Supervised non-negative matrix factorization for audio source separation

Pablo Sprechmann[1], Alex M. Bronstein[2], and Guillermo Sapiro[3]

[1] New York University, pablo.sprechmann@nyu.edu
[2] Tel Aviv University & Duke University, bron@eng.tau.ac.il
[3] Duke University, guillermo.sapiro@duke.edu

Summary. Source separation is a widely studied problems in signal processing. Despite the permanent progress reported in the literature it is still considered a significant challenge. This chapter first reviews the use of non-negative matrix factorization (NMF) algorithms for solving source separation problems, and proposes a new way for the supervised training in NMF. Matrix factorization methods have received a lot of attention in recent year in the audio processing community, producing particularly good results in source separation. Traditionally, NMF algorithms consist of two separate stages: a training stage, in which a generative model is learned; and a testing stage in which the pre-learned model is used in a high level task such as enhancement, separation, or classification. As an alternative, we propose a task-supervised NMF method for the adaptation of the basis spectra learned in the first stage to enhance the performance on the specific task used in the second stage. We cast this problem as a bilevel optimization program efficiently solved via stochastic gradient descent. The proposed approach is general enough to handle sparsity priors of the activations, and allow non-Euclidean data terms such as β-divergences. The framework is evaluated on speech enhancement.

Key words: Supervised learning, tast-specific learning, bilevel optimization, NMF, speech enhancement, source separation.

1 Introduction

The problem of isolating or enhancing an audio signal recorded in a noisy environment has been widely studied in the signal processing community [1, 2]. It becomes particularly challenging in the presences of non-stationary background noise, which is a very common situation in many applications encountered, e.g., in mobile telephony. In this chapter we address the problem of monaural source separation by applying matrix factorization algorithms on a transformed domain given by time-frequency representations of the signals.

The decomposition of time-frequency representations, such as the power or magnitude spectrogram, in terms of elementary atoms of a dictionary, has

become a popular tool in audio processing. While many matrix factorization approaches have been used, models imposing non-negativity in their parameters have been proven to be significantly more effective for modeling complex audio mixtures. The non-negativity constraint ensures a parts-based decomposition [3], in which the elementary atoms can be thought as constructive building blocks of the input signal corresponding to interpretable spectral patterns of recurrent events. Non-negative matrix factorization (NMF) [3], and its probabilistic counterpart, the probabilistic latent component analysis (PLCA) [4], are the first instances of a great variety of approaches proposed over the last few years, see [5] for a recent reveiw.

NMF can be applied with different levels of supervision [6, 7]. In this work we are interested in the supervised use of NMF, in which it is assumed that one has access to example audio signals at a training stage. In this setting, NMF is used to take advantage of the available data by pre-computing dictionaries that accurately represent the input signals. NMF has been successfully used in a great variety of audio processing problems ranging from music information retrieval to speech processing. In most approaches, the trained dictionaries are used to facilitate a high-level task, such as speech separation [8, 9, 10, 11, 12], robust automatic speech recognition [13, 14], and bandwidth extension [15, 16], among many others. In the great majority of these approaches the dictionaries are pre-trained independently as a separate initial step not adapted to the subsequent (and ultimate) high level task. Initial works have recently shown the benefit of incorporating the actual objective of source separation into the training of the model, for example in NMF [17, 18] and deep neural network based separation [19]. It is worth mentioning that, in the context of classification, NMF has been also trained optimized in a discriminate way [20, 21].

In this chapter we discuss in detail a supervised dictionary learning scheme that can be tailored for different specific high level tasks [17]. Following recent ideas proposed in the context of sparse coding [22], our training scheme is formulated as a bilevel optimization problem, which can be efficiently solved using standard stochastic optimization techniques. We use speech denoising as an example illustrating the power of the proposed framework. However, this technique is general and can be used for various audio applications involving NMF. We also show that these ideas can be employed in general regularized versions of NMF.

This chapter is organized as follows. In Section 2 we begin by briefly summarizing NMF (and several of its commonly used extensions) in the context of audio source separation. We present the proposed supervised NMF framework in Section 3 and describe how to solve the asociated optimization problem in Section 4. Experimental results are presented in Section 5. In Section 6 we conclude the paper and discuss future lines of work.

2 Source separation via NMF

We consider the setting in which we observe a temporal signal $x(t)$ that is the sum of two speech signals $x_i(t)$, with $i = 1, 2$,

$$x(t) = x_1(t) + x_2(t), \tag{1}$$

and we aim at finding estimates $\widehat{x_i}(t)$. Let us define $\mathbf{x} \in \mathbb{R}^N$, a sampled version of the input signal satisfying, $x[n] = x(\frac{n}{f_s})$. with $n = 1, \ldots, N$, where f_s is the sampling rate.

NMF-based source separation techniques typically operate in two stages. First, the signal is represented in a feature space given by a non-linear analysis operator, typically defined (in the case of audio signals) as the magnitude of a time-frequency representation such as the Short-Time Fourier Transform (STFT). Then, a synthesis operator, given by the NMF, is applied to produce an unmixing in the feature space. The separation is obtained by inverting these representations. Performing the separation in the non-linear representation is key to the success of the algorithm. The magnitude of the STFT is in general sparse (simplifying the separation process) and invariant to variations in the phase (local translations), thus freeing the NMF model from learning this irrelevant variability. This comes at the expense of inverting the unmixed estimates in the feature space, which is a well known problem usually referred to as the phase recovery problem [23].

Let us denote by $\mathbf{V} = \Phi(\mathbf{x}) \in \mathbb{R}^{m \times n}$ a time frequency representation of $\mathbf{x}$, comprising m frequency bins and n (usually overlapping) temporal frames. When the feature extractor Φ is able to produce sparse representations of the sources (such as in the STFT), the following approximation holds,

$$\Phi(\mathbf{x}) \approx \Phi(\mathbf{x}_1) + \Phi(\mathbf{x}_2),$$

for sufficiently distinct signals. The sum is approximate due to the non-linear effects of the phase. In such a setting, NMF attempts to find the non-negative activations $\mathbf{H}_i \in \mathbb{R}^{q \times n}$, $i = 1, 2$, best representing the different components in two non-negative dictionaries $\mathbf{W}_i \in \mathbb{R}^{m \times q}$. This task is achieved through the solution of the minimization problem

$$\min_{\mathbf{H}_i \geq 0} D(\mathbf{V} | \sum_{i=1,2} \mathbf{W}_i \mathbf{H}_i) + \lambda \sum_{i=1,2} \psi(\mathbf{H}_i) \ . \tag{2}$$

The first term in the optimization objective is a divergence measuring the dissimilarity between the input data $\mathbf{V}$ and combination of the estimated channels. Typically, this data fitting term is assumed to be separable,

$$D(\mathbf{A}|\mathbf{B}) = \sum_{i,j} D(a_{ij}|b_{ij}).$$

Significant attention has been devoted in the literature to the case in which the scalar divergence D in the right-hand side belongs to the family of the β-divergences [24],

$$D_\beta(a|b) = \begin{cases} \frac{a}{b} - \log\frac{a}{b} - 1 & : \beta = 0, \\ a\log a/b + (a-b) & : \beta = 1, \\ \frac{1}{\beta(\beta-1)}(a^\beta + (\beta-1)b^\beta - \beta a b^{\beta-1}) & : \text{otherwise.} \end{cases}$$

This family includes the three most widely used cost functions in NMF: the squared Euclidean distance ($\beta = 2$), the Kullback-Leibler divergence ($\beta = 1$), and the Itakura-Saito divergence ($\beta = 0$). For $\beta \geq 1$, the divergence is convex. The case of $\beta = 0$ is attractive despite the lack of convexity, due to the scale-invariance of the Itakura-Saito divergence, which makes the NMF procedure insensitive to volume changes [25].

The second term in the minimization objective is included to promote some desired structure of the activations. This is done using a designed regularization function ψ, whose relative importance is controlled by the parameters λ.

Once the optimal activations are solved for, the spectral envelopes of each source are estimated as $\mathbf{W}_i\mathbf{H}_i$. Since these estimated spectrum envelopes contains no phase information, a subsequent phase recovery stage is necessary. When the non-linearity is imposed as the magnitude of an invertible transform, $\mathcal{F}$, such as the STFT, a simple filtering strategy can be used [12]. In this case we have $\Phi(\mathbf{x}) = |\mathcal{F}\{\mathbf{x}\}|$, where $\mathcal{F}\{\mathbf{x}\} \in \mathbb{C}^{m\times n}$ is a complex matrix. This strategy resembles Wiener filtering and has demonstrated very good results in practice. The recovered spectral envelopes are used to build soft masks to filter the input mixture signal,

$$\hat{\mathbf{x}}_i = \mathcal{F}^{-1}\left\{\mathbf{M}_i \circ \mathcal{F}\{\mathbf{x}\}\right\}, \quad \text{with} \quad \mathbf{M}_i = \frac{(\mathbf{W}_i\mathbf{H}_i^*)^p}{\sum_{j=1,2}(\mathbf{W}_j\mathbf{H}_j^*)^p}, \tag{3}$$

where $\mathbf{H}_i^*$ are the optimal activations obtained after solving (2), where multiplication denoted $\circ$, division, and exponentials are element-wise operations. The parameter p defines the smoothnes of the mask. Note that when p goes to infinity, the mask becomes binary, choosing for each bin the larger of the two signals.

In this section we assumed that the dictionaries for each source were available beforehand for performing the demixing. This corresponds to a supervised version of NMF, in which the dictionaries for each source are trained independently from available training data. Specifically, this is achieved by solving

$$\min_{\mathbf{H}_i, \mathbf{W}_i \geq \mathbf{0}} D(\mathbf{V}_i|\mathbf{W}_i\mathbf{H}_i) + \lambda\,\psi(\mathbf{H}_i) \tag{4}$$

on a training set $\mathbf{V}_i$ of feature representations of the unmixed signals for each source.

As mentioned above, the underlying assumption is that the signals forming the mixture, and consequently the learned dictionaries, are sufficiently distinct to be unambiguously decomposed into $\mathbf{V} \approx \sum_{i=1,2}\mathbf{W}_i\mathbf{H}_i$. However, this assumption is often violated in practice, for which we would want to have

the dictionaries $\mathbf{W}_i$ as incoherent as possible. In other words, the independently trained dictionaries do not ensure that the solutions $\mathbf{W}_1\mathbf{H}_1$ and $\mathbf{W}_2\mathbf{H}_2$ obtained from (2) will resemble the original components of the mixture.

2.1 Case study

The method proposed in this paper, described in Section 3, can be applied to a large family of approaches following the supervised NMF paradigm. In this paper, we opted to use a sparsity-regularized version of NMF as a case study. In this case, the regularizer ψ in (2) is given by the columns-wise ℓ_1 norm,

$$\psi(\mathbf{H}) = \lambda\|\mathbf{H}\|_1 + \frac{\mu}{2}\|\mathbf{H}\|_2^2. \tag{5}$$

For technical reasons, that will be clear in Section 4, we also include an ℓ_2 regularizer on the activations.

3 Supervised NMF

As was discussed in the previous section, the optimization problem (5) is merely a proxy to the desired estimation problem. Standard dictionary learning applied independently to each source does not guarantee that its solutions will produce the best estimate of the unmixed sources even on mixtures created from the training data. Ideally, we would like to train dictionaries that explicitly maximize the performance directly on the source separation problem. In this section we describe a way of better posing this problem in the context of NMF.

Given a mixed input signal, $\mathbf{x}$, the method described in Section 2 defines an estimator of the signal components $\hat{\mathbf{x}}_i(\mathbf{W}_1, \mathbf{W}_2, \mathbf{x})$, where we made explicit their dependence on the dictionaries and the input signal. Ideally we would like to train the signal dictionaries to minimize the expected estimation risk of the estimation, for example, in terms of the mean squared error (MSE),

$$\{\mathbf{W}_i\}_{i=1,2} = \underset{\mathbf{W_i}\geq 0}{\operatorname{argmin}} \sum_{i=1,2} \mathbb{E}_{\mathbf{x}_1,\mathbf{x}_2}\Big\{\|\mathbf{x}_i - \hat{\mathbf{x}}_i(\mathbf{W}_1, \mathbf{W}_2, \mathbf{x}_1 + \mathbf{x}_2)\|^2\Big\}.$$

Assuming that the signals are independent, we can write this expression as,

$$\{\mathbf{W}_i\}_{i=1,2} = \underset{\mathbf{W_i}\geq 0}{\operatorname{argmin}} \iint \sum_{i=1,2} \|\mathbf{x}_i - \hat{\mathbf{x}}_i(\mathbf{W}_1, \mathbf{W}_2, \mathbf{x}_1 + \mathbf{x}_2)\|^2 dP(\mathbf{x}_1)dP(\mathbf{x}_2),$$

where P are the distributions of each source. In practice, these distributions are latent; a common strategy to overcome this problem is to approximate the expected risk by computing the empirical risk over a finite set of training examples sampled from the source distributions. In what follows, we denote

by $\mathcal{X}_i$ the available sets of training signals for each source. Then, the empirical risk is given by

$$\{\mathbf{W}_i\}_{i=1,2} = \underset{\mathbf{W_i}\geq 0}{\operatorname{argmin}} \frac{1}{|\mathcal{X}|} \sum_k \sum_{i=1,2} \|\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k(\mathbf{W}_1, \mathbf{W}_2, \mathbf{x}^k)\|_2^2 \tag{6}$$

where the first sum (with the index k) goes over the elements in the product set, $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$, containing all possible pairs of training signals. We used $\mathbf{x}^k = \mathbf{x}_1^k + \mathbf{x}_2^k$ to simplify the notation. While the empirical risk measures the performance of the estimators over the training set, the expected risk measures the expected performance over new data samples following the same distribution, that is, the generalization capabilities of the model. We can expect a good generalization when sufficient representative training data are available in advance.

When the feature space is given by an invertible transformation, the MSE in (6) can be computed in the (complex) transformed domain. From Parseval's theorem it follows that (6) is equivalent to

$$\{\mathbf{W}_i\}_{i=1,2} = \underset{\mathbf{W_i}\geq 0}{\operatorname{argmin}} \frac{1}{|\mathcal{X}|} \sum_k \sum_{i=1,2} \|\mathcal{F}\{\mathbf{x}_i^k\} - \mathbf{M}_i(\mathbf{W}_1, \mathbf{W}_2, \mathbf{x}^k)\mathcal{F}\{\mathbf{x}^k\}\|^2. \tag{7}$$

Note that the transformed representations $\mathcal{F}\{\mathbf{x}_i^k\}$ of the signals are complex.

As it was discussed in Section 2, the standard setting for supervised NMF estimates the signal dictionaries independently solving (4) for each source. This approximation is pragmatic rather than principled, since the empirical loss given in (6) (or (7)) is difficult to compute. While the estimators $\hat{\mathbf{x}}_i$ (or the masks $\mathbf{M}_i$) are functions of the dictionaries and the mixture signal, they cannot be computed in closed form as they depend on the solution of the optimization problem (2). Such optimization problems are referred to as *bilevel*. In the following section we describe how to solve the bilevel NMF dictionary learning problem when the divergence used in (2) is a convex $\beta-$divergence with appropiate regularization.

Finally, we note that another dificulty posed by the proposed training regime (common to any discriminative approach to source separation [18, 19]) is that the estimation of the dictionaries needs to be computed over the product set rather than each training set independently. This naturally increases the computational load of the training stage, however, it might not be a serious limitation as this can be done in an offline manner without affecting the computational load at testing time.

4 Optimization

As in any empirical risk minimization task, both formulations (6) and (7), are written as the average over a training set of a given cost function. We are

going to adopt the formulation in the frequency domain, given in (7), since it has the aditional advantage that can be easily separable on a frame-wise maner.

For now, we will assume that the regularizer in (2) is frame-wise spearable, and defer the discussion of the more general case to Section 4.3. In this way, the cost function of the NMF problem also becomes frame-wise separable. In order to aleviate the notation, we are going to write the minimization of the empirical risk over a collection of frames rather than the actual audio signals. With this notation, the training data are composed by the set $\mathcal{X}_f$ containg pairs of frames of the form $(\mathbf{f}_1^j, \mathbf{f}_2^j)$, being $\mathbf{f}_i^j \in \mathbb{C}^m$ the $j-$th frame in the collection, corresponding to one column of the time frequency representation, $\mathcal{F}\{\mathbf{x}_i^k\}$, of some signal, $\mathbf{x}_i^k$, in the original training set of signals $\mathcal{X}_i$. Now we denote the mixture as $\mathbf{f}^j = \mathbf{f}_1^j + \mathbf{f}_2^j$. Let us define the loss function

$$\ell(\mathbf{f}_1, \mathbf{f}_2, \mathbf{W}_1, \mathbf{W}_2, \mathbf{h}_1^*, \mathbf{h}_2^*) = \sum_{i=1,2} \|\mathbf{f}_i - \mathbf{M}_i(\mathbf{W}_1, \mathbf{W}_2, \mathbf{f}, \mathbf{h}_1^*, \mathbf{h}_2^*)\, \mathbf{f}\|^2, \quad (8)$$

where we made explicit the dependency of ℓ and the masks on the optimal activations $\mathbf{h}_1^*$ and $\mathbf{h}_2^*$. These optimal activations are themselves functions of the input mixture and the dictionaries, $\mathbf{h}_i^* = \mathbf{h}_i^*(\mathbf{f}, \mathbf{W}_1, \mathbf{W}_2)$, and are obtained by solving the frame-wise version of (2) given by,

$$\{\mathbf{h}_i^*\}_{i=1,2} = \underset{\mathbf{h}_i \geq 0}{\operatorname{argmin}}\ D_\beta(\mathbf{v} | \sum_{i=1,2} \mathbf{W}_i \mathbf{h}_i) + \sum_{i=1,2} \lambda \psi(\mathbf{h}_i)\ , \quad (9)$$

where, following previous notation, $\mathbf{v} = \Phi(\mathbf{f})$, and we explicitly wrote a ridge regression term controled by the non-negative parameter μ. This is included to guarantee that (9) is strictly convex and has a unique solution. The supervised NMF problem can be stated as the optimization program given by

$$\{\mathbf{W}_i\}_{i=1,2} = \underset{\mathbf{W_i} \geq 0}{\operatorname{argmin}} \frac{1}{|\mathcal{X}_f|} \sum_j \ell(\mathbf{f}_1^j, \mathbf{f}_2^j, \mathbf{W}_1, \mathbf{W}_2, \mathbf{h}_1^*, \mathbf{h}_2^*). \quad (10)$$

This optimization problem is referred to as bilevel, with (10) and (9) being the high and low level problems, respectively. It is important to notice that while (10) depends on knowing the ground truth demixing, (9) only depends on the mixture signal, hence matching exactly the situation encountered at testing. As NMF itself, this bilevel optimization problem is non-convex. Hence, we aim at finding a good local minimizer. In what follows, we describe the general optimization algorithm used for this purpose.

4.1 Stochastic gradient descent

Problem (9) has a unique solution when $\beta \geq 1$ and $\mu > 0$, due to the strict convexity of the objective. In this situation, a local minimizer of (10) can be found via (projected) stochastic gradient descent (SGD) [26]. SGD is a

gradient descent optimization algorithm for minimizing an objective function expressed as a sum or average of some training data of an almost-everywhere differentiable function. At each iteration, the gradient of the objective function is approximated using a randomly picked sub-sample.

At iteration j we randomly draw a sample pair from the training set of frames $\mathcal{X}_f$ and sum them together to obtain a mixture sample in the feature space, $\mathbf{v}^j = \Phi(\mathbf{f}^j)$. Then the combined dictionary at iteration $j+1$, $\mathbf{W}^{j+1} = [\mathbf{W}_1^{j+1}, \mathbf{W}_2^{j+1}]$, is obtained by

$$\mathbf{W}^{j+1} \leftarrow \mathcal{P}(\mathbf{W}^j - \eta_j \nabla_{\mathbf{W}} \ell(\mathbf{f}_1^j, \mathbf{f}_2^j, \mathbf{W}_1^j, \mathbf{W}_2^j, \mathbf{h}_1^{*j}, \mathbf{h}_2^{*j}),$$

where $0 \leq \eta_i \leq \eta$ is a decreasing sequence of step-sizes, and $\mathcal{P}$ is a projection operator making the argument matrix be non-negative with column having the norm smaller or equal than one. Note that the learning requires the gradient $\nabla_{\mathbf{W}} \ell$, which in turn relies (via the chain rule) on the gradients of $\nabla_{\mathbf{M}_i} \ell$, $\nabla_{\mathbf{h}_i^*} \mathbf{M}_i$, and $\nabla_{\mathbf{W}} \mathbf{h}_i^*(\mathbf{v}, \mathbf{W})$. As in the context of dictionary learning for sparse coding [22], even though the $\mathbf{h}_i^*$ are obtained by solving a non-smooth optimization problem, they are almost everywhere differentiable, and one can compute their gradient with respect to $\mathbf{W}$ in a closed form. In the next section, we summarize the derivation of the gradients $\nabla_{\mathbf{W}} \ell$.

Following [22], we use a step size of the form $\eta_i = \eta \min(1, i_0/i)$ in all our experiments, which means that a fixed step size is used during the first i_0 iterations, after which it decays according to the $1/i$ annealing strategy. We set in all our experiments i_0 to be half of the total number of iterations. However, other standard tools commonly used in SGD optimization, such as momentum, could also be used. A common heuristic used in practice for accelerating the convergence speed of SGD algorithms consists randomly drawing several samples (a mini batch) at each iteration instead of a single one. A natural initialization of the speech and noise dictionaries is the individual training via the solution of (4), as in standard supervised NMF denoising.

4.2 Gradient computation

Let us denote by ρ the objective function in (9),

$$\rho(\mathbf{W}, \mathbf{h}) = D_\beta(\mathbf{v}|\mathbf{W}\mathbf{h}) + \sum_{i=1,2} \lambda \psi(\mathbf{h}_i) + \mu ||\mathbf{h}_i||_2^2,$$

where, for simplicity, we define the vector $\mathbf{h} = [\mathbf{h}_1; \mathbf{h}_2]$ (using Matlab-like notation), containing the column-concatenated activations for each source, such that the product of $\mathbf{h}$ with the row-concatenated matrix $\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2]$ is well defined. Let us denote by Λ the active set of the solution of (9), that is, the indeces of the non-zero coefficients of $\mathbf{h}^*$. We use the sub-index Λ to indicate the sub-vector restricted to the active set, e.g., $\mathbf{h}_\Lambda^*$. The first-order optimality conditions of (9) require the derivatives with respect to $\mathbf{h}_\Lambda$ to be zero,

$$\mathbf{h}^* \geq 0, \quad \nabla_{\mathbf{h}}\rho(\mathbf{W}, \mathbf{h}^*) \geq 0, \quad \mathbf{h}^* \circ \nabla_{\mathbf{h}}\rho(\mathbf{W}, \mathbf{h}^*) = \mathbf{0}, \tag{11}$$

where $\circ$ denotes element-wise multiplication (Hadamard product). For each coefficient in the active set of any stationary point of (9), the partial derivative of ρ with respect to that coefficient needs to be zero. Hence, if we look only at the active set we have,

$$[\nabla_{\mathbf{h}}\rho(\mathbf{W}, \mathbf{h}^*)]_{\Lambda} = \mathbf{W}_{\Lambda}^{\mathrm{T}}\mathbf{\Phi} + \lambda\nabla_{\mathbf{h}} \sum_{i=1,2} \psi(\mathbf{h}_i^*)_{\Lambda} + \mu\, \mathbf{h}_{\Lambda}^* = \mathbf{0}, \tag{12}$$

where $\mathbf{W}_{\Lambda}$ is the matrix retaining only the columns of the dictionary associated with the active set, and $\mathbf{\Phi} = (\mathbf{W}_{\Lambda}\mathbf{h}_{\Lambda}^*)^{\beta-2} \circ (\mathbf{W}_{\Lambda}\mathbf{h}_{\Lambda}^* - \mathbf{v})$. When ψ is the ℓ_1 norm as in the case of study described in Section 2.1, the derivative of the regularization term, $\nabla_{\mathbf{h}}\psi(\mathbf{h}_i) = \mathbf{p}$, is equal to a constant vector that assumes the value of one on the coefficients of Λ and zero otherwise.

For a given coordinate, say indexed by r, the conditions given in (11) imply three cases, either only one of $[\mathbf{h}^*]_r$ or $[\nabla_{\mathbf{h}}\rho(\mathbf{W}, \mathbf{h}^*)]_r$ are zero or both are. As it was shown in the sparse coding context [22], a key observation is that, almost surely, the set of active constraints in the solution of (9) remains constant on a local neighborhood of $\mathbf{v}$ and $\mathbf{W}$. That is, for small changes in the dictionary, the active set Λ remains constant. The only points in which $\mathbf{h}^*$ is non-diferentiable are points where the active set changes.

Hence, we know that only the gradient $\nabla_{\mathbf{W}_{\Lambda}}\mathbf{h}^*$ will be non-zero, that is, changes in the columns of $\mathbf{W}$ that do not affect the coefficients in Λ do not affect the cost function. Since we cannot write $\mathbf{h}^*$ in closed form as a function of $\mathbf{W}$, we need to perform implicit differentiation. Taking the derivative in (12) with respect to $\mathbf{W}_{\Lambda}$ we obtain,

$$d\mathbf{W}_{\Lambda}^{\mathrm{T}}\boldsymbol{\phi} + \mathbf{W}_{\Lambda}^{\mathrm{T}}\mathbf{\Phi}(d\mathbf{W}_{\Lambda}\mathbf{h}_{\Lambda}^* + \mathbf{W}_{\Lambda}d\mathbf{h}_{\Lambda}^*) + \mu\, d\mathbf{h}_{\Lambda}^* = \mathbf{0}, \tag{13}$$

where we used d to denote the differentials, and

$$\mathbf{\Phi} = \mathrm{diag}\big((\mathbf{W}_{\Lambda}\mathbf{h}_{\Lambda}^*)^{\beta-2} + (\beta-2)(\mathbf{W}_{\Lambda}\mathbf{h}_{\Lambda}^*)^{\beta-3} \circ (\mathbf{W}_{\Lambda}\mathbf{h}_{\Lambda}^* - \mathbf{v})\big). \tag{14}$$

We can obtain an expression for $d\mathbf{h}_{\Lambda}^*$ from (13) as,

$$d\mathbf{h}_{\Lambda}^* = \mathbf{Q}\,(d\mathbf{W}_{\Lambda}^{\mathrm{T}}\boldsymbol{\phi} + \mathbf{W}_{\Lambda}^{\mathrm{T}}\mathbf{\Phi}d\mathbf{W}_{\Lambda}\mathbf{h}_{\Lambda}^*), \tag{15}$$

where $\mathbf{Q} = (\mathbf{W}_{\Lambda}^{\mathrm{T}}\mathbf{\Phi}\mathbf{W}_{\Lambda} + \mu\mathbf{I})^{-1}$. Note that the size of the matrix being inverted is given by the sparsity level of the representation. Now we can proceed to compute the gradient of the loss function in with respect to the dictionary. Invoking the chain rule, we have

$$\nabla_{\mathbf{W}}\ell = \mathrm{trace}(\nabla_{\mathbf{h}^*}\ell^{\mathrm{T}}d\mathbf{h}^*) + \nabla_{\mathbf{W}}\hat{\ell}, \tag{16}$$

where $\nabla_{\mathbf{W}}\hat{\ell}$ represents the gradient of ℓ with respect to $\mathbf{W}$ assuming $\mathbf{h}^*$ fixed. To compute the gradient $\nabla_{\mathbf{h}^*}\ell$ one has to also use the chain rule considering

the definition of the masks given in (3). Combining (15) and (16) and using the properties of the trace function, it follows that

$$\nabla_{\mathbf{W}}\ell = \boldsymbol{\phi}\,\boldsymbol{\xi}^{\mathrm{T}} + \boldsymbol{\Phi}\mathbf{W}_{\Lambda}\boldsymbol{\xi}{\mathbf{h}^*_{\Lambda}}^{\mathrm{T}} + \nabla_{\mathbf{W}}\hat{\ell}, \tag{17}$$

where $\boldsymbol{\xi} = \mathbf{Q}\nabla_{\mathbf{h}^*}\ell$.

4.3 Implementation details

There are a few important implementation that need to be considered in practice. First, the $\beta-$divergences are not differentiable at zero when $\beta leq 2$. A common way to solve this problem is to consider a translated version of the divergence insted, which is obtained by adding a small constant in the second argument,

$$\tilde{D}_{\beta}(a|b) = D_{\beta}(a|b+\delta)$$

where $\delta > 0$ is a small constant. In our experiments we used $\delta = 0.001$. It is worth mentioning that this is common practice even in every setting of NMF in order to avoid instabilities produced by extremely large values.

During the iterations of the SGD algorithm, the estimation of the gradient of the cost function on the current sample (or mini-batch) requires the computation of the optimal activations $\mathbf{h}^*$ by solving (9). The precision with which this activations are computed is very important for obtaning meaningful gradients. In that sense, it is preferable to use algorithms with fast converge rates, for example the least angle regression (LARS) in the case of $\beta = 2$ [27], or the alternating method of multipliers (ADMM) [28] in the case of $\beta \leq 2$. While running multiplicative algorithms for a small number of iterations produces sactifactory results when running NMF for separation, their slow convergence rate makes them extremely unefficient in this case, requiring a very large number of iterations for computing meaningful gradients.

5 Experimental results

Data sets. We evaluated the separation performance of the proposed methods on a subset of the GRID dataset [29]. Three randomly chosen sets of distinct clips each were used for training (500 clips), validation (10 clips), and testing (50 clips). The clips were resampled to 8 KHz. For the noise signals we used the AURORA corpus [30], which contains six categories of noise recorded from different real environments (street, restaurant, car, exhibition, train, and airport). Three sets of distinct clips each were used for training (15 clips), validation (3 clips), and testing (15 clips).

Evaluation measures. As the evaluation criteria, we used the *source-to-distortion ratio* (SDR), *source-to-interference ratio* (SIR), and *source-to-artifact ratio* (SAR) from the BSS-EVAL metrics [31]. We also computed the

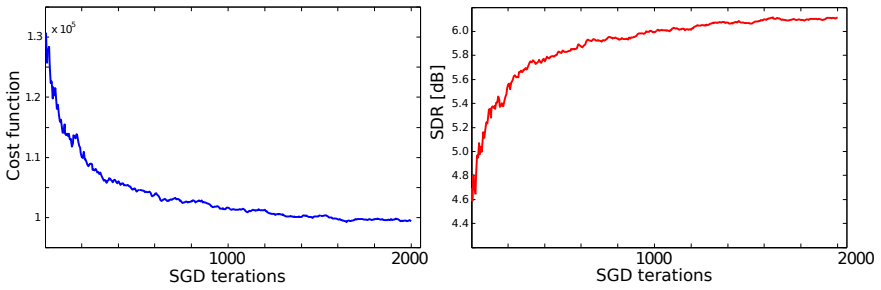

Fig. 1. Evolution of the average high level cost function (left) and the average SDR (in dB) on the validation set (mixed at $SNR = 0dB$) with the SGD iterations for task-specific NMF with $\beta = 1$.

standard *signal-to-noise ratio* (SNR). When dealing with several frames, we computed a global score (GSDR, GSIR, GSAR and GSNR) by averaging the metrics over all test clips from the same speaker and noise weighted by the clip duration.

The goal of this experiment was to apply the proposed approach in the context of audio denoising. Here the noise is considered as a source and modeled explicitly. We used dictionaries of size 60 and 10 atoms for representing the speech and the noise, respectively. These values were obtained using cross-validation. We used different values of the parameter λ for the signal and the noise, namely $\lambda_{\mathrm{s}} = 0.1$ for speech and $\lambda_{\mathrm{n}} = 0$ for the noise (the latter means that no sparsity was promoted in the representation of the noise) and $\mu = 0.001$. As an example, we used $\beta = 1$ and $\beta = 0$, and $\alpha = 0$ in the high level cost (10). For the SGD algorithm we used $\eta = 0.1$ and minibatch of size 50. These were obtained by trying several values of during a small number of iterations, keeping those producing the lowest error on a small validation set. All training signals where mixed at 5 dB.

Results. Figure 1 shows the evolution of the high level cost (10) and the SDR on the validation set with the SGD iterations. The algorithm converges to a dictionary that achieves about 2 dB better SDR on the validation set, this behaivior is also verified on the test set. Tables 1 and 2 show results for the proposed approach on the test setting. We compare the performance of standard supervised sparse-NMF (referred simply as NMF) against the performance of the same model trained in the proposed task-specific manner (referred as TS-NMF) on denoising two with different SNR levels. Observe that the task-specific supervision leads to improvements in performance, maintaining (at $5dB$ SNR) the improvements observed on the validation set. Interestingly, the method also works when using $\beta = 0$ (Itakura-Saito), even if the developments in Section 4 are technically not valid in this case, since the divergence is not convex. While the non-convexity of the problem implies that there might be multiple minimums, we initialize the pursuit algorithm always with

Table 1. Average performance (in dB) for NMF and proposed supervised NMF methods measured in terms of SDR, SIR, SAR and SNR. Speech and noise were mixed at $5dB$ of SNR. The standard deviation of each result is shown in brackets.

	SDR	SIR	SAR	SNR
NMF $\beta = 1$	7.5 [1.5]	13.7 [0.9]	8.9 [1.7]	8.2 [1.3]
TS-NMF $\beta = 1$	9.5 [1.4]	15.2 [0.7]	11.0 [1.7]	10.0 [1.2]
TS-NMF $\beta = 0$	8.6 [1.3]	14.1 [1.2]	10.3 [1.5]	9.1 [1.1]

Table 2. See description of Table 1. In this case, speech and noise were mixed at $0dB$ of SNR.

	SDR	SIR	SAR	SNR
NMF $\beta = 1$	4.5 [1.1]	9.3 [0.9]	6.8 [1.2]	5.8 [0.8]
TS-NMF $\beta = 1$	6.3 [1.0]	11.9 [0.7]	8.0 [1.1]	7.2 [0.8]
TS-NMF $\beta = 0$	5.2 [1.2]	12.0 [1.7]	6.6 [1.2]	6.3 [0.9]

the exact same initial condition (all zeros).Intuitively, one can expect that a small perturbation on the dictionary will the local minims of the solution change slightly and consequently the algorithm will still converge to the same (perturbed) minimum.

6 Discussion

In this chapter we reviewed the use of NMF for solving source separationg problems. We discussed different ways of solving the supervised training of the NMF model and proposed an algorithm for the task-supervised training of NMF models following the ideas introduced in [22] in the context of sparse coding. Unlike standard supervised NMF, the proposed approach matches the optimization objective used at the train and testing stages. In this way, the dictionaries can be trained to optimize the performance of the specific task. We cast this problem as bilevel optimization that can be efficiently solved via stochastic gradient descent. The proposed approach allows non-Euclidean data terms such as β-divergences. A simple case study of sparse-NMF with task specific supervision demonstrates promising results.

Acknowledgments

Work partially supported by ONR, NSF, NGA, AFOSR, BSF, ARO, and ERC.

References

1. P. C. Loizou, *Speech Enhancement: Theory and Practice*, vol. 30, CRC, 2007.
2. E. Hänsler and G. Schmidt, *Speech and Audio Processing in Adverse Environments*, Springer, 2008.
3. D.D. Lee and H.S. Seung, "Learning parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
4. P. Smaragdis, B. Raj, and M. Shashanka, "A probabilistic latent variable model for acoustic modeling," *NIPS*, vol. 148, 2006.
5. P. Smaragdis, C. Fevotte, G Mysore, N. Mohammadiha, and M. Hoffman, "Static and dynamic source separation using nonnegative factorizations: A unified view," *Signal Processing Magazine, IEEE*, vol. 31, no. 3, pp. 66–75, 2014.
6. Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka, "Supervised and semi-supervised separation of sounds from single-channel mixtures," in *Independent Component Analysis and Signal Separation*, pp. 414–421. Springer, 2007.
7. N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 10, pp. 2140–2151, 2013.
8. M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *INTERSPEECH*, Sep 2006.
9. M. V. S. Shashanka, B. Raj, and P. Smaragdis, "Sparse Overcomplete Decomposition for Single Channel Speaker Separation," in *ICASSP*, 2007.
10. C. Joder, F. Weninger, F. Eyben, D. Virette, and B. Schuller, "Real-time speech separation by semi-supervised nonnegative matrix factorization," in *LVA/ICA*, 2012, pp. 322–329.
11. Z. Duan, G. J. Mysore, and P. Smaragdis, "Online plca for real-time semi-supervised source separation," in *LVA/ICA*, 2012, pp. 34–41.
12. M. N. Schmidt, J. Larsen, and F.-T. Hsiao, "Wind noise reduction using non-negative sparse coding," in *MLSP*, Aug 2007, pp. 431–436.
13. J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Trans. on Audio, Speech, and Lang. Proc.*, vol. 19, no. 7, pp. 2067–2080, 2011.
14. F. Weninger, M. Wöllmer, J. T. Geiger, B. Schuller, J. F. Gemmeke, A. Hurmalainen, T. Virtanen, and G. Rigoll, "Non-negative matrix factorization for highly noise-robust asr: To enhance or to recognize?," in *ICASSP*, 2012, pp. 4681–4684.
15. D. Bansal, B. Raj, and P. Smaragdis, "Bandwidth expansion of narrowband speech using non-negative matrix factorization," in *INTERSPEECH*, 2005, pp. 1505–1508.
16. J. Han, G. J. Mysore, and B. Pardo, "Audio imputation using the non-negative hidden markov model," in *LVA/ICA*, 2012, pp. 347–355.
17. Pablo Sprechmann, Alex M Bronstein, and Guillermo Sapiro, "Supervised non-euclidean sparse nmf via bilevel optimization with applications to speech enhancement," in *HSCMA*. IEEE, 2014, pp. 11–15.
18. F. Weninger, J. Le Roux, J. R Hershey, and S. Watanabe, "Discriminative NMF and its application to single-channel source separation," *Proc. of ISCA Interspeech*, 2014.
19. P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *ICASSP*, 2014, pp. 1562–1566.

20. N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Discriminative non-negative matrix factorization for multiple pitch estimation," in *ISMIR*. Citeseer, 2012, pp. 205–210.
21. T. Ben Yakar, P. Sprechmann, R. Litman, A. M. Bronstein, and G. Sapiro, "Bilevel sparse models for polyphonic music transcription," in *ISMIR*, 2013, pp. 65–70.
22. J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 4, pp. 791–804, 2012.
23. R. W. Gerchberg and W. Owen Saxton, "A practical algorithm for the determination of the phase from image and diffraction plane pictures," *Optik*, vol. 35, pp. 237–246, 1972.
24. C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural Computation*, vol. 23, no. 9, pp. 2421–2456, 2011.
25. C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the itakura-saito divergence. with application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, Mar. 2009.
26. B. Colson, P. Marcotte, and G. Savard, "An overview of bilevel optimization," *Annals of Operations Research*, vol. 153, no. 1, pp. 235–256, 2007.
27. Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al., "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.
28. D. L. Sun and C. Fvotte, "Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014.
29. M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *J. of the Acoustical Society of America*, vol. 120, pp. 2421, 2006.
30. D. Pearce and H.-G. Hirsch, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *INTERSPEECH*, 2000, pp. 29–32.
31. E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. on Audio, Speech, and Lang. Proc.*, vol. 14, no. 4, pp. 1462–1469, 2006.