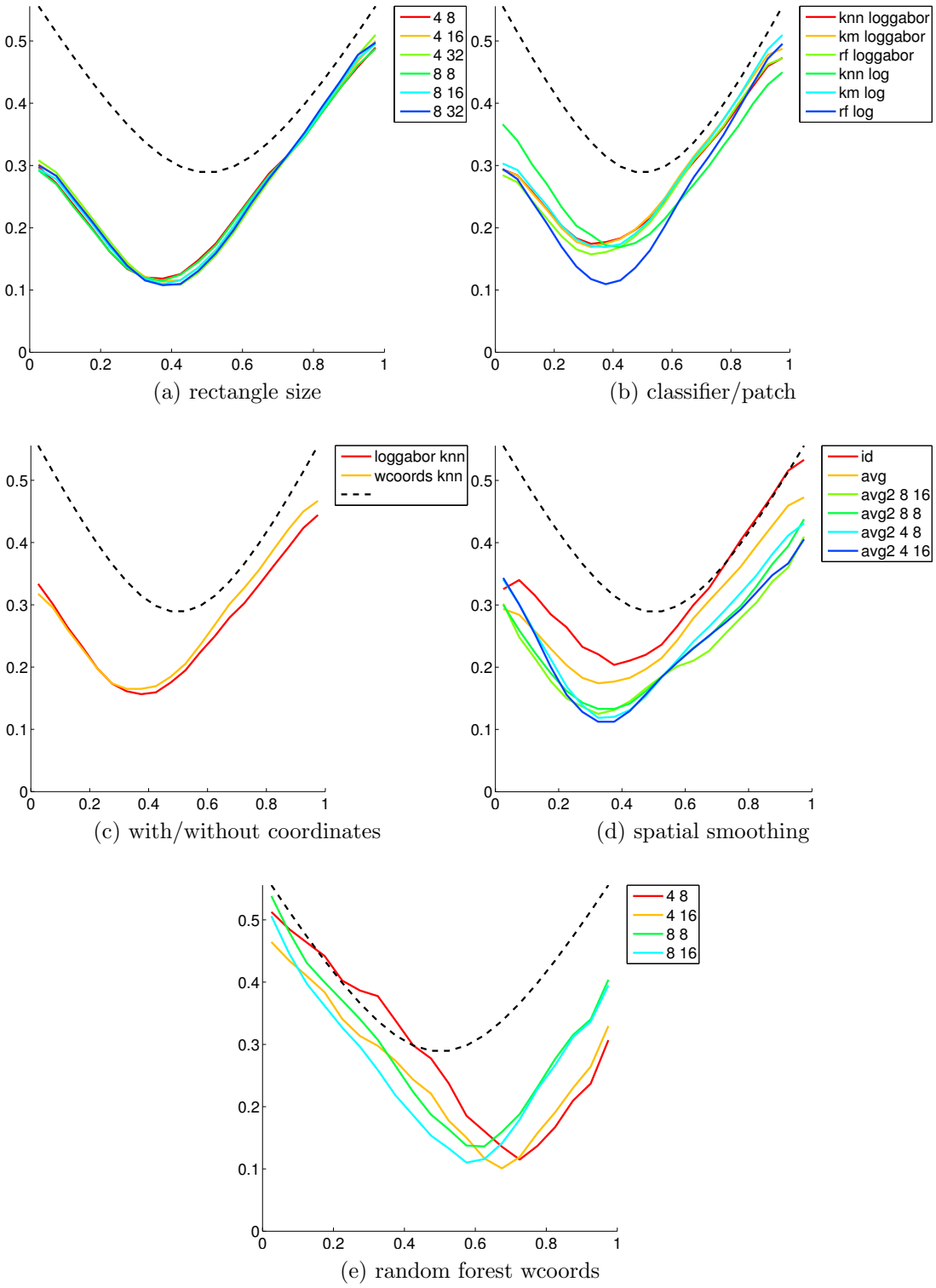
We plot in Figure 4.10 the expected distance $\sqrt{\mathbb{E}((\hat{y}-y)^2|y)}$ of predicted values $\hat{y}_n$ conditional to observed values y, as a function of y. The dashed curves correspond to a uniformly random predictor, for which the mean error can be computed explicitly:

$$\sqrt{\mathbb{E}((y-\hat{y})^2|y)} = \sqrt{\frac{1}{12} + (y-\frac{1}{2})^2}\,. \tag{4.8}$$

Like the other prediction curves, it has a bell shape and is minimum for $y = \frac{1}{2}$.

The default classifier is a random forest fed with raw patches with a log transform taken on rectangular 8×32 time-frequency windows. As we can see, while our classifiers are able to retrieve patches that correspond to isolated sources accurately, they make more mistakes when mixing coefficients are equal. The distribution of the Wiener coefficients provides one explanation for that : Wiener

[4] `www.di.ens.fr/~lefevrea/annot.html`



(a) rectangle size

(b) classifier/patch

(c) with/without coordinates

(d) spatial smoothing

(e) random forest wcoords

Figure 4.10: Mean distance of the predicted Wiener coefficient $\hat{y}$ to the observed value y, as a function of the true value, for various combinations of features/algorithms. Dashed lines are the theoretical error achieved by the uniform predictor (pure chance prediction).

coefficients tend to concentrate around 0 and 1, so mixed patches count less in the minimization criterion. The parameters we put in the grid are the size of windows (width and height), the type of predictor (K-nearest neighbor, k-means based and random forest), the type of features, and finally the amount of spatial averaging of the Wiener coefficients.

We found that two parameters were crucial in reducing the prediction error : choosing random forests and averaging masking coefficients y_i over small time-frequency neighborhoods of the same size as that used for extracting features x_i. While this biases the predictions y_i towards 0.5 as we can see on Figure 4.9, it also allows predicting more accurately time-frequency bins where sources have equal amplitudes $\hat{V}_{fn}^{(g)}$: in this sense, smoothing the training data points y_i decreases the false prediction error. However, as the size of the neighborhood increases, false prediction error stays the same but true prediction errors $\mathbb{E}(\|y-\hat{y}\|_2|y=\pm 1)$ increase.

Finally, using time-frequency coordinates (f,n) as features did not improve prediction error significantly.

4.5.2 Extension of annotated NMF with more than one dominant source

In this section, we show that our framework may be extended to handle time-only annotations when there are more than 2 sources. In this case, we cannot make the assumption that only one source is dominant per time frame anymore. We adapt our formulation accordingly in Section 4.5.2.1, and provide experimental results on a three instrument track from the QUASI database.

4.5.2.1 Mathematical formulation

$\mathcal{L}$ is the set of annotated time indices n. While in the time-frequency annotations case we allow only one active source at each annotated time-frequency bin, in the case of time annotations, there may be several. For each time frame n, the set pres(n) denotes all sources that are annotated as active and abs(n) all sources that are annotated as inactive, so that $pres(n) \bigcup abs(n) = \{1,\dots,G\}$. Problem 4.5 is then naturally generalized to :

$$\begin{array}{rc} \min & \sum_{f,n} d_{IS}(V_{fn},\hat{V}_{fn}) + \dots \\ & \lambda \sum_{n\in\mathcal{L}} \sum_f d_{IS}(V_{fn}, \sum_{g\in \mathrm{pres}(n)} \hat{V}_{fn}^{(g)}) + \dots \\ & \lambda \sum_{n\in\mathcal{L}} \sum_f d_{IS}(0, \sum_{g\in \mathrm{abs(n)}} \hat{V}_{fn}^{(g)}) \,, \\ \text{subject to} & W \geq 0 \, H \geq 0 \,. \end{array} \tag{4.9}$$

where $\hat{V}_{fn}^{(g)} = \sum_k W_{fk}^{(g)} H_{kn}^{(g)}$.

4.5.2.2 Source separation of three instrumental tracks

We display here separation results on a mix of three instruments. The task is particularly difficult because the drum source is always active, as can be seen on

Figure 4.11. Compared to random estimates, the baseline and time-annotated methods manage to obtain better estimates of piano and guitar. However, only time-frequency annotations might still improve upon those results, which are far below standards.

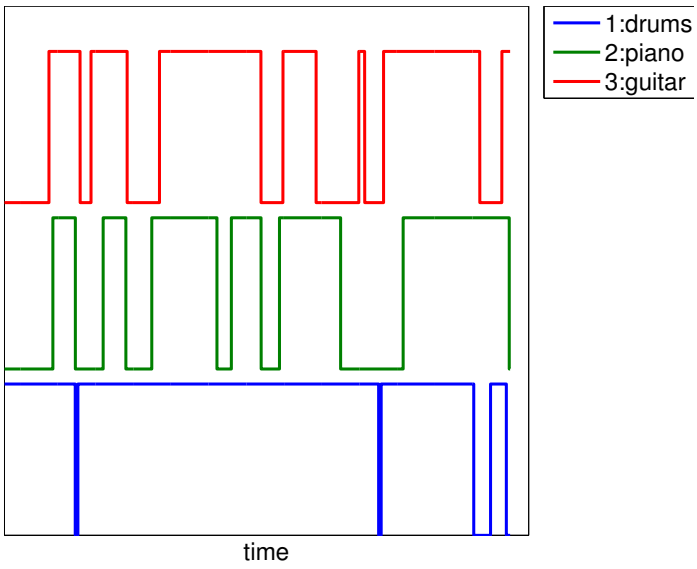


Figure 4.11: Time annotations used for the three instruments' track.

	user	baseline	self	oracle
sdr1	5.30	9.53	8.79	17.26
sdr2	-7.72	-6.73	-11.06	6.57
sdr3	-4.86	-7.80	-13.37	4.99
sir1	21.98	13.43	8.79	24.12
sir2	-1.90	0.89	-11.06	19.58
sir3	5.04	1.17	-13.37	18.81
sar1	5.42	11.99	261.79	18.28
sar2	-2.34	-3.31	261.79	6.84
sar3	-3.21	-4.75	261.79	5.24
% ann.	100.00	0.00	0.00	0.00

Table 4.6: Source separation results for a track with three instruments

As we can see on Table 4.6, time only annotations are not sufficient to retrieve good enough source estimates.

4.5.3 Handling uncertainty in automatic annotations

While the detectors proposed in Section 4.3 masking coefficients $M_{fn}^{(g)} \in [0, 1]$, values such as $(0.5, 0.5)$ do not really indicate that sources are mixed in these

proportions. Instead, we interpret a value of $(0.5, 0.5)$ as a measure of uncertainty : the probability that source 1 is dominant in this time-frequency bin is 0.5. This time-frequency bin should thus be discarded from the set of labelled time-frequency bins $\mathcal{L}$. In this Section, we introduce a weighting term to take into account this uncertainty.

We introduce a weighting term μ_{fn}, that depends on M, in order to compensate for uncertainty in our estimate of $V_{fn}^{(g)}$. Assume the time-frequency index (f, n) is fixed and we observe $X = \sum_k x_k$, where $x_k \sim \mathcal{N}(0, v_k)$. Then the distribution of $x = (x_1, \ldots, x_k)$ conditional on the value of X is also a Gaussian distribution. The expectation and variances are as follows :

$$\begin{aligned} \mathbb{E}x &= mX \\ \mathbb{E}(x - \mathbb{E}x)(x - \mathbb{E}x)^\top &= v(\operatorname{diag} m - mm^\top)\,. \end{aligned} \tag{4.10}$$

where $m_k = \frac{v_k}{\sum_l v_l}$ and $v = \sum_k v_k$. In particular, each individual prediction has variance $m_k(1 - m_k)v$. We propose then to compute the weight μ_{fn} as :

$$\mu_{fn} = 1 - \frac{G}{G-1} \sum_g M_{fn}^{(g)}(1 - M_{fn}^{(g)})\,. \tag{4.11}$$

Thus $0 \leq \mu_{fn} \leq 1$ and $\mu_{fn} = 0$ if all $M_{fn}^{(g)}$ are equal, μ compensates for the fact that Wiener coefficients bring us a lot of information although it is not perfect.

As a consequence, we use the following complete formulation of semi-supervised NMF in our algorithm:

$$\min_{W \geq 0, H \geq 0} \quad \textstyle\sum_{(f,n)} d_{IS}(V_{fn}, \hat{V}_{fn}) + \lambda \sum_{(f,n)\in\mathcal{L}} \mu_{fn} \sum_{g\in\mathcal{G}} d_{IS}(M_{fn}^{(g)} V_{fn}, \hat{V}_{fn}^{(g)}), \tag{4.12}$$

where

$$\tilde{V}_{fn}^{(g)} = M_{fn}^{(g)} V_{fn} \qquad \hat{V}_{fn}^{(g)} = \sum_k W_{fk}^{(g)} H_{kn}^{(g)}\,. \tag{4.13}$$

It only differs from 4.12 by the additional term μ_{fn} in the second term. This implies only minor changes in the algorithm itself. In practice, if there are two sources and masking coefficients take values in $(0.5, 0.5)$, then $\mu_{fn} = 0$ and the penalty term for time-frequency bin (f, n) disappears.

4.5.4 Predicting source specific time intervals

To improve the reliability of our detectors, we try predicting time-only annotations on long audio tracks from the QUASI database. We first show that time-only annotations may be cast as an averaged version of time-frequency annotations, so that we may combine the output of any of the detectors proposed

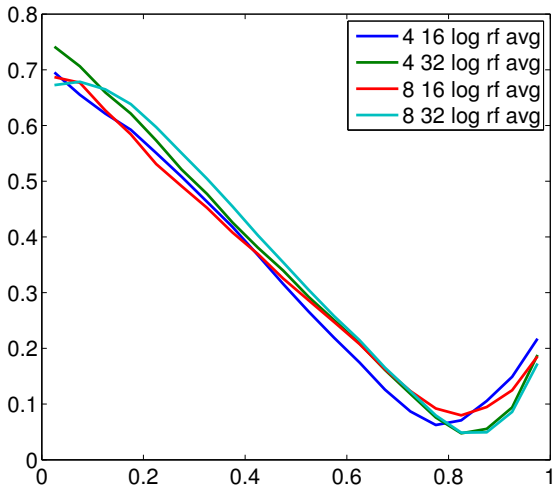


Figure 4.12: Average prediction error $(\hat{\mu}_n^{(1)} - \mu_n^{(1)})^2$ plotted against true $\mu_n^{(1)}$, for various detectors trained on the SISEC database.

previously to predict time annotations. We then examine how these results comply with manual annotations, since those have very small error. Finally we compare source separation results when combining these automatic time annotations with our NMF algorithm.

Define by $\mu_n^{(g)}$ the relative amplitude of source (g) at time n, i.e., the ratio of the energy of source (g) to the total energy of the signal. The following relationship holds (in expectation w.r.t. to the Gaussian random variables $S_{fn}^{(g)}$):

$$\mu_n^{(g)} = \frac{\sum_f |S_{fn}^{(g)}|^2}{\sum_{f,g'} |S_{fn}^{(g')}|^2} = \frac{\sum_f \mu_{fn}^{(g)} V_{fn}}{\sum_f V_{fn}} \,. \tag{4.14}$$

Thus $\mu_n^{(g)}$ is a weighted average of the same Wiener coefficients that are the output of our detectors. In practice, in order to save computational resources, we only predict a fraction of the indices f, so we compute approximate averages with only those indices. Figure 4.12 displays prediction error averaged over all tracks from the QUASI database, for various detectors that were trained on the SISEC database. Results could be further improved by using data from the QUASI database.

4.6 Related work

Although our formulation of semi-supervised NMF is entirely novel, we were strongly inspired by work on matrix completion : matrix factorizations is one among other techniques used to perform matrix completion [Srebro and Jaakkola, 2003] or solve the problem of inpainting [Bertalmío et al., 2000]. Our work on

supervised detection was inspired by [Yu and Slotine, 2009], who have built audio classifiers based on measures of local time-frequency information.

Our approach is closely related to discriminative learning of time-frequency masks proposed in [Emiya et al., 2009], although we did not know of this approach at the time we published our work. We were inspired by methods proposed [Lagrange et al., 2008, Bach and Jordan, 2004], which are unsupervised, in contrast to ours : instead of estimating a model and comparing it to data, only pairwise comparison between data points are used. More information is used than in our setting to compute those comparisons : in particular, both papers rely on some kind of multi-pitch detection scheme to compare time-frequency bins that are far apart in frequency coordinates.

4.7 Conclusion

We have introduced in this chapter an extension of NMF that shares similarities with inpainting and allows incorporating user-provided side information, that is specific to the mixed signal at hand. Our framework generalizes the case of using time annotations or score information to initialize H with zeros where sources are known to be absent.

A striking difference between time-frequency annotations and time annotations is the performance quickly reaches its limits even when 100% of the track is annotated. In contrast, if 100% of *correct* time-frequency annotations are provided, the source separation result are perfect. As we have shown in Section 4.4.2, even with 20% of annotations, the obtained estimates are very satisfactory.

This calls for new exciting directions in source separation. Indeed, the task of annotating time-frequency regions is very hard, even for trained users having both a good ear and a good knowledge of time-frequency representations. It is therefore important to develop automatic annotations tool which might complement user annotations. To this end, we have experimented with simple regression techniques, such as random forests, using local time-frequency information.

Several improvements might be made to our automatic annotation unit : incorporating features from multipitch detection was indeed critical in [Bach and Jordan, 2004, Lagrange et al., 2008]. Another drawback of our detectors is that masking coefficients are heavily biased towards 0.5 : in our opinion this is due to the use of the ℓ_2 loss. The logistic loss is better suited to predicting coefficients whose distribution is strongly concentrated around 0 and 1.

We view our contribution to training automatic detectors as a starting point to multiple working directions : in addition to technical improvements to the particular system we have proposed, there may be other ways to build automatic annotation tools.

One idea would be to alternate annotations and runs of semi-supervised NMF, then in the next rounds correct wrong annotations, etc. However, in our own experience, correcting annotations is hard, because the contribution of a small time-frequency region to the global source estimate is difficult to “hear apart”.

However, it is worth thinking about the concept of reinforcement learning, which in our case means that annotations would be successively proposed by the system and corrected by the user.

Chapter 5

Conclusion

In this thesis we have made several contributions to audio source separation. These contributions are independent and complementary, each one of them tackles a different aspect of learning for audio signals, but they may be combined together seamlessly. Each of them opens perspectives for future work.

First we have experimented in Chapter 2 with group-sparsity penalized NMF in signals which in some parts are mixed and in other parts feature isolated sources. In this case, group-sparsity allows identifying which segment are mixed and which segment are isolated. Actually, source specific models may also be learnt in a more general setting : in the case where each source is modelled by only one dictionary element, then provided that for each source, there is at least one long segment where this and only this source is missing, it was shown in [Laurberg et al., 2008a] that learning NMF on the mixed signal is equivalent to learning NMF on each signal in isolation and then using that to separate sources. If more than one component per source is needed, theoretical results must still be investigated. We have proposed a criterion to select the penalty parameter based on tools from statistical theory. It is a competitive alternative to cross-validation, and may be used to select parameters as soon as a generative model of the data is provided. Compared to selection methods based on probabilistic extensions (Bayesian nonparametric methods)[Zhou et al., 2009, Tan and Févotte, 2009, Hoffmann et al., 2010b], it is more computationally demanding but on the other hand it may be used out of the box without the need to modify the algorithm used to estimate the dictionary and activation coefficients.

There are still algorithmic challenges in sparse NMF, namely finding algorithms which do not suffer from absorbing zeroes. Current algorithms are still slow when compared to running NMF with the Euclidean loss. Alternatives to multiplicative updates based on active set algorithms [Kim and Park, 2008] or block coordinate descent [Gillis and Glineur, 2012] are interesting candidates that may be studied in the general case of β-divergences. Once efficient solutions are found for sparse NMF, more elaborate sparsity inducing terms should be investigated. Recent contributions have been made to take into account local structure, with smoothness penalties or hidden Markov models [Mysore et al., 2010, Févotte, 2011a, Virtanen, 2007, Vincent et al., 2010b, Virtanen et al., 2008]. But it is also important to look at larger time intervals, at the level of sentences in speech, of bars and verses in music.

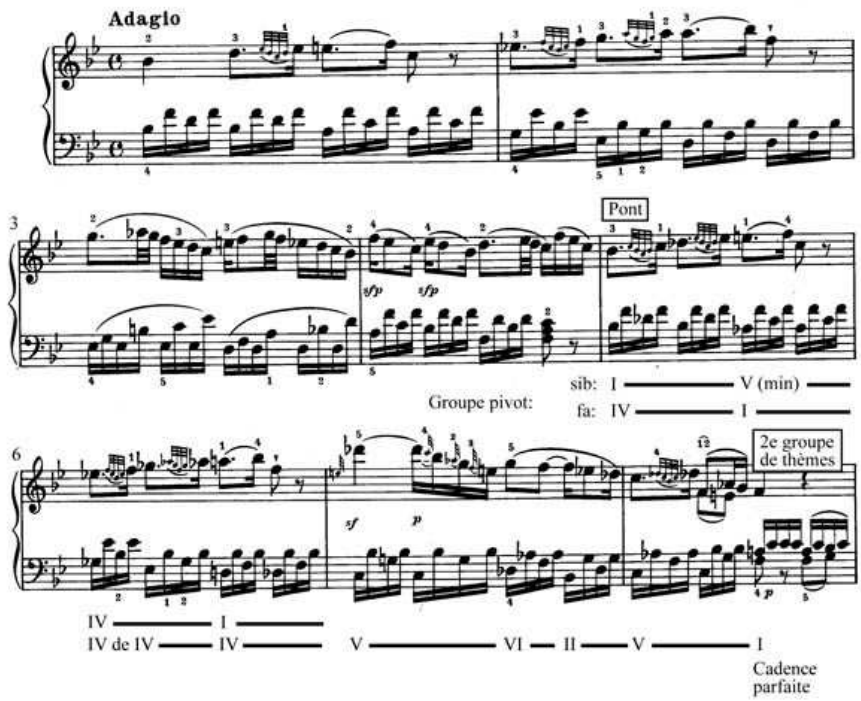


Figure 5.1: Excerpt of sheet music from second Mozart Piano Sonata K. 332.

Using score information to perform source separation was proposed before, and is still an active topic of research. The main problem is that of performing audio to score alignment : dynamic time warping, hidden Markov models [Cont, 2006, Montecchio and Cont, 2011] and conditional random fields [Joder et al., 2011] were investigated. Independently from the problem of alignment, parsing scores and extracting prior information about the most likely sequences of notes in polyphonic music might be of substantial help in polyphonic transcription. One might think of such information as clues on the kind of initial points to try when estimating activation coefficients. In order to encode such priors, dependencies in the range of a quarter of a second, i.e. 10 time bins, should be taken into account, which is already intractable for Markov models. In the last years, efficient methods have been proposed to overcome the limitation of Markov models : conditional random fields (CRF) [Lafferty et al., 2001] provide a framework to model dependency between latent variables at a given time and observations at any point in the future or the past, while still assuming a Markov property for latent variables.

Our second contribution in Chapter 3 is an online algorithm to learn dictionaries adapted to the Itakura-Saito divergence. We show that it allows a ten times speedup for signals longer than three minutes, in the small dictionary setting. It also allows running NMF on signals longer than an hour which was previously impossible. The approach we take in our contribution [Lefèvre et al., 2011b] was inspired from [Mairal et al., 2010], who additionally show in the Euclidean setting that limit points of the algorithm are stationary points of an abstract

function, provided minimal assumptions on the distribution of the data. Their algorithm is itself inspired by incremental EM algorithms [Neal and Hinton, 1998] which have been around for a long time now. Compared to popular stochastic gradient descent algorithms [Robbins and Monro, 1951], the method of averaging auxiliary functions is more robust in the sense that it does not depend on the careful tuning of a learning rate. Extensions of online NMF were proposed, in the Euclidean setting, to the case of convolutive NMF [Wang et al., 2011]. In the case of the Itakura-Saito divergence, additional penalties such as the group structured penalty proposed in Chapter 2 may be handled straightforwardly.

Online dictionary learning opens exciting prospects for dictionary learning on large scale audio databases. For instance, instrument specific dictionaries might be learnt on full-length databases such as the Iowa database. It would be interesting to extend this work to overcomplete dictionaries with sparsity penalties, so that the number of components in the dictionary can match tens of hours of audio recordings. This in turn implies additional algorithmic challenges when estimating sparse decomposition coefficients.

Our third contribution, presented in Chapter 4, goes back to short signals and blind separation : we introduce in NMF additional constraints on the estimates of the source spectrograms, which are related to the optimal masking coefficients. Other approaches to user informed source separation had been proposed before [Wang, 2009, Hennequin et al., 2011, Durrieu and Thiran, 2012]. Our contribution is closer in spirit to inpainting techniques based on matrix factorization [Roux et al., 2011, Adler et al., 2012]. In an ideal setting where part of the ground truth optimal masks is known, we achieve state-of-the-art source separation results, while being robust to small but significant amounts of errors in the masking coefficients. In order to retrieve optimal masking coefficients we appeal to techniques from computational audio scene analysis (CASA) and obtain promising results, which suggest interesting connections between CASA techniques and blind source separation. A quantized nearest-neighbor algorithm was also used in [Emiya et al., 2009] to predict masking coefficients based on training data consisting of mixed signals plus known sources. This approach is an interesting "discriminative" counterpart to the more "generative" approach of learning one dictionary per source and using concatenated dictionaries at test time. User annotations give satisfactory results, but are very time consuming and require a lot of training. Automatic tools should be provided to help the user and propagate his annotations quickly. As regards automatic annotations, the tools we have provided in this thesis need additional work : in particular, using more adapted losses based e.g., on generalized linear models [MacCullagh and Nelder, 1989] may reduce the bias observed in Section 4.4.3. Moreover, more features such as multipitch information should be added to obtain comparable results to [Bach and Jordan, 2004, Lagrange et al., 2008]. In a broader perspective, taking into account dependencies between labels may be done in two ways : either pre-clustering the spectrogram in pitch tracks and then predicting annotations for each pitch track independently, as is done for instance in [Reyes-Gomez and Jojic, 2011], or introducing local dependencies between labels in the framework

of Markov random fields. Structured output methods [Joachims, 2005] also offer the possibility of modelling dependencies between latent variables.

Audio source separation is an example of the rich interplay between machine learning, optimization and signal processing. Starting from the open problem of single-channel source separation, machine learning offers generic tools to learn models either based on training data or in an unsupervised fashion. Adapting these tools and scaling them to the size of audio collections in turn poses significant optimization issues. Algorithmic solutions open the way for more imaginative solutions : how can we use additional data, additional information ? What are the new machine learning tools we need to use this information and solve our problem. This interplay between many fields of applied mathematics is at the heart of the contributions we have presented in this thesis.

Appendix A

Projected gradient descent

The following theorem may be found in [Bertsekas, 1999]. For simplicity we assume that f is twice continuously differentiable although the results presented here hold under weaker assumptions.

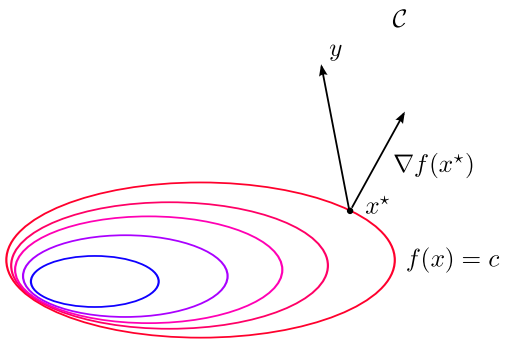


Figure A.1: Illustration of the stationarity property for a constrained minimum problem. There is no feasible direction $y - x$ that locally decreases the value of $f(x^\star)$.

Definition 1. *Let $f : \mathbb{R}^n \to R$ be a twice differentiable function. Consider the following optimization problem*

$$\min_{x \in \mathcal{C}} f(x) \tag{A.1}$$

where $\mathcal{C}$ is a convex set. $x^\star \in \mathcal{C}$ is a stationary point (or local minimum) of Problem A.1 if :

$$\forall y \in \mathcal{C}, (y - x^\star)^\top \nabla f(x^\star) \geq 0\,. \tag{A.2}$$

This condition is necessary but not sufficient.

Theorem 4. *Define projected gradient descent algorithm with diminishing step size :*

$$x^{(0)} \in \mathcal{C} \tag{A.3}$$

$$x^{(t)} = (x^{(t-1)} - \mu^{(t)} \nabla f(x^{(t-1)}))_+ \tag{A.4}$$

where $(\cdot)_+$ *is the projection operator onto* $\mathcal{C}$ *and*

$$\mu^{(t)} \to 0 \quad \sum_{t=1}^{\infty} \mu^{(t)} = \infty$$

If there exists a constant L *such that* $\|\nabla^2 f(x)\|_2 \leq L$ *for all* x*, then every limit point of projected gradient descent is a stationary point of Problem A.1.*

This is one of several variants of projected gradient descent. In particular, we did not use the Armijo rule to update the step size.

Appendix B

Description of music databases used in this thesis

We used two publicly available databases in our experiments : the QUASI database[1], and the SISEC database for Professionally Produced Music Recordings[2] (PPMR). For each database, mixes and source files were processed as follows : All source tracks were down-sampled from 44100 Hz to 16000 Hz, and down-mixed to mono by taking the averaging of left and right channels. A voice track and accompaniment track are then created by aggregating the various source files, and then a final mix is created by summing the two tracks.

Table B.1b displays the train set, i.e. tracks provided along with full sources. Table B.1c displays the test set, i.e. full tracks provided without accompanying sources. We could only use the train set which we split in two to validate user informed NMF, as we need ground truth sources to compute performance measures. As we can see on Table B.1b, the PPMR task is particularly difficult because tracks are very short : only ten seconds per track. Moreover, listening to those tracks reveals that they are very different from each other : different instruments, different music genres, different recording conditions. For instance, there is very little in common between the acoustic track "Tamy - Que pena tanto faz", where sources are an acoustic guitar and a female voice with little or no post-processing, and "Fort Minor - Remember the name", a rap track, with a saturated bass, a piano loop which was certainly created with commercial software, two male voices, and additional sound effects. This is one example of the wide variations between tracks, which is one of the reason why training a dictionary with NMF on train tracks is not sufficient to yield satisfactory results.

In the QUASI database, on the other hand, there is no train and test set, and full tracks are released along with the sources so that extensive training may be performed. In our opinion this is a major step forward as it will emphasize overfitting problems and favor algorithms tailored for large scale learning.

[1] http://www.tsi.telecom-paristech.fr/aao/en/2012/03/12/quasi/

[2] http://sisec.wiki.irisa.fr

Track (artist - title)	Duration (hh:mm:ss.cs)
another dreamer-one we love	00:03:25.32
carl leth-the world is under attack	00:05:08.67
fort minor-remember the name	00:03:48.35
glen philips-the spirit of shackleton	00:04:04.98
jims big ego-mix tape	00:03:03.29
nine inch nails-good soldier	00:03:22.97
shannon hurley-sunrise	00:03:14.07
ultimate nz tour	00:02:21.12
vieux farka-ana	00:04:09.76

(a) Listing of the QUASI database

Track (artist - title)	Duration (hh:mm:ss.cs)
bearlin-roads	00:00:14.00
tamy-que pena tanto faz	00:00:13.00
another dreamer-the ones we love	00:00:25.00
fort minor-remember the name	00:00:24.00
ultimate nz tour	00:00:18.00

(b) Listing of the SISEC database (short recordings)

Track (artist - title)	Duration (hh:mm:ss.cs)
another dreamer-the ones we love	00:03:25.32
fort minor-remember the name	00:03:48.35
ultimate nz tour	00:02:21.12

(c) Listing of the SISEC database (full recordings)

Table B.1: Listing of the databases used in our source separation tasks

Appendix C

Detailed results of semi-supervised NMF

We provide in this appendix track by track source separation results for the semi-supervised NMF algorithm presented in Chapter 4. Audio samples are available online[1]. A detailed inspection reveals that user annotations consistently outperform NMF with ideally permuted components (which requires knowing the ground truth !). The average gain is small in SDR (1dB) but very significant in terms of Source to Interference Ratio (4dB on average). Remind that for the SISEC database, user annotations are time-frequency while in the QUASI database they are time-only. Source 1 is always the accompaniment, and source 3 is always the voice.

In both scenarios, we compare five methods :

auto : Automatic annotations and semi-supervised NMF. The best detector from Table 4.3 was chosen.

user : User annotations and semi-supervised NMF (time-frequency annotations for SISEC, manual annotations for QUASI). We tried $K \in \{5, 10, 20\}$ for the SISEC database and $\{10, 20, 50\}$ for the QUASI database, as well as $\lambda \in \{1, 10, 100\}$, and selected parameters yielding highest SDR for fair comparison with the baseline.

baseline : Run NMF and permute factors to obtain optimal SDR. We set $K = 8$ because it already takes a 10 times as long to evaluate SDRs for all permutation on a single track as it takes to run semi-supervised NMF.

self : set $s^{(g)} = \frac{1}{G}x$ as estimates for the sources, it serves to estimate the difficulty of the source separation problem for a given database.

oracle : results obtained with Wiener coefficients computed from the ground truth. In addition we display track by track annotation accuracy for user annotations, for comparison with Table 4.2. For each method, we ran NMF three times for 1000 iterations to avoid local minima, and kept the run with the lowest objective cost value.

For each track, the column self gives an idea of how hard it is to provide good source estimates. The results obtained by **self** gives an idea of the difficulty of the problem. Since the mixture itself is given as an estimate for both source, the SAR is always $+\infty$ (finite values in Tables C.1-C.2 are due to rounding errors for

[1] `www.di.ens.fr/ lefevra/annot.html`

numbers smaller than $10^{-}16$). As we can see, in each track the voice is always harder to extract than the accompaniment.

Overall results have been discussed in Section 4.4.4. Detailed results on the SISEC database (Table C.1) reveal that automatic annotations often yield results below the "significance threshold" indicated by **self**. On the other hand, that of the baseline and of user annotations are systematically above that. User annotations yield consistently equal or better SDR values than the baseline, which suggest that user annotations allow to group components. Recall that the baseline consists in NMF with optimally permuted components, which requires knowing the ground truth.

For longer tracks such as those in the QUASI database, the results obtained by all methods are are not always above **self**. One reason for this is that there are large intervals in which only one source is active, so that providing the mix itself as an estimate of the sources makes sense. A more careful comparison would consist in measuring SDR only where sources are really mixed.

	auto	**user**	**baseline**	**self**	**oracle**
SDR1	2.44	8.74	8.20	5.07	15.65
SDR2	2.35	3.18	0.86	-5.11	10.34
SIR1	4.87	21.85	12.05	5.07	23.84
SIR2	6.08	13.62	3.47	-5.11	23.36
SAR1	7.36	8.99	10.76	273.30	16.38
SAR2	5.71	3.77	5.93	273.30	10.58
% ann.	10.37	22.61	0.00	0.00	0.00

(a) Track 1

	auto	**user**	**baseline**	**self**	**oracle**
SDR1	0.22	2.07	2.66	-2.72	11.78
SDR2	-0.65	6.51	6.21	2.54	14.52
SIR1	2.39	24.87	7.22	-2.72	23.81
SIR2	3.60	9.74	9.22	2.54	24.77
SAR1	6.25	2.11	5.29	280.02	12.08
SAR2	2.97	9.75	9.71	280.02	14.97
% ann.	6.31	10.33	0.00	0.00	0.00

(b) Track 2

	auto	**user**	**baseline**	**self**	**oracle**
SDR1	-0.61	5.09	4.37	2.68	13.27
SDR2	-0.84	1.02	-0.02	-2.79	10.42
SIR1	1.75	10.79	6.41	2.68	20.90
SIR2	1.94	10.43	3.60	-2.79	20.99
SAR1	5.39	6.80	9.54	268.33	14.13
SAR2	4.57	1.92	4.02	268.33	10.86
% ann.	8.79	29.64	0.00	0.00	0.00

(c) Track 3

	auto	**user**	**baseline**	**self**	**oracle**
SDR1	1.83	8.92	9.41	7.35	18.43
SDR2	1.17	-0.39	-0.61	-7.34	10.82
SIR1	3.67	17.05	13.98	7.35	27.44
SIR2	6.65	11.60	4.05	-7.34	26.46
SAR1	7.99	9.73	11.45	295.03	19.02
SAR2	3.46	0.19	2.65	295.03	10.95
% ann.	9.29	16.68	0.00	0.00	0.00

(d) Track 4

Table C.1: Track by track results on the SISEC database

	auto	user	baseline	self	oracle
SDR1	7.89	8.66	7.33	7.51	16.76
SDR2	-5.95	-1.85	-2.84	-7.54	8.83
SIR1	8.04	13.93	14.18	7.51	24.04
SIR2	-5.48	7.86	1.03	-7.54	22.84
SAR1	23.22	10.36	8.50	278.94	17.68
SAR2	10.49	-0.70	1.97	278.94	9.03
% ann.	5.08	100.00	0.00	0.00	0.00

(a) Track 1

	auto	user	baseline	self	oracle
SDR1	6.24	8.51	7.95	5.17	17.67
SDR2	-1.88	2.50	0.83	-5.16	12.34
SIR1	6.41	18.51	11.50	5.17	28.09
SIR2	-1.19	11.13	6.38	-5.16	27.95
SAR1	21.35	9.03	10.77	283.12	18.09
SAR2	10.11	3.46	3.15	283.12	12.47
% ann.	8.25	100.00	0.00	0.00	0.00

(b) Track 2

	auto	user	baseline	self	oracle
SDR1	4.54	3.39	-0.68	4.38	15.89
SDR2	-3.98	-2.26	-3.06	-4.40	11.33
SIR1	4.70	11.26	13.21	4.38	24.92
SIR2	-3.36	3.28	-1.48	-4.40	24.53
SAR1	20.35	4.48	-0.30	251.42	16.48
SAR2	9.80	0.84	5.91	251.42	11.56
% ann.	5.85	100.00	0.00	0.00	0.00

(c) Track 3

	auto	user	baseline	self	oracle
SDR1	8.63	9.70	9.46	7.77	17.47
SDR2	-4.66	0.43	-1.27	-7.74	9.33
SIR1	8.88	19.07	14.04	7.77	25.31
SIR2	-4.04	7.73	2.98	-7.74	24.21
SAR1	21.62	10.29	11.49	264.99	18.26
SAR2	9.65	2.00	2.55	264.99	9.49
% ann.	7.86	100.00	0.00	0.00	0.00

(d) Track 4

	auto	user	baseline	self	oracle
SDR1	12.36	13.38	8.26	12.22	19.14
SDR2	-10.51	-3.94	-9.23	-12.23	6.11
SIR1	12.78	16.19	15.97	12.22	24.00
SIR2	-10.06	13.04	-6.10	-12.23	21.63
SAR1	22.95	16.70	9.17	279.84	20.88
SAR2	10.10	-3.64	0.72	279.84	6.27
% ann.	6.67	100.00	0.00	0.00	0.00

(e) Track 5

	auto	user	baseline	self	oracle
SDR1	-0.26	0.70	1.99	-0.27	10.04
SDR2	-0.14	-9.04	2.45	0.25	10.32
SIR1	-0.22	10.61	6.44	-0.27	18.56
SIR2	0.47	-6.84	3.71	0.25	18.62
SAR1	24.17	1.52	4.81	262.04	10.75
SAR2	11.52	2.62	9.99	262.04	11.07
% ann.	7.37	100.00	0.00	0.00	0.00

(f) Track 6

	auto	user	baseline	self	oracle
SDR1	9.23	7.15	7.49	8.44	17.71
SDR2	-5.55	-26.57	-3.94	-8.49	8.82
SIR1	9.56	10.75	16.46	8.44	25.32
SIR2	-4.96	-18.25	-2.30	-8.49	24.21
SAR1	21.08	10.00	8.18	264.30	18.54
SAR2	9.56	-7.56	5.42	264.30	8.97
% ann.	8.56	100.00	0.00	0.00	0.00

(g) Track 7

	auto	user	baseline	self	oracle
SDR1	5.45	9.24	8.56	4.49	20.40
SDR2	-1.96	4.15	3.40	-4.48	15.85
SIR1	5.62	20.06	18.66	4.49	34.73
SIR2	-1.41	14.77	10.81	-4.48	34.65
SAR1	20.51	9.66	9.07	262.99	20.57
SAR2	11.03	4.69	4.62	262.99	15.91
% ann.	5.64	100.00	0.00	0.00	0.00

(h) Track 8

Table C.2: Track by track results on the QUASI database

Bibliography

S.A. Abdallah and M.D. Plumbley. Polyphonic transcription by non-negative sparse coding of power spectra. In *International Conference on Music Information Retrieval (ISMIR)*, 2004.

Amir Adler, Valentin Emiya, G. Maria Jafari, Michael Elad, Rémi Gribonval, and Mark D. Plumbley. Audio Inpainting. *IEEE Transactions on Audio, Speech and Language Processing*, 2012.

M. Aharon, M. Elad, and A. Bruckstein. K-SVD: Design of dictionaries for sparse representation. In *Signal Processing with Adaptative Sparse Structured Representations (SPARS)*, 2005.

F. Bach and M.I. Jordan. Blind one-microphone speech separation: A spectral learning approach. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

R. Badeau, G. Richard, and B. David. Sliding window adaptive svd algorithms. *IEEE Transactions on Signal Processing*, 2004.

L. Benaroya and F. Bimbot. Wiener based source separation with hmm/gmm using a single sensor. In *International Conference in Independant Component Analysis (ICA)*, 2003.

L. Benaroya, F. Bimbot, and R. Gribonval. Audio source separation with a single sensor. *IEEE Transactions on Audio Speech and Language Processing*, 2006.

S. Bengio, F. Pereira, Y. Singer, and D. Strelow. Group sparse coding. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2000.

D. Bertsekas. *Nonlinear Programming*. Athena Scientific Belmont, MIT, 1 edition, 1999.

S. Bucak and B. Gunsel. Incremental subspace learning via non-negative matrix factorization. *Pattern Recognition*, 2009.

W. Buntine. Variational extentions to em and multinomial PCA. In *European Conference on Machine Learning*, 2002.

B. Cao, D. Shen, J.T. Sun, X. Yang, and Z. Chen. Detect and track latent factors with online nonnegative matrix factorization. In *International Joint Conference on Artifical Intelligence (IJCA)*, 2007.

Y. Cao, P. Eggermont, and S. Terebey. Cross Burg entropy maximization and its application to ringing suppression in image reconstruction. *IEEE Transactions on Image Processing*, 1999.

O. Cappé, C. Févotte, and D. Rohde. Algorithme em en ligne simulé pour la factorisation non-négative probabiliste. In *Colloque du GRETSI*, 2011.

J-F Cardoso. Blind signal separation: statistical principles. *Proceedings of the IEEE, special issue on blind identification and estimation*, 1998.

M.A. Casey and A. Wetsner. Separation of mixed audio sources by independent subspace analysis. In *Internation Computer Music Conference (ICMC)*, 2000.

A.T. Cemgil, P. Peeling, O. Dikmen, and S.J. Godsill. Prior structures for time-frequency energy distributions. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2007.

P. Comon. Independent component analysis, a new concept ? *Signal Processing, Elsevier*, 1994.

A. Cont. Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical hmms. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2006.

M.E. Daube-Witherspoon and G. Muehllehner. An iterative image space reconstruction algorithm suitable for volume ect. *IEEE Transactions on Medical Imaging*, 1986.

L. Daudet. Sparse and structured decompositions of signals with the molecular matching pursuit. *IEEE Transactions on Audio Speech and Language Processing*, 2006.

A. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 1977.

A. Dessein, A. Cont, and G. Lemaitre. Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence. In *International Conference on Music Information Retrieval (ISMIR)*, 2010.

O. Dikmen and A.T. Cemgil. Gamma markov random fields for audio source modelling. In *Signal Processing with Adaptative Sparse Structured Representations (SPARS)*, 2009.

C. Ding, T. Li, and M.I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

Z. Duan, G.J. Mysore, and P. Smaragdis. Online PLCA for real-time semi-supervised source separation. In *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2012.

J.-L. Durrieu. *Automatic Transcription and Separation of the Main Melody in Polyphonic Music Signals*. PhD thesis, Telecom ParisTech, 2010.

J.-L. Durrieu and J.-P. Thiran. Musical audio source separation based on user-selected f0 track. In *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2012.

J.-L. Durrieu, G. Richard, B. David, and C. Févotte. Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE Transactions on Audio Speech and Language Processing*, 2010.

D.P.W. Ellis. Sinewave and sinusoid+noise analysis/synthesis in Matlab, 2003. URL `http://www.ee.columbia.edu/~dpwe/resources/matlab/sinemodel/`. online web resource.

V. Emiya, E. Vincent, and R. Gribonval. An investigation of discrete-state discriminant approaches to single-sensor source separation. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2009.

C. Févotte. Majorization-minimization algorithm for smooth Itakura-Saito nonnegative matrix factorization. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2011a.

C. Févotte. Majorization-minimization algorithm for smooth Itakura-Saito nonnegative matrix factorizations. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2011b.

C. Févotte and A.T. Cemgil. Nonnegative matrix factorizations as probabilistic inference in composite models. In *European Signal Processing Conference (EUSIPCO)*, 2009.

C. Févotte and J. Idier. Algorithms for nonnegative matrix factorization with the beta-divergence. *Neural Computation*, 2011.

C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis. *Neural Computation*, 2009.

D. FitzGerald, M. Cranitch, and E. Coyle. On the use of the beta divergence for musical source separation. In *Irish Signals and Ssytems Conference*, 2009.

J. Ganseman, P. Scheunders, and S. Dixon. Improving plca-based score-informed source separation with invertible constant-q transforms. In *European Signal Processing Conference (EUSIPCO)*, 2012.

Q. Geng, H. Wang, and J. Wright. On the local correctness of ℓ_1 minimization for dictionary learning. Technical report, Microsoft Research Asia, 2011.

N. Gillis and F. Glineur. Accelerated multiplicative updates and hierarchical als algorithms for nonnegative matrix factorization. *Neural Computation*, 2012.

R. Gribonval and K. Schnass. Dictionary Identification - Sparse Matrix-Factorisation via ℓ_1-Minimisation. *IEEE Transactions on Information Theory*, 2010.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition edition, 2009.

R.J. Hathaway. Another interpretation of the EM algorithm for mixture distributions. *Statistics and Probability Letters*, 1986.

R. Hennequin. *Décomposition de spectrogrammes musicaux informée par des modèles de synthèse spectrale*. PhD thesis, Telecom ParisTech, 2010.

R. Hennequin, B. David, and R. Badeau. Score informed audio source separation using a parametric model of non-negative spectrogram. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2011.

M.D. Hoffmann, D.M. Blei, and F. Bach. Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*, 2010a.

M.D. Hoffmann, D.M. Blei, and P. Cook. Bayesian nonparametric matrix factorization for recorded music. In *International Conference on Machine Learning (ICML)*, 2010b.

A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 1999.

M.G. Jafari and M.D. Plumbley. Speech denoising based on a greedy adaptive dictionary algorithm. In *European Signal Processing Conference (EUSIPCO)*, 2009.

G. Jang, T.W. Lee, J-F Cardoso, E. Oja, and S. Amari. A maximum likelihood approach to single-channel source separation. *Journal of Machine Learning Research*, 2003.

R. Jenatton, F. Bach, and J.-Y. Audibert. Structured variable selection with sparsity-inducing norms. Technical report, arXiv, 2009.

R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research*, 2011a.

R. Jenatton, R. Gribonval, and F. Bach. Local analysis of sparse coding in the presence of noise. In *NIPS Workshop on Sparse Representation and Low-rank Approximation*, 2011b.

R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 2011c.

T. Joachims. A support vector method for multivariate performance measures. In *International Conference on Machine Learning (ICML)*, pages 377–384, 2005.

C. Joder, S. Essid, and G. Richard. A conditional random field framework for robust and scalable audio-to-score matching. *IEEE Transactions on Audio Speech and Language Processing*, 2011.

H. Kameoka, N. Ono, K. Kashino, and S. Sagayama. Complex nmf: A new sparse representation for acoustic signals. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2009.

H. Kim and K. Park. Non-negative matrix factorization based on alternating non-negativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications*, 2008.

B. King and L. Atlas. Single-channel source separation using simplified-training complex matrix factorization. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010.

B. King and L. Atlas. Single-channel source separation using complex matrix factorization. *IEEE Transactions on Audio Speech and Language Processing*, 2011.

J. Kovačevič, V.K. Goyal, and M. Vetterli. *Signal Processing : Fourier and Wavelet Representations*. Released under the Attribution-NonCommercial-NoDerivs 3.0 Unported License, 2012.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*, 2001.

M. Lagrange, L.G. Martins, J. Murdoch, and G. Tzanetakis. Normalized cuts for predominant melodic source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 2008.

H. Laurberg, M.G. Christensen, M.D. Plumbley, L.K. Hansen, and S.H. Jensen. Theorems on positive data: on the uniqueness of nmf. *Computer Intelligence and Neuroscience*, 2008a.

H. Laurberg, M.N. Schmidt, M.G. Christensen, and S.H. Jensen. Structured non-negative matrix factorization with sparsity patterns. In *Asilomar Conference on Signals, Systems and Computers*, 2008b.

J. Le Roux, E. Vincent, Y. Mizunoo, H. Kameoka, N. Ono, and S. Sagayama. Consistent Wiener filtering: Generalized time-frequency masking respecting spectrogram consistency. In *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2010.

D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 1999.

D.D. Lee and H.S. Seung. Algorithms for nonnegative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

A. Lefèvre, F. Bach, and C. Févotte. Itakura-Saito nonnegative matrix factorization with group sparsity. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2011a.

A. Lefèvre, F. Bach, and C. Févotte. Online algorithms for nonnegative matrix factorization with the Itakura-Saito divergence. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2011b.

A. Lefèvre, F. Bach, and C. Févotte. Semi-supervised NMF with time-frequency annotations for single-channel source separation. In *International Conference on Music Information Retrieval (ISMIR)*, 2012.

E. Lehmann and J.P. Romano. *Testing Statistical Hypotheses (Springer Texts in Statistics)*. Springer, 3rd edition, 2005.

C.-J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 2007a.

C.-J. Lin. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, 2007b.

L.B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 1974.

P. MacCullagh and J.A. Nelder. *Generalized Linear Models*. Chapman & Hall, 1989.

L. Mackey, A. Talwalkar, and M.I. Jordan. Divide-and-conquer matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

J. Mairal. *Sparse Coding for Machine Learning, Image Processing and Computer Vision*. PhD thesis, École Normale Supérieure de Cachan, 2011.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 2010.

J. Mairal, R. Jenatton, G. Obozinski, and F. Bach. Convex and network flow optimization for structured sparsity. *Journal of Machine Learning Research*, 2011.

S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way.* Academic Press, 3rd edition, 2008.

N. Montecchio and A. Cont. A unified approach to real time audio-to-score and audio-to-audio alignment using sequential montecarlo inference techniques. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2011.

K. Murao, M. Nakano, Y. Kitano, N. Ono, and S. Sagayama. Monophonic instrument sound segregation by clustering nmf components based on basis similarity and gain disjointness. In *International Conference on Music Information Retrieval (ISMIR)*, 2010.

G. Mysore, P. Smaragdis, and B. Raj. Non-negative hidden markov modeling of audio with application to source separation. In *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2010.

R.M. Neal and G.E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in Graphical Models*, 1998.

B.A Olshausen and D.J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 1997.

A. Ozerov, C. Févotte, and M. Charbit. Factorial scaled hidden markov model for polyphonic audio representation and source separation. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2009.

A. Ozerov, C. Févotte, R. Blouet, and J.-L. Durrieu. Multichannel nonnegative tensor factorization with structured constraints for user-guided audio source separation. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2011.

A. Ozerov, E. Vincent, and F. Bimbot. A general flexible framework for the handling of prior information in audio source separation. *IEEE Transactions on Audio Speech and Language Processing*, 2012.

T.W. Parsons. Separation of speech from interfering speech by means of harmonic selection. *Journal of the Royal Acoustical Society of America*, 1976.

M.D. Plumbley, S.A. Abdallah, T. Blumensath, and M.E. Davies. Sparse representations of polyphonic music. *Elsevier Signal Processing*, 2006.

M.D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M.E. Davies. , "sparse representations in audio & music: from coding to source separation". *IEEE*, 2010.

T. F. Quatieri and R.G. Danisewic. An approach to co-channel talker interference suppression using a sinusoidal model for speech. *IEEE Transactions on Acoustics Speech and Signal Processing*, 1990.

Z. Rafii and B. Pardo. A simple music/voice separation method based on the extraction of the repeating musical structure. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2011.

I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Computer Vision and Pattern Recognition*, 2010.

C. Raphael and Y. Han. A classifier-based approach to score-guided music audio source separation. *Computer Music Journal*, 2008.

M. Reyes-Gomez and N. Jojic. Signal separation by efficient combinatorial optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

W.H. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 1972.

H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 1951.

J. Le Roux, H. Kameoka, N. Ono, A. de Cheveigné, and S. Sagayama. Computational auditory induction as a missing-data model-fitting problem with bregman divergences. *Speech Communication (Special issue on Perceptual and Statistical Audition)*, 2011.

S. Roweis. One microphone source separation. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics Speech and Signal Processing*, 1978.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2000.

P. Smaragdis. User guided audio selection from complex sound mixtures. In *ACM Symposium on User Interface Software and Technology (UIST)*, 2009.

P. Smaragdis and J.C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2003.

P. Smaragdis, B. Raj, and M.V. Shashanka. Supervised and semi-supervised separation of sounds from single-channel mixtures. In *International Conference in Independant Component Analysis (ICA)*, 2007.

P. Sprechmann, I. Ramirez, P. Cancela, and G. Sapiro. Collaborative sources identification in mixed signals via hierarchical sparse modeling. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2011.

P. Sprechmann, A. Bronstein, and G. Sapiro. Real-time online singing voice separation from monaural recordings using robust low-rank modelling. In *International Conference on Music Information Retrieval (ISMIR)*, 2012.

N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *International Conference on Machine Learning (ICML)*, 2003.

V.Y.F Tan and C. Févotte. Automatic relevance determination in nonnegative matrix factorization. In *Workshop on Signal Processing with Adaptive Sparse Structured Representations*, 2009.

R. Tibshirani. Regression shrinkage and selection via the lasso. j. *Journal of the Royal Statistical Society : series B*, 1996.

M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, 1999.

E. Vincent, R. Gribonval, and C. Févotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio Speech and Language Processing*, 2006.

E. Vincent, N. Bertin, and R. Badeau. Adaptive harmonic spectral decomposition for multiple pitch estimation. *IEEE Transactions on Audio Speech and Language Processing*, 2010a.

E. Vincent, N. Bertin, and R. Badeau. Enforcing Harmonicity and Smoothness in Bayesian Non-negative Matrix Factorization Applied to Polyphonic Music Transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, 2010b.

T.O. Virtanen. Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio Speech and Language Processing*, 2007.

T.O. Virtanen, A.T. Cemgil, and S.J. Godsill. Bayesian extensions to nonnegative matrix factorisation for audio signal modelling. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2008.

B. Wang. Musical audio stream separation. Master's thesis, Queen Mary, University of London, 2009.

D. Wang, R. Vipperla, and N. Evans. Online pattern learning for convolutive non-negative sparse coding. In *Interspeech*, 2011.

G. Yu and J.J. Slotine. Audio classification from time-frequency texture. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2009.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society : series B*, 2006.

Y. Zhang and M.S. Scordilis. Effective online unsupervised adaptation of gaussian mixture models and its application to speech classification. *Pattern Recognition*, 2008.

G. Zhou, Z. Yang, S. Xie, and J.-M. Yang. Online blind source separation using incremental nonnegative matrix factorization with volume constraint. *IEEE Transactions on Neural Networks*, 2011.

M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin. Non-parametric bayesian dictionary learning for sparse image representations. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.