A convex formulation for informed source separation in the single channel setting.

Augustin Lefèvre, François Glineur, P.-A. Absil

Abstract

Blind audio source separation is well-suited for the application of unsupervised techniques such as Nonnegative Matrix Factorization (NMF). It has been shown that on simple examples, it retrieves sensible solutions even in the single-channel setting, which is highly ill-posed. However, it is now widely accepted that NMF alone cannot solve single-channel source separation, for real world audio signals. Several proposals have appeared recently for systems that allow the user to control the output of NMF, by specifying additional equality constraints on the coefficients of the sources in the time-frequency domain. In this article, we show that matrix factorization problems involving these constraints can be formulated as convex problems, using the nuclear norm as a low-rank inducing penalty. We propose to solve the resulting nonsmooth convex formulation using a simple subgradient algorithm. Numerical experiments confirm that the nuclear norm penalty allows the recovery of (approximately) low-rank solutions that satisfy the additional user-imposed constraints. Moreover, for a given computational budget, we show that this algorithm matches the performance or even outperforms state-of-the art NMF methods in terms of the quality of the estimated sources.

1. Introduction

Single-channel source separation is an underdetermined problem, commonly used as a pre-processing technique for higher-level tasks (speech recognition in complex environments, polyphonic music transcription, etc.). While exact source recovery cannot be expected in general, a key ingredient in source separation techniques consists in assuming some form of redundancy in the data, which renders the problem overdetermined. This is typically done by representing audio tracks in the time-frequency domain as low-rank matrices. Nonnegative matrix factorization was first applied to audio signals for polyphonic transcription [SB03], although it was already used in other fields [PT94, LS99].

An important idea underlying matrix factorization techniques for audio signals is that they recover a representation of signals in terms of template signals modulated by location-dependent gains. In the field of music signal processing, this idea was supported by experiments on simple music signals [FBD09]. In computer vision, similar experiments suggested a that part-based representation

of visual objects could be retrieved by NMF [LS99]. The miracle of part-based representation no longer works for real music or speech signals, because they cannot be assumed to satsify the low-rank hypothesis, but it has spawned several interesting research tracks: parameterized templates were introduced in [VBB10] in order to match the harmonic structure of many musical instruments ; probabilistic models and penalty functions to favor smooth time-varying gains in [Vir07, Fév11] ; Markov models, to stabilize the recognition of vowels in speech processing [MSR10].

In parallel to these research tracks, linear models for audio signals have also been the subject of many contributions. These models rely on the library approach (or dictionary approach), where audio templates correspond to actual signals stored offline in libraries, each specific to an instrument. The University of Iowa's electronic music studios, for instance, have made available recordings of isolated notes for many popular instruments: violin, piano, cello, more generally instruments belonging to the family of woodwind, brass, or string instruments. Due to the large size of the libraries, there are many ways to represent any audio signals as a linear combination of audio templates. Thus, in the library approach, structured decompositions are introduced, based on simple principles: if an instrument is present in the mix, only a few of its templates should be used at the same time [SSR09] ; in the case where the sources are unknown group structures are employed to select the appropriate libraries [BPSS10].

More recently, several contributions have been made to take into account prior information specific to the target mix signal: manual segmentation of audio tracks [OFBD11], MIDI aligned music scores [GSD12, HBD10], time-aligned pitch estimates for the singing voice [DT12]. A common trait of these methods is that they are all based on a simple extension of NMF : annotations are used to specify equality constraints in the matrix of activation coefficients in NMF, setting them to known values. Thus, annotations help learn a source specific dictionary on segments of the recording where only that source is active: in this way, manual segmentation of audio signals allows a blind source separation task to be cast as a supervised linear model. In [GSD12], prior information consists in the score that the music follows. Digital music synthesizers are used to provide a rough guess of the sources. All these contributions are now identified as the category of *informed source separation* methods. The formulation proposed in this article belongs to this category.

While time segmentation of audio signals allows to use supervised learning techniques, it is not always applicable. Instead, one can always rely on a universal property of natural signals: they have a very sparse representation in the time-frequency domain. This property, dubbed W-disjoint orthogonality, is at the heart of several source separation techniques in the multiple microphone setting [YR04, AGB10].

In a previous contribution [LBF12], we formulated as a problem of non-negative matrix factorization (NMF) with additional equality constraints, consistently with the strong tradition in audio source separation. Results on the SISEC database showed that we can obtain state-of-the-art results while annotating only a fraction of the spectrogram ; since user annotation is difficult

and time-consuming, we also experimented with automatic annotation methods, relying on supervised learning. Interaction with the user has been further explored in [BM13, FBR12]. Although based on a slightly different technique, called probabilistic latent component analysis (PLCA), the formulation used in [BM13] can be viewed as NMF where dissimilarity between observations and the model is measured with a Kullback-Leibler divergence.

While it gives satisfactory results, NMF is hard to solve: for typical values of the "rank" parameter used in audio, algorithms cannot be guaranteed to converge to globally optimal solutions, and there is no alternative but to resort to algorithms that converge to local minima. In practice, this means that several initial points should be tried and the best be selected on a principled basis. One would be tempted to replace the strict low-rank constraint by a convex penalty function favoring low-rank solutions.

The main contribution of this article is to show that we can replace NMF by a matrix approximation problem involving nonnegativity constraints, low-rank inducing penalty functions and constraints on the coefficients of the solutions to model additional information provided by the user, i.e. annotations. The main advantage of such a formulation is that one can borrow tools from the field of convex optimization to construct algorithms that retrieve source estimates of similar, if not better quality, for a comparable computational budget, as shown in preliminary results [LAG13]. In this article, we give a detailed presentation of a subgradient algorithm used to solve the proposed formulation, and show that it has the desired effect of finding solutions that are (approximately) low-rank. Our second contribution, which we detail in Section 5.1, is related to the way we let the user specify annotations: by restricting the set of annotated time-frequency coefficients to those whose target values is zero, we show that our formulation can gain in robustness, at a small sacrifice in terms of generality.

The rest of this article is organized as follows: in Section 2, we review of well-established techniques for single-channel source separation : time-frequency transforms, filtering techniques for source estimates recovery, and evaluation metrics. In Section 3, we introduce a formulation of informed source separation using nonnegative matrix factorization which was previously proposed [LBF12]. In Section 3.2, we discuss a convex formulation of annotation-informed source separation, dubbed `AISS_lownuc`, in the form of a low-rank matrix approximation problem with a low-rank inducing penalty term, and equality constraints. User-provided annotations are encoded as equality constraints, and those are key to the success of our formulation. After presenting in Section 4 our algorithm for `AISS_lownuc`, we investigate in Section 5 the impact of various choices of annotations, and demonstrate the benefits of our convex formulation compared with NMF.

2. Time-frequency analysis and audio source separation

This section is a brief introduction to audio source separation. In Section 2.1, we present time-frequency transforms, which allow to transform a one-dimensional audio signal into a two-dimensional object, frequency and time

being now the dimensions of variation. The matrix factorization problem that we introduce is indeed posed in the time-frequency domain, so that an input time-frequency matrix is separated as a sum of matrices, which are interpreted as source terms (as illustrated in Figure 3). We refer the reader to textbooks such as [OS75] for a complete presentation of time-frequency transforms and their many applications, such as modifying the duration of an audio signal of modifying its pitch.

Next, we explain in Section 2.2 how to transform those source estimates back as audio signals, using *time-frequency masking* : early proposals for source separation recognized filtering as the best way to avoid artifacts due to inexact solutions [SB03, FBD09]. In Section 2.3, we summarize evaluation metrics for audio source separation [VGF06], and define the notion of oracle estimates, in controlled experiments where the true source signals are known in advance.

2.1. Time-frequency representation of audio signals

Single-channel source separation consists in recovering a certain number of unknown source signals from measurements of their sum. The first step in single-channel source separation consist in finding a representation of the source signals that enhances their redundancy. As we shall explain in this section, this is done by computing their spectrogram, which is a time-frequency representation. Time-frequency representations of audio signals are sparse and redundant, which is key to the success of blind source separation.

The computation of spectrograms is illustrated on Figure 1: short time segments are extracted from the signal and multiplied coefficientwise by a window function. Successive windows overlap by a fraction of their length, which is usually taken as 50%. On each of these segments, a Fourier transform is computed. Thus, from a one-dimensional signal $x \in \mathbb{R}^T$, we obtain a complex matrix C of size $F \times N$ where $FN \simeq 2T$ (because of the 50% overlap between windows). These preliminary steps correspond to computing the short time Fourier transform (STFT):

$$C_{fn} = \sum_{t=1}^{F} x_{t+(n-1)H} w_t \exp\left(-\frac{2(f-1)\pi(t-1)}{F}\right)$$

for all $f \in \{1 \dots F\}$, and $n \in \{1 \dots N\}$. The so-called hop size H determines the overlap between successive windows, $w \in \mathbb{R}^F$ is a window function, and N is chosen to match the size of the signal. To make this possible, that the signal should be appropriately zero-padded beforehand. We refer the reader to textbooks such as [OS75] for more explanations. Finally, we take $Y_{fn} = |C_{fn}|^2$, in order to obtain approximate invariance to translations of the signal. Coefficient Y_{fn} measures the amount of energy of the signal at frequency f and time index n in the time-frequency plane. This magnitude is represented as a color code in Figure 1: blue for small coefficients, and red for high coefficients.

The length `winlen` of the segments, or *window length* determines the shape of the spectrogram as a matrix. The number of rows corresponds to the frequency resolution: indeed, letting f_s be the sampling rate of the audio signal,

consecutive rows correspond to consecutive frequencies that are $\frac{f_s}{\text{winlen}}$ Hz apart. On the other hand, the number of columns determines the time resolution. For instance, if a signal is sampled at the typical rate of 44100 Hz, and the window length is 2048 (which corresponds to approximately 50 ms), the frequency resolution is 20 Hz. If, the window length is 1024, the frequency resolution will be only 10 Hz, but the time resolution will be 25 ms.

An important property of the STFT operator is that we can reconstruct a signal exactly from its STFT samples, through a so-called "*inverse*" short time Fourier transform (iSTFT). However, modifying the coefficients of a signal in the time-frequency domain does not guarantee that the "inverse" signal will be well-defined. In other words, the adjoint of the STFT operator has a nontrivial kernel.

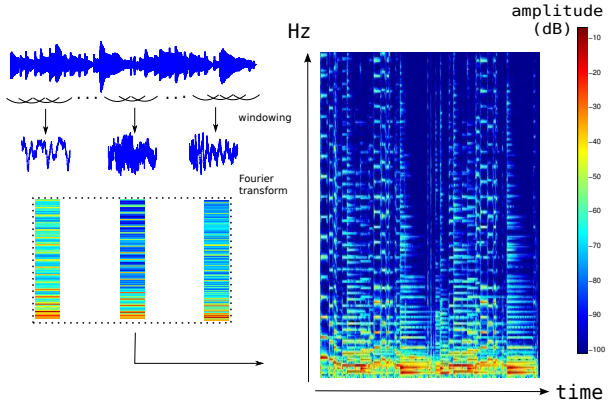


Figure 1: Time-frequency operators enhance sparsity in audio signals

Spectrograms of natural signals have two important properties. The first, depicted in Figure 1 is their sparsity: only a fraction of the STFT coefficients have significant magnitude. The second property is redundancy: there are many similar columns in the spectrogram, which reflects an intuitive notion of vocabulary for musical sounds or speech.

2.2. Time-frequency masking and source separation

Computing spectrograms approximately preserves the property that the observed signal is the sum of the source signals. Again defining, the spectrogram of the mix $Y_{fn} = |C_{fn}|^2$, we have $Y \simeq \sum_g X_g$, where X_g is the spectrogram of source g. Note that even if we assume that the mixed signal is the sum of source signals, in the time domain ($x = \sum_g s_g$), we can only guarantee $Y \simeq \sum_g X_g$ approximately since Y and X_g are nonlinear transforms of x and s_g, respectively. Still, this approximate summation property is good enough and observed in practice.

Now, suppose we have an estimate of the spectrogram $\hat{X}_g$ for each source g. To obtain an estimate of the time domain signals $s_g \in \mathbb{R}^T$, we first estimate their

STFT through *time-frequency masking* $\hat{S}_{g,fn} = \dfrac{\hat{X}_{g,fn}}{\sum_{g'} \hat{X}_{g',fn}} C_{fn}$, and recover $\hat{s}_g$ from $\hat{S}_g$ by inverse STFT. Note that the masking coefficients

$$\mathrm{mask}_{g,fn} = \frac{\hat{X}_{g,fn}}{\sum_{g'} \hat{X}_{g',fn}} \tag{1}$$

are nonnegative and sum to one each pairl (f, n): consequently, we have $\sum_g \hat{S}_g = C$ and $\sum_g \hat{s}_g = x$, i.e. we obtain a source separation that reconstructs the mixed signal exactly.

We now come to a fundamental property on which our contribution relies: since each source spectrogram $S_{g,fn}$ is sparse, the masking coefficients $\mathrm{mask}_{g,fn}$ are most likely equal to either 0 or 1. Indeed, we display in Figure 2 color codes indicating in bright color those points (f, n) for which masking coefficients are close to 0 or 1. We also scale the brightness according to the magnitude of the spectrogram, so that the most important points are emphasized. As we can see, a significant proportion of the spectrogram is either black (no source is active), or colored (only one source contributes). A smaller fraction of the points is white, those are the points where several sources contribute simultaneously. This property, dubbed W-disjoint orthogonality, is at the heart of state-of-the art methods in multichannel audio source separation algorithms [YR04]. This property implies that although the support of the sources are neither disjoint in the time domain, nor in the frequency domain, they are disjoint in the time-frequency domain.

Thus, if we provide for each source a guess of a subset of those points where its contribution is negligible, we would already obtain good source estimates. In the multichannel setting, [YR04, AGB10] use unsupervised learning techniques to provide such a guess. Those techniques cannot be applied in the single-channel case.

In recent contributions [BM13, LBF12], a subset of the coefficients of the spectrogram of each source is constrained to pre-specified target values. Matrix factorization techniques can then be used to complete the picture, as will be explained in Section 3.2. One point raised by those contributions is that source estimates are very sensitive to the chosen target values. However, if we trust the W-disjoint orthogonality property, we can safely choose to impose constraints only when the coefficients are equal to zero, and let other coefficients free for further investigation. This possibility will be discussed further in Section 5.1.

Before we proceed to a formal presentation of matrix factorization, let us discuss evaluation procedures in audio source separation.

2.3. Evaluation of source separation results

Single-channel source separation is an open problem in audio signal processing: it is highly ill-posed, so that several assumptions must be made to avoid situations were recovering source estimates is not sensible. As mentioned before, we must assume that the number and type of sources is known, and that the

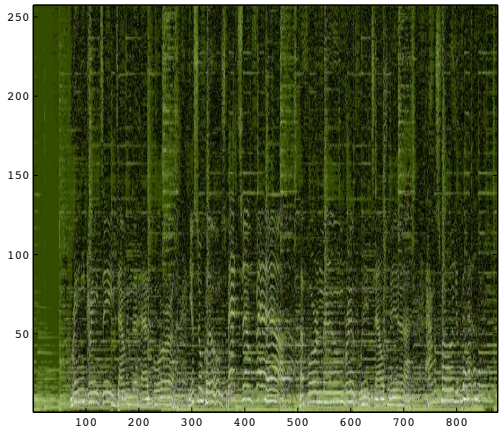


Figure 2: The W-disjoint orthogonality property (see text for details)

recorded signal is a linear instantaneous mixture of the source signals. This is a standard hypothesis, and in this report we will additionally assume that *the mix is balanced*. More precisely, we assume that all source signals have equal ℓ_2 norm: this is to avoid such situations where one of the source signals is so small that it is unreasonable to try to recover it.

Listening tests provide the most valuable insights on the quality of source estimates $\hat{s}_g$. In some cases, they are also the true performance measure, for instance if we are to remix sources or make a karaoke version of a song. Purely quantitative criteria, on the other hand, are objective and require little enough computational resource so we can use them to track the performance of a sequential algorithm, or determine the influence of some parameter on source separation performance. Benchmark evaluations use both types of criterion, in an attempt to find an ideal perceptual measure of quality.

Computing quantitative criteria implies that we have true source signals at our disposal. The simplest performance criterion is the signal to noise ratio (SNR) for each source g:

$$SNR_g = 10 \log_{10} \frac{\|\hat{s}_g\|_2^2}{\|s_g - \hat{s}_g\|_2^2} \,. \tag{2}$$

In practice, the SNR is not close to a perceptual metric, because it does not tolerate some benign deformations of the target signal : for instance, if the estimate $\hat{s}_g$ was simply a scaled version of the true source signal λs_g then perceptually the result would be perfect, but the SNR would be $10 \log_{10} \frac{\lambda^2}{(1-\lambda)^2}$ which can be arbitrarily low. Another quality measure, the source to distortion ratio (SDR), was first proposed in [VGF06] to allow such deformations.

	method 1	**random**	**blind**	**oracle**
SDR1	7.83	-2.02	-0.02	18.43
SDR2	6.17	1.01	-0.04	12.82

Table 1: Example of source separation results on one audio track

More precisely, the idea behind SDR is to estimate the contribution of each source in a given estimate, through a linear model:

$$\hat{s}_g = \sum_l \lambda_l s_l + \varepsilon \tag{3}$$

where each $\lambda_l \in \mathbb{R}$ is interpreted as a gain applied to each source and ε is an error term. Thus, $\sum_{l \neq g} \lambda_l s_l$ can be interpreted as undesirable interferences (hearing other sources than the desired one), and ε is often interpreted as artefacts introduced by the algorithm. Coefficients λ_l are estimated by least-squares regression, and three metrics are then computed:

$$\begin{aligned}
\mathrm{SDR}_g =& 20 \log_{10} \left(\frac{\|\lambda_g s_g\|_2}{\|\hat{s}_g - \lambda_g s_g\|_2} \right) \\
\mathrm{SIR}_g =& 20 \log_{10} \left(\frac{\|\lambda_g s_g\|_2}{\|\sum_{l \neq g} \lambda_l s_l\|_2} \right)) \\
\mathrm{SAR}_g =& 20 \log_{10} \left(\frac{\|\lambda_g s_g\|_2}{\|\varepsilon\|_2} \right))
\end{aligned}$$

The source-to-distortion measure (SDR) is an overall measure of quality of the source estimate. The source-to-interference-ratio (SIR) and source-to-artifacts-ratio allow a finer diagnosis of the error in the estimate. If the index g is omitted, SDR simply refers to the average $\frac{1}{G} \sum_g \mathrm{SDR}_g$ over all sources.

The SDR of a proposed method must always be compared to that obtained when using the mixed signal as a source estimate (in fact, the mixed signal divided by the number of sources G so that the sum of estimates is equal to the mix): we will refer to this as the *blind guess*. This way, we measure the improvement accomplished rather than an absolute value that is not always meaningful. Another interesting point of comparison is if we use the true spectrograms $X_{g,fn}$ to compute the masking coefficients in Equation 1 : in a sense, this corresponds to the ideal performance of our method.

Table 1 is an example of evaluation: method 1 is compared to a random method, to the blind guess and an oracle estimator ; SDR is displayed for both sources. As we can see, the blind guess yields an SDR close to 0: this is because we use a balanced mix, whereas in the unbalanced case, the blind guess actually becomes a good guess of one of the sources (and a perfect case if only one source is present in the mix). Consequently, we can average SDR over all sources to measure an overall performance, and omit column **blind**.

Other deformations can be allowed in the `bss_eval` toolbox released by [VGF06], but they are more useful in a multichannel setting.

Oracle estimates. When the masking coefficients of Equation 1 are computed using the ground truth values of $X_{g,fn}$, we will refer to the obtained estimates as *oracle* estimates. The notion of oracle estimates has been further studied in [VGP07], so that the term "oracle" actually corresponds to the optimal estimates of the sources that could be found using any method relying on time-frequency masking. The oracle estimates that we defined in this Section are not optimal in this sense, but they are close enough so that they can be considered as an upper-bound of the quality we can achieve using our proposed formulation.

3. Formulations for informed source separation

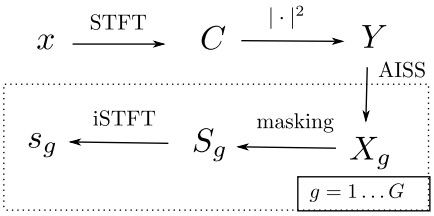


Figure 3: Overview of the source separation process.

Figure 3 summarizes the flow of our source separation procedure: we apply a short-time Fourier transform (STFT) to the observed signal, and take the square modulus, thus discarding the complex phase. Our formulation, which we dub, annotation informed source separation (`AISS`) is then performed. Estimates of the sources S_g are then computed in the time-frequency domain by masking, and mapped back to waveforms s_g.

Now that we have explained the role of time-frequency transforms, we can restrict our attention to a problem of matrix approximation. The observed spectrogram is stored in matrix $Y \in \mathbb{R}^{F \times N}$, where F is the number of frequency bins, and N the number of time bins. We assume that $Y \approx \sum_{g=1}^{G} X_g$, where each matrix $X_g \in \mathbb{R}^{F \times N}$ is the power spectrogram of source g. For each source g, a binary parameter $M_{g,fn} \in \{0, 1\}$ indicates whether we impose a target value to $X_{g,fn}$. Target values are specified by an additional parameter $T_{g,fn}$. For instance, if we want to impose the constraint $X_{g,fn} = 3$, we set $M_{g,fn} = 1$ and $T_{g,fn} = 3$. If on the other hand, we want to optimize $X_{g,fn}$, then we simply set $M_{g,fn} = 0$, regardless of the value of $T_{g,fn}$.

In Equation 4 we display an example where the top row of X_1 and bottom row of X_2 are constrained to have all zeros, and the remaining entries are left free, since the corresponding entries of M are all equal to zero. The values in T will be referred to as *target values*.

$$M_1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad T_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad X_1 = \begin{pmatrix} 0 & 0 & 0 \\ \times & \times & \times \\ \times & \times & \times \end{pmatrix}$$
$$M_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad T_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad X_2 = \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ 0 & 0 & 0 \end{pmatrix} \quad (4)$$

Thus, our problem takes as input a matrix of observations $Y \in \mathbb{R}^{F \times N}$, a sparse three-way array $M \in \mathbb{R}^{F \times N \times G}$, and another sparse three-way array $T \in \mathbb{R}^{F \times N \times G}$. The pair (M, T) represents *user annotations*. We will always assume in the rest of this article that $F \leq N$.

We focus in this article on the numerics of informed source separation, regardless of how annotations were acquired. Recently, there has been much effort in building (semi-)automatic tools to gather annotations, either through a graphical user interface[1] [HDB11, DT12, LBF12, BM13], or using a precalibrated classifier such as random forests[LBF12]. In the latter contribution, it is shown that for simple benchmarks where the sources to be separated are singing voice and accompaniment music, it is possible to retrieve a significant fraction of annotations (up to 20%) with limited error.

We first present a summary of nonnegative matrix approximation with the Itakura-Saito divergence and explicit factors. In previous work [LBF12], we introduced annotation-informed source separation using this technique. In Section 3.2, we introduce a convex formulation of low-rank approximation. This formulation, together with the algorithm presented in Section 4, is the main contribution of this article.

3.1. Constrained nonnegative matrix factorization

As discussed in Section 2.1, matrix factorization is useful in capturing the redundancy of high-dimensional data sets. Each source g is modelled as a product of factors $D_g \in \mathbb{R}^{F \times K}$ with the corresponding activation coefficients $A_g \in \mathbb{R}^{K \times N}$: $X_g = D_g A_g$. The number K of columns of D must be fixed in advance. Since only the sum $Y = \sum_g X_g$ is observed, blind source separation techniques proceed by computing a matrix factorization $Y = DA$ and assign each column of D to one and only one source, thereby obtaining one dictionary D_g and one matrix of activation coefficients A_g for each source. In our setting, annotations act as incomplete observations of each source term X_g.

Allowing inexact observations, estimates of D_g and A_g are obtained by solv-

[1] see also `https://www.youtube.com/watch?v=Rd3prIkO5bg` for a video demonstration

ing the following optimization problem [LBF12]:

$$\begin{array}{ll} \min & L(Y, \sum_{g=1}^{G} D_g A_g)\,, \\ \text{s.t.} & M_g \odot (D_g A_g) = M_g \odot T_g\,, \\ & D_g \geq 0 \\ & A_g \geq 0\,. \end{array} \tag{5}$$

where $L(Y, \hat{Y})$ is a measure of dissimilarity between Y and $\hat{Y}$. $\odot$ stands for pointwise multiplication. Additional nonnegativity constraints are imposed to ensure that each source estimate is nonnegative: indeed, as we saw in Section 2, we need nonnegative estimates in order to form time-frequency masks. Note that imposing nonnegativity of the factors is stronger than imposing nonnegativity of the source terms X_g. For typical values of K in audio source separation, Problem 5 is nonconvex and multimodal, so that only local minima can be computed. In practice, one obtains good source estimates by starting from many initial points and selecting the best solutions, at the cost of increasing the computing time.

As for the choice of $L(Y, \hat{Y})$, the Itakura-Saito divergence was used in [LBF12]:

$$L(Y, \hat{Y}) = \sum_{fn} \frac{Y_{fn}}{\hat{Y}_{fn}} + \log \frac{Y_{fn}}{\hat{Y}_{fn}} - 1$$

The Itakura-Saito is a popular choice in the audio source separation community. Another choice is the generalized Kullback-Leibler divergence. However, such measures of dissimilarity require an additional level of complexity in the optimization. An algorithm to retrieve good candidate solutions was developed in [LBF12], specifically to handle this aspect. In this article, we will use a simpler dissimilarity measure.

3.2. A convex formulation with low-rank inducing penalties

Instead of fixing the rank of the source terms X_g, we introduce a *penalty* function favoring low-rank solutions. Denoting the nuclear norm by $\|X\|_\star$, i.e. the sum of singular values of X, the optimization problem becomes:

$$\begin{array}{ll} \min_X & \overbrace{\frac{1}{2}\|Y - \sum_{g=1}^{G} X_g\|_F^2}^{\hat{f}(X)} + \overbrace{\lambda \sum_{g=1}^{G} \|X_g\|_*}^{\psi(X)} \\ \text{subject to} & M_g \odot X_g = M_g \odot T_g \text{ for all } g = 1 \ldots G \\ & X_g \in \mathcal{C} \text{ for all } g = 1 \ldots G \end{array} \tag{6}$$

Dissimilarity between the input spectrogram and its approximation is now measured with a Frobenius norm : $\|X\|_F = \sqrt{\sum_{f,n} X_{fn}^2}$. This choice is made for convenience, and we will leave the use of more involved dissimilarity measures for future work.

To any rectangular matrix $X \in \mathbb{R}^{F\times N}$, we can associate a singular value decomposition (SVD) $X = P\Sigma Q^\top$, where $P \in \mathbb{R}^{F\times F}$ and $Q \in \mathbb{R}^{F\times N}$ have orthonormal columns and Σ is a diagonal matrix with nonnegative diagonal coefficients $\sigma_1 \geq \ldots \sigma_F$. : the rank of X is thus equal to the number of nonzero elements of Σ. Coefficients σ_i are known as the *singular values* of X.

Instead, of fixing the rank of X_g to a known value, one might attempt to penalize the number of nonzero singular values of X_g, bu that function is not convex. The sum of singular values, also known as nuclear norm $\|X\|_* = \sum_{f=1}^{F}$ has the advantage of being convex with respect to X : we expect that for increasing values of λ, solutions of Problem 6 have many small singular values, and only a few large singular values, hence they will be nearly low rank. This will be further discussed in the experimental section.

Nonnegativity of the solutions X_g is often required in source separation, e.g. when magnitude or power spectrograms are used to compute estimates of the source signals. However, the nonnegativity constraints may be dropped if other time-frequency operators are used (like the modified discrete cosine transform[OS75]). For flexibility, we will allow both possibilities and consider either case $\mathcal{C} = \mathbb{R}^{F\times N}$ or $\mathcal{C} = \mathbb{R}_+^{F\times N}$. In the latter case, the source terms are required to have nonnegative coefficients. We also introduce the notation

$$\mathcal{Q} = \{X, X_g \in \mathcal{C}, M_g \circ X_g = M_g \circ T_g,\ \forall g = 1 \ldots G\}$$

for the *feasible* set. Set $\mathcal{Q}$ is convex, and we will make frequent use of projections on $\mathcal{Q}$ in the following.

The objective function $f(X)$ is a composite function, i.e. it is the sum of a smooth function $\hat{f}(X)$ and a non-smooth function $\psi(X)$: thus, in our formulation, the advantage of dealing with a convex objective function is slightly counterbalanced by the fact that it is also non-smooth, which leads to a more complex optimization problem.

4. Subgradient projection algorithm

As we introduced a convex model (convex objective function and convex feasible region), we can now look for global optima of the informed source separation problem. Since we are dealing with a nonsmooth objective function, we recall a few preliminary notations and basic facts in Section 4.1, before we proceed to a detailed presentation of a subgradient algorithm in Sections 4.2-4.3.

4.1. Notations and basic definitions

In the following, we will consider the vector space $\mathbb{R}^{F\times N\times G}$ endowed with scalar product $\langle A, B\rangle = \sum_g \mathrm{Tr} A_g^\top B_g$, and the induced norm

$$\|A - B\| = \sqrt{\sum_g \|A_g - B_g\|_F^2}$$

where $\|\cdot\|_F$ is the Frobenius norm on $\mathbb{R}^{F\times N}$. We will also make use of the spectral norm of matrices of size $F \times N$:

$$\|W\|_{\text{op}} = \max\{\|Wu\|_2 \text{ s.t. } \|u\|_2 \leq 1\}\,,$$

At times, we might also refer to this norm as the operator norm. Recall that the spectral norm is equal to the largest singular value of W.

We define $\hat{f}(X) = \frac{1}{2}\|Y - \sum_g X_g\|_F^2$ the quadratic part of the objective function, and $\psi(X) = \lambda \sum_g \|X_g\|_*$ the non-differentiable part. Objective function f is defined over all $\mathbb{R}^{F\times N\times G}$, it is convex and Lipschitz continuous, so its directional derivatives are defined in every direction [Roc70]:

$$f'(X;D) = \lim_{t\downarrow 0} \frac{f(X+tD) - f(X)}{t}\,.$$

Note that although the objective function is convex, there may be several globally optimal solutions.

Quantity Z is a *subgradient* of f at X if it satisfies one of the following statements, which are equivalent:

$$\forall Y,\ f(Y) - f(X) \geq \langle Z, Y - X\rangle\,, \tag{7}$$

$$\forall D,\ f'(X;D) \geq \langle Z, D\rangle\,.$$

The set of subgradients of f at X is the subdifferential $\partial f(X)$. It is closed, convex, and it is bounded because $\text{dom} f = \mathbb{R}^{F\times N\times G}$. It is possible to make the link between subgradients and directional derivatives even stronger. In fact, one can show [Sho85], using the theorem of separating hyperplanes, that the inequality in Equation (7) is tight, for an appropriate choice of subgradient:

$$f'(X;D) = \begin{array}{l}\max \langle Z, D\rangle \\ Z \in \partial f(X)\,.\end{array}$$

Subgradients generalize the gradient of differentiable functions: indeed, if f is differentiable then $\partial f(X) = \{\nabla f(X)\}$. As we can see in Equation (7), subgradients extend the well-known result that a convex function is lower-bounded by its tangent at a given point.

4.2. Subgradients and SVD

The subgradient projection algorithm is a simple method for generic non-smooth problems. Given an initial point X^0, and a choice of subgradient Z^k at every step, a subgradient iteration consists in taking a step in the opposite direction of the subgradient, and projecting the result on the feasible set:

$$X^{k+1} = \Pi(X^k - hZ^k) \tag{8}$$

where $h > 0$ is the *step size*, and $\Pi : \mathbb{R}^{F\times N\times G} \mapsto \mathbb{R}^{F\times N\times G}$ is the projection on the feasible set Q. We will also use the notation Π_g to refer to the projection of

each block on its own set of constraints, so $\Pi(X) = (\Pi_g(X_g))_{g=1\dots G}$. Note that in our case, the projection is simple:

$$\Pi(X)_{g,fn} = \begin{cases} T_{g,fn} & \text{if } M_{g,fn} = 1 \\ X_{g,fn} & \text{otherwise} \end{cases} \qquad \text{if } \mathcal{C} = \mathbb{R}^{F\times N\times G}$$

$$\Pi(X)_{g,fn} = \begin{cases} T_{g,fn} & \text{if } M_{g,fn} = 1 \\ (X_{g,fn})_+ & \text{otherwise} \end{cases} \qquad \text{if } \mathcal{C} = \mathbb{R}_+^{F\times N\times G}$$

with the notation $(x)_+ = \max(0, x)$. As we can see, if f is differentiable, we recover the usual gradient projection method. General properties of subgradient algorithms are discussed in Section 4.3.

The gradient of $\hat{f}$ with respect to each matrix X_g is readily seen to be:

$$\nabla \hat{f}(X) = \sum_g X_g - Y\,,$$

which corresponds to the opposite of the residual $R = Y - \sum_g X_g$. Subgradients of ψ are obtained by forming the singular value decomposition (SVD) of source terms X_g. Let us denote the SVD of each source term:

$$\begin{gathered} X_g = P_g \Sigma_g Q_g^\top\,, \\ P_g^\top P_g = I\,, \\ Q_g^\top Q_g = I\,, \\ \Sigma_g = \mathrm{Diag}(\sigma_1, \dots, \sigma_F)\,. \end{gathered}$$

Since $F \leq N$, it is best to compute an 'economy size' SVD, i.e. $P_g \in \mathbb{R}^{F\times F}$, $Q_g \in \mathbb{R}^{F\times N}$, $\Sigma_g \in \mathbb{R}^{F\times F}$. Recall that matrices P_g and Q_g are orthogonal and Σ_g is diagonal.

With these notations at hand, the following are necessary and sufficient conditions for $Z \in \mathbb{R}^{F\times N\times G}$ to be a subgradient of f at X [RFP10] :

$$\begin{gathered} Z_g = -R + \lambda P_g Q_g^\top + W_g\,, \\ W_g^\top X_g = 0\,, \\ W_g X_g^\top = 0\,, \\ \|W_g\|_{\mathrm{op}} \leq \lambda\,. \end{gathered} \tag{9}$$

$W_g \in \mathbb{R}^{F\times N}$ is a free variable subject to the constraints that its row space and column space are respectively orthogonal to those of X_g, and that its operator norm is less than λ.

Since the subgradient algorithm only requires to choose an arbitrary subgradient at each iteration, we simply choose $W = 0$.

Algorithm 1 Subgradient algorithm for informed source separation.

Input: $Y \in \mathbb{R}^{F\times N}$, $M \in \{0,1\}^{F\times N\times G}$, $T \in \mathbb{R}^{F\times N\times G}$, $\lambda > 0$, $h^0 > 0$
  $k \leftarrow 0$
  Initialization : $X_g \leftarrow \Pi_g(Y/G)$ for $g = 1 \dots G$.
  repeat
    $k \leftarrow k + 1$
    Store current iterate $\tilde{X} \leftarrow X$
    Update step length $h \leftarrow \frac{h^0}{\sqrt{k+1}}$
    $R \leftarrow Y - \sum_g X_g$
    for $g = 1, \dots, G$ **do**
      $(P, \Sigma, Q) \leftarrow \mathrm{svd}(X_g, \text{'econ'})$
      $X_g \leftarrow X_g + h(R - \lambda P Q^\top)$.
      $X_g \leftarrow \Pi_g(X_g)$
    end for
  until $\|X - \tilde{X}\| \leq \epsilon$

4.3. Overview and comments

Algorithm 1 sums up the subgradient projection algorithm that we use in our experiments. The user should specify in adavance whether $\mathcal{C} = \mathbb{R}^{F\times N}$ or $\mathcal{C} = \mathbb{R}_+^{F\times N}$. In the latter case, it should also be checked that $T_{g,fn} \geq 0$.

Defining an appropriate stopping criterion is quite involved, in view of the non-differentiability of the objective function and the presence of constraints. We choose $\|X^{k+1} - X^k\| \leq \epsilon$ as a stopping criterion. Recall from Section 4.1 that $\|A - B\| = \sqrt{\sum_g \|A_g - B_g\|_F^2}$.

Iteration complexity. The cost of each iteration of the subgradient projection method is dominated by the computation of one SVD for each matrix X. This amounts to $O(G(F^2N + F^3))$ floating point operations [GL96]. We split the update of the gradient term $\nabla f(X^k)$ in two places: the residual term is computed only once as it appears in all blocks $\nabla f(X^k)_g \in \mathbb{R}^{F\times N}$, while SVDs are computed one by one for each X_g, so that only one triplet (P, Σ, Q) need actually be stored in memory. Projecting on the constraints $M_g \circ X_g = 0$ is linear in the number of *nonzero entries* of $M = (M_1; \dots; M_G)$: this is typically very cheap as the proportion of nonzero entries is only a fraction of the total number of entries in M. Projecting on nonnegativity constraints is linear in the number of entries of each matrix X_g. Note that the order in which we project does not matter.

Rate of convergence. Choosing the step size in the subgradient method is not a trivial task: indeed, unless we can prove that our choice of subgradient ensures a decrease of the cost function at each step, it is not possible to select the step size by line search. In general, we can still guarantee convergence to a minimum

point of Problem 6 as long as [Nes03, Theorem 3.2.2]:

$$\sum_{k=0}^{+\infty} h^k = +\infty$$
$$\sum_{k=0}^{+\infty} (h^k)^2 < +\infty$$

One possible choice, which is often advocated, is to fix h^0 arbitrarily and choose $h^k = \frac{h^0}{\sqrt{k+1}}$. In this case, the approximation error $f(X^k) - f(X^*) \leq \epsilon$ decreases, in function of the number of iterations, as $O\left(\frac{\ln(k+1)}{\sqrt{k+1}}\right)$.

5. Numerical experiments

We have presented a convex formulation for informed source separation and derived an algorithm to solve it, based on the subgradient method. Our aim in this experimental section is to show that it compares favourably with NMF on a benchmark of professionally produced music recordings. Before we can do so, however, we will examine in Sections 5.1 - 5.4 a certain number of factors that influence the performance of our algorithm. One may skip these sections at first reading and proceed to the comparison with NMF in Section 5.5, after taking a look at the paragraph on *Audio settings*.

As mentioned earlier, an important point raised by [LBF12, BM13] is the sensitivity of solutions to the quality of annotations. We discuss in Section 5.1 two possible choices of annotations: either imposing zeroes in source terms, letting large amplitude coefficients $X_{g,fn}$ be determined by the low-rank approximation ; or imposing large amplitude coefficients as well, which would imply in practice to have precise estimates of the masking coefficients.

We discuss step size selection in Section 5.2. In Section 5.3, we check that the nuclear norm term in our convex formulation has the desired effect of favoring low-rank solutions. In Section 5.4, we examine the influence of the sparsity-inducing parameter λ on the quality of source estimates. We perform experiments on a randomly selected audio track to discuss these points.

Finally, we compare in Section 5.5 our convex formulation and the NMF formulation presented in Section 3.1 on a benchmark database from the SISEC evaluation campaign, on which state-of-the-art methods in source separation are evaluated every year. Still on this benchmark, we evaluate the influence of the proportion of annotations on source separation results. Finally, the relationship between the proportion of annotations and the quality of source estimates is made precise in Section 5.6.

Audio settings. All experiments are performed on real audio signals. Unless explicitly mentioned, simulations are performed on a randomly chosen audio track from the SISEC database of professional music recordings. The duration of the track is between 10 and 25 seconds. All tracks were sampled at 16 kHz, STFTs computed with sinebell windows of length 512 samples (i.e. 32 ms), with an overlap of 50% between windows. This yields input matrices Y with 512 rows and between 800 and 2000 columns. The number of rows is reduced

choice	run 1	run 2	run 3	run 4	run 5	mean	std
$T = 0$	5.17	5.03	4.98	5.11	5.04	5.06	0.08
T general	4.61	4.58	4.50	4.28	4.66	4.52	0.15

Table 2: Imposing only zeroes in T yields better SDR values with high probability.

from 512 to 257 by exploiting Hermitian symmetries in the Fourier transform. In all tracks, the two sources are accompaniment $g = 1$ and voice $g = 2$.

5.1. Choice of annotations

Since we have at our disposal the ground truth sources, we can compute ideal target values for the annotations: $M_{g,fn} = 1$, and $T_{g,fn}$ is obtained by Equation 1. In this case, there is no need to solve Problem 6, since the feasible set reduces to one point. In this case we obtain an oracle estimate. In reality, we only expect a fraction of the coefficients of M to be equal to 1. In [BM13, LBF12], estimates of the target values $T_{g,fn}$ are provided by the user, or using an external "plugin" [LBF12].

Since the solutions of Problem (6) are quite sensitive to specific values of T, one possibility is to only pick $M_{g,fn} = 1$ when the associated target value is $T_{g,fn} = 0$. In view of the W-disjoint orthogonality property discussed in Section 2.1, we expect that a large fraction of the coefficients in T fall in this subset, so that good enough source estimates can be retrieved.

We compare here these two possible choices of annotations in a controlled experimental setting, where coefficients of M can be selected according to the known value of T. In the first scenario, we sample each $M_{g,fn}$ at random in $\{0, 1\}$ with probability p of choosing 1. Corresponding target values are chosen as $T_{g,fn} = \text{mask}_{g,fn} V_{fn}$: this way, we impose a proportion p of the entries of X to correspond to the oracle estimates [VGP07].

The second scenario consists in imposing only zero values, i.e. $T_{g,fn} = 0$ for all (f, n, g). Only a fraction of the FNG entries of X can be constrained, in this case. For fair comparison, for a given value of p, we sample M and T as follows: first we threshold some values of T as $T_{g,fn} = 0$ if $\text{mask}_{g,fn} < \max_h \text{mask}_{h,fn}$. We then compute the proportion ρ of zero entries of T, and sample $M_{gfn} = 1$ with probability $\frac{p}{\rho}$. In this way, we know that the expected number of constraints is always $pFNG$ for small enough values of p. When $G = 2$, this means that $p < \frac{1}{2}$: we cannot annotate more than 50% of the $2FN$ entries, since either source 1 or source 2 is dominant. If $G = 3$, p can be up to 66%, and so on. In any case, this restriction is not too strong since we are expecting our system to work with small fractions of annotations.

We display in Table 2 the average SDR obtained on 5 simulated annotations masks, with either choice (T general or $T = 0$), and $p = 0.2$. Note that the actual number of constraints relative to FNG was equal to 0.2 up to 10^{-2}. It turns out that even though the target values $T = 0$ are not exact, it is better

to constrain entries of X_g to be small than constrain entries of X to be equal to a large and inexact value.

5.2. Influence of step size in subgradient descent

The subgradient method we use is sensitive to the choice of the initial step size. Figure 4 displays the objective function value versus the allowed CPU time for the subgradient algorithm: as we can see, too small values of h^0 yield suboptimal solutions. On the other hand, too high values of h^0 not only yield suboptimal solutions after 60 seconds, but also give very bad results in early iterations.

One should bear in mind that the choice of step size is not so crucial if one allows enough CPU time, since iterates are guaranteed to converge to the global minimum ($X \to X^*$). If the allowed CPU time (or number of iterations) is limited, a practical strategy consists in running a few tens of iterations with various values of h^0, keep the one which yields the lowest objective cost value, and use it for the rest of the allowed time. See also Section 4.2 for more comments about step size selection.

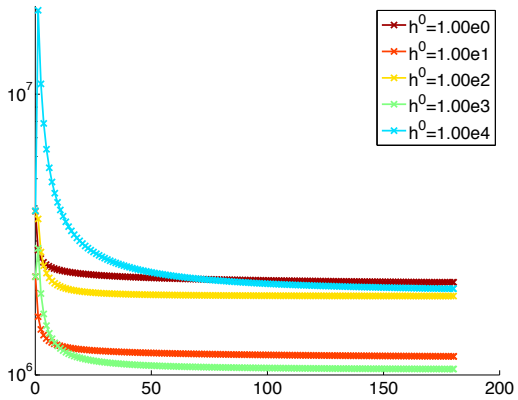


Figure 4: Evolution of the objective function value VS allowed CPU time, for various values of h^0.

5.3. Influence of λ on singular value profile

Figure 5 displays the singular value profile of one of the estimated sources X_g, for various values of the sparsity-inducing penalty parameter λ. The magnitude of singular values is displayed in log scale so that we can compare the influence of λ more precisely. Although the magnitude of a particular singular value is not interpretable, it is useful to count the number of singular values less than an arbitrary threshold. For instance, we can see that the number of singular values

less than 10^{-1} is around 60 for $\lambda = 10^{-1}$, 190 for $\lambda = 10^{2}$, and 0 for $\lambda = 10^{-2}$. This is consistent with our expectation that the nuclear norm penalty favors solutions that are approximately low rank.

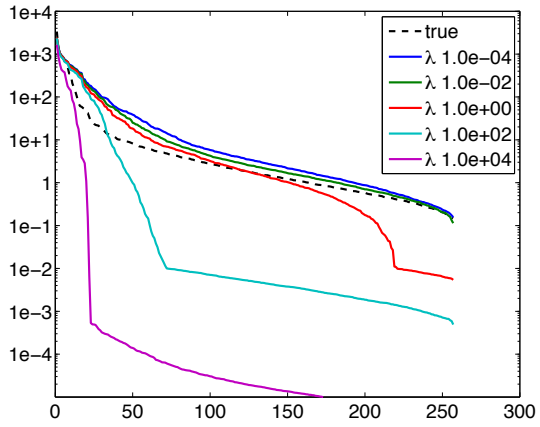


Figure 5: Magnitude of singular values in decreasing order, for various values of λ. Dotted line is the true singular value profile.

5.4. Sensitivity of solutions to λ

Choosing regularization parameters (number of latent components, coefficient of structured penalties, etc.) in unsupervised techniques such as sparse PCA or the present problem is known to be a very challenging problem. In practice, one can only try a few values and conduct listening tests to determine the best value. In evaluation contexts, where the sources are known, this can be replaced by selecting the values that yield the highest SDR value. As we can see in Table 3, there is much practical importance in tuning λ, once all other parameters are fixed, since it can lead to an improvement of 3 dB over a randomly chosen value.

For $\lambda = 0$, solutions of Problem 6 cannot be expected to yield much improvement, other than that given by imposing annotations. As $\lambda \to +\infty$, we observe in Table 3 that SDR decreases after a certain point. Thus, a certain amount of trial and error is required to find a satisfactory value of λ, but this value is not too sensitive as we can see on Table 4 : choosing λ in the order of 10^1 seems to yield satisfactory performance.

5.5. Comparison of our formulation with NMF on the SISEC database

We are now ready to compare our formulation (dubbed `lownuc`) with the constrained NMF proposed in [LBF12]. We reproduce in this subsection the experiment of [LAG13]. We make this comparison on the whole SISEC PPMR

λ	10^{-4}	10^{-2}	10^{0}	10^{2}	10^{4}
SDR	3.98	4.82	7.33	7.16	4.40

Table 3: Sensitivity of SDR to λ.

λ	10^{-2}	10^{-1}	10^{0}	10^{1}	10^{2}
SDR track 1	3.57	5.95	6.40	6.41	6.22
SDR track 2	4.79	7.93	8.35	8.33	8.17
SDR track 3	2.85	4.36	5.41	5.54	5.56

Table 4: Sensitivity of SDR to λ for three tracks on a finer grid of λ values.

	SDR	**SIR**	**SAR**
lazy	3.47	4.91	10.22
nmf	7.93	16.19	8.82
lownuc	8.78	16.02	9.95
oracle	12.54	22.73	13.03

Table 5: Average results on SISEC database using 40% of annotations. See text for a description of the row labels.

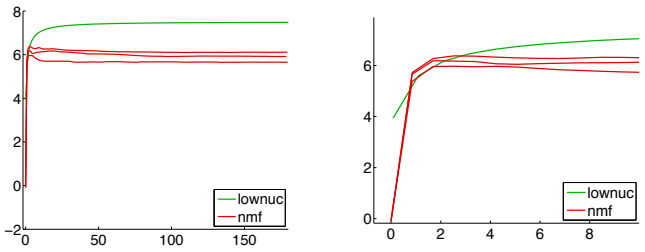
database[ANV+12], which consists of 5 audio tracks with duration between 10 to 25 seconds each. For fairness, parameters λ (for `lownuc`) and K (for NMF) were both selected for maximum value of $\sum_g \mathrm{SDR}_g$, out of a finite number of trial values. Target values T are not constrained to be 0. Again, remember that a given value of p corresponds to $pFNG$ equality constraints, irrespectively of the choice of annotations we make.

We include in our comparison the "oracle" estimates, computed using the true values of the source spectrograms: those are an upper-bound on the accuracy of our method. A sanity check is to compare also to a simple candidate: projecting $X_g = \frac{1}{G}Y$ on the feasible set $\mathcal{Q}$ of Problem (6). As we can see in Table 5, both `nmf` and `lownuc` improve substantially over lazy estimates, with `lownuc` outperforming `nmf` by roughly 0.85 dB on average. In our interpretation, this is because the nonnegativity constraints that we impose are weaker than in NMF.

Despite the simplicity of the subgradient scheme, our approach is also attractive in terms of computing time, as illustrated on Figure 6(a). We display here the SDR against the allowed CPU time. Since NMF is a nonconvex problem, we display the SDR for several initial points (red curves). A closer look at the first few seconds of each run (Figure 6(b)) shows that the subgradient method improves over NMF as soon as the allowed CPU time budget is more than ten seconds of computations.

5.6. Influence of the proportion of annotations

We can now evaluate the results of our approach for various values of p. According to our previous discussion in Section 5.1, we only pick $M_{g,fn} = 1$



(a) Overall (b) First few seconds

Figure 6: (Left) Evolution of SDR as a function of CPU time (in seconds), for (green) our method and (red) NMF started from several initial points.

when $T_{g,fn} = 0$. Hyperparameters were tuned as follows: $\lambda = 1$ as it seems to give good results on all tracks. A CPU time budget of 180 seconds was allowed. Again, remember that a given value of p corresponds to $pFNG$ equality constraints, irrespectively of the choice of annotations we make.

p	SDR	SIR	SAR
5 %	2.33	3.35	10.80
10 %	3.85	5.79	9.37
20 %	6.07	9.99	8.81
30 %	7.78	13.85	9.23
50 %	11.88	26.58	12.07
oracle	12.54	22.73	13.03

Table 6: Mean results on the SISEC database, as the proportion of annotation increases.

Table 6 displays source separation results achieved by `lownuc`. All instrument tracks were mixed together to form the accompaniment (source 1), and vocal tracks (that is, lead vocal and backing vocals if any) mixed together as source 2. Complete results are available online [2]. As we can see, satisfactory results are obtained with as little as 20% of annotations. For 50% of annotations, the computed masks are close to the oracle masking coefficients.

6. Conclusion and perspectives

We have introduced a convex formulation for annotation informed source separation, dubbed `AISS_lownuc`. Instead of explicitly looking for low-rank source estimates, we use nuclear norm terms to favor solutions that are close

[2] `http://www.di.ens.fr/~lefevrea/neurocomp_demo.zip`

to low-rank, as demonstrated in the experimental section. One feature of our formulation is that it handles nonnegativity constraints as an option, which allows several choices of time-frequency operators, such as the modified discrete cosine transform[OS75].

In order to simplify our investigations, we have used a Frobenius norm to measure dissimilarity between the observed spectrogram and its model. In future work, we will consider dissimilarity measures that are more frequently used in the audio source separation community, such as the Kullback-Leibler divergence [BM13].

Our algorithm for `AISS_lownuc` is based on subgradient iterations. Ongoing experiments using smoothing techniques [Nes05] suggest that we can further improve the quality of source estimates by approximating the nonsmooth objective function by a differentiable one with Lipschitz continuous gradient. Another interesting research direction would be to use inexact subgradients, by computing only partial SVDs. By choosing an appropriate algorithm for SVD, we expect that this will reduce the complexity of each iteration.

Experiments on a benchmark of professionally produced music recordings [ANV+12] suggest that imposing zeroes in the source estimates is sufficient and even better than using target values for high coefficients. This is a useful complement to the heuristics proposed in [LBF12, BM13]. The formulation we have proposed for `AISS` is competitive with NMF: even at early iterations, it produces source estimates of superior quality, based on well-established criteria [VGF06].

As expected, the nuclear norm term in our formulation favors approximately low-rank solutions. It is difficult to decide whether exact thresholding may be accomplished, since in our Problem the proximal operator is difficult to compute.

From the theoretical point of view, our convex formulation is definitely more attractive than NMF since we can guarantee that our algorithm converges to a global optimum whereas multiplicative updates algorithms for NMF can only be shown to convergenge to local minima (when they do converge) : furthermore, we can rely on standard rates of convergence for subgradient algorithms to apply to our formulation.

To conclude, we have shown that interactions with user specified constraints allow to construct well-posed semi-supervised learning techniques. User assisted source separation methods are the state of the art in single-channel audio source separation, so it is worth considering algorithms that solve general formulations which would allow to incorporate various user specifications as constraints or penalty functions. Thus, the convex formulation presented in this article opens the possibility of even smarter tools in audio software for creative purposes as well as advanced sound engineering tasks.

7. References

References

[AGB10] S. Arberet, R. Gribonval, and F. Bimbot, *A robust method to count and locate audio sources in a multichannel underdetermined mixture*, IEEE Transactions on Signal Processing (2010), 121–133.

[ANV+12] Shoko Araki, Francesco Nesta, E. Vincent, Zbynek Koldovsky, Guido Nolte, Andreas Ziehe, and Alexis Benichoux, *The 2011 Signal Separation Evaluation Campaign (SiSEC2011): - Audio source separation* -, 10th Int. Conf. on Latent Variable Analysis and Signal Separation (LVA/ICA) (Tel Aviv, Israel), March 2012, pp. 414–422.

[BM13] N.J. Bryan and G.J. Mysore, *Interactive refinement of supervised and semi-supervised sound source separation estimates*, ICASSP, 2013.

[BPSS10] S. Bengio, F. Pereira, Y. Singer, and D. Strelow, *Group Sparse Coding*, Advances in Neural Information Processing Systems (NIPS), 2010, Experimental results not convincing. groups are across observations, for each image. Codes are nonnegative, but not the dictionary.

[DT12] J.-L. Durrieu and J.-P. Thiran, *Musical Audio Source Separation Based on User-Selected F0 Track*, International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA), Mar. 2012.

[FBD09] C. Févotte, N. Bertin, and J.-L. Durrieu, *Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis*, Neural Computation **21** (2009), no. 3, 793–830.

[FBR12] B. Fuentes, R. Badeau, and G. Richard, *Blind Harmonic Adaptive Decomposition applied to supervised source separation*, European Signal Processing Conference (EUSIPCO), IEEE, 2012, pp. 2654–2658.

[Fév11] C. Févotte, *Majorization-minimization algorithm for smooth Itakura-Saito nonnegative matrix factorization*, International Conference on Acoustics Speech and Signal Processing (ICASSP) (Prague, Czech Republic), May 2011.

[GL96] G.H. Golub and C.F. Van Loan, *Matrix Computations*, John Hopkins University Press, 1996.

[GSD12] J. Ganseman, P. Scheunders, and S. Dixon, *Improving PLCA-based score-informed source separation with invertible constant-Q transforms*, European Signal Processing Conference (EUSIPCO), 2012.

[HBD10] R. Hennequin, R. Badeau, and B. David, *NMF with time-frequency activations to model non stationary audio events*, International Conference on Acoustics Speech and Signal Processing (ICASSP), March 2010.

[HDB11] R. Hennequin, B. David, and R. Badeau, *Score informed audio source separation using a parametric model of non-negative spectrogram*, ICASSP, 2011.

[LAG13] A. Lefèvre, P.-A. Absil, and F. Glineur, *A nuclear norm-based convex formulation for informed source separation*, European Symposium on Artificial Neural Networks, 2013.

[LBF12] A. Lefèvre, F. Bach, and C. Févotte, *Semi-supervised NMF with time-frequency annotations for single-channel source separation*, International Conference on Music Information Retrieval (ISMIR), 2012.

[LS99] D.D. Lee and H.S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature **401** (1999), no. 6755, 788–791.

[MSR10] G.J. Mysore, P. Smaragdis, and B. Raj, *Non-negative hidden Markov modeling of audio with application to source separation*, International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA), 2010.

[Nes03] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer, 2003.

[Nes05] Yu. Nesterov, *Smooth minimization of non-smooth functions*, Mathematical Programming (2005), 127–152.

[OFBD11] A. Ozerov, C. Févotte, R. Blouet, and J.-L. Durrieu, *Multichannel nonnegative tensor factorization with structured constraints for user-guided audio source separation*, International Conference on Acoustics Speech and Signal Processing (ICASSP) (Prague, Czech Republic), May 2011.

[OS75] A. Oppenheimer and R. Schafer, *Digital signal processing*, Prentice-Hall, 1975.

[PT94] P. Paatero and U. Tapper, *Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values*, Environmetrics (1994).

[RFP10] B. Recht, M. Fazel, and P.A. Parrilo, *Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization*, SIAM Review. (2010).

[Roc70] R.T. Rockafellar, *Convex analysis*, Princeton University Press, 1970.

[SB03] P. Smaragdis and J.C. Brown, *Non-negative matrix factorization for polyphonic music transcription*, IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2003.

[Sho85] N. Z. Shor, *Minimization methods for nondifferentiable functions*, Springer-Verlag, 1985.

[SSR09] P. Smaragdis, M.V. Shashanka, and B. Raj, *A sparse non-parametric approach for single channel separation of known sounds.*, Advances in Neural Information Processing Systems (NIPS), December 2009.

[VBB10] E. Vincent, N. Bertin, and R. Badeau, *Enforcing Harmonicity and Smoothness in Bayesian Non-negative Matrix Factorization Applied to Polyphonic Music Transcription*, IEEE Transactions on Audio, Speech, and Language Processing **18** (2010), no. 3, 538–549.

[VGF06] E. Vincent, R. Gribonval, and C. Févotte, *Performance measurement in Blind Audio Source Separation*, IEEE Transactions on Audio Speech and Language Processing **14** (2006), no. 4.

[VGP07] E. Vincent, R. Gribonval, and M.D. Plumbley, *Oracle estimators for the benchmarking of source separation algorithms*, Signal Processing **87** (2007), no. 8.

[Vir07] T.O. Virtanen, *Monaural Sound Source Separation by Non-Negative Matrix Factorization with Temporal Continuity and Sparseness Criteria*, IEEE Transactions on Audio Speech and Language Processing **15** (2007), no. 3.

[YR04] O. Yilmaz and S. Rickard, *Blind separation of speech mixtures via time-frequency masking*, IEEE Transactions on Signal Processing (2004).