Incremental Multi-Source Recognition with Non-Negative Matrix Factorization

Master's Thesis

Arnaud Dessein

arnaud.dessein@ircam.fr

June 23, 2009. Paris.

Revised on February 09, 2010 to correct some errors and typos.

PARCOURS MASTER 2
ATiAM
Parcours multi-mentions du Master Sciences et Technologies
Université Pierre et Marie Curie - Paris 6
en collaboration avec TELECOM ParisTech et l'Ircam

ircam
Centre
Pompidou

This master's thesis is dedicated to incremental multi-source recognition using non-negative matrix factorization. A particular attention is paid to providing a mathematical framework for sparse coding schemes in this context. The applications of non-negative matrix factorization problems to sound recognition are discussed to give the outlines, positions and contributions of the present work with respect to the literature. The problem of incremental recognition is addressed within the framework of non-negative decomposition, a modified non-negative matrix factorization scheme where the incoming signal is projected onto a basis of templates learned off-line prior to the decomposition. As it appears that sparsity is one of the main issue in this context, a theoretical approach is followed to overcome the problem. The main contribution of the present work is in the formulation of a sparse non-negative matrix factorization framework. This formulation is motivated and illustrated with a synthetic experiment, and then addressed with convex optimization techniques such as gradient optimization, convex quadratic programming and second-order cone programming. Several algorithms are proposed to address the question of sparsity. To provide results and validations, some of these algorithms are applied to preliminary evaluations, notably that of incremental multiple-pitch and multiple-instrument recognition, and that of incremental analysis of complex auditory scenes.

Keywords: multi-source recognition, incremental system, non-negative matrix factorization, sparsity, convex optimization.

Cette thèse de master est dédiée à la factorisation en matrices non-négatives pour la reconnaissance incrémentale multi-source. Une attention toute particulière est attachée à fournir un cadre mathématique pour contrôler la parcimonie dans ce contexte. Les applications des problèmes de factorisation en matrices non-négatives à la reconnaissance des sons sont discutées pour dessiner les grandes lignes ainsi que la position et les contributions du présent travail par rapport à la littérature. Le problème de la reconnaissance incrémentale est attaqué dans un cadre de décomposition non-négative, une modification du problème standard de factorisation en matrices non-négatives où le signal est projeté sur une base de modèles apprise avant la décomposition. La question de parcimonie ressortant comme l'un des principaux problèmes dans ce contexte, elle est abordée par une approche théorique. La contribution principale de ce travail consiste en la formulation d'un cadre de factorisation parcimonieuse en matrices non-négatives. Cette formulation est motivée et illustrée par une expérience synthétique, et approchée par des techniques d'optimisation convexe comme l'optimisation par gradient, la programmation quadratique convexe et la programmation conique de second ordre. Plusieurs algorithmes sont proposés pour attaquer le problème de la parcimonie. Des résultats et validations sont proposés en appliquant certains de ces algorithmes à des évaluations préliminaires, notamment à la reconnaissance multi-pitch et multi-instrument incrémentale, et à l'analyse incrémentale de scènes sonores complexes.

Mots-clés : reconnaissance multi-source, système incrémental, factorisation en matrices non-négatives, parcimonie, optimisation convexe.

Acknowledgments

I would like to thank my two tutors at IRCAM, Arshia Cont and Guillaume Lemaitre, whose expertise, understanding and patience were a great help during this master's thesis. Their trust is also to be reckoned with, and I am grateful to them for the exciting subject they proposed me to work on.

I would also like to acknowledge the other ATIAM students at IRCAM, Baptiste, Benjamin, John, Javier, Julien, Philippe and Pierre, for the good moments we spent together. A special thanks goes out to Julien, my office mate during the internship.

I would like to thank other people I met at IRCAM, especially Mondher Ayari, Stephen Barras and Julien Tardieu for the warm welcome.

Finally, I would like to thank my parents and Eve, for their support along this master's thesis and the whole scholar year.

Contents

List of Algorithms

List of Figures

Is perception of the whole based on perception of its parts?

Lee & Seung (1999)

Introduction

Non-negative matrix factorization is a technique for data decomposition and analysis that was made popular by Lee & Seung (1999). The main philosophy of this technique is to build up the observed data in an additive manner, so that cancellation is not allowed. The technique has been applied to various problems such as face recognition, semantic analysis of text documents and audio analysis among others, for which it has proven to yield relevant results and to give meaningful part-based representations of the analyzed objects. In the present work, non-negative matrix factorization is applied to incremental multi-source recognition.

This study is an extension to the framework addressed by Cont et al. (2007) who proposed a real-time system for multi-pitch and multi-instrument recognition based on non-negative matrix factorization. In that work, it has been pointed out that in pattern recognition situations and in absence of structural *a priori* information on the signal, controlling the sparsity of the solutions is of great importance to achieve better results. The current study aims at explicitly addressing the question of sparsity and optimization of non-negative matrix factorization problems, leaving more complete applicative evaluations for future work.

Following this introduction, the present study focuses on theoretical frameworks employing non-negative matrix factorization techniques with explicit sparsity controls. Once these theoretical frameworks developed, we are able to address more complex optimization schemes which are applied in the context of incremental multi-source recognition.

Chapter 1 introduces the state-of-the-art in non-negative matrix factorization and its extensions. General applications of these methods to sound recognition are also motivated and discussed to give the outlines, positions and contributions of the present work with respect to the literature. Chapter 2 discusses the question of introducing an explicit control of sparsity in the case of non-negative factorization. The main contribution of the present work is in the formulation of a sparse non-negative matrix factorization framework. This formulation is motivated and illustrated with a synthetic experiment, and then addressed with convex optimization techniques such as gradient optimization, convex quadratic programming and second-order cone programming. Several algorithms are proposed in this framework. These algorithms are then applied to preliminary evaluations in Chapter 3. The developed algorithms are first validated on a synthetic experiment. Applications to multi-pitch and multi-instrument recognition, and analysis of complex auditory scenes are then discussed. We leave more rigorous evaluations on complex auditory scenes for future works.

1. State-of-the-Art

This chapter aims at introducing the state-of-the-art in non-negative matrix factorization, its extensions, and their applications in sound recognition. This should provide the necessary background to understand the framework of the present study and its position in the existing literature. The chapter is organized as follows. In Section 1.1, we introduce the background of non-negative matrix factorization and formulate the standard factorization problem. We also give an overview of three common classes of algorithms used to solve this problem, namely alternating least-squares, gradient descent and multiplicative updates. In Section 1.2, we review several extensions to the standard non-negative matrix factorization problem, in terms of modified cost functions, constraints or models. In Section 1.3, we present and discuss the state-of-the-art in sound recognition with non-negative matrix factorization. We focus on incremental source recognition, and define the position of the present work within this context to expose its outlines and contributions.

1.1. Non-negative matrix factorization

1.1.1. Introduction

Non-negative matrix factorization (NMF) is a low-rank approximation technique for unsupervised multivariate data decomposition, such as *vector quantization* (VQ), *principal component analysis* (PCA) or *independent component analysis* (ICA) (Lee & Seung, 1999). Given an $n \times m$ real matrix $\mathbf{V}$ and a positive integer $r < \min(n, m)$, these techniques try to find a factorization of $\mathbf{V}$ into an $n \times r$ real matrix $\mathbf{W}$ and an $r \times m$ real matrix $\mathbf{H}$ such that:

$$\mathbf{V} \approx \mathbf{WH} \tag{1.1}$$

The multivariate data to decompose is stacked into $\mathbf{V}$, whose columns represent the different observations, and whose rows represent the different variables. Each column $\mathbf{v}_j$ of $\mathbf{V}$ can be expressed as $\mathbf{v}_j \approx \mathbf{W}\mathbf{h}_j = \sum_i h_{ij}\mathbf{w}_i$, where $\mathbf{h}_j$ and $\mathbf{w}_i$ are respectively the j-th column of $\mathbf{H}$ and the i-th column of $\mathbf{W}$. The columns of $\mathbf{W}$ then form a *basis* and each column of $\mathbf{H}$ is the *decomposition* or *encoding* of the corresponding column of $\mathbf{V}$ into this basis. The rank r of the factorization is generally chosen such that $(n + m)r \ll nm$, so $\mathbf{WH}$ can be thought of as a compression or reduction of $\mathbf{V}$. In the sequel, matrices are denoted by uppercase bold letters. Lowercase bold letters denote column or row vectors, while lowercase plain letters denote scalars. Where these conventions clash, the intended meaning should be clear enough from the context.

As mentioned by Lee & Seung (1999), NMF, PCA, ICA and VQ share the same linear model expressed in Equation 1.1, but differ in the assumptions on the data and its factorization. In VQ, a hard winner-take-all constraint is imposed, *i.e.* there is a single non-null encoding coefficient per observation, so that the basis vectors represent mutually exclusive prototypes. In PCA, the basis is constrained to be orthogonal in the sense that basis vectors are constrained to be uncorrelated. In ICA, the basis vectors are constrained to be statistically independent. In both PCA and ICA, cancellation is allowed in order to decompose the data, *i.e.* encoding coefficients can be either positive or negative, thus it is possible to construct an observation by addition or subtraction of the basis vectors. However, when the data is non-negative, this may be counter-intuitive and negative values of elements cannot be interpreted (*e.g.* value of pixels, occurrence of words, magnitude spectrum of sounds).

Compared to PCA and ICA, cancellation is not allowed in NMF. The data is supposed to be non-negative, and the basis and encodings are constrained to be non-negative, so that an observation is constructed only additively. These assumptions have participated in the growing interest for NMF since the technique was made popular by Lee & Seung (1999) with applications to facial images and semantic analysis of text documents[1]. They succeeded in obtaining part-based representations of the underlying data (*e.g.* eyes, nose, eyebrows, mouth for the facial images). They also made a parallel with perception, claiming that there is psychological and physiological evidence for part-based representations in the brain, and that the non-negativity constraints could represent the fact that firing rates of neurons are never negative.

1.1.2. Standard problem

In Equation 1.1, the rank of factorization r is supposed to be less than $\min(n, m)$ so as to avoid trivial solutions. Therefore, the factorization $\mathbf{WH}$ may be inexact, *i.e.* differ from $\mathbf{V}$, so the factorization is only approximate. The aim is then to find the best factorization with respect to a given goodness-of-fit measure $\mathcal{C}$ called *cost function* or *objective function*. In the standard formulation, the Frobenius norm is used to define the following cost function:

$$\mathcal{C}(\mathbf{W}, \mathbf{H}) = \frac{1}{2}\|\mathbf{V} - \mathbf{WH}\|_F^2 = \frac{1}{2}\sum_j \|\mathbf{v}_j - \mathbf{W}\mathbf{h}_j\|_2^2 = \frac{1}{2}\sum_{i,\,j}\left(v_{ij} - [\mathbf{WH}]_{ij}\right)^2 \tag{1.2}$$

Thus, the NMF problem can be expressed as a constrained optimization problem:

$$\begin{array}{ll} \text{Given} & \mathbf{V} \in \mathbb{R}_+^{n\times m},\ r \in \mathbf{N}^*\ \text{ s.t. }\ r < \min(n, m) \\ \text{minimize} & \frac{1}{2}\|\mathbf{V} - \mathbf{WH}\|_F^2\ \text{ w.r.t. }\ \mathbf{W}, \mathbf{H} \\ \text{subject to} & \mathbf{W} \in \mathbb{R}_+^{n\times r},\ \mathbf{H} \in \mathbb{R}_+^{r\times m} \end{array} \tag{1.3}$$

The factorization may be approximate, and the solution of the problem given in Equation 1.3 is not unique. Some theoretical work about necessary or sufficient conditions for an exact or unique factorization has been done (Donoho & Stodden, 2004; Theis et al., 2005; Laurberg

[1] However, NMF can be traced back to the work of Paatero & Tapper (1994) who used the confusing and unfortunate term of *positive matrix factorization*.

et al., 2008). The uniqueness of the solution must be considered up to a permutation and a diagonal rescaling. Given optimal matrices $\mathbf{W}$ and $\mathbf{H}$, for any $r \times r$ non-negative invertible matrix $\mathbf{P}$ such that $\mathbf{P}^{-1}$ is also non-negative, $\mathbf{WP}$ and $\mathbf{P}^{-1}\mathbf{H}$ are other optimal matrices. It is easy to show that such a matrix $\mathbf{P}$ is necessarily the product of a permutation and a strictly positive diagonal rescaling matrices. But other solutions that differ from such a transformation of $\mathbf{W}$ and $\mathbf{H}$ may also exist.

These remarks point out the fact that the optimization problem in Equation 1.3 has not a unique global minimizer. This is in part due to the non-convexity of the cost function $\mathcal{C}$ in both $\mathbf{W}$ and $\mathbf{H}$. Because of this non-convexity, it is also possible that the cost function exhibits local minima. Therefore, the problem is not easy to solve and several algorithms have been developed in an attempt to achieve a good factorization.

1.1.3. Algorithms

Three common algorithms used in NMF are presented here. For more details, the interested reader can refer to Berry et al. (2007). We focus here on explaining the optimization techniques, and discuss other issues elsewhere.

Alternating least squares

The *alternating least squares* algorithms were the first to be used to solve NMF problems (Paatero & Tapper, 1994; Paatero, 1997). They are based on the property that although the cost function $\mathcal{C}$ is not convex in both $\mathbf{W}$ and $\mathbf{H}$, it is convex in $\mathbf{W}$ and $\mathbf{H}$ separately. Thus, the idea is to update $\mathbf{W}$ and $\mathbf{H}$ in turn by minimizing $\mathcal{C}$ respectively w.r.t. $\mathbf{W}$ or $\mathbf{H}$ until convergence. For the first update, either $\mathbf{W}$ or $\mathbf{H}$ needs to be initialized. In most cases, $\mathbf{W}$ is initialized and $\mathbf{H}$ is updated first, but the opposite is also possible. The two alternating minimizations are both constrained *least squares* problems, more precisely *non-negative least squares* problems:

$$\mathbf{H} \leftarrow \underset{\mathbf{H}\in\mathbb{R}_+^{r\times m}}{\arg\min} \|\mathbf{V} - \mathbf{WH}\|_F^2 \qquad \mathbf{W} \leftarrow \underset{\mathbf{W}\in\mathbb{R}_+^{n\times r}}{\arg\min} \|\mathbf{V} - \mathbf{WH}\|_F^2 \tag{1.4}$$

They can be solved exactly with a non-negative least squares algorithm, or approximatively with a projected least squares algorithm that projects the unconstrained least squares solution to the non-negative matrices orthant, *i.e.* sets to zero the negative coefficients of the unconstrained solution.

In general, alternating least squares algorithms are fast to converge. Using a non-negative least squares algorithm to solve the problems in Equation 1.4 gives a global minimizer of the corresponding problem at each iteration, and guarantees the convergence to a local minimum of the NMF problem. Using a projected least squares algorithm aids speed and sparsity, but does not guarantee to give a global minimizer of the non-negative least squares problems, nor to decrease the cost function at each iteration, what may lead to oscillations or to a convergence towards a point which is not a local minimum of the NMF problem.

Gradient descent

The *gradient descent* algorithms are a particular case of *additive updates* algorithms whose principle is to give additive update rules so as to progress in a direction, called *learning direction*, where the cost function $\mathcal{C}$ is decreasing. In gradient descent, the learning direction is expressed using the gradient of $\mathcal{C}$. For the standard NMF problem, the following additive update rules can be deduced for the coefficients of $\mathbf{W}$ and $\mathbf{H}$:

$$h_{ij} \leftarrow h_{ij} - \mu_{ij} \frac{\partial \mathcal{C}(\mathbf{W}, \mathbf{H})}{\partial h_{ij}} \qquad\qquad w_{ij} \leftarrow w_{ij} - \eta_{ij} \frac{\partial \mathcal{C}(\mathbf{W}, \mathbf{H})}{\partial w_{ij}} \tag{1.5}$$

where $\mu_{ij} \geqslant 0$ and $\eta_{ij} \geqslant 0$ are the respective *learning rates* or *steps of progression* of h_{ij} and w_{ij}. The gradient coordinates are given by:

$$\frac{\partial \mathcal{C}(\mathbf{W}, \mathbf{H})}{\partial h_{ij}} = \left[\mathbf{W}^T\mathbf{W}\mathbf{H} - \mathbf{W}^T\mathbf{V}\right]_{ij} \qquad \frac{\partial \mathcal{C}(\mathbf{W}, \mathbf{H})}{\partial w_{ij}} = \left[\mathbf{W}\mathbf{H}\mathbf{H}^T - \mathbf{V}\mathbf{H}^T\right]_{ij} \tag{1.6}$$

The update rules are applied in turn until convergence. As a special case when all η_{ij} and μ_{ij} are equal, the learning direction is exactly the opposite direction of the gradient, *i.e.* the learning is done in the direction of the steepest descent.

The main problem of the gradient descent algorithms is the choice of the steps. Indeed, they should be small enough to reduce the cost function, but not too small for quick convergence. Moreover, the updates do not guarantee that coefficients are non-negative. To alleviate these problems, dynamic programming and back-tracking are often used to choose correct steps that decrease $\mathcal{C}$ and give non-negative coefficients. The non-negativity constraints can also be enforced using projected gradient algorithms that project $\mathbf{W}$ and $\mathbf{H}$ to the non-negative matrices orthant after each update.

Multiplicative updates

The *multiplicative updates* algorithms for NMF were introduced by Lee & Seung (1999, 2001) as an alternative to the additive updates algorithms such as gradient descent. The multiplicative updates are however derived from the gradient descent scheme, with judiciously chosen descent steps that lead to the following update rules:

$$h_{ij} \leftarrow h_{ij} \times \frac{\left[\mathbf{W}^T\mathbf{V}\right]_{ij}}{\left[\mathbf{W}^T\mathbf{W}\mathbf{H}\right]_{ij}} \qquad\qquad w_{ij} \leftarrow w_{ij} \times \frac{\left[\mathbf{V}\mathbf{H}^T\right]_{ij}}{\left[\mathbf{W}\mathbf{H}\mathbf{H}^T\right]_{ij}} \tag{1.7}$$

Like in gradient descent, these update rules are applied in turn until convergence. To avoid potential divisions by zero and negative values due to numerical imprecision, it is possible in practice to add a small constant ε to the numerator and denominator, or to use the non-linear operator $\max(x, \varepsilon)$.

Compared to gradient descent algorithms, multiplicative updates are easy to implement and guarantee the non-violation of the non-negativity constraints if $\mathbf{W}$ and $\mathbf{H}$ are initialized with non-negative coefficients. However, despite Lee & Seung's claims that multiplicative updates

converge to a local minimum of the cost function, several authors remarked that the proof shows that the cost function is non-increasing under these updates, which is slightly different from the convergence to a local minimum (*e.g.* Berry et al., 2007). Compared to alternating least squares algorithms, multiplicative updates are computationally more expensive and undergo slow convergence time. Finally, since a null coefficient in $\mathbf{W}$ or $\mathbf{H}$ remains null under the updates, the algorithm can easily get stuck into a poor local minimum.

1.2. Extensions

A flourishing literature exists about extensions to the standard NMF problem and algorithms detailed in Section 1.1. We do not seek to cover all the work here, but we rather try to give a general structured viewpoint of the possible ways to extend the standard problem. For further information, the interested reader can refer as a starting point to the outstanding work of Cichocki & Zdunek (2006) who provide a MATLAB toolbox with a wide range of common and new algorithms for extended NMF problems.

1.2.1. Cost functions

The standard NMF problem can be extended by using other cost functions than the cost defined in Equation 1.2. We overview some of these extensions here on.

Divergences

In a more general setting, the Frobenius norm can be replaced with a *divergence* $\mathcal{D}$, which generalizes the notion of distance, to define the following cost function:

$$\mathcal{C}(\mathbf{W}, \mathbf{H}) = \mathcal{D}(\mathbf{V} \parallel \mathbf{WH}) \tag{1.8}$$

Usually, the divergences used are *separable*, *i.e.* the divergence $\mathcal{D}$ between two matrices $\mathbf{A}$ and $\mathbf{B}$ is the sum of the element-wise divergences $d(a_{ij} \parallel b_{ij})$ where d is a given associated between-scalar divergence:

$$\mathcal{D}(\mathbf{A} \parallel \mathbf{B}) = \sum_{i,\,j} d(a_{ij} \parallel b_{ij}) \tag{1.9}$$

The cost defined in Equation 1.2 with the Frobenius norm is a special case of divergence. It is equivalent to the divergence $\mathcal{D}_E$ constructed with the between-scalar *Euclidean divergence* d_E defined as:

$$d_E(a \parallel b) = \frac{1}{2}(a-b)^2 \tag{1.10}$$

Another widespread divergence $\mathcal{D}_{KL}$ is constructed with the *generalized Kullback-Leibler divergence* d_{KL} defined as:

$$d_{KL}(a \parallel b) = a \log \frac{a}{b} - a + b \tag{1.11}$$

Recently, the divergence $\mathcal{D}_{IS}$ was used in the context of audio analysis by Févotte et al. (2009) who developed a Bayesian framework for NMF with this divergence. The divergence $\mathcal{D}_{IS}$ is associated with the between-scalar *Itakura-Saito divergence* defined as:

$$d_{IS}(a \parallel b) = \frac{a}{b} - \log \frac{a}{b} - 1 \tag{1.12}$$

Like the Euclidean divergence d_E, the divergences d_{KL} and d_{IS} are lower bounded by zero and vanish if and only if $a = b$, but are not distances since they are not symmetric. These divergences are special cases of wider classes of divergences that can also be used and for which algorithms have been developed: Bregman divergences (Dhillon & Sra, 2005), Csiszár's divergences (Cichocki et al., 2006) and Amari's α-divergences (Cichocki et al., 2008).

It is also possible to use a *weighted divergence* $\mathcal{D}^{(\Omega)}$ in which the element-wise divergences are weighted:

$$\mathcal{D}^{(\Omega)}(\mathbf{A} \parallel \mathbf{B}) = \sum_{i,\,j} \omega_{ij}\; d(a_{ij} \parallel b_{ij}) \tag{1.13}$$

where Ω is a weighting matrix with coefficients $\omega_{ij} > 0$. This can help to emphasize certain parts of the data to decompose. This idea was proposed by Guillamet et al. (2003), then developed by Blondel et al. (2005) and generalized by Dhillon & Sra (2005).

Penalties

One can extend the cost function further by adding penalties $\mathcal{J}_{\mathbf{W}}(\mathbf{W})$ and $\mathcal{J}_{\mathbf{H}}(\mathbf{H})$ on the structure of $\mathbf{W}$ and $\mathbf{H}$. The general form of the cost function then becomes:

$$\mathcal{C}(\mathbf{W}, \mathbf{H}) = \mathcal{D}(\mathbf{V} \parallel \mathbf{WH}) + \lambda_{\mathbf{W}} \mathcal{J}_{\mathbf{W}}(\mathbf{W}) + \lambda_{\mathbf{H}} \mathcal{J}_{\mathbf{H}}(\mathbf{H}) \tag{1.14}$$

where the parameters $\lambda_{\mathbf{W}}, \lambda_{\mathbf{H}} \geqslant 0$ are set by the user to control the trade-off between the reconstruction error and the penalties on $\mathbf{W}$ and $\mathbf{H}$.

The penalty terms are used to obtain particular regularizations of $\mathbf{W}$ and $\mathbf{H}$, and are often problem-dependent (*e.g.* temporally smooth encoding coefficients, orthogonal or localized basis vectors). The interested reader can refer to Buciu (2008) for an overview of several penalties used in the literature. However, regardless of the application, a particular kind of regularization is often desired, namely the *sparseness* of the factors $\mathbf{W}$ and $\mathbf{H}$. Hoyer (2002) was the first to propose a way to control sparseness of $\mathbf{W}$ and $\mathbf{H}$ using penalty terms. We develop the idea of controlling sparseness in Chapter 2.

1.2.2. Constraints

A second way to extend the standard problem is to enforce more constraints than non-negativity on the factors $\mathbf{W}$ and $\mathbf{H}$, and/or to relax the non-negativity constraints as described below.

Additional constraints

The most widespread additional constraint is to enforce either the columns of $\mathbf{W}$ or the rows of $\mathbf{H}$ to be of unit-norm. In general, the ℓ_1-norm or ℓ_2-norm is used. We have seen that the solution of standard NMF is not unique and that a diagonal rescaling of $\mathbf{W}$ and $\mathbf{H}$ gives another solution. The additional unit-norm constraint helps to avoid the ambiguity of the solution up to such a rescaling. The normalization can in most cases be done by an appropriate diagonal rescaling of $\mathbf{W}$ and $\mathbf{H}$ after each update of the matrix constrained to be of unit-norm[2].

A second interesting constraint was proposed by Hoyer (2004) to control the sparseness of the factors $\mathbf{W}$ and $\mathbf{H}$. The idea was later generalized by Heiler & Schnörr (2005a,b, 2006) to give more control on sparseness and include other constraints. We develop these approaches in Chapter 2.

A third additional constraint of interest is the situation where one wants to project $\mathbf{V}$ onto a basis $\mathbf{W}$ which is known and fixed. This can be seen as the additional constraint that $\mathbf{W}$ must be equal to a given fixed matrix. Throughout this document, we call this approach *non-negative decomposition* (ND). We discuss the use of ND in the context of incremental multi-source recognition in Section 1.3.

Relaxed constraints

The main philosophy of NMF is to build up the observed data additively. In the standard formulation, $\mathbf{V}$ is supposed to be non-negative and $\mathbf{W}$ constrained to be non-negative, but one may also want to allow the observed data to take negative values even if it is still built only additively. Ding et al. (2006) developed a mean of relaxing the non-negativity constraints to use real matrices $\mathbf{V}$ and $\mathbf{W}$. In the present study, we propose in Appendix A a way to use complex matrices $\mathbf{V}$ and $\mathbf{W}$ while keeping the non-negativity constraints on the encoding matrix $\mathbf{H}$.

Another constraint that can be relaxed is the rank of factorization r. In the standard problem, r is supposed to be less than $\min(n, m)$ so as to avoid trivial solutions. However, for some extensions of the standard problem, the addition of penalties or constraints can move the trivial solutions aside the global minima. One may then want to choose a rank of factorization greater than n or m. This is typically the case when sparseness penalties or constraints are added, or when $\mathbf{W}$ is fixed.

1.2.3. Models

A third way to extend the standard problem is to modify the linear model of matrix factorization given in Equation 1.1. Several modified models were proposed in the literature, but for the sake of concision we do not present all of them here. We introduce some models that are relevant

[2] In some particular extended problems however, a rescaling may modify the value of the cost or make additional constraints to be violated. Thus if the unit-norm constraint is desired, an *ad hoc* method must be used.

to the context of incremental sound recognition. The interested reader can refer to (Li & Ding, 2006) as reference for further information about extended NMF models.

In the context of signal processing, where the columns of $\mathbf{V}$ are often successive observations along time, Smaragdis (2004) remarked that NMF is well adapted to the analysis of static objects but not of time-varying objects. He thus proposed *non-negative matrix factor deconvolution* (NMFD) to overcome this issue. The idea was further developed by Mørup & Schmidt (2005) for sound analysis, with *non-negative matrix factor 2-D deconvolution* (NMF2D), to take not only time into consideration but also frequency shifts. Another model extension called *incremental non-negative matrix factorization* (INMF) was proposed very recently by Bucak & Günsel (2009) to overcome the off-line nature of standard NMF and consider the incoming signals as data streams.

Welling & Weber (2001) proposed to extend NMF to tensors for *non-negative tensor factorization* (NTF). NTF was used in numerous applications and two MATLAB toolboxes are available to perform NTF (Cichocki & Zdunek, 2006; Friedlander, 2006). In the context of sound analysis, NTF has been applied to source separation (FitzGerald et al., 2008; Ozerov & Févotte, 2009) where tensors are helpful to deal with multi-channel signals, time-varying objects, and express sound transformations such as frequency shifts.

1.3. Application to sound recognition

Due to the practical nature of NMF algorithms, they have been largely applied to problems in vision (*e.g.* face and object recognition), sound analysis (*e.g.* source separation, automatic transcription), biomedical data analysis, and text or email classification among others (Buciu, 2008). Overall, NMF has been mostly used in the general domain of pattern recognition. We introduce here the background of NMF in sound recognition, and discuss its applications within the context of the present work.

1.3.1. Background

In the context of sound analysis, NMF and its extensions have been used in several applications. In general, the matrix $\mathbf{V}$ is a time-frequency representation of the sound to analyze. The rows and columns represent respectively different frequencies and successive time-frames. As the columns $\mathbf{v}_j$ of $\mathbf{V}$ can be decomposed as $\mathbf{v}_j \approx \sum_i h_{ij}\mathbf{w}_i$ (see Section 1.1), the factorization is easy to interpret: each basis vector $\mathbf{w}_i$ contains a *spectral template*, and the encoding coefficients h_{ij} represent the *activation coefficients* of the i-th template $\mathbf{w}_i$ at the j-th time-frame.

Several sound representations have been used in the literature: the magnitude or power spectrum computed by different means (*e.g.* Fourier transform, constant-Q transform, instantaneous frequency estimation, filter-bank), the magnitude modulation spectrum, etc. One could argue that none of these representations is additive, and that anyway, there do not exist any additive non-negative representation of sounds (just consider two sinusoids in phase opposition). How-

ever, under certain conditions of phase independence between the different sources, some of the above-mentioned representations can approximatively be considered as additive. For example, Parry & Essa (2007) discussed the additivity of the spectrogram.

Concerning sound recognition, NMF has been widely used in polyphonic music transcription, where the sounds to recognize are notes (*e.g.* Smaragdis & Brown, 2003; Abdallah & Plumbley, 2004). Several problem-dependent extensions have been developed to this end such as enforcing an additional purely harmonic constraint (Raczyński et al., 2007), or constructing an harmonic/inharmonic model with adaptive tuning (Vincent et al., 2008). In these approaches, the pitches of the basis vectors are either known in advance (*e.g.* for the case of an additional purely harmonic constraint) or computed with a pitch detector. The attacks are detected with an *ad hoc* method (*e.g.* activation threshold, onset detector). These approaches rely in general on the off-line nature of NMF, some authors however used NMF in the context of incremental multi-source recognition, we review this approach here on.

1.3.2. Incremental multi-source recognition

To overcome the off-line nature of NMF and design an incremental system, the traditional approach is to use non-negative decomposition (ND, see Section 1.2), *i.e.* to constrain the basis vectors $\mathbf{W}$ to be equal to a given matrix of templates which is in general learned off-line prior to the factorization. These templates represent the different sources that the system should recognize (*e.g.* several notes of the same instrument or of different instruments). During the factorization, each incoming time-frame $\mathbf{v}_j$ can be decomposed onto these templates $\mathbf{v}_j \approx \mathbf{W}\mathbf{h}_j$ where $\mathbf{W}$ is kept constant and only the activation coefficients $\mathbf{h}_j$ at the current time-frame are updated. A NMF problem with the input matrix $\mathbf{v}_j$ and the templates $\mathbf{W}$ is thus solved at each time-frame j to give the current activation coefficients $\mathbf{h}_j$.

The first authors to use ND for incremental multi-source recognition are Sha & Saul (2005) who proposed a system to identify the presence and determine the pitch of one or more voices in real-time. Their method is based on an instantaneous frequency representation of the incoming signal which is decomposed in real-time on templates learned off-line. To make recognition more robust, a template for unvoiced speech is added. A threshold heuristic is used to determine the activated voiced templates, and unvoiced speech is detected with a classifier on $\mathbf{v}_j$ and $\mathbf{h}_j$. The system was later adapted for sight-reading evaluation of solo instrument by Cheng et al. (2008). For each note of the instrument, five templates are learned off-line using NMF with a power spectrum representation. The five templates represent the variations in timbre due to onset, offset and three dynamics that are taken into account.

Concerning automatic transcription, a similar system was used by Paulus & Virtanen (2005) for drum transcription. A five-band spectrum is used to represent the sounds and one template is learned off-line using NMF for each drum instrument. The detection of the instruments is done with an onset detector on the encoding coefficients. An incremental system for transcription of polyphonic music was proposed by Niedermayer (2008). The learning is also done using NMF, one template is learned for each note, and the system is tested on piano music.

A real-time alignment of audio to score system for polyphonic music was proposed by Cont (2006). The system is tested on piano music, using an instantaneous frequency representation. For each note of the piano, a template is learned off-line using NMF. More precisely, a training sample $\mathbf{V}^{(k)}$ of each note k is factorized into $\mathbf{W}^{(k)}\mathbf{H}^{(k)}$, with a rank of factorization $r = 2$ and the second column of $\mathbf{W}^{(k)}$ fixed as white noise. The first column of each matrix $\mathbf{W}^{(k)}$ is kept as a template in $\mathbf{W}$ for the real-time decomposition. To help generalization and robustness during the decomposition, a noise template is also added to $\mathbf{W}$, and a sparseness constraint similar to the one proposed by Hoyer (2004) is used. This approach was further developed by Cont et al. (2007) for real-time multi-pitch and multi-instrument recognition, using a modulation spectrum representation instead of an instantaneous frequency estimation.

1.3.3. Position of the present work

The present study is dedicated to incremental multi-source recognition using NMF. We are not only interested in multi-instrument and multi-pitch recognition for polyphonic music transcription, but also in the analysis of everyday auditory scenes with overlapping sound sources and background noise. The state-of-the-art presented above shows two general approaches to extend the unsupervised off-line NMF for incremental sound recognition.

The first one is to use NMF to factorize the signal to analyze, and then classify the basis vectors found with an *ad hoc* method (*e.g.* pitch detector for polyphonic music transcription). This approach may be adapted for incremental recognition using INMF (Bucak & Günsel, 2009), a recent incremental model extension of NMF (see Section 1.2). In the case of polyphonic music transcription, the main problem we see is that INMF may not be well-suited to extract notes as basis vectors. To overcome this issue, it would be interesting to combine INMF with the additional purely harmonic constraint proposed by Raczyński et al. (2007), or the harmonic/inharmonic model proposed by Vincent et al. (2008). We think that this approach could perform well for the transcription of a solo polyphonic instrument. However, it may be too limited for a generalization to multi-pitch and multi-instrument recognition, and for the recognition of everyday sound sources in a complex auditory scene.

The second approach is to use ND. Templates are learned off-line and the incoming signal is decomposed incrementally onto these templates which are kept fixed. This is the approach followed by all the incremental sound recognition systems we have found in the literature, and the approach we chose in the present work. The main problem we see is the robustness against noise or unknown sound events, and the power of generalization if the sound events to recognize are different from the sounds used for template learning. In the context of automatic classification of musical instrument segments, Benetos et al. (2006) proposed to include between-class information for better discrimination and generalization, and developed an incremental system combining NMF with a classifier. However, this approach cannot be easily adapted to a multi-source application such as ours.

We have chosen to address the problem of robustness and generalization from a theoretical viewpoint. As the control of sparseness in ND seems to be a predominant issue to us, we have focused on methods to obtain a *sparse non-negative decomposition* (SND). We have considered

extended NMF problems with penalties and additional constraints to control sparseness explicitly. As a result, these problems are more complex than the standard NMF problem, and we have developed specific optimization techniques to solve them.

We can establish a parallel between SND and the framework of non-negative sparse representations (*e.g.* Bruckstein et al., 2008). In the two approaches, the incoming signal is decomposed on a fixed basis with a few active coefficients. In SND however, we have no assumption on the rank of factorization r which can be less or greater than the number of variables n, and we do not seek a reconstructive but a discriminative decomposition. This means that we do not want to reconstruct the incoming signal perfectly with few basis vectors, but to find the few basis vectors that are present in the signal. In Chapter 2, we motivate the approach of SND, expose the theoretical framework and develop specific algorithms for controlling sparseness in this context.

We focus on NMF with the Euclidean distance. This helps us to consider sparsity within a geometrical framework and use optimization techniques such as second-order cone and convex quadratic programming. Using the Euclidean distance, we are also able to relax the non-negativity constraints on $\mathbf{V}$ and $\mathbf{W}$ to use complex representations as shown in Appendix A. These theoretical developments are validated on synthetic data, and applied to multi-pitch multi-instrument recognition and to the analysis of a complex auditory scene in Chapter 3. Some perspectives for future work, such as the use of other cost functions and extended NMF models, are discussed later.

2. Controlling Sparsity

Sparsity is an important way to reduce the space of plausible factorizations in NMF applied to complex problems. Concerning sound recognition, it has been shown that more rigorous and complex optimization schemes with explicit considerations for sparsity are necessary to deal with real-world problems (Cont et al., 2007). In this chapter, we address the issue of controlling sparsity from a theoretical viewpoint, with specific optimization techniques. In Section 2.1, we define sparsity and review several measures of sparseness. We also motivate the need for an explicit control of sparseness in NMF, and illustrate our points with a synthetic experiment. In Section 2.2, we introduce two optimization techniques to overcome the lack of control on sparseness in NMF, namely projected gradient optimization and second-order cone programming. In Section 2.3, we focus on controlling sparseness in non-negative decomposition (ND) to obtain a sparse non-negative decomposition (SND). We adapt the previous optimization methods and develop specific algorithms. In particular, we propose modified gradient descent algorithms and we introduce another optimization technique, namely convex quadratic programming. These theoretical developments are validated on synthetic data, and applied to multi-pitch multi-instrument recognition and to the analysis of a complex auditory scene in Chapter 3. In Appendix A, we also show how to relax the non-negativity constraints in the developed algorithms, so as to use complex matrices $\mathbf{V}$ and $\mathbf{W}$ in SND. The practical interest of such an extension is discussed later.

2.1. Preliminaries

2.1.1. Sparseness and its measures

The simplest definition of *sparseness* (or *sparsity*) is that a vector is *sparse* when most of its elements are null. However, there is no consensus on how sparseness should actually be defined and measured, with the result that numerous sparseness measures have been proposed. A comparative review of commonly used measures is done in Karvanen & Cichocki (2003). The idea is that a vector is sparse when it is not dense, *i.e.* much of its energy is packed into a few components, and so the different measures of sparseness try to quantify how much of the energy is packed into a few components. In the sequel, we employ the term norm in a wide sense, remembering that the ℓ_p-norms are not norms in the strict terms for $p < 1$.

Given a vector $\mathbf{x}$ of length n, the sparseness measure that corresponds to the naive definition

is based on the ℓ_0-norm defined as the number of non-zero elements:

$$\mathrm{sp}(\mathbf{x}) = \frac{\|\mathbf{x}\|_0}{n} = \frac{\mathrm{card}\,\{i\colon x_i \neq 0\}}{n} \tag{2.1}$$

This measure increases as $\mathbf{x}$ becomes less sparse, and is comprised between 0 for the null vector and 1 for any vector $\mathbf{x}$ with n non-null coefficients.

This basic measure is only applicable in noiseless situations. In practice, $\|\cdot\|_0$ is often replaced with $\|\cdot\|_{0,\,\varepsilon}$ to define:

$$\mathrm{sp}(\mathbf{x}) = \frac{\|\mathbf{x}\|_{0,\,\varepsilon}}{n} = \frac{\mathrm{card}\,\{i\colon |x_i| \geqslant \varepsilon\}}{n} \tag{2.2}$$

where $\varepsilon > 0$ is a threshold that takes into account the presence of noise. The choice of ε should depend on the noise variance which is not easy to estimate, so it is often chosen by trial-and-error in practice. Another problem is that this measure is non-differentiable in the interior of $\mathbb{R}^n_+$, denoted by $\mathbb{R}^n_{++}$ in the sequel, and thus cannot be used with several optimization techniques such as gradient descent. To overcome this problem, $\|\cdot\|_{0,\,\varepsilon}$ is often approximated with a hyperbolic tangent function:

$$\|\mathbf{x}\|_{0,\,\varepsilon} \approx \sum_i \tanh\left(|a x_i|^b\right) \tag{2.3}$$

where $a > 0$ and $b \geqslant 1$ are constant parameters chosen to approximate $\|\cdot\|_{0,\,\varepsilon}$ more or less smoothly.

Other sparseness measures differentiable on $\mathbb{R}^n_{++}$ can also be defined using the ℓ_p-norms for $0 < p \leqslant 1$, leading to:

$$\mathrm{sp}(\mathbf{x}) = \frac{\|\mathbf{x}\|_p^p}{n} = \frac{\sum_i |x_i|^p}{n} \tag{2.4}$$

In the context of NMF, Hoyer (2004) introduced an interesting sparseness measure which writes:

$$\mathrm{sp}(\mathbf{x}) = \frac{\sqrt{n} - \|\mathbf{x}\|_1/\|\mathbf{x}\|_2}{\sqrt{n}-1} = \frac{\sqrt{n} - \sum_i |x_i| \,/\sqrt{\sum_i {x_i}^2}}{\sqrt{n}-1} \tag{2.5}$$

From now on, $\mathrm{sp}(\mathbf{x})$ denotes this particular sparseness measure. On the contrary of the other measures, $\mathrm{sp}(\mathbf{x})$ increases as $\mathbf{x}$ becomes sparser and has the interesting property of being scale-independent (but thus is not defined for the null vector). Another interesting property comes from the following inequalities:

$$\frac{1}{\sqrt{n}}\|\mathbf{x}\|_1 \leqslant \|\mathbf{x}\|_2 \leqslant \|\mathbf{x}\|_1 \tag{2.6}$$

where the lower and upper bounds are respectively obtained if and only if all components of $\mathbf{x}$ are equal up to their signs, and if and only if $\mathbf{x}$ has a single non-null component. As a result, $\mathrm{sp}(\mathbf{x})$ is comprised between 0 for any vector with all components equal up to the signs, and 1 for any vector with a single non-null component, interpolating smoothly between the two bounds. Moreover, $\mathrm{sp}(\mathbf{x})$ is differentiable on $\mathbb{R}^n_{++}$ and the gradient coordinates are given by:

$$\frac{\partial \mathrm{sp}(\mathbf{x})}{\partial x_i} = \frac{1}{\sqrt{n}-1} \cdot \frac{x_i\|\mathbf{x}\|_1 - \|\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^3} \tag{2.7}$$

Like the other presented measures, $\mathrm{sp}(\mathbf{x})$ is *symmetric*, *i.e.* invariant against permutation of the components of $\mathbf{x}$, which is a desirable property for a sparseness measure. Another desirable property for sparseness measures is the *Schur-convexity* or the *Schur-concavity* (Marshall & Olkin, 1979), if the measure respectively increases or decreases as sparseness increases (Kreutz-Delgado & Rao, 1997). The ℓ_p-norms for $0 \leqslant p \leqslant 1$ are both symmetric and concave so they are Schur-concave, and $\mathrm{sp}(\mathbf{x})$ is Schur-convex (Heiler & Schnörr, 2005b). In short, a sparseness measure having this property ensures that for two vectors $\mathbf{x} \preccurlyeq \mathbf{y}$, where $\preccurlyeq$ is the *majorization* partial order, $\mathbf{x}$ is sparser than $\mathbf{y}$.

2.1.2. Motivations

One of the most appreciated properties of NMF is that it usually reveals a sparse and part-based decomposition of the data. However, this is rather a side-effect than a goal, and one cannot control sparseness explicitly in standard NMF. Several application-dependent ways to obtain sparse representations have been suggested in the literature. We focus here on more general approaches to control sparseness, where we do not have any other *a priori* (*e.g.* quasi-orthogonal or localized basis vectors) than the existence of a sparse representation of the data. Such approaches have been considered by Hoyer (2002, 2004); Heiler & Schnörr (2005a,b, 2006).

In the present study, we are mainly interested in obtaining a sparse decomposition on a learned basis $\mathbf{W}$, that is we seek a sparse encoding matrix $\mathbf{h}_j$ at each time-frame j. We are thus more interested in devising algorithms for the specific case of *sparse non-negative decomposition* (SND) rather than *sparse non-negative matrix factorization* (SNMF). We assume here that sparseness is necessary to obtain a relevant decomposition and devise a robust system capable of generalization. This assumption has already been discussed in the context of polyphonic music transcription with SND by Cont (2006); Cont et al. (2007). We develop its meaning and implications in the context of the present study here on.

For the specific problem of music transcription, the price to pay for the simplicity of the standard NMF formulation is the multiplicity of solutions for the factors $\mathbf{W}$ and $\mathbf{H}$. Assuming no mathematical independence over parts in $\mathbf{W}$, and considering simple time-frequency representations in $\mathbf{V}$, this amounts to common octave and harmonic errors during decoding and recognition. To overcome this problem, we can use the plausible assumption that the correct solution for a given input $\mathbf{V}$ uses the minimum number of available part representations in $\mathbf{W}$ to avoid common octave and harmonic errors. More specifically, this intuition about the structure of the desired results amounts to using a sparse coding scheme.

In a more general term, sound recognition paradigms can be seen as dimensionality reduction algorithms where the original high-dimensional space of sound representations (*e.g.* audio features) is being mapped to a much smaller space of desired classes. In such problems, it is generally desirable to obtain sparse results by nature. The issue becomes more serious if different classes are allowed to overlap over the geometry of the problem with presence of noise, causing uncertainties during recognition tasks. In such realistic cases, controlling sparsity of the solution could reduce the space of plausible results and increase the economy of class usage during reconstruction phases of any NMF formulation. Following an illustration of these

motivations, we introduce specific optimization techniques to control sparsity in NMF. We will concentrate on the particular case of SND later.

2.1.3. Illustration

We repeat here an experiment proposed by Paatero (1997) to illustrate the lack of control on sparsity even in lab situations. The experiment consists in creating a synthetic non-negative matrix $\mathbf{V} = \mathbf{WH} + |\mathbf{N}|$, by mixing sparse basis vectors $\mathbf{W}$ with encoding coefficients $\mathbf{H}$ and adding some positive noise $|\mathbf{N}|$. The synthetic matrix $\mathbf{V}$ is then analyzed with NMF, and the estimated basis vectors $\widehat{\mathbf{W}}$ and encoding coefficients $\widehat{\mathbf{H}}$ are compared with the ground truth $\mathbf{W}$ and $\mathbf{H}$ to see if NMF succeeded in recovering them.

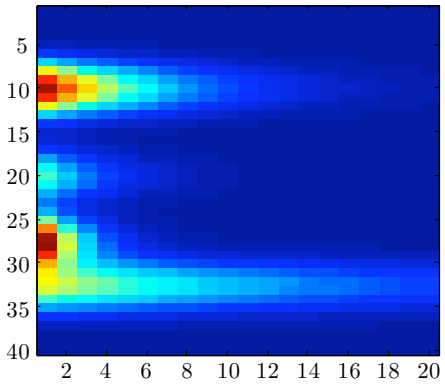
The data set is designed to resemble spectroscopic experiments in chemistry and physics. The input matrix $\mathbf{V} \in \mathbb{R}_+^{40\times 20}$ is created with four Gaussian distributions for the columns of the matrix $\mathbf{W} \in \mathbb{R}_+^{40\times 4}$, four exponential distributions for the rows of the matrix $\mathbf{H} \in \mathbb{R}_+^{4\times 20}$, and Gaussian noise $\mathbf{N} \sim \mathcal{N}(0, 0.01)$ for the matrix $\mathbf{N} \in \mathbb{R}_+^{40\times 20}$. The four Gaussian distributions are chosen such that $\text{sp}(\mathbf{w}_j) = 0.6877$ for all j. The ground truth $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{H}$ of this experiment are shown in Figures 2.1(a) and 2.1(b).

The optimization problem to solve is the same as the standard NMF problem 1.3, with the additional constraints that the estimated basis vectors have a ℓ_2-norm equal to 1, leading to the following formulation:

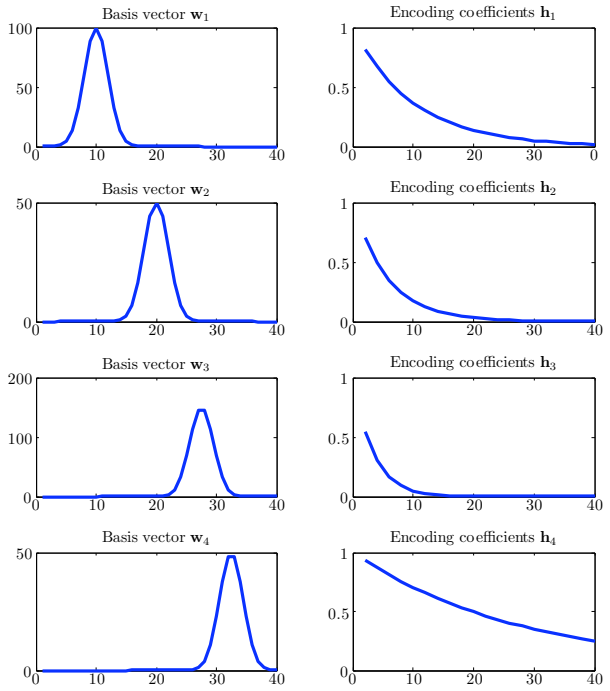
$$\begin{array}{ll} \text{Given} & \mathbf{V} \in \mathbb{R}_+^{n\times m},\ r \in \mathbf{N}^*\ \text{ s.t. }\ r < \min(n, m) \\ \text{minimize} & \dfrac{1}{2}\|\mathbf{V} - \mathbf{WH}\|_F^2\ \text{ w.r.t. }\ \mathbf{W}, \mathbf{H} \\ \text{subject to} & \mathbf{W} \in \mathbb{R}_+^{n\times r},\ \mathbf{H} \in \mathbb{R}_+^{r\times m},\ \|\mathbf{w}_j\|_2 = 1\ \ \forall j \end{array} \tag{2.8}$$

Algorithm 2.1 is used to solve this problem. In this algorithm and in the sequel, the matrix element-wise multiplication and division are respectively denoted by $\odot$ and $\oslash$. Algorithm 2.1 is the standard NMF algorithm with multiplicative updates (see Section 1.1) and an additional step of diagonal rescaling to ensure the additional constraints on $\mathbf{W}$. A rank of factorization $r = 4$ and a number of 50000 iterations are chosen for the analysis.

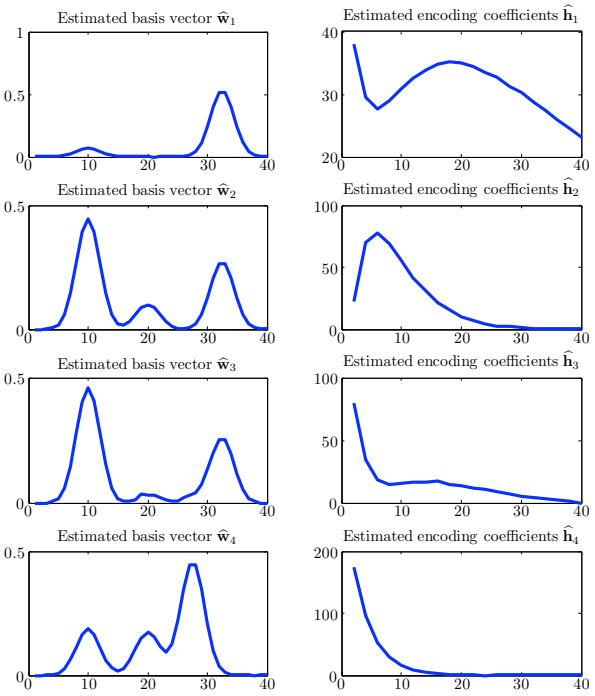
Results obtained from this factorization are shown in Figure 2.1(c). They reveal that NMF has not recovered correctly $\mathbf{W}$ and $\mathbf{H}$, and that the estimated basis vectors $\widehat{\mathbf{w}}_j$ are not sparse enough. By increasing the number of iterations or employing alternative algorithms such as alternating least squares algorithm (see Section 1.1), NMF will not recover the ground truth factors either. In Section 3.1, we repeat Paatero's experiment with two of the algorithms developed in this chapter, and show that an explicit control of sparsity can help to recover the ground truth factors, where NMF failed.



(a) Input matrix $\mathbf{V} = \mathbf{WH} + |\mathbf{N}|$.



(b) Ground truth matrices $\mathbf{W}$ and $\mathbf{H}$.



(c) Estimated matrices $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{H}}$.

Figure 2.1.: Paatero's experiment with non-negative matrix factorization. The input matrix $\mathbf{V}$ represented in Figure 2.1(a) is obtained by mixing $\mathbf{W}$ with $\mathbf{H}$ and adding a small amount of noise $\mathbf{N}$. The columns of $\mathbf{W}$ and the rows of $\mathbf{H}$ are represented in Figure 2.1(b). The matrix $\mathbf{V}$ is analyzed with Algorithm 2.1 which outputs the estimated basis vectors $\widehat{\mathbf{W}}$ and encoding coefficients $\widehat{\mathbf{H}}$ represented in Figure 2.1(c). The experiment shows that NMF did not succeed in recovering the ground truth matrices, and that the estimated basis vectors are not sparse enough.

Algorithm 2.1 NMF with multiplicative updates and diagonal rescaling.

Input: $\mathbf{V} \in \mathbb{R}_+^{n \times m}$, $r < \min(n, m)$
Output: $\mathbf{W}, \mathbf{H}$ that try to solve the optimization problem in Equation 2.8

Initialize $\mathbf{W}$ and $\mathbf{H}$ with strictly positive random values or with an *ad hoc* method
$\mathbf{W} \leftarrow \mathbf{WD}$ with $\mathbf{D} > 0$ diagonal s.t. the resulting $\mathbf{W}$ has columns of ℓ_2-norm equal to 1
$\mathbf{H} \leftarrow \mathbf{D}^{-1}\mathbf{H}$
repeat
  $\mathbf{H} \leftarrow \mathbf{H} \odot (\mathbf{W}^T\mathbf{V} + \varepsilon) \oslash (\mathbf{W}^T\mathbf{WH} + \varepsilon)$
  $\mathbf{W} \leftarrow \mathbf{W} \odot (\mathbf{VH}^T + \varepsilon) \oslash (\mathbf{WHH}^T + \varepsilon)$
  $\mathbf{W} \leftarrow \mathbf{WD}$ with $\mathbf{D} > 0$ diagonal s.t. the resulting $\mathbf{W}$ has columns of ℓ_2-norm equal to 1
  $\mathbf{H} \leftarrow \mathbf{D}^{-1}\mathbf{H}$
until convergence

2.2. Sparseness in non-negative matrix factorization

In this section, we present two optimization methods to include an explicit control of sparseness in NMF. The aim is not to propose an exhaustive range of algorithms for SNMF, but to introduce algorithms that are easily adaptable for the particular case of SND for which a more exhaustive range of specific algorithms are proposed in Section 2.3.

2.2.1. Projected gradient optimization

The *projected gradient* algorithms principle is to apply a gradient descent scheme (see Section 1.1) with an additional step of projection after each update. The updated variables are projected onto the *feasible set*, *i.e.* the set made of the points that verify the constraints. If the cost does not decrease, then back-tracking is used, the step size(s) is (are) reduced, and new updates of the variables are calculated and projected onto the feasible set until the cost decreases. Otherwise, the step size(s) can be slightly increased for the next iteration and the algorithm continues until convergence.

Projected gradient optimization has been used by Hoyer (2004) to control sparseness in NMF. Hoyer proposed to enforce additional constraints on $\mathbf{W}$ and/or $\mathbf{H}$, more precisely to enforce $\mathbf{W}$ and/or $\mathbf{H}$ to have a desired sparseness $\text{sp}(\mathbf{W}) = s_w$, $\text{sp}(\mathbf{H}^T) = s_h$, with $0 \leqslant s_w, s_h \leqslant 1$ chosen by the user. As a diagonal rescaling does not change the cost function nor the sparseness of $\mathbf{W}$ and $\mathbf{H}^T$, we can also constrain the columns of $\mathbf{W}$ to have a ℓ_2-norm equal to 1. The optimization problem then writes:

$$\begin{array}{ll} \text{Given} & \mathbf{V} \in \mathbb{R}_+^{n \times m},\ r \in \mathbf{N}^*\ \text{ s.t. }\ r < \min(n, m) \\ & s_w\ \text{ and/or }\ s_h\ \text{ s.t. }\ 0 \leqslant s_w, s_h \leqslant 1 \\ \text{minimize} & \frac{1}{2}\|\mathbf{V} - \mathbf{WH}\|_F^2\ \text{ w.r.t. }\ \mathbf{W}, \mathbf{H} \\ \text{subject to} & \mathbf{W} \in \mathbb{R}_+^{n \times r},\ \mathbf{H} \in \mathbb{R}_+^{r \times m},\ \|\mathbf{w}_j\|_2 = 1\ \ \forall j \\ & \text{sp}(\mathbf{W}) = s_w\ \text{ and/or }\ \text{sp}(\mathbf{H}^T) = s_h \end{array} \tag{2.9}$$

This problem can be solved with Algorithm 2.2 adapted from Hoyer (2004) for the unit-norm constraints on the columns of $\mathbf{W}$. The unit-norm constraint can be enforced by a diagonal rescaling step at the end of each iteration and does not need to be taken into account during the projected gradient scheme. If no sparseness constraint is enforced on either $\mathbf{W}$ or $\mathbf{H}$, then the standard multiplicative updates (Section 1.1) are applied respectively on $\mathbf{W}$ or $\mathbf{H}$. The step sizes μ and η are adaptive, they are chosen at each iteration by dynamic programming as explained above. They are both initialized at 1 before the iterations and multiplied by 1.2 at the end of each iteration. During back-tracking, the corresponding step size is divided by 2 until the updated factor makes the cost decrease.

Algorithm 2.2 SNMF with projected gradient optimization and diagonal rescaling.

Input: $\mathbf{V} \in \mathbb{R}_+^{n \times m}$, $r < \min(n, m)$, s_w and/or s_h s.t. $0 \leqslant s_w, s_h \leqslant 1$
Output: $\mathbf{W}, \mathbf{H}$ that try to solve the optimization problem in Equation 2.9

Initialize $\mathbf{W}$ and $\mathbf{H}$ with strictly positive random values or with an *ad hoc* method
if sp($\mathbf{W}$) is constrained **then**
  $\mathbf{W} \leftarrow \pi_{s_w}(\mathbf{W})$
end if
if sp($\mathbf{H}^T$) is constrained **then**
  $\mathbf{H}^T \leftarrow \pi_{s_h}(\mathbf{H}^T)$
end if
$\mathbf{W} \leftarrow \mathbf{WD}$ with $\mathbf{D} > 0$ diagonal s.t. the resulting $\mathbf{W}$ has columns of ℓ_2-norm equal to 1
$\mathbf{H} \leftarrow \mathbf{D}^{-1}\mathbf{H}$
repeat
  if sp($\mathbf{H}^T$) is constrained **then**
    $\mathbf{H}^T \leftarrow \pi_{s_h}\big(\mathbf{H}^T - \mu \cdot (\mathbf{H}^T\mathbf{W}^T - \mathbf{V}^T)\mathbf{W}\big)$ with μ chosen s.t. the cost decreases
  else
    $\mathbf{H} \leftarrow \mathbf{H} \odot (\mathbf{W}^T\mathbf{V} + \varepsilon) \oslash (\mathbf{W}^T\mathbf{W}\mathbf{H} + \varepsilon)$
  end if
  if sp($\mathbf{W}$) is constrained **then**
    $\mathbf{W} \leftarrow \pi_{s_w}\big(\mathbf{W} - \eta \cdot (\mathbf{W}\mathbf{H} - \mathbf{V})\mathbf{H}^T\big)$ with η chosen s.t. the cost decreases
  else
    $\mathbf{W} \leftarrow \mathbf{W} \odot (\mathbf{V}\mathbf{H}^T + \varepsilon) \oslash (\mathbf{W}\mathbf{H}\mathbf{H}^T + \varepsilon)$
  end if
  $\mathbf{W} \leftarrow \mathbf{WD}$ with $\mathbf{D} > 0$ diagonal s.t. the resulting $\mathbf{W}$ has columns of ℓ_2-norm equal to 1
  $\mathbf{H} \leftarrow \mathbf{D}^{-1}\mathbf{H}$
until convergence

In Algorithm 2.2, a projection operator π_s is used to project the factors onto the feasible set (excluding the unit-norm constraints). This operator projects each column of the input matrix on the intersection of the non-negative orthant and the cone of sparsity s. Hoyer proposes to do this projection at constant ℓ_2-norm, that is the projection has the same ℓ_2-norm as the input vector $\mathbf{x}$. The projection is thus equivalent to finding $\mathbf{y}$, the closest non-negative vector to $\mathbf{x}$ such that $\|\mathbf{y}\|_2 = \|\mathbf{x}\|_2 = l_2$ and $\|\mathbf{y}\|_1 = ((1-s)\sqrt{n} + s)\ l_2 = l_1$ so that $\mathbf{y}$ has the desired sparseness s. Algorithm 2.3 represents this projection scheme with the following procedures[1].

[1] Due to the symmetries of the ℓ_1 and ℓ_2-norms, the algorithm can be extended straightforward for a solution unconstrained in sign. It suffices to compute $\mathbf{y}$ for $|\mathbf{x}|$ and then re-enter the signs of $\mathbf{x}$ into $\mathbf{y}$.

The input vector $\mathbf{x}$ is first projected onto the hyperplane $\sum_i x_i = l_1$ to give $\mathbf{y}$. Then $\mathbf{y}$ is projected onto the l_2-hypersphere under the constraint that it stays in the hyperplane. In other terms, $\mathbf{y}$ is projected onto the intersection of the l_1-sum-constrained-hyperplane and the l_2-hypersphere, *i.e.* on a hypercircle whose center $\mathbf{c}$ is a vector with all components equal to $\frac{l_1}{n}$. The projection onto this hypercircle is done by moving radially outward from $\mathbf{c}$ to the hypersphere, what requires solving the following quadratic equation for $\alpha \geqslant 0$:

$$\|\mathbf{y} + \alpha(\mathbf{y} - \mathbf{c})\|_2^2 = {l_2}^2 \iff \alpha^2 \|\mathbf{y} - \mathbf{c}\|_2^2 + 2\alpha \langle \mathbf{y}, \mathbf{y} - \mathbf{c} \rangle + \|\mathbf{y}\|_2^2 - {l_2}^2 = 0 \tag{2.10}$$

If the projection $\mathbf{y}$ on the hypercircle is non-negative, then it is the solution. Otherwise, the components that are negative are set to zero, and the algorithm is repeated with these components fixed null. In principle, as many as n iterations may be needed, but in practice, the algorithm converges much faster.

Algorithm 2.3 Non-negative ℓ_1-ℓ_2 projection.

Input: $\mathbf{x} \in \mathbb{R}^n$, $0 \leqslant l_1 < \sqrt{n}\, l_2 \leqslant \sqrt{n}\, l_1$
Output: $\mathbf{y}$ the closest vector to $\mathbf{x}$ for $\|\cdot\|_2$ s.t. $\mathbf{y} \in \mathbb{R}^n_+$, $\|\mathbf{y}\|_1 = l_1$, $\|\mathbf{y}\|_2 = l_2$

$y_j \leftarrow x_j + \frac{l_1 - \sum_i x_i}{n} \quad \forall j$
$J \leftarrow \emptyset$
loop
  $c_j \leftarrow \frac{l_1}{n - \operatorname{card} J} \quad \forall j \notin J$
  $\mathbf{y} \leftarrow \mathbf{y} + \alpha(\mathbf{y} - \mathbf{c})$ where α is the positive root of Equation 2.10
  $J' \leftarrow \{j : y_j < 0\}$
  if $J' = \emptyset$ **then**
    return $\mathbf{y}$
  end if
  $J \leftarrow J \cup J'$
  $y_j \leftarrow 0 \quad \forall j \in J$
  $k \leftarrow \frac{l_1 - \|\mathbf{y}\|_1}{n - \operatorname{card} J}$
  $y_j \leftarrow y_j + k \quad \forall j \notin J$
end loop

2.2.2. Second-order cone programming

Second-order cone programming (SOCP) is a framework for *convex optimization* (Boyd & Vandenberghe, 2004) that addresses problems of the form:

$$\begin{aligned} &\text{minimize} && \mathbf{f}^T\mathbf{x} \text{ w.r.t. } \mathbf{x} \in \mathbb{R}^n \\ &\text{subject to} && \|\mathbf{A}_i\,\mathbf{x} + \mathbf{b}_i\|_2 \leqslant {\mathbf{c}_i}^T\mathbf{x} + \mathbf{d}_i \quad \forall i \end{aligned} \tag{2.11}$$

The constraints $\|\mathbf{A}_i\,\mathbf{x} + \mathbf{b}_i\|_2 \leqslant {\mathbf{c}_i}^T\mathbf{x} + \mathbf{d}_i$ are *second-order cone constraints*. They allow for example to model linear inequalities (*e.g.* non-negativity of $\mathbf{x}$) for $\mathbf{A}_i$ and $\mathbf{b}_i$ null. Geometrically, they require the affine functions $\binom{\mathbf{A}_i\,\mathbf{x}+\mathbf{b}_i}{{\mathbf{c}_i}^T\mathbf{x}+\mathbf{d}_i}$ to lie in the *second-order cone* of $\mathbb{R}^{n+1}$ defined as $\mathcal{L}^{n+1} = \left\{ \binom{\mathbf{x}}{t} : \|\mathbf{x}\|_2 \leqslant t \right\}$.

For example, the following non-negative least squares problem is a particular case of SOCP:

$$\arg\min_{\mathbf{W}\in\mathbb{R}_+^{n\times r}} \|\mathbf{V}-\mathbf{W}\mathbf{H}\|_F^2 \tag{2.12}$$

The standard formulation can be easily deduced once we have remarked that the problem can be reformulated using a second-order cone, the vectorization operator $\mathrm{vec}(\cdot)$ and the Kronecker product $\otimes$, leading to:

$$\begin{array}{ll} \text{minimize} & t \text{ w.r.t. } \mathbf{W},\, t \\ \text{subject to} & \mathbf{W}\in\mathbb{R}_+^{n\times r},\ \begin{pmatrix} \mathrm{vec}(\mathbf{V}) - (\mathbf{H}^T\otimes I)\,\mathrm{vec}(\mathbf{W}) \\ t \end{pmatrix} \in \mathcal{L}^{nm+1} \end{array} \tag{2.13}$$

From a computational viewpoint, second-order cone programs are convex and robust solvers are available. In our implementations we employed the open source SeDuMi 1.1.R3 solver (Sturm, 2001) in combination with the open source YALMIP R20090505 modeling language (Löfberg, 2004).

SOCP has been used by Heiler & Schnörr (2005a,b, 2006) in the context of SNMF, to generalize the approach of Hoyer (2004). The idea is to relax the hard sparseness constraints imposed on $\mathbf{W}$ and $\mathbf{H}^T$ by allowing them to lie between two sparsity cones, rather than projecting them onto one sparsity cone. As a diagonal rescaling does not change the cost function nor the sparseness of $\mathbf{W}$ and $\mathbf{H}^T$, we can also constrain the columns of $\mathbf{W}$ to have a ℓ_2-norm equal to 1. The optimization problem then becomes:

$$\begin{array}{ll} \text{Given} & \mathbf{V}\in\mathbb{R}_+^{n\times m},\ r\in\mathbf{N}^*\ \text{ s.t. }\ r<\min(n,m) \\ & 0\leqslant s_w^{min}<s_w^{max}\leqslant 1\ \text{ and }\ 0\leqslant s_h^{min}<s_h^{max}\leqslant 1 \\ \text{minimize} & \dfrac{1}{2}\|\mathbf{V}-\mathbf{W}\mathbf{H}\|_F^2\ \text{ w.r.t. }\ \mathbf{W},\mathbf{H} \\ \text{subject to} & \mathbf{W}\in\mathbb{R}_+^{n\times r},\ \mathbf{H}\in\mathbb{R}_+^{r\times m},\ \|\mathbf{w}_j\|_2=1\ \ \forall j \\ & s_w^{min}\leqslant \mathrm{sp}(\mathbf{W})\leqslant s_w^{max},\ s_h^{min}\leqslant \mathrm{sp}(\mathbf{H}^T)\leqslant s_h^{max} \end{array} \tag{2.14}$$

To simplify this formulation, let us consider the following convex sets parametrized by a sparsity parameter s:

$$\mathcal{C}(s)=\left\{\mathbf{x}\in\mathbb{R}^n\colon \begin{pmatrix} \mathbf{x} \\ \frac{1}{c_{n,s}}\cdot\mathbf{e}^T\mathbf{x} \end{pmatrix}\in\mathcal{L}^{n+1}\right\}\ \text{ with }\ c_{n,s}=(1-s)\sqrt{n}+s \tag{2.15}$$

where $\mathbf{e}$ denotes a column vector full of ones. The intersection of $\mathcal{C}(s)$ with the non-negative orthant $\mathbb{R}_+^n$ is exactly the set of the non-negative vectors with a sparseness less than s:

$$\mathbb{R}_+^n\cap\mathcal{C}(s)=\{\mathbf{x}\in\mathbb{R}_+^n\colon \mathrm{sp}(\mathbf{x})\leqslant s\} \tag{2.16}$$

To represent the feasible set (excluding the unit-norm constraints), we can combine the convex non-negativity constraint with the convex upper bound constraint $\mathbb{R}_+^n\cap\mathcal{C}(s^{max})$, and impose

the reverse-convex lower bound constraint by subsequently removing $\mathcal{C}(s^{min})$. We thus define the following sets:

$$\begin{aligned}\mathcal{C}_w(s) &= \left\{\mathbf{W} \in \mathbb{R}^{n\times r} \colon \mathbf{w}_j \in \mathcal{C}(s) \;\; \forall j \in \{1,\ldots,r\}\right\} \\ \mathcal{C}_h(s) &= \left\{\mathbf{H} \in \mathbb{R}^{r\times m} \colon \mathbf{h}_i{}^T \in \mathcal{C}(s) \;\; \forall i \in \{1,\ldots,r\}\right\}\end{aligned} \tag{2.17}$$

The problem 2.14 can now be re-written as follows:

$$\begin{aligned}\text{minimize} \quad & \frac{1}{2}\|\mathbf{V}-\mathbf{WH}\|_F^2 \;\text{ w.r.t. }\; \mathbf{W},\mathbf{H} \\ \text{subject to} \quad & \mathbf{W} \in \left(\mathbb{R}_+^{n\times r} \cap \mathcal{C}_w(s_w^{max})\right) \setminus \mathcal{C}_w(s_w^{min}),\; \|\mathbf{w}_j\|_2 = 1 \;\; \forall j \\ & \mathbf{H} \in \left(\mathbb{R}_+^{r\times m} \cap \mathcal{C}_h(s_h^{max})\right) \setminus \mathcal{C}_h(s_h^{min})\end{aligned} \tag{2.18}$$

This formulation explicitly introduces reverse-convex constraints for $\mathbf{W}$ and $\mathbf{H}$. In the standard NMF problem, we have seen that the individual optimization with respect to $\mathbf{W}$ or $\mathbf{H}$ is convex. The introduction of reverse-convex constraints however makes the problem more complex. As a result, not only the joint optimization with respect to $\mathbf{W}$ and $\mathbf{H}$ is not convex, but also individual optimizations with respect to $\mathbf{W}$ or $\mathbf{H}$.

To solve this optimization problem, Heiler & Schnörr (2005b, 2006) proposed an alternating minimization scheme similar to the alternating least squares algorithms in Section 1.1, but where the least squares problems are replaced with reverse-convex problems as shown in Algorithm 2.4. The unit-norm constraint can be enforced by a diagonal rescaling step at the end of each iteration.

Algorithm 2.4 SNMF with alternating reverse-convex minimization and diagonal rescaling.

Input: $\mathbf{V} \in \mathbb{R}_+^{n\times m},\; r < \min(n,m),\; 0 \leqslant s_w^{min} < s_w^{max} \leqslant 1 \;\text{ and }\; 0 \leqslant s_h^{min} < s_h^{max} \leqslant 1$
Output: $\mathbf{W},\mathbf{H}$ that try to solve the optimization problem in Equation 2.18

1: Initialize $\mathbf{W}$ with strictly positive random values or with an *ad hoc* method
2: Project $\mathbf{W}$ anywhere in $\left(\mathbb{R}_+^{n\times r} \cap \mathcal{C}_w(s_w^{max})\right) \setminus \mathcal{C}_w(s_w^{min})$
3: $\mathbf{W} \leftarrow \mathbf{WD}$ with $\mathbf{D} > 0$ diagonal s.t. the resulting $\mathbf{W}$ has columns of ℓ_2-norm equal to 1
4: **repeat**
5: $\quad \mathbf{H} \leftarrow \arg\min_{\mathbf{H}\in\mathcal{F}_h} \|\mathbf{V}-\mathbf{WH}\|_F^2$ where $\mathcal{F}_h = \left(\mathbb{R}_+^{r\times m} \cap \mathcal{C}_h(s_h^{max})\right) \setminus \mathcal{C}_h(s_h^{min})$
6: $\quad \mathbf{W} \leftarrow \arg\min_{\mathbf{W}\in\mathcal{F}_w} \|\mathbf{V}-\mathbf{WH}\|_F^2$ where $\mathcal{F}_w = \left(\mathbb{R}_+^{n\times r} \cap \mathcal{C}_w(s_w^{max})\right) \setminus \mathcal{C}_w(s_w^{min})$
7: $\quad \mathbf{W} \leftarrow \mathbf{WD}$ with $\mathbf{D} > 0$ diagonal s.t. the resulting $\mathbf{W}$ has columns of ℓ_2-norm equal to 1
8: $\quad \mathbf{H} \leftarrow \mathbf{D}^{-1}\mathbf{H}$
9: **until** convergence

A powerful framework for global optimization called *reverse-convex programming* (RCP) (Tuy, 1987) has been developed to solve reverse-convex problems such as the ones used for the updates of $\mathbf{W}$ and $\mathbf{H}$ in Algorithm 2.4. However, for the large-scale problems that are often considered in machine learning, more realistic methods that find a local solution need to be used. As the two reverse-convex problems in Algorithm 2.4 are equivalent up to a transposition, it suffices to focus on the problem for the update of $\mathbf{W}$:

$$\begin{aligned}\text{minimize} \quad & \frac{1}{2}\|\mathbf{V}-\mathbf{WH}\|_F^2 \;\text{ w.r.t. }\; \mathbf{W} \\ \text{subject to} \quad & \mathbf{W} \in \left(\mathbb{R}_+^{n\times r} \cap \mathcal{C}_w(s_w^{max})\right) \setminus \mathcal{C}_w(s_w^{min})\end{aligned} \tag{2.19}$$

Two approaches were proposed by Heiler & Schnörr to solve this specific problem, namely the *tangent plane approximation* algorithm and the *sparsity maximization* algorithm. Both rely on approximating the reverse-convex problem in Equation 2.19 by a sequence of convex problems that lead to a local optimal solution. However, the two algorithms have not the same convergence properties concerning the sequence of alternating optimizations in Algorithm 2.4. The sparsity maximization algorithm guarantees convergence to a local optimal solution of the problem 2.18, whereas the tangent plane approximation algorithm may oscillate in rare situations (but it does lead to a local optimal solution of the problem 2.18 when it converges). In our experiments, we implemented the both and obtained similar results. For the sake of concision, we only detail the tangent plane approximation algorithm for three reasons: (1) it is much faster than the sparsity maximization algorithm, (2) it can easily be modified using convex quadratic programming instead of SOCP as explained in Section 2.3, and (3) in the case of SND we are not concerned with the problem of potential oscillation due to alternating optimization since $\mathbf{W}$ is fixed.

The tangent plane approximation algorithm as shown in Algorithm 2.5 solves a sequence of SOCPs where the reverse-convex constraint is linearized, that is the min-sparsity cone is approximated by its tangent planes at some particular points. This algorithm can be discussed as follows.

As a first step, $\mathbf{W}$ is initialized by solving a SOCP without the min-sparsity constraints:

$$\begin{aligned} &\text{minimize} && t \text{ w.r.t. } \mathbf{W},\ t \\ &\text{subject to} && \mathbf{W} \in \mathbb{R}_+^{n\times r} \cap \mathcal{C}_w(s_w^{max}),\ \begin{pmatrix} \operatorname{vec}(\mathbf{V}) - (\mathbf{H}^T \otimes I)\operatorname{vec}(\mathbf{W}) \\ t \end{pmatrix} \in \mathcal{L}^{nm+1} \end{aligned} \tag{2.20}$$

If the columns of the resulting $\mathbf{W}$ all have a sparseness greater than s_w^{min}, then $\mathbf{W}$ is a global optimal solution of the problem 2.19 and the algorithm terminates. Otherwise, the indexes of the columns $\mathbf{w}_j$ that violate the min-sparsity constraint are added to the set J. The columns $\mathbf{w}_j$ for $j \in J$ are projected onto the min-sparsity cone, and the exterior normals $\mathbf{n}_j$ of the tangent planes at these projections $\mathbf{p}_j$ are computed. Another SOCP is solved with additional tangent plane constraints for the columns $\mathbf{w}_j$ with $j \in J$. This SOCP can be formulated as follows:

$$\begin{aligned} &\text{minimize} && t \text{ w.r.t. } \mathbf{W},\ t \\ &\text{subject to} && \mathbf{W} \in \mathbb{R}_+^{n\times r} \cap \mathcal{C}_w(s_w^{max}),\ \begin{pmatrix} \operatorname{vec}(\mathbf{V}) - (\mathbf{H}^T \otimes I)\operatorname{vec}(\mathbf{W}) \\ t \end{pmatrix} \in \mathcal{L}^{nm+1} \\ & && \langle \mathbf{n}_j, \mathbf{w}_j - \mathbf{p}_j \rangle \geqslant 0 \ \ \forall j \in J \end{aligned} \tag{2.21}$$

J is updated and a new SOCP is solved until $\mathbf{W}$ becomes feasible. Once $\mathbf{W}$ is feasible, it is a local solution of the problem in Equation 2.19. The process can then be repeated until convergence to give a better local optimal solution, but there is no guarantee to reach a global optimal solution. Throughout the algorithm, J keeps in memory the indexes for which $\mathbf{w}_j$ violated at least once the min-sparsity constraint along the updates of $\mathbf{W}$, and thus the indexes of the columns that must be constrained to lie outside the volume delimited by the min-sparsity cone.

The projections onto the min-sparsity cone are considered at constant ℓ_2-norm. They can be done with Algorithm 2.3, but Heiler & Schnörr also proposed an efficient way to approximate

them by an exponentiation and rescaling, that is each component x_i is replaced with ${x_i}^{\alpha}$ where $\alpha \geqslant 1$ is chosen such that $\mathrm{sp}(\mathbf{x}) = s_w^{min}$, and $\mathbf{x}$ is then rescaled to have unaffected ℓ_2-norm.

The exterior normal $\mathbf{n} = \nabla\mathcal{C}(s_w^{min})(\mathbf{p})$ of a tangent plane at $\mathbf{p} \in \mathcal{C}(s_w^{min})$ can be calculated analytically since $\mathcal{C}(s_w^{min})$ is parametrized by $f(\mathbf{x}) = 0$ with $f(\mathbf{x}) = \mathrm{sp}(\mathbf{x}) - s_w^{min}$. Thus, the direction of a normal at $\mathbf{p} \in \mathcal{C}(s_w^{min})$ is given by $\nabla f(\mathbf{p}) = \nabla\mathrm{sp}(\mathbf{p})$. As the gradient $\nabla\mathrm{sp}(\mathbf{p})$ points outwards from the volume delimited by the min-sparsity cone (because sparseness increases in its direction), $\mathbf{n}$ is given by $\nabla\mathrm{sp}$ (see Equation 2.7) evaluated at $\mathbf{p}$ and normalized.

Algorithm 2.5 Tangent plane approximation.

Input: $\mathbf{V} \in \mathbb{R}_+^{n\times m}$, $\mathbf{H} \in \mathbb{R}_+^{r\times m}$, $0 \leqslant s_w^{min} < s_w^{max} \leqslant 1$
Output: $\mathbf{W}$ local optimal solution of the RCP 2.19

Initialize $\mathbf{W}$ by solving the SOCP 2.20
$J \leftarrow \emptyset$
$J' \leftarrow \{j : \mathbf{w}_j \in \mathcal{C}(s_w^{min})\}$
if $J' = \emptyset$ **then**
  return $\mathbf{W}$
else
  repeat
    repeat
      $J \leftarrow J \cup J'$
      $\mathbf{p}_j = \pi_{s_w^{min}}(\mathbf{w}_j) \ \ \forall j \in J$
      $\mathbf{n}_j \leftarrow \nabla\mathcal{C}(s_w^{min})(\mathbf{p}_j) \ \ \forall j \in J$
      Update $\mathbf{W}$ by solving the SOCP 2.21
      $J' \leftarrow \{j : \mathbf{w}_j \in \mathcal{C}(s_w^{min})\}$
    until $J' = \emptyset$
  until convergence
end if

In our implementation of the tangent plane approximation algorithm, we had to add small constants ε in several inequalities of the SOCPs in Equation 2.20 and 2.21. This helped prevent from numerical imprecisions, mainly of the solutions output by the solver, that sometimes caused a violation of the non-negativity and sparseness constraints. This also helped staying in the interior (in a topological sense) of the non-negative orthants where $\nabla\mathrm{sp}$ is properly defined. Finally, the attentive reader could have remarked that removing the min-sparsity cones implies $\mathrm{sp} < s_{w/h}^{min}$ and not $\mathrm{sp} \leqslant s_{w/h}^{min}$ as considered in 2.14. However, because of the numerical imprecisions, there is a slight tolerance around the values of $s_{w/h}^{min}$ and $s_{w/h}^{max}$ and the strict inequality does not make sense in practice.

2.3. Sparse non-negative decomposition

In this section, we adapt the previously presented optimization methods and develop specific algorithms for the case of SND. We recall that SND aims at decomposing successive column vectors $\mathbf{v}_j$ on a fixed matrix $\mathbf{W}$ of learned templates, and where the encoding coefficients $\mathbf{h}_j$ are wanted as sparse as possible. To simplify the notations, we restrict without lack of generality

to the case where there is only one column vector $\mathbf{v}$ to decompose as $\mathbf{v} \approx \mathbf{W}\mathbf{h}$. We now present the optimization techniques we have developed to obtain this decomposition.

2.3.1. Gradient optimization

The projected gradient approach used for SNMF in Section 2.2 can be readily adapted for SND. The corresponding optimization problem can be formulated as follows:

$$\begin{array}{ll} \text{Given} & \mathbf{v} \in \mathbb{R}^n_+,\ \mathbf{W} \in \mathbb{R}^{n\times r}_+,\ 0 \leqslant s \leqslant 1 \\ \text{minimize} & \frac{1}{2}\|\mathbf{v} - \mathbf{W}\mathbf{h}\|_2^2 \ \text{ w.r.t. } \mathbf{h} \\ \text{subject to} & \mathbf{h} \in \mathbb{R}^r_+,\ \mathrm{sp}(\mathbf{h}) = s \end{array} \tag{2.22}$$

This problem can be solved by modifying Algorithm 2.2 accordingly, as shown in Algorithm 2.6.

Algorithm 2.6 SND with projected gradient optimization.

Input: $\mathbf{v} \in \mathbb{R}^n_+,\ \mathbf{W} \in \mathbb{R}^{n\times r}_+,\ 0 \leqslant s \leqslant 1$
Output: $\mathbf{h}$ that tries to solve the optimization problem in Equation 2.22

1: Initialize $\mathbf{h}$ with non-negative random values or with an *ad hoc* method
2: $\mathbf{h} \leftarrow \pi_s(\mathbf{h})$
3: **repeat**
4: $\quad \mathbf{h} \leftarrow \pi_s\big(\mathbf{h} - \mu \cdot \mathbf{W}^T(\mathbf{W}\mathbf{h} - \mathbf{v})\big)$ with μ chosen s.t. the cost decreases
5: **until** convergence

We found that other ways to control sparsity with gradient optimization could also be used for the particular case of SND. Since $\mathbf{W}$ is fixed, no unit-norm constraint is needed anymore. As a result, we can make use not only of scale-invariant terms but also of scale-dependent terms to create penalties $\mathcal{J}(\mathbf{h})$ and include them into the cost function. The optimization problem with a penalty can be formulated as follows:

$$\begin{array}{ll} \text{Given} & \mathbf{v} \in \mathbb{R}^n_+,\ \mathbf{W} \in \mathbb{R}^{n\times r}_+,\ \lambda \geqslant 0 \\ \text{minimize} & \frac{1}{2}\|\mathbf{v} - \mathbf{W}\mathbf{h}\|_2^2 + \lambda\, \mathcal{J}(\mathbf{h}) \ \text{ w.r.t. } \mathbf{h} \\ \text{subject to} & \mathbf{h} \in \mathbb{R}^r_+ \end{array} \tag{2.23}$$

For the penalties, the idea is to employ the sparseness measures differentiable on $\mathbb{R}^n_{++}$ that were presented in Section 2.1. In the present work, we propose three different penalties: (1) $\mathcal{J}(\mathbf{h}) = \sum_i \tanh\left(|ah_i|^b\right)$, (2) $\mathcal{J}(\mathbf{h}) = \|\mathbf{h}\|_p^p$, and (3) $\mathcal{J}(\mathbf{h}) = \left(\sigma\|\mathbf{h}\|_2 - \|\mathbf{h}\|_1\right)^2$ where $\sigma = (1-s)\sqrt{r} + s$ and $0 \leqslant s \leqslant 1$. The latter is not a sparseness measure but a measure on how close to s is $\mathrm{sp}(\mathbf{h})$. It is null if and only if $\mathrm{sp}(\mathbf{h}) = s$, and increases as $\mathrm{sp}(\mathbf{h})$ moves away from s. We did not use $\mathrm{sp}(\mathbf{h})$ with gradient optimization because we found a better suited optimization technique with convex quadratic programming to include this term as a penalty.

The problem in Equation 2.23 can be solved with Algorithm 2.6, which consists in a projected gradient scheme, with projection on the non-negative orthant using the $\max(\varepsilon, \cdot)$ operator, and where the step size is chosen by dynamic programming such that the cost decreases.

Algorithm 2.7 SND with projected gradient optimization and penalty.

Input: $\mathbf{v} \in \mathbb{R}_+^n,\ \mathbf{W} \in \mathbb{R}_+^{n \times r},\ \lambda \geqslant 0$
Output: $\mathbf{h}$ local optimal solution of the optimization problem in Equation 2.23
1: Initialize $\mathbf{h}$ with non-negative random values or with an *ad hoc* method
2: **repeat**
3: $\quad \mathbf{h} \leftarrow \max\left(\varepsilon, \mathbf{h} - \mu \cdot \left(\mathbf{W}^T(\mathbf{W}\mathbf{h} - \mathbf{v}) + \lambda \nabla \mathcal{J}(\mathbf{h})\right)\right)$ with μ chosen s.t. the cost decreases
4: **until** convergence

2.3.2. Convex quadratic programming

Convex quadratic programming (CQP) is a specific field of SOCP that addresses problems of the form:

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} \ \text{ w.r.t. } \ \mathbf{x} \in \mathbb{R}^n \\ &\text{subject to} && \mathbf{A}\mathbf{x} \leqslant \mathbf{b} \end{aligned} \tag{2.24}$$

where $\mathbf{P}$ is supposed to be positive-semidefinite. Such a problem is convex, and if $\mathbf{P}$ is positive-definite and the feasible set is non-empty, then the problem is strictly convex and thus has a unique global optimum.

For example, the following regularized non-negative least squares problem is a particular case of CQP:

$$\arg\min_{\mathbf{h} \in \mathbb{R}_+^r} \frac{1}{2}\|\mathbf{v} - \mathbf{W}\mathbf{h}\|_2^2 + \lambda_1\|\mathbf{h}\|_1 + \frac{\lambda_2}{2}\|\mathbf{h}\|_2^2 \tag{2.25}$$

where $\lambda_1, \lambda_2 \geqslant 0$ are regularization parameters. It can be reformulated as follows:

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\mathbf{h}^T(\mathbf{W}^T\mathbf{W} + \lambda_2\mathbf{I})\mathbf{h} + (\lambda_1\mathbf{e}^T - \mathbf{v}^T\mathbf{W})\mathbf{h} \ \text{ w.r.t. } \ \mathbf{h} \\ &\text{subject to} && \mathbf{h} \in \mathbb{R}_+^r \end{aligned} \tag{2.26}$$

The ℓ_1-norm regularization term is introduced to penalize less sparse vectors since it decreases with sparseness (see Section 2.1). The ℓ_2-norm regularization term is a particular case of *Tikhonov regularization* which is often used in CQP (Boyd & Vandenberghe, 2004) because it makes $\mathbf{W}^T\mathbf{W} + \lambda_2\mathbf{I}$ positive-definite for $\lambda_2 > 0$ and thus makes the problem strictly convex.

As suggested by Heiler & Schnörr (2006), we have modified the tangent plane approximation algorithm presented in Section 2.2 to replace the SOCPs with CQPs. From a computational viewpoint, CQPs can be solved more efficiently than SOCPs. In the implementation, we have used the `quadprog` function available in the MATLAB optimization toolbox. We have also found that the use of CQPs instead of SOCPs allows the introduction of additional penalty

terms to control sparseness more precisely. The problem we have considered is the following:

$$\begin{array}{ll} \text{Given} & \mathbf{V} \in \mathbb{R}_+^{n\times m},\ \mathbf{W} \in \mathbb{R}_+^{n\times r},\ \lambda_1, \lambda_2, \lambda_s \geqslant 0,\ 0 \leqslant s_{min} < s_{max} \leqslant 1 \\ \text{minimize} & \frac{1}{2}\|\mathbf{v} - \mathbf{W}\mathbf{h}\|_2^2 + \lambda_1\|\mathbf{h}\|_1 + \frac{\lambda_2}{2}\|\mathbf{h}\|_2^2 - \lambda_s \text{sp}(\mathbf{h}) \ \text{ w.r.t. } \mathbf{h} \\ \text{subject to} & \mathbf{h} \in \mathbb{R}_+^r,\ s_{min} \leqslant \text{sp}(\mathbf{h}) \leqslant s_{max} \end{array} \tag{2.27}$$

With the notations for SOCP defined in Section 2.2, this problem can be formulated as follows:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|\mathbf{v} - \mathbf{W}\mathbf{h}\|_2^2 + \lambda_1\|\mathbf{h}\|_1 + \frac{\lambda_2}{2}\|\mathbf{h}\|_2^2 - \lambda_s \text{sp}(\mathbf{h}) \ \text{ w.r.t. } \mathbf{h} \\ \text{subject to} & \mathbf{h} \in \left(\mathbb{R}_+^r \cap \mathcal{C}(s_{max})\right) \setminus \mathcal{C}(s_{min}) \end{array} \tag{2.28}$$

To solve this problem, we developed Algorithm 2.8 that we call *multiple tangent plane approximation* (MTPA). The idea is to solve a sequence of CQPs where both the reverse-convex and convex sparsity constraints are linearized, that is the min and max-sparsity cones are approximated by their tangent planes at some particular points. $\text{sp}(\mathbf{h})$ is also linearized, that is we approximate $\text{sp}(\mathbf{h})$ by its first-order Taylor expansion around a particular point.

The principle is the same as for the tangent plane approximation algorithm (see Section 2.2) except that the max-sparsity cone is linearized and that the cost function is extended to include regularization terms. We can use CQPs instead of SOCPs leading to advantages in terms of efficiency. Leaving out the three regularization terms in Equation 2.28, our algorithm coincides with the idea suggested by Heiler & Schnörr (2006) and can readily be used instead of the tangent plane approximation algorithm to solve the problem 2.18. Leaving out the ℓ_1 and ℓ_2-norm regularization terms and the sparseness constraints s_{min}, s_{max}, the problem becomes closely related to the relaxed problem proposed in Heiler & Schnörr (2005a), except that our cost function can be interpreted as a separable divergence with an additional penalty term and that we can use CQPs instead of SOCPs. The algorithm can be explained as follows.

As a first step, $\mathbf{h}$ is initialized by solving a CQP without the sparsity constraints and without the $\text{sp}(\mathbf{h})$ regularization term:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{h}^T(\mathbf{W}^T\mathbf{W} + \lambda_2\mathbf{I})\mathbf{h} + (\lambda_1\mathbf{e} - \mathbf{W}^T\mathbf{v})^T\mathbf{h} \ \text{ w.r.t. } \mathbf{h} \\ \text{subject to} & \mathbf{h} \in \mathbb{R}_+^r \end{array} \tag{2.29}$$

If $\mathbf{h}$ has a sparseness equal to s_{max} or ($\lambda_s = 0$ and $\mathbf{h}$ is feasible), then $\mathbf{h}$ is a global optimal solution of the problem in Equation 2.28 and the algorithm terminates. Otherwise, if $\lambda_s > 0$, we iteratively solve the following CQP until convergence, where $\mathbf{h}_0$ is the previous solution around which we linearize $\text{sp}(\mathbf{h}) \approx \text{sp}(\mathbf{h}_0) + \nabla\text{sp}(\mathbf{h}_0)^T(\mathbf{h} - \mathbf{h}_0)$:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{h}^T(\mathbf{W}^T\mathbf{W} + \lambda_2\mathbf{I})\mathbf{h} + \left(\lambda_1\mathbf{e} - \mathbf{W}^T\mathbf{v} - \lambda_s\nabla\text{sp}(\mathbf{h}_0)\right)^T\mathbf{h} \ \text{ w.r.t. } \mathbf{h} \\ \text{subject to} & \mathbf{h} \in \mathbb{R}_+^r \end{array} \tag{2.30}$$

If the resulting $\mathbf{h}$ lies between the min and the max-sparsity cones, then the algorithm terminates. Otherwise, we set the booleans b_{min} and b_{max} true respectively if $\text{sp}(\mathbf{h}) < s_{min}$ and

if $\text{sp}(\mathbf{h}) > s_{max}$ and update $\mathbf{h}$ by solving the following CQP with additional tangent plane constraints:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\mathbf{h}^T(\mathbf{W}^T\mathbf{W} + \lambda_2\mathbf{I})\mathbf{h} + (\lambda_1\mathbf{e} - \mathbf{W}^T\mathbf{v})\mathbf{h} \ \text{ w.r.t. } \ \mathbf{h} \\ \text{subject to} \quad & \mathbf{h} \in \mathbb{R}^r_+ \\ & \langle \mathbf{n}_{min}, \mathbf{h} - \mathbf{p}_{min} \rangle \geqslant 0 \ \text{ if } \ b_{min} \ \text{ is true} \\ & \left\langle \mathbf{n}^{(j)}_{max}, \mathbf{h} - \mathbf{p}^{(j)}_{max} \right\rangle \leqslant 0 \ \ \forall j \in [1, \dots, N] \end{aligned} \tag{2.31}$$

If $\lambda_s > 0$, a sequence of CQPs is then solved until convergence, with additional tangent plane constraints and approximation of the $\text{sp}(\mathbf{h})$ regularization term around the previous solution $\mathbf{h}_0$:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\mathbf{h}^T(\mathbf{W}^T\mathbf{W} + \lambda_2\mathbf{I})\mathbf{h} + \left(\lambda_1\mathbf{e} - \mathbf{W}^T\mathbf{v} - \lambda_s\nabla\text{sp}(\mathbf{h}_0)\right)^T\mathbf{h} \ \text{ w.r.t. } \ \mathbf{h} \\ \text{subject to} \quad & \mathbf{h} \in \mathbb{R}^r_+ \\ & \langle \mathbf{n}_{min}, \mathbf{h} - \mathbf{p}_{min} \rangle \geqslant 0 \ \text{ if } \ b_{min} \ \text{ is true} \\ & \left\langle \mathbf{n}^{(j)}_{max}, \mathbf{h} - \mathbf{p}^{(j)}_{max} \right\rangle \leqslant 0 \ \ \forall j \in [1, \dots, N] \end{aligned} \tag{2.32}$$

Once this sequence has converged, the booleans, the projections and the normals are updated, and the scheme is repeated until $\mathbf{h}$ becomes feasible. Once $\mathbf{h}$ is feasible, the process can be repeated until convergence to give a better solution. We hope to give a full proof of convergence of this algorithm in future work.

Throughout the algorithm, the booleans b_{min} and b_{max} respectively keep in memory whether $\mathbf{h}$ violated at least once the min-sparsity constraint and whether the current $\mathbf{h}$ violates the max-sparsity constraint, and thus whether the min-sparsity constraint need to be enforced and whether an additional max-tangent plane constraint need to be considered. N keeps in memory the number of times $\mathbf{h}$ violated the max-sparsity constraint, and thus the number of max-tangent planes constraints that are needed. The projections onto the min and max-sparsity cones, and the exterior normals can be computed as in the tangent plane approximation algorithm (see Section 2.2). We present applications of Algorithm 2.8 in Chapter 3.

Algorithm 2.8 SND with the multiple tangent plane approximation algorithm.

Input: $\mathbf{v} \in \mathbb{R}^n_+,\ \mathbf{W} \in \mathbb{R}^{n\times r}_+,\ \lambda_1, \lambda_2, \lambda_s \geqslant 0,\ 0 \leqslant s_{min} < s_{max} \leqslant 1$
Output: $\mathbf{h}$ that tries to solve the optimization problem in Equation 2.28

Initialize $\mathbf{h}$ by solving the CQP in Equation 2.29
if $\mathrm{sp}(\mathbf{h}) = s_{max}$ or ($\lambda_s = 0$ and $\mathbf{h}$ is feasible) **then**
  return $\mathbf{h}$
end if
if $\lambda_s > 0$ **then**
  repeat
    $\mathbf{h}_0 \leftarrow \mathbf{h}$
    Update $\mathbf{h}$ by solving the CQP in Equation 2.30
  until convergence
end if
if $s_{min} \leqslant \mathrm{sp}(\mathbf{h}) \leqslant s_{max}$ **then**
  return $\mathbf{h}$
end if
$b_{min} \leftarrow$ **false**
$N \leftarrow 0$
repeat
  repeat
    if $\mathrm{sp}(\mathbf{h}) < s_{min}$ **then**
      $b_{min} \leftarrow$ **true**
    end if
    if $\mathrm{sp}(\mathbf{h}) > s_{max}$ **then**
      $b_{max} \leftarrow$ **true**
    else
      $b_{max} \leftarrow$ **false**
    end if
    if b_{min} **then**
      $\mathbf{p}_{min} \leftarrow \pi_{s_{min}}(\mathbf{h})$
      $\mathbf{n}_{min} \leftarrow \nabla\mathcal{C}(s_{min})(\mathbf{p}_{min})$
    end if
    if b_{max} **then**
      $N \leftarrow N + 1$
      $\mathbf{p}^{(N)}_{max} \leftarrow \pi_{s_{max}}(\mathbf{h})$
      $\mathbf{n}^{(N)}_{max} \leftarrow \nabla\mathcal{C}(s_{max})(\mathbf{p}_{max})$
    end if
    Update $\mathbf{h}$ by solving the CQP in Equation 2.31
    if $\lambda_s > 0$ **then**
      repeat
        $\mathbf{h}_0 \leftarrow \mathbf{h}$
        Update $\mathbf{h}$ by solving the CQP in Equation 2.32
      until convergence
    end if
  until $\mathbf{h}$ is feasible
until convergence

3. Results

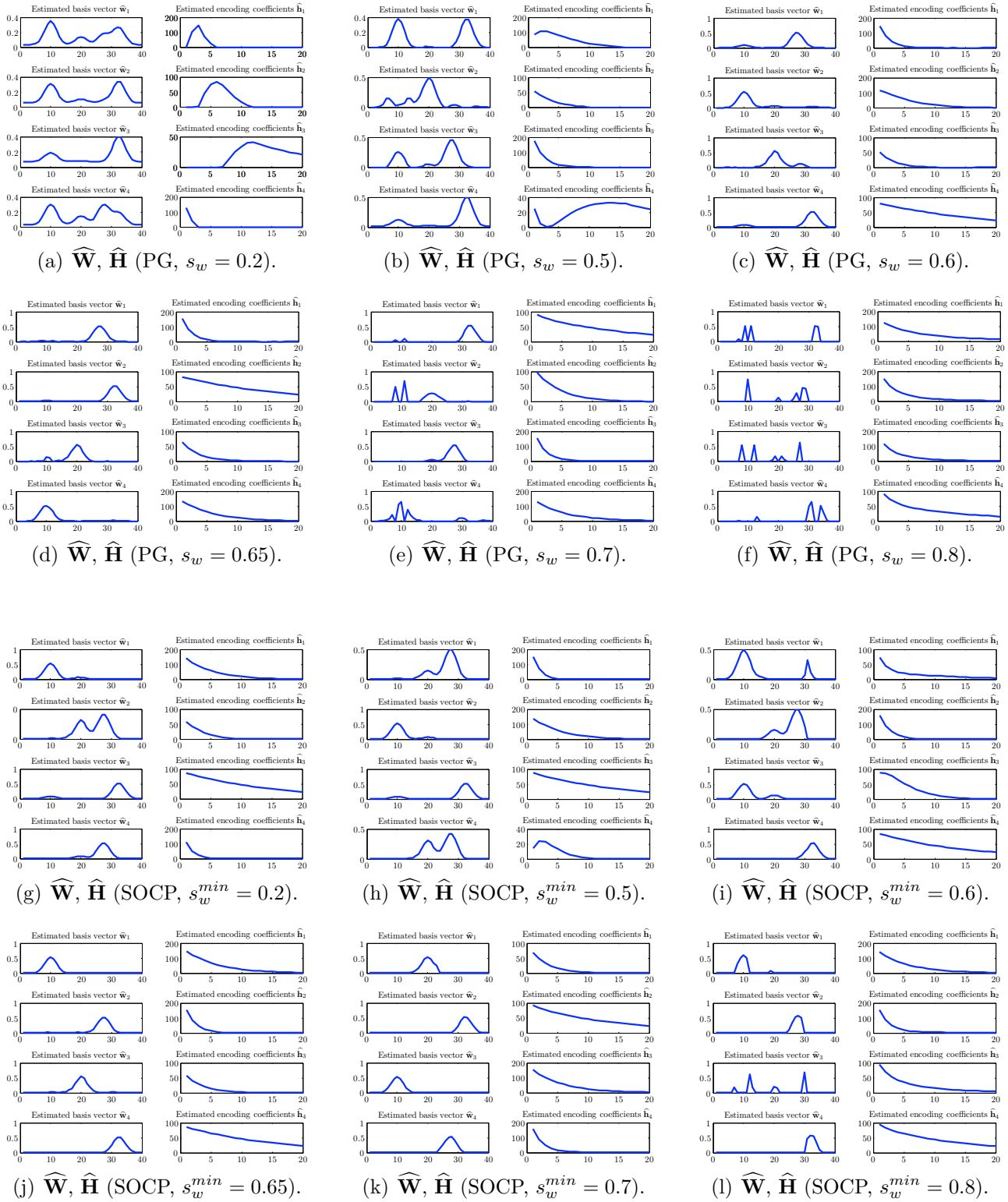
In Chapter 2, we have shown how to control sparsity in non-negative decomposition from a theoretical viewpoint. The need for such a control has also been motivated and illustrated in the context of the present work. Several algorithms for sparse non-negative decomposition (SND) have been developed. In this chapter, we now focus on applying the devised algorithms on a synthetic task for validation, and on more concrete and realistic tasks for illustration. In Section 3.1, the synthetic experiment proposed by Paatero (1997) is repeated to illustrate our points about the necessity of a control of sparseness in NMF. This experiment reveals that an explicit control of sparsity helps to overcome the limits of standard NMF. In Section 3.2, we concentrate on multi-pitch and multi-instrument recognition. We explain the learning step where templates are determined for each note of the instruments. The system is then evaluated on a real transcription task with recorded polyphonic music. In Section 3.3, the system is applied to the analysis of complex auditory scenes. Templates are learned for different everyday sound events, and a realistic auditory scene is created and analyzed.

3.1. Paatero's experiment

In this section, we repeat the experiment proposed by Paatero (1997) to illustrate the necessity of controlling sparseness even in lab situations. The experiment has been explained and performed with NMF in Section 2.1. It has underlined the real need for an explicit control of sparsity. The algorithms for SNMF proposed in Section 2.2 are here applied to the same task to assess their reliability compared to NMF.

The same data set for $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{H}$ is used (see Figures 2.1(a) and 2.1(b)). A rank of factorization $r = 4$ is chosen for the analysis. No sparseness constraint is enforced on $\widehat{\mathbf{H}}$, that is s_h is not specified for the projected gradient method, and we set $s_h^{min} = 0$, $s_h^{max} = 1$ for the SOCP method. No sparseness upper bound is enforced on $\widehat{\mathbf{W}}$ for the SOCP method, that is we set $s_w^{max} = 1$. We thus only consider one sparsity parameter, s_w for PG and s_w^{min} for SOCP, for which we tried the values 0.2, 0.5, 0.6, 0.65, 0.7, 0.8 (remembering that the sparseness of the ground truth basis vectors is 0.6877). For the two methods, the alternating updates of $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{H}}$ are stopped when the cost function does not change from more than a termination tolerance $\varepsilon = 10^{-7}$. The results are represented in Figure 3.1. They show that an *adhoc* control of sparseness allows to recover the ground truth factors, where standard NMF failed.

As the sparseness parameter used increases, the estimated basis vectors get sparser and closer to the ground truth, and for a value of 0.65, $\mathbf{W}$ and $\mathbf{H}$ are recovered quasi-exactly by the two methods (see Figures 3.1(d) and 3.1(j)).



(a) $\widehat{\mathbf{W}}$, $\widehat{\mathbf{H}}$ (PG, $s_w = 0.2$). (b) $\widehat{\mathbf{W}}$, $\widehat{\mathbf{H}}$ (PG, $s_w = 0.5$). (c) $\widehat{\mathbf{W}}$, $\widehat{\mathbf{H}}$ (PG, $s_w = 0.6$).

(d) $\widehat{\mathbf{W}}$, $\widehat{\mathbf{H}}$ (PG, $s_w = 0.65$). (e) $\widehat{\mathbf{W}}$, $\widehat{\mathbf{H}}$ (PG, $s_w = 0.7$). (f) $\widehat{\mathbf{W}}$, $\widehat{\mathbf{H}}$ (PG, $s_w = 0.8$).

(g) $\widehat{\mathbf{W}}$, $\widehat{\mathbf{H}}$ (SOCP, $s_w^{min} = 0.2$). (h) $\widehat{\mathbf{W}}$, $\widehat{\mathbf{H}}$ (SOCP, $s_w^{min} = 0.5$). (i) $\widehat{\mathbf{W}}$, $\widehat{\mathbf{H}}$ (SOCP, $s_w^{min} = 0.6$).

(j) $\widehat{\mathbf{W}}$, $\widehat{\mathbf{H}}$ (SOCP, $s_w^{min} = 0.65$). (k) $\widehat{\mathbf{W}}$, $\widehat{\mathbf{H}}$ (SOCP, $s_w^{min} = 0.7$). (l) $\widehat{\mathbf{W}}$, $\widehat{\mathbf{H}}$ (SOCP, $s_w^{min} = 0.8$).

Figure 3.1.: Paatero's experiment with sparse non-negative matrix factorization. This reveals that an explicit control of sparsity in NMF may help to recover the ground truth matrices $\mathbf{W}$ and $\mathbf{H}$, where NMF failed. The ground truth factors however are not always recovered, depending on the parameters.

On the contrary of the projected gradient method, the SOCP method seems to already lead to a sparse $\widehat{\mathbf{W}}$ with small values 0.2 or 0.5 of the sparsity parameter (see Figures 3.1(a), 3.1(b), 3.1(g) and 3.1(h)). This can be explained easily: the PG method enforces the sparseness by projecting onto the sparsity cone even if factors with a greater sparseness could explain the data better, whereas the SOCP method is more flexible and allows the factors to have a sparseness greater than a specified lower bound.

With a sparsity parameter of 0.6, which is slightly less than the sparseness of the ground truth basis vectors, the projected gradient method seems to recover better the basis vectors than the SOCP method (see Figures 3.1(c) and 3.1(i)). We ran the algorithms several times and observed that the projected gradient method was quite robust and not sensitive to the initialization, despite the SOCP method. The same phenomenon is also observed with a sparsity parameter of 0.65, which is very close to the sparseness of the ground truth basis vectors. For that value, the projected gradient method appeared quite robust, whereas the SOCP method sometimes did not recover the basis vectors, as shown in Figure 3.2 which represents two different runs of the SOCP method.

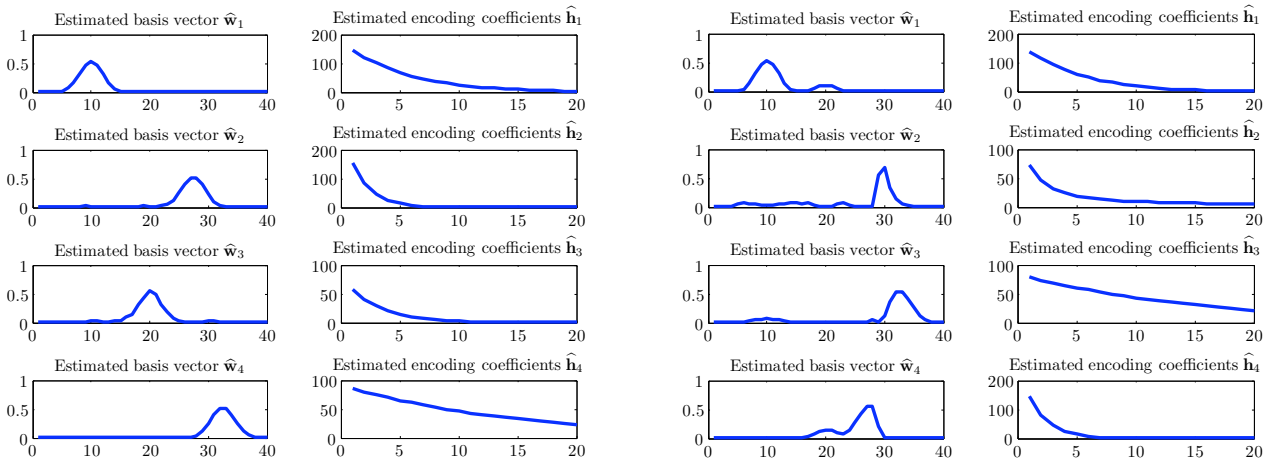


Figure 3.2.: Two runs of the SOCP method on Paatero's data. This shows that for a sparsity parameter of 0.65, which is close to the sparseness of the ground truth basis vectors, the SOCP method does not always recover the ground truth factors.

As the sparsity parameter increases and exceeds the sparseness of the ground truth basis vectors for the values 0.7 or 0.8, the SOCP method seems more robust than the projected gradient method which completely fails in recovering the basis vectors (see Figures 3.1(e), 3.1(f), 3.1(k) and 3.1(l)). We ran the algorithms several times and observed that the projected method was much more sensitive to the initialization than the SOCP method in that case.

Overall, these results confirm our assumption that controlling sparseness is necessary to obtain a relevant factorization. Sparseness sometimes also helps robustness of the algorithm in the presence of noise. However, the experiment shows that the algorithms can be sensible to the parameters and the initialization, so that the ground truth basis vectors are not always recovered. This phenomenon has also been observed by Heiler & Schnörr (2006).

Our results suggest that it could be interesting to use the projected gradient method as a way to initialize $\mathbf{W}$ in the SOCP method. This would help in the general and practical cases where

the sparseness of the factors is not known in advance. Alternative methods such as replacing the additional constraints with penalties could also be tried (Hoyer, 2002; Heiler & Schnörr, 2005a). These approaches need to be investigated further in the context of SNMF, but we now concentrate on applications of SND.

3.2. Multi-pitch and multi-instrument recognition

3.2.1. Introduction

The problem of multiple-pitch detection has been largely investigated for speech and music signals. The literature contains a wide variety of methods to address this problem (de Cheveigné, 2006). NMF has already been used in this context, either with off-line approaches (Smaragdis & Brown, 2003; Abdallah & Plumbley, 2004; Raczyński et al., 2007; Vincent et al., 2008; Bertin et al., 2009) or with incremental approaches (Cont, 2006; Niedermayer, 2008).

In this section, we are not only interested in incremental multi-pitch detection, but also in multi-instrument recognition. This means that we want to detect both the pitches and the instruments that played the corresponding notes. Our approach follows the work initiated by Cont et al. (2007) for multi-pitch and multi-instrument recognition with SND. The idea is to use a recently introduced signal representation scheme based on the *modulation spectrum* (Sukittanon et al., 2004). We explain the interest of this representation here on.

In the context of instrument classification, features are in general extracted from short-term analysis windows. However, these features do not efficiently capture longer term variations of the signals. Such variations are yet important since they can represent discriminative characteristics of the sound classes. For example, it has been shown that the phase coupling properties of instruments are characteristic of the instrument (Dubnov & Rodet, 2003). The modulation spectrum representation contains such longer term information about the signal, more precisely information about the time variation of the spectrum.

Modulation spectrum is a joint frequency representation $P(\eta, \omega)$, *i.e.* a two-dimensional representation of the modulation frequency η and the acoustic frequency ω. Several techniques can be used to obtain a modulation spectrum representation (*e.g.* Atlas & Janssen, 2005; Sukittanon et al., 2005). We employ here an estimation of the modulation spectrum based on two successive transformations: (1) a magnitude spectrogram calculated with a short-time Fourier transform to give a joint time-frequency representation $S(t, \omega)$ of the input signal, (2) a magnitude spectrum, obtained with a Fourier transform, applied along the time dimension of the spectrogram $S(t, \omega)$ to estimate the joint frequency representation $P(\eta, \omega)$. The interpretation of the modulation spectrum $P(\eta, \omega)$ is straightforward. The values for $\eta = 0$ correspond to stationary information about the signal, whereas the values for $\eta > 0$ correspond to non-stationary information. In the particular case of instruments, the representation $P(\eta, \omega)$ contains thus information about the degree of modulation of the partials.

As the modulation spectrum is non-negative and gives information about the long-term varia-

tions of the signal, it is an interesting representation to consider in the context of multi-pitch and multi-instrument recognition with SND. Furthermore, Atlas & Janssen (2005) discussed the additivity of the modulation spectrum, and Cont et al. (2007) provided an illustrative example to show the superposition of the modulation spectra of two instruments playing together. For these reasons, we adopt the modulation spectrum as a representation front-end for our system. We now show how modulation spectrum templates are learned for the incremental decomposition with SND.

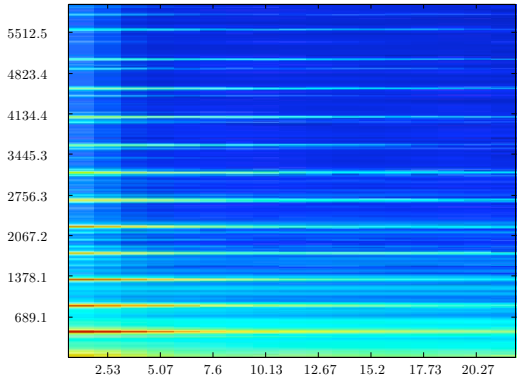
3.2.2. Learning the templates

The step of learning templates for each note of the instruments is important for the system to work properly. Indeed, the templates should: (1) represent well the sounds and what is invariant in the sounds so as to help generalization, and (2) allow for a good discrimination between the different classes of sounds so as to help robustness. Without these two properties, no matter how sophisticated is the incremental decomposition system, it will fail to deal with real-world problems.

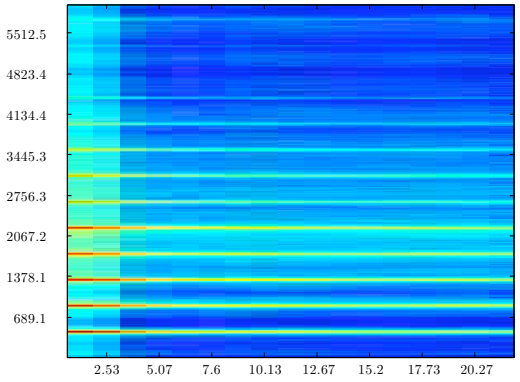
We have used here the templates learned with the method and the analysis parameters employed by Cont et al. (2007) to provide a fair comparison of his decomposition algorithm with ours. Moreover, his method has proven to provide relevant templates as illustrated in the paper. Hence we do not seek to develop new techniques for template learning in the present study. We discuss however other possible approaches for the learning step later.

To summarize the learning step employed by Cont et al. (2007), it consists in learning one modulation spectrum template k for each note of each instrument. This template is learned by applying NMF on a training sample of the corresponding note. More precisely, the training sample is decomposed into successive time-frames in each of which the modulation spectrum is computed. The modulation spectra in the different frames are vectorized to give column vectors that are then stacked in a matrix $\mathbf{V}^{(k)}$. NMF with the generalized Kullback-Leibler divergence is applied to $\mathbf{V}^{(k)}$ which is factorized into $\mathbf{W}^{(k)}\mathbf{H}^{(k)}$, with a rank of factorization $r = 2$ and the second column of $\mathbf{W}^{(k)}$ fixed as white noise.

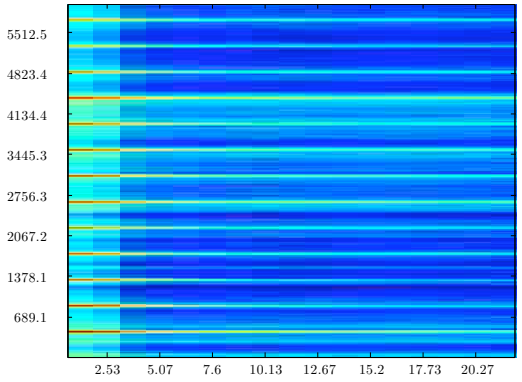
The templates are computed for the piano, the flute and the violin. The samples for the piano are taken from the RWC database (Goto et al., 2002), and the samples for the flute and the violin are taken from the SOL database (Ballet et al., 1999). Three templates obtained with the described learning scheme for the A4 of the piano, the flute and the violin are shown in Figure 3.3. This reveals that NMF has succeeded in learning the modulation characteristics of the three instruments, and that the templates contain between-instrument discriminative information which is of interest for the incremental decomposition system.



(a) Template for the A4 of the piano.



(b) Template for the A4 of the flute.



(c) Template for the A4 of the violin.

Figure 3.3.: Learned templates for the A4 of three different instruments. Modulation spectrum templates for the A4 of the piano, the flute and the violin are learned with the method proposed in Cont et al. (2007). The x-axis represents the modulation frequency η in hertz and the y-axis the acoustic frequency ω in hertz. The three modulation spectrum templates are different, meaning that the three instruments have their own modulation characteristics, and that the templates contain discriminative information.

3.2.3. Evaluation on recorded music

In this section, we subjectively evaluate the proposed system on a multi-pitch and multi-instrument recognition task, that is we seek a transcription in form of a piano-roll presentation. We focus on one of the algorithms developed in Section 2.3, Algorithm 2.8, in comparison to simple gradient optimization techniques as proposed in (Cont et al., 2007).

Evaluation of multi-pitch and multi-instrument recognition systems are often presented as two different tasks and are generally difficult. The general procedure often consists in evaluating the output of the system against reference annotated music. Some databases containing annotated music does exist, but are in the best case dedicated to polyphonic mono-instrument music (often piano music), such that several systems for multi-instrument recognition are evaluated without the pitch information (*e.g.* Essid et al., 2006; Kitahara et al., 2006). Other systems are evaluated with a manual mixing of single note recordings of different instruments (*e.g.* Livshin & Rodet, 2006).

We aim at evaluating results against acoustic recordings despite the difficulties of the approach. To compare Algorithm 2.8 with the algorithm proposed by Cont et al. (2007), we process in the same way as in the paper, and use the same analysis parameters. We demonstrate the performances of the systems with a subjective evaluation on a short musical excerpt shown in a piano roll representation in Figure 3.4, and leave an objective evaluation for future work.

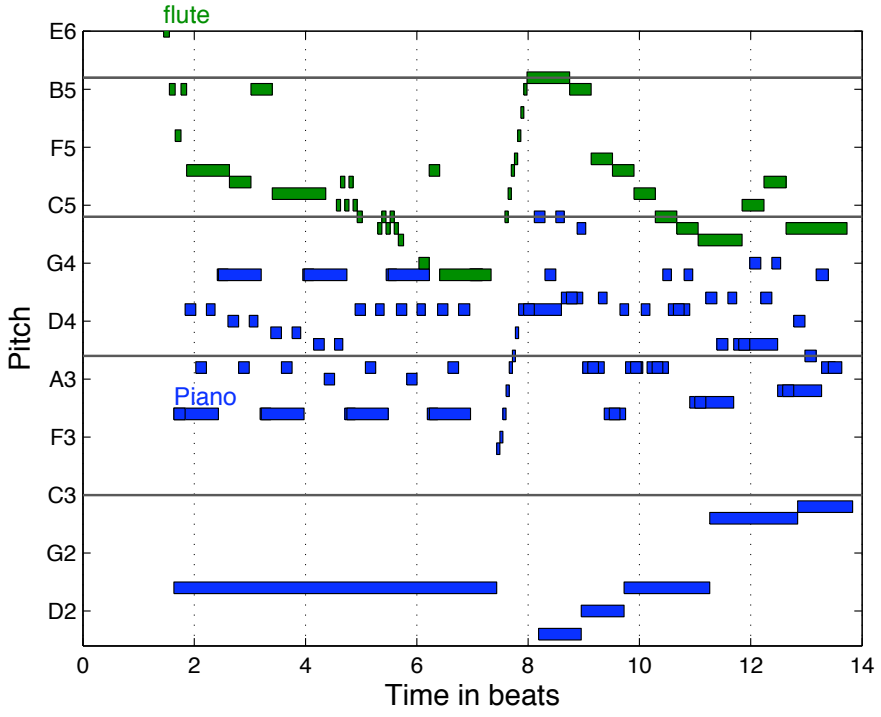


Figure 3.4.: Piano roll of the MIDI score from Poulenc's *Sonata for Flute and Piano*.

Figures 3.5(a) and 3.5(b) show performances of the algorithm in (Cont et al., 2007) and Algorithm 2.8 respectively on a recording of a musical phrase from Poulenc's *Sonata for Flute and Piano*. A piano roll representation of the MIDI score of the same phrase is represented

in Figure 3.4 for visual reference. The templates used for the incremental decomposition are computed as explained previously. A noise template is also added to absorb not only transients and non-harmonic structures, but also the background noise and the reverberation due to real-world recording situation. For the decomposition with Algorithm 2.8, we have only considered the two parameters $\lambda_1 = 1$ and $\lambda_2 = 0.01$, that is s_{min}, s_{max} and λ_s are respectively set to 0, 1 and 0. Changing other parameters slightly would not alter the results of our algorithm.

Comparing Figures 3.5(a) and 3.5(b) alone clearly shows that Algorithm 2.8 has less tendency to use other templates than the active ones in the reference. This is a direct consequence of (1) the ability of Algorithm 2.8 to control the degree of sparsity freely within a given range in contrast to (Cont et al., 2007) where sparsity is a rigid parameter, and (2) more efficient optimization techniques in Algorithm 2.8 compared to (Cont et al., 2007) as discussed previously.

3.3. Analysis of complex auditory scenes

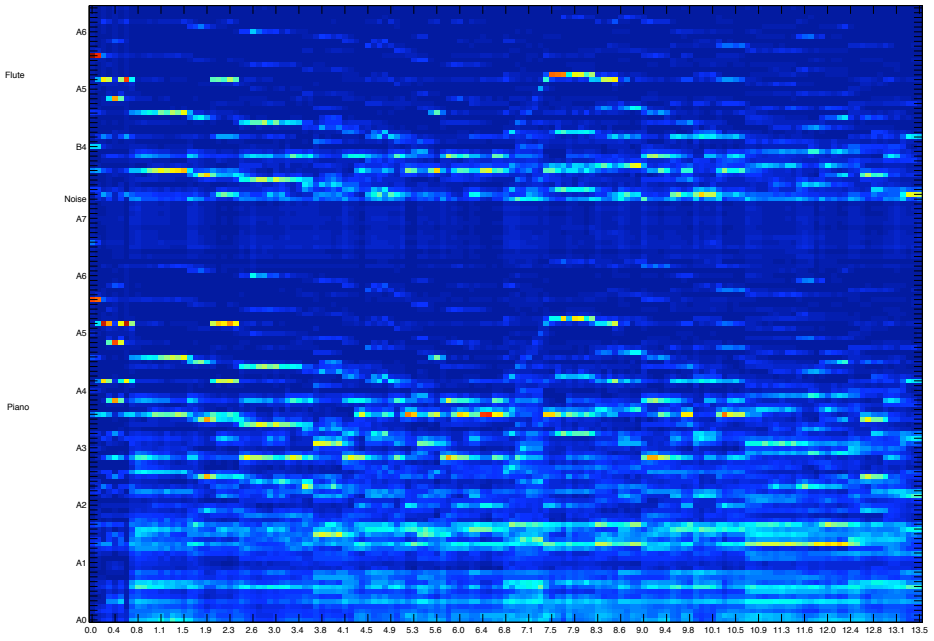
3.3.1. Introduction

Analysis of complex auditory scenes has received much attention mainly with the development of *computational auditory scene analysis* (CASA) (Wang & Brown, 2006). The field of CASA is wide and deals with various real-world problems such as source separation, automatic polyphonic music transcription, recognition of speech in noisy environments, environmental sound recognition in realistic scenarios among others.

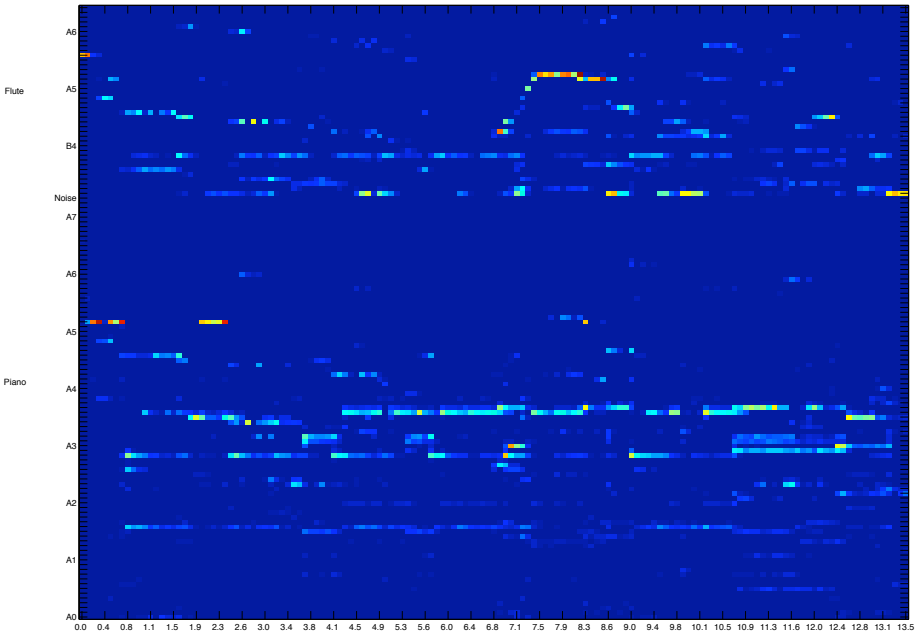
In this section, we showcase the problem of environmental sound recognition in complex auditory scenes, using sparse non-negative decomposition techniques. To the best of our knowledge there is no application of NMF to the analysis of complex auditory scenes in the literature. Our goal here is not to propose a universal framework to address the problems of CASA but rather to demonstrate capabilities of simple sparse non-negative methods for such tasks.

We concentrate on a basic example to assess the reliability of the proposed system. The idea is to create a realistic scenario with several sound sources and background noise. The proposed system is able to deal with complex scenes where the sound sources overlap. We choose however to begin with a less complex scene where the sound sources do not overlap, but where we still consider background noise. This should provide a preliminary example to test the system, and assess its drawbacks. Future improvements of the system for the analysis of more complex scenes will be addressed later.

The context of this experiment is more or less the same as in Section 3.2, by employing templates learned off-line in a decomposition setup, but with different representation front-ends. In the context of environmental sound recognition, as for the task of instrument recognition, features are in general extracted from short-term analysis windows. For a recent review of the common methods for environmental sound classification, we refer the reader to (Chu et al., 2009). We have used a simple time-frequency representation with the short-time Fourier transform. We discuss later the possibility to use other sound representations.



(a) Activation coefficients obtained with the algorithm in (Cont et al., 2007).



(b) Activation coefficients obtained with Algorithm 2.8.

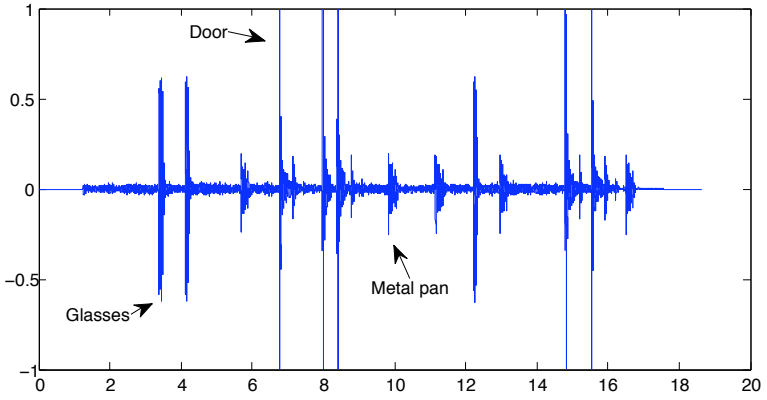
Figure 3.5.: Subjective evaluation with recorded music. The system with Algorithm 2.8 is evaluated and compared to the algorithm in (Cont et al., 2007) on a real recording of a musical phrase from Poulenc's *Sonata for Flute and Piano*. The results obtained with these two algorithms are shown respectively on Figures 3.5(a) and 3.5(b). These figures represent the respective encoding matrices $\mathbf{H}$ obtained during the decomposition. The x-axis corresponds to the time while the y-axis corresponds to the pitches of the instruments (flute on top and piano at the bottom, separated by the noise template) sorted in ascending order.

3.3.2. Validation of the system

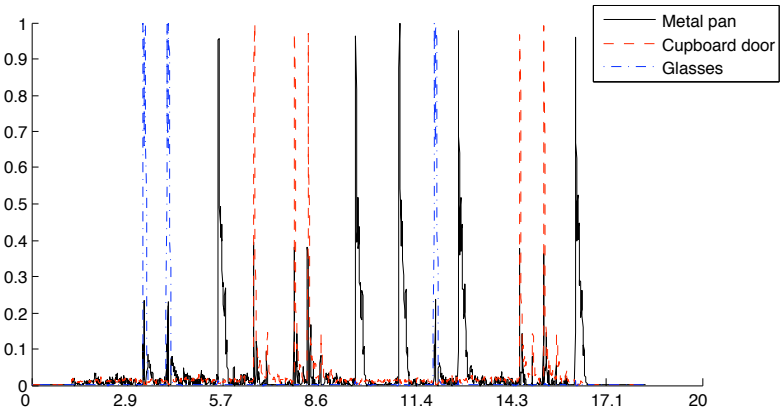
We aim at testing the proposed system on a realistic complex auditory scene. The learning phases as before is achieved with the method from Cont et al. (2007), except that instead of the modulation spectrum we use a short-time Fourier transform, with a window length of 2048 samples and a hop size of 256 samples, for a sampling rate of 44100 hertz. The templates are computed for three different environmental sounds which were used by Houix et al. (2007) in the context of experimental classification of everyday sounds. The three chosen sounds represent common sound events that occur in the everyday life context of a kitchen: the clinking of glasses, the closing of a door cupboard, and the scraping of a metal pan. In the sequel, these sounds are referenced respectively with the labels S_1, S_2 and S_3.

We have mixed S_1, S_2 and S_3 with some background noise recorded in the real-world context of a railway station by Tardieu (2006) to provide the problem setup. The resulting mix constitutes a complex auditory scene which is shown as a waveform representation in Figure 3.6(a). This reference scene is composed sequentially as follows: S_1, S_1, S_3, S_2, S_2, S_2, S_3, S_3, S_1, S_3, S_2, S_2, S_3. The scene is analyzed with Algorithm 2.8 using $\lambda_1 = 2.5$, $\lambda_2 = \lambda_s = 0$, $s_{min} = 0$ and $s_{max} = 1$. The template matrix $\mathbf{W}$, contains only the three templates for S_1, S_2 and S_3, without a noise template. The analysis parameters of the short-time Fourier transform during the incremental analysis are the same as the ones used for learning the templates.

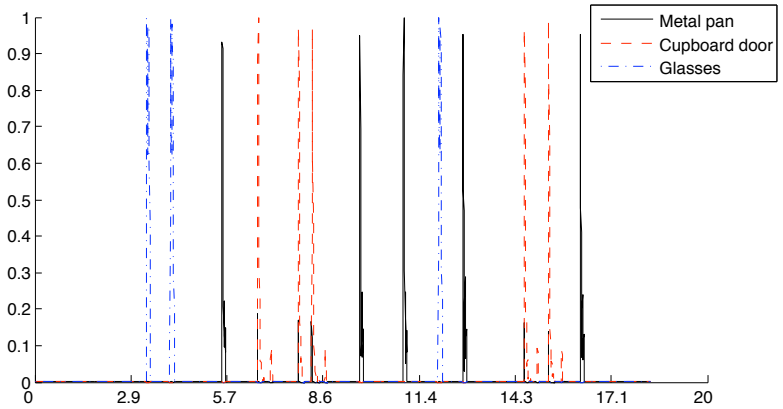
The results of the incremental analysis are shown in Figure 3.6. Figures 3.6(b) and 3.6(c) show the normalized activation coefficients $\mathbf{h}_j$ corresponding to each sound template j over time, as a result of non-sparse and sparse non-negative decomposition respectively. Following the time lines in both figures, it can be seen that the system has correctly identified the existence of each sound source. However, comparing Figures 3.6(b) and 3.6(c), we can observe more clear activities on the sparse version. Notably in the time interval between 2.9 and 5.0 seconds, the non-sparse version reports activations for two templates whereas the sparse version correctly reports one activation. This is a direct consequence of adaptive sparse scaling of Algorithm 2.8, leading to a more robust system.



(a) Waveform representation of the complex auditory scene.



(b) Activation coefficients using non-sparse ND.



(c) Activation coefficients using sparse ND with Algorithm 2.8.

Figure 3.6.: Analysis of a complex auditory scene. An auditory scene is created by mixing three environmental sounds with some background noise. The scene is then analyzed with non-sparse and sparse non-negative decomposition. The resulting activation coefficients show that the two versions have succeeded in identifying the sound sources. The sparse version however seems more robust than the non-sparse version.

Conclusion

Summary of the work

The present work has been dedicated to incremental multi-source recognition using non-negative matrix factorization, with an important focus on providing a mathematical framework for sparse coding schemes. Following an introduction of NMF problems and optimization techniques in Chapter 1, we discussed their applications to sound recognition and positioned our work in this context. We chose to address the problem of incremental recognition within the framework of non-negative decomposition, a modified NMF problem where the incoming signal is projected onto a basis of templates learned off-line prior to the decomposition. We have more specifically extended the framework proposed by Cont et al. (2007) in the context of non-negative matrix factorization for multi-pitch and multi-instrument recognition, where the need for an explicit control of sparsity has been underlined.

The main contribution of the present work is in the formulation of a sparse NMF framework where the sparseness of the factors is adaptively scaled to the data, whereas it has been assumed fixed in several previous approaches. We addressed this problem in Chapter 2 by employing convex optimization techniques. In this chapter, the need for sparsity controls in the context of NMF was motivated and illustrated with a synthetic experiment showing the limits of standard NMF even in lab situations. To overcome the issue, projected gradient optimization and second-order cone programming were introduced and employed to develop sparse NMF algorithms. Furthermore, alternative algorithms using gradient optimization and convex quadratic programming were also revised for the particular case of non-negative decomposition.

Following this theoretical development, some of the proposed algorithms were applied to preliminary evaluations in Chapter 3. The synthetic experiment that had shown the limits of standard NMF in Chapter 2 was repeated using two sparse NMF algorithms, validating both algorithms and the initial assumption of a real need for sparsity control to achieve desired results. We took one step further by applying the proposed algorithms to two sample real-world applications, notably that of incremental multiple-pitch and multiple-instrument recognition, and that of incremental analysis of complex auditory scenes. In both applications, we affirmed the interest for sparsity controls for recognition purposes. We leave a more rigorous evaluation, using common reference databases on the described problems for future work.

Perspectives

The present work mainly focused on proposing more rigorous optimization techniques with explicit controls for sparsity. However, to better approach the real-world problems of computational auditory scene analysis, other issues should also be taken into account. We try to expose some of the most important issues for future development of this work.

One of the main problems with NMF approaches is the implicit assumption that templates in $\mathbf{W}$ are considered inherently as stationary. Despite this assumption, NMF approaches have been successful in generalizing their knowledge in uncertain situations thanks to their inherent sparsity. One way to overcome this limitation is to consider front-end representations that are able to capture variability of the data over a short time-span. This was inherently the reason behind our choice of employing modulation spectrum representations in Section 3.2 for music transcription. This representation appeared relevant for the between-instrument discrimination. While this can address to some extent the problem of musical instrument discrimination, it will be not sufficient for complex auditory object analysis with long-term structural variations (*e.g.* a car engine sound). We believe that a more rigorous approach to address the non-stationarity of real-world objects would be to consider the temporality of object templates directly within the optimization scheme and representations of NMF. Such an approach was proposed by Smaragdis (2004) with an extended model of NMF capable of dealing with time-varying objects. Another possible approach for the case of non-negative decomposition would be to combine NMF with a state representation of sounds, and use one template for each state (*e.g.* attack and sustain part of a note). Regularization terms could then be used to express the potential transitions between the different states. These two approaches should be investigated further.

The optimization schemes proposed in this work, focused on the problem of sparsity control within a geometric space with the Euclidean distance. However, other structural components could be considered more rigorously for specific applications. Among these are the cost functions used in the optimization techniques which we did not consider thoroughly in this study. Other works suggest significant improvement of results by reformulating problems using other divergences than the Euclidean distance, leading to other interpretations of NMF problems. For example, Févotte et al. (2009) proved the relevance of the Itakura-Saito divergence in the context of sound analysis, and interpreted NMF problems with this divergence within a Bayesian framework. There is a close relationship between the choice of cost function and the geometry of the solution onto which we employ our optimization schemes. This relationship must be thoroughly formulated within each optimization framework, left out in the present work for future development.

Besides the optimization algorithms, the inherent additivity of the NMF framework invites other potential representation schemes to be considered for our applications. In the present work, we have considered a modulation spectrum and a magnitude spectrum representations. In the future, we would like to address the use of a wavelet transform in combination with the modulation spectrum, to provide a multi-scale analysis of the spectro-temporal modulations as proposed in (Sukittanon et al., 2005). Complex representations using the framework proposed in Appendix A could also be used. For example, keeping the phase information in the second transform of the modulation spectrum could provide a more informative representation that

would not only consider the amplitude modulation of the different partials, but also the phase coupling. Another idea would be the extension of the frequency shift model proposed by Mørup & Schmidt (2005) to a phase shift model, so as to keep the phase information in the frequency representations. In a more general setup, the extension of NMF to tensors could provide a new intuition about the geometry of both the problems and the representations, allowing for instance to use multi-channel information, or to keep the fundamental geometry of some multi-dimensional representations that need to be vectorized in the standard approach (*e.g.* the modulation spectrum).

Finally, we would like to improve the power of generalization and the robustness of the proposed system. Concerning the question of generalization, we think that second-order cone programming could help relaxing the hard constraints imposed on $\mathbf{W}$ which is kept fixed during the decomposition. The idea would be to have an adaptive template matrix $\mathbf{W}$ whose columns are allowed to move inside the volume delimited by a cone centered around the respective learned templates. A similar idea has already been proposed by Heiler & Schnörr (2006) in the context of supervised classification with NMF. Future work should address the adaptation of the approach to sparse non-negative decomposition. Concerning robustness, we believe that investigating further the use of a noise template would be of interest. We have thought of adding a column in $\mathbf{W}$ that would be updated in a similar scheme as in incremental NMF (Bucak & Günsel, 2009). This could help to absorb not only the background noise but also some unknown present sources, which is important in a real-world scenario where we do not know all the present sources in advance.

A. Relaxation of the non-negativity constraints

In this appendix, we show how to relax the non-negativity constraints on $\mathbf{V}$ and $\mathbf{W}$ so as to use complex matrices. In the case of standard NMF, the issue can be addressed easily using an alternating least squares scheme as described in Section 1.1, but where only the least squares problem for the update of $\mathbf{H}$ is constrained. We concentrate on the more demanding step of relaxing the non-negativity constraints on $\mathbf{v}$ and $\mathbf{W}$ in the different problems of SND formulated in Section 2.3.

Using the Euclidean distance and the presented optimization techniques, we can relax the non-negativity constraints on $\mathbf{v}$ and $\mathbf{W}$ in the algorithms for SND developed in Section 2.3. Considering complex matrices $\mathbf{v}$, $\mathbf{W}$ and $\mathbf{h}$, we can rewrite the Euclidean distance as follows:

$$\frac{1}{2}\|\mathbf{v}-\mathbf{W}\mathbf{h}\|_2^2 = \frac{1}{2}\|\mathbf{v}\|_2^2 + \frac{1}{2}\|\mathbf{W}\mathbf{h}\|_2^2 - \Re\mathfrak{e}\left(\langle \mathbf{v}, \mathbf{W}\mathbf{h}\rangle\right) \tag{A.1}$$

$$= \frac{1}{2}\mathbf{v}^H\mathbf{v} + \frac{1}{2}\mathbf{h}^H\mathbf{W}^H\mathbf{W}\mathbf{h} - \Re\mathfrak{e}\left(\mathbf{v}^H\mathbf{W}\mathbf{h}\right) \tag{A.2}$$

Assuming now that $\mathbf{h}$ is non-negative, we can deduce the following expression:

$$\frac{1}{2}\|\mathbf{v}-\mathbf{W}\mathbf{h}\|_2^2 = \frac{1}{2}\mathbf{v}^H\mathbf{v} + \frac{1}{2}\mathbf{h}^T\mathbf{W}^H\mathbf{W}\mathbf{h} - \Re\mathfrak{e}\left(\mathbf{v}^H\mathbf{W}\right)\mathbf{h} \tag{A.3}$$

As the matrix $\mathbf{W}^H\mathbf{W}$ is positive-semidefinite, we have $\mathbf{h}^T\mathbf{W}^H\mathbf{W}\mathbf{h} \geqslant 0$, and in particular we have $\mathbf{h}^T\mathbf{W}^H\mathbf{W}\mathbf{h} \in \mathbb{R}$. As a result, we can derive the following equalities:

$$\mathbf{h}^T\mathbf{W}^H\mathbf{W}\mathbf{h} = \Re\mathfrak{e}\left(\mathbf{h}^T\mathbf{W}^H\mathbf{W}\mathbf{h}\right) = \mathbf{h}^T\Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{W}\right)\mathbf{h} \tag{A.4}$$

We notice that since $\Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{W}\right)$ is symmetric and that $\mathbf{h}^T\Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{W}\right)\mathbf{h} = \mathbf{h}^T\mathbf{W}^H\mathbf{W}\mathbf{h} \geqslant 0$, we can deduce that $\Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{W}\right)$ is positive-semidefinite. This property will be helpful to extend the MTPA algorithm to complex matrices $\mathbf{v}$ and $\mathbf{W}$. But for the moment, we derive a last expression for the Euclidean distance:

$$\frac{1}{2}\|\mathbf{v}-\mathbf{W}\mathbf{h}\|_2^2 = \frac{1}{2}\mathbf{v}^H\mathbf{v} + \frac{1}{2}\mathbf{h}^T\Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{W}\right)\mathbf{h} - \Re\mathfrak{e}\left(\mathbf{v}^H\mathbf{W}\right)\mathbf{h} \tag{A.5}$$

The latter expression allows to relax the non-negativity constraints on $\mathbf{v}$ and $\mathbf{W}$ in both the gradient optimization and the CQP algorithms we developed for SND.

Gradient optimization algorithms are derived by calculating the gradient of the Euclidean distance with respect to $\mathbf{h}$. Using the expression in Equation A.5, the gradient is calculated as follows:

$$\frac{1}{2}\,\nabla_{\mathbf{h}}\|\mathbf{v}-\mathbf{W}\mathbf{h}\|_2^2 = \frac{1}{2}\Big(\Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{W}\right) + \Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{W}\right)^T\Big)\mathbf{h} - \Re\mathfrak{e}\left(\mathbf{v}^H\mathbf{W}\right)^T \tag{A.6}$$

As the matrix $\mathbf{W}^H\mathbf{W}$ is hermitian, the matrix $\Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{W}\right)$ is symmetric and the gradient becomes:

$$\frac{1}{2}\nabla_{\mathbf{h}}\|\mathbf{v}-\mathbf{W}\mathbf{h}\|_2^2 = \Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{W}\right)\mathbf{h} - \Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{v}\right) \tag{A.7}$$

We can now extend the presented gradient optimization algorithms to complex matrices $\mathbf{v}$ and $\mathbf{W}$. It suffices to replace the gradient $\mathbf{W}^T\mathbf{W}\mathbf{h} - \mathbf{W}^T\mathbf{v}$ for real matrices, with the gradient $\Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{W}\right)\mathbf{h} - \Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{v}\right)$ for complex matrices in the Algorithms 2.6 and 2.7.

The extension to complex matrices $\mathbf{v}$ and $\mathbf{W}$ is more direct for the MTPA algorithm with CQPs. As stated above, the matrix $\Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{W}\right)$ is positive-semidefinite. Therefore, it suffices to replace $\mathbf{W}^T\mathbf{W}$ and $\mathbf{W}^T\mathbf{v}$ respectively with $\Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{W}\right)$ and $\Re\mathfrak{e}\left(\mathbf{W}^H\mathbf{v}\right)$ in the problems given in Equations 2.29, 2.30, 2.31 and 2.32 which are still CQPs under these modifications.

To conclude, all the algorithms we have developed for SND can be extended to complex matrices $\mathbf{v}$ and $\mathbf{W}$. The different SND problems that we can solve with these algorithms are summed up in Table A.1.

Problem	Cost function	Constraints
SND PG1	$\frac{1}{2}\|\mathbf{v}-\mathbf{W}\mathbf{h}\|_2^2 + \lambda(\sigma\|\mathbf{h}\|_2 - \|\mathbf{h}\|_1)^2$	$\mathbf{h}\in\mathbb{R}_+^r$
SND PG2	$\frac{1}{2}\|\mathbf{v}-\mathbf{W}\mathbf{h}\|_2^2 + \lambda\sum_i \tanh\left(\|ah_i\|^b\right)$	$\mathbf{h}\in\mathbb{R}_+^r$
SND PG3	$\frac{1}{2}\|\mathbf{v}-\mathbf{W}\mathbf{h}\|_2^2 + \lambda\|\mathbf{h}\|_p^p$	$\mathbf{h}\in\mathbb{R}_+^r$
SND PG	$\frac{1}{2}\|\mathbf{v}-\mathbf{W}\mathbf{h}\|_2^2$	$\mathrm{sp}(\mathbf{h})=s,\ \mathbf{h}\in\mathbb{R}_+^r$
SND SCQP	$\frac{1}{2}\|\mathbf{v}-\mathbf{W}\mathbf{h}\|_2^2 + \lambda_1\|\mathbf{h}\|_1 + \frac{\lambda_2}{2}\|\mathbf{h}\|_2^2 - \lambda_s\mathrm{sp}(\mathbf{h})$	$s_{min}\leqslant\mathrm{sp}(\mathbf{h})\leqslant s_{max},\ \mathbf{h}\in\mathbb{R}_+^r$

Table A.1.: Different sparse non-negative decomposition problems. The name of the problems specifies the optimization technique needed to solve them: (PG) projected gradient and (SCQP) sequential convex quadratic programming with the MTPA algorithm. The corresponding cost functions and constraints are also given. All of these problems can be extended to complex matrices $\mathbf{v}$ and $\mathbf{W}$.

Bibliography

Abdallah, S. A. & Plumbley, M. D. (2004). Polyphonic music transcription by non-negative sparse coding of power spectra. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)* (pp. 318–325). Barcelona, Spain.

Atlas, L. E. & Janssen, C. (2005). Coherent modulation spectral filtering for single-channel music source separation. In *IEEE International Conference in Acoustics and Speech Signal Processing (ICASSP)*.

Ballet, G., Borghesi, R., Hoffmann, P., & Lévy, F. (1999). Studio Online 3.0: An Internet "killer application" for remote access to IRCAM sounds and processing tools. In *Journées d'informatique musicale* Paris.

Benetos, E., Kotropoulos, C., Lidy, T., & Rauber, A. (2006). Testing supervised classifiers based on non-negative matrix factorization to musical instrument classification. In *14th European Signal Processing Conference*.

Berry, M. W., Browne, M., Langville, A., Pauca, V. P., & Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(1), 155–173.

Bertin, N., Badeau, R., & Vincent, E. (2009). *Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription.* Technical report, TELECOM ParisTech.

Blondel, V., Ho, N.-D., & Van Dooren, P. (2005). Algorithms for weighted non-negative matrix factorization.

Boyd, S. & Vandenberghe, L. (2004). *Convex Optimization.* Cambridge University Press.

Bruckstein, A. M., Elad, M., & Zibulevsky, M. (2008). On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations. *Information Theory, IEEE Transactions on*, 54(11), 4813–4820.

Bucak, S. S. & Günsel, B. (2009). Incremental subspace learning via non-negative matrix factorization. *Pattern Recognition*, 42(5), 788–797.

Buciu, I. (2008). Non-negative matrix factorization, a new tool for feature extraction: Theory and applications. In *Proceedings of ICCCC 2008*, volume 3 (pp. 67–74).

Cheng, C.-C., Hu, D. J., & Saul, L. K. (2008). Nonnegative matrix factorization for real time musical analysis and sight-reading evaluation. In *Acoustics, Speech and Signal Processing 2008, IEEE International Conference on* (pp. 2017–2020). Las Vegas, NV, USA.

Chu, S., Narayanan, S., & Kuo, C.-C. J. (2009). Environmental sound recognition with time-frequency audio features. IEEE Transactions on Speech, Audio, and Language Processing (in press).

Cichocki, A., Lee, H., Kim, Y.-D., & Choi, S. (2008). Nonnegative matrix factorization with α-divergence. *Pattern Recognition Letters*, 29, 1433–1440.

Cichocki, A. & Zdunek, R. (2006). NMFLAB/NTFLAB – MATLAB Toolbox for Non-Negative Matrix/Tensor Factorization. `http://www.bsp.brain.riken.jp/ICALAB/nmflab.html`.

Cichocki, A., Zdunek, R., & Amari, S.-i. (2006). Csiszar's divergences for non-negative matrix factorization: Family of new algorithms. In Springer (Ed.), *Independent Component Analysis and Blind Signal Separation (ICA 2006)*, volume 3889 of *Lecture Notes in Computer Science* (pp. 32–39). Charleston, SC, USA.

Cont, A. (2006). Realtime multiple pitch observation using sparse non-negative constraints. In *International Symposium on Music Information Retrieval (ISMIR)* Victoria, Canada.

Cont, A., Dubnov, S., & Wessel, D. (2007). Realtime multiple-pitch and multiple-instrument recognition for music signals using sparse non-negative constraints. In *Proceedings of the 10th International Conference on Digital Audio Effects (DAFx-07)* Bordeaux, France.

de Cheveigné, A. (2006). Multiple f0 estimation. In D.-L. Wang & G. J. Brown (Eds.), *Computational Auditory Scene Analysis: Principles, Algorithms and Applications* (pp. 45–72). IEEE Press / Wiley, 2006.

Dhillon, I. S. & Sra, S. (2005). Generalized nonnegative matrix approximations with bregman divergences. In *Neural Information Processing Systems* (pp. 283–290).: MIT Press.

Ding, C. H. Q., Li, T., & Jordan, M. I. (2006). *Convex and semi-nonnegative matrix factorizations for clustering and low-dimension representation*. Technical report, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA, USA.

Donoho, D. & Stodden, V. (2004). When does non-negative matrix factorization give a correct decomposition into parts? In S. Thrun, L. K. Saul, & B. Schölkopf (Eds.), *Advances in Neural Information Processing Systems, Proceedings of the 2003 Conference (NIPS 2004)*, volume 16 Cambridge, MA: MIT Press.

Dubnov, S. & Rodet, X. (2003). Investigation of phase coupling phenomena in sustained portion of musical instruments sound. *Acoustical Society of America Journal*, 113, 348–359.

Essid, S., Richard, G., & David, B. (2006). Instrument recognition in polyphonic music based on automatic taxonomies. *IEEE Transactions on Audio, Speech & Language Processing*, 14(1), 68–80.

FitzGerald, D., Cranitch, M., & Coyle, E. (2008). Extended nonnegative tensor factorisation models for musical sound source separation. *Computational Intelligence and Neuroscience*, 2008, 15 pages.

Friedlander, M. P. (2006). BCLS: A large-scale solver for bound-constrained least squares. `http://www.cs.ubc.ca/~mpf/bcls/`.

Févotte, C., Bertin, N., & Durrieu, J.-L. (2009). Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural Computation*, 21(3), 793–830.

Goto, M., Hashiguchi, H., Nishimura, T., & Oka, R. (2002). RWC Music Database: Popular, Classical, and Jazz Music Databases. In *3rd International Conference on on Music Information Retrieval (ISMIR)* (pp. 287–288).

Guillamet, D., Vitrià, J., & Schiele, B. (2003). Introducing a weighted non-negative matrix factorization for image classification. *Pattern Recognition Letters*, 24, 2447–2454.

Heiler, M. & Schnörr, C. (2005a). Learning non-negative sparse image codes by convex programming. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*.

Heiler, M. & Schnörr, C. (2005b). Reverse-convex programming for sparse image codes. In Springer (Ed.), *Proceedings of Energy Minimization Methods in Computer Vision and Pattern Recognition, 5th International Workshop*, volume 3757 of *Lecture Notes in Computer Science* (pp. 600–616). St. Augustine, FL, USA.

Heiler, M. & Schnörr, C. (2006). Learning sparse representations by non-negative matrix factorization and sequential cone programming. *Journal of Machine Learning Research*, 7, 1385–1407.

Houix, O., Lemaitre, G., Misdariis, N., & Susini, P. (2007). *Everyday sound classification: Experimental classification of everyday sounds*. Deliverable (CLOSED project) 4.1 part 2, IRCAM, Paris.

Hoyer, P. O. (2002). Non-negative sparse coding. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on* (pp. 557–565). Martigny, Switzerland.

Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5, 1457–1469.

Karvanen, J. & Cichocki, A. (2003). Measuring sparseness of noisy signals. In *4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)* (pp. 125–130). Nara, Japan.

Kitahara, T., Komatani, K., Ogata, T., Okuno, H. G., & Goto, M. (2006). A missing feature approach to instrument identification in polyphonic music. In *IEEE ICASSP* Toulouse.

Kreutz-Delgado, K. & Rao, B. D. (1997). *A General Approach to Sparse Basis Selection: Majorization, Concavity, and Affine Scaling*. Technical report, University of California, San Diego.

Laurberg, H., Christensen, M. G., Plumbley, M. D., Hansen, L. K., & Jensen, S. H. (2008). Theorems on positive data: On the uniqueness of NMF. *Computational Intelligence and Neuroscience*, 2008, 9 pages.

Lee, D. D. & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755), 788–791.

Lee, D. D. & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing, Proceedings of the 2000 Conference (NIPS 2000)*, volume 13 (pp. 556–562). Cambridge, MA: MIT Press.

Li, T. & Ding, C. (2006). The relationships among various nonnegative matrix factorization methods for clustering. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining* (pp. 362–371). Washington, DC, USA: IEEE Computer Society.

Livshin, A. & Rodet, X. (2006). The significance of the nonharmonic "noise" versus the harmonic series for musical instrument recognition. In *International Symposium on Music Information Retrieval (ISMIR)*.

Löfberg, J. (2004). YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference* Taipei, Taiwan. `http://control.ee.ethz.ch/~joloef/wiki/pmwiki.php`.

Marshall, A. W. & Olkin, I. (1979). *Inequalities: Theory of Majorization and Its Application*. Academic Press.

Mørup, M. & Schmidt, M. N. (2005). *Nonnegative matrix factor 2-D deconvolution (NMF2D) and sparse NMF2D (SNMF2D)*. Technical report, Institute for Mathematical Modelling, Technical University of Denmark.

Niedermayer (2008). Non-negative matrix division for the automatic transcription of polyphonic music. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)* (pp. 544–549).

Ozerov, A. & Févotte, C. (2009). Multichannel nonnegative matrix factorization in convolutive mixtures. with application to blind audio source separation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'09)*.

Paatero, P. (1997). Least squares formulation of robust non-negative factor analysis. *Chenometrics and Intelligent Laboratory Systems*, 37(1), 23–35.

Paatero, P. & Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2), 111–126.

Parry, R. M. & Essa, A. I. (2007). Phase-aware non-negative spectrogram factorization. In Springer (Ed.), *Proceedings of the 7th International Conference on Independent Component Analysis and Signal Separation (ICA 2007)*, volume 4666 of *Lecture Notes in Computer Science* (pp. 536–543). London, UK.

Paulus, J. & Virtanen, T. (2005). Drum transcription with non-negative spectrogram factorisation. In *Proceedings of the 13th European Signal Processing Conference (EUSIPCO 05)* Antalya, Turkey.

Raczyński, S. A., Ono, N., & Sagayama, S. (2007). Harmonic nonnegative matrix approximation for multipitch analysis of musical sounds. In *Proceedings of ASJ Autumn Meeting* (pp. 827–830).

Sha, F. & Saul, L. K. (2005). Real-time pitch determination of one or more voices by nonnegative matrix factorization. *Advances in Neural Information Processing Systems*, 17, 1233–1240.

Smaragdis, P. (2004). Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In Springer (Ed.), *Proceedings of the 5th International Conference on Independent Component Analysis and Blind Signal Separation (ICA 04)*, volume 3195 of *Lecture Notes in Computer Science* (pp. 494–499). Granada, Spain.

Smaragdis, P. & Brown, J. C. (2003). Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (pp. 177–180). New Paltz, NY.

Sturm, J. F. (2001). *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones (updates version 1.05).* Technical report, Department of Econometrics, Tilburg University, Tilburg, The Netherlands. `http://sedumi.ie.lehigh.edu/`.

Sukittanon, S., Atlas, L. E., & Pitton, J. W. (2004). Modulation-scale analysis for content identification. *IEEE Transactions on Signal Processing*, 52(10), 3023–3035.

Sukittanon, S., Atlas, L. E., Pitton, J. W., & Filali, K. (2005). Improved modulation spectrum through multi-scale modulation frequency decomposition. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 4 (pp. 517–520).

Tardieu, J. (2006). *De l'ambiance à l'information sonore dans un espace public. Méthodologie et réalisation appliquée aux gares.* PhD thesis, Université de Paris 6.

Theis, F. J., Stadlthanner, K., & Tanaka, T. (2005). First results on uniqueness of sparse non-negative matrix factorization. In *Proceedings of the 13th European Signal Processing Conference (EUSIPCO '05)* Antalya, Turkey.

Tuy, H. (1987). Convex programs with an additional reverse convex constraint. *Journal of Optimization Theory and Applications*, 52, 463–486.

Vincent, E., Bertin, N., & Badeau, R. (2008). Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on* (pp. 109 –112).

Wang, D. L. & Brown, G. J., Eds. (2006). *Computational auditory scene analysis: Principles, algorithms and applications.* IEEE Press/Wiley-Interscience.

Welling, M. & Weber, M. (2001). Positive tensor factorization. *Pattern Recognition Letters*, 22(12), 1255–1261.