A MAJORIZATION-MINIMIZATION ALGORITHM WITH PROJECTED GRADIENT UPDATES FOR TIME-DOMAIN SPECTROGRAM FACTORIZATION

Hideaki Kagami[†], *Hirokazu Kameoka*[‡], *Masahiro Yukawa*[†]

[†] Dept. Electronics and Electrical Engineering, Keio University, Japan.
[‡] NTT Communication Science Laboratories, NTT Corporation, Japan.

ABSTRACT

We previously introduced a framework called time-domain spectrogram factorization (TSF), which realizes nonnegative matrix factorization (NMF)-like source separation in the time domain. This framework is particularly noteworthy in that, while maintaining the ability of NMF to obtain a parts-based representation of magnitude spectra, it allows us to (i) circumvent the commonly made assumption with the NMF approach that the magnitude spectra of source components are additive and (ii) take account of the interdependence of the phase/amplitude components at different time-frequency points. In particular, the second factor has been overlooked despite its potential importance. Our previous study revealed that the conventional TSF algorithm was relatively slow due to large matrix inversions, and the early stopping of the algorithm often resulted in poor separation accuracy. To overcome this problem, this paper presents an iterative TSF solver using projected gradient updates. Simulation results show that the proposed TSF approach yields higher source separation performance than NMF and the other variants including the original TSF.

Index Terms— Audio source separation, non-negative matrix factorization(NMF), projected gradient method

1. INTRODUCTION

Many sound recordings are mixtures of multiple sound sources. Audio source separation, i.e., the inverse operation of the mixing process, has long been a formidable challenge in the field of audio signal processing [1].

For speech enhancement tasks, i.e., audio source separation tasks where the mixture is given by speech and noise, deep neural network-based approaches have recently proved powerful [2]. A recently proposed deep-network-based approach [3] has further made it possible to deal with "cocktail party" scenarios where the interference is also speech. Although these methods have been shown to work well when a large number of training samples are available, the supervised/semi-supervised non-negative matrix factorization (NMF) approach [4–6] still remains attractive for audio source separation tasks particularly when only a limited number of training data are available. With the NMF approach, the magnitude (or power) spectrogram of a mixture signal, represented as a non-negative matrix $\boldsymbol{Y}$, is factorized into a product of two non-negative matrices $\boldsymbol{H}$ and $\boldsymbol{U}$. This can be interpreted as approximating the observed spectra at each time frame as a linear sum of basis spectra scaled by time-varying amplitudes, and amounts to approximating the observed spectrogram as the sum of rank-1 spectrograms. The sequence of observed spectra can be approximated reasonably well when each basis spectrum expresses the spectrum of an underlying audio event that occurs frequently over the entire observed range. Thus, with music signals, each basis spectrum usually becomes the spectrum of a frequently used pitch in the music piece. In a supervised/semi-supervised setting, NMF is first applied to train the basis spectra of each sound source using the individually recorded audio samples. At test time, NMF is applied to the spectrogram of a test mixture signal, where the subsets of the basis spectra are fixed at the pretrained spectra. In this way, the signals of the underlying source components can be separated out using the Wiener filter obtained with the estimated spectrograms of the individual sources.

Although the NMF approach has been shown to be successful, one limitation is that it assumes the additivity of magnitude (or power) spectra, which holds only approximately, and does not take account of phase information. To overcome this limitation, we have previously proposed a framework called complex NMF [7], where the complex spectrum observed at each time frame is modeled as the sum of components, each of which is described by the multiplication of a static basis spectrum, a time-varying amplitude and a time-varying phase spectrum. With a similar motivation, Parry *et al.* [8] and Févotte *et al.* [9] have independently proposed a generative model of the complex spectrogram obtained with the short-time Fourier transform (STFT) of a mixture signal, where the power spectrogram of each component is modeled as a rank-1 matrix and the phase spectrogram is treated as uniformly distributed latent variables. It can be shown that when each element of the complex spectrogram is assumed to independently follow a zero-mean complex normal distribution, the maximum likelihood estimation of the model parameters amounts to fitting the NMF model to an observed power spectrogram using the Itakura-Saito (IS) divergence as a goodness-of-fit criterion. This approach is called IS-NMF. A similar kind of generative model using a complex Cauchy distribution instead of a complex normal distribution has recently been proposed [10].

Although these phase-aware NMF variants treat each element of the phase spectrogram as an independent parameter (or latent variable), in fact the phases of time-frequency components are constrained and dependent on each other. This is because the spectrograms obtained with typical time-frequency transforms (such as the STFT and the wavelet transform) are redundant representations of a signal. For example, the STFT spectrogram is computed by concatenating the Fourier transforms of overlapping short-time frames of the signal. Hence, all the elements of the STFT spectrogram must satisfy a certain condition to ensure that the waveforms within the overlapping segment of consecutive frames are consistent [11]. The shortcomings of the complex NMF and IS-NMF frameworks are that they fail to take account of this kind of redundancy. Moreover, the time domain signal converted from the estimated complex spectrogram is in general not "optimal" unless the redundancy is taken into

This work was supported by the Support Center for Advanced Telecommunications Technology Research (SCAT) and JSPS Grants-in-Aid (15K06081, 15K13986, 15H02757, 26730100).

978-1-5090-4117-6/17/$31.00 ©2017 IEEE

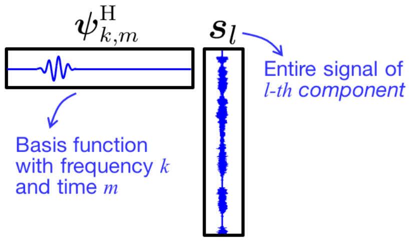

Fig. 1. Illustration of time-frequency basis functions.

account, since the complex spectrogram of the converted signal is no longer the same as the original complex spectrogram.

To exploit the intrinsically redundant structure of spectrograms explicitly, we previously introduced a framework called time-domain spectrogram factorization (TSF) [12], which realizes NMF-like source separation in the time domain. This framework is particularly noteworthy in that, while maintaining the ability to obtain a parts-based representation of magnitude spectra, it directly decomposes a time-domain signal into each underlying component. Postprocessing to obtain a time domain signal from the estimated spectrogram as typically required in the conventional frameworks is implicitly involved in the optimization process. However, our previous study revealed that the conventional TSF algorithm was relatively slow due to large matrix inversions, and the early stopping of the algorithm often resulted in poor separation accuracy. This paper proposes an iterative algorithm leveraging the projected gradient method to solve the TSF optimization problem in a tractable manner.

2. PROBLEM FORMULATION

2.1. Objective function

We denote an observed signal at time n by $y[n]$, and the signal of the entire period by $\boldsymbol{y} = [y[1], \dots, y[N]]^{\mathsf{T}} \in \mathbb{R}^N$, where $(\cdot)^{\mathsf{T}}$ stands for the vector (matrix) transpose. While the NMF approach decomposes an observed magnitude spectrogram into the sum of rank-1 spectrograms, TSF decomposes $\boldsymbol{y}$ into the sum of L signal components:

$$\boldsymbol{y} = \sum_{l=1}^{L} \boldsymbol{s}_l, \tag{1}$$

such that the magnitude spectrogram of each component is as close to a rank-1 structure as possible. Here, we use $\boldsymbol{\psi}_{k,m} \in \mathbb{C}^N$ to denote a basis function for the time-frequency transform, where k and m are the frequency and time indices, respectively. By using $\boldsymbol{\psi}_{k,m}$, the magnitude spectrogram of $\boldsymbol{s}_l$ can be written as $|\boldsymbol{\psi}^{\mathsf{H}}_{k,m}\boldsymbol{s}_l|$, where $(\cdot)^{\mathsf{H}}$ stands for the Hermitian transpose of a vector. Hence, we formulate TSF as the optimization problem of minimizing

$$\mathcal{J}(\boldsymbol{\theta}) := \sum_{l,k,m} \frac{1}{\beta_{l,k,m}} \left(|\boldsymbol{\psi}^{\mathsf{H}}_{k,m}\boldsymbol{s}_l| - H_{k,l}U_{l,m}\right)^2 + 2\lambda \sum_{l,m} |U_{l,m}|^p \,,$$
$$\text{subject to } \boldsymbol{y} = \sum_{l} \boldsymbol{s}_l, \tag{2}$$

with respect to $\boldsymbol{\theta} = \{\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{S}, \boldsymbol{\beta}\}$ where $\boldsymbol{H} = \{H_{k,l}\}, \boldsymbol{U} = \{U_{l,m}\}$, $\boldsymbol{S} = \{\boldsymbol{s}_l\}$ and $\boldsymbol{\beta} = \{\beta_{l,k,m}\}$ with $\beta_{l,k,m} > 0$ satisfying $\sum_l \beta_{l,k,m} = 1$. This problem can be seen as a weighted least squares problem where $\beta_{l,k,m}$ is the reciprocal of the weight. The importance of this weight parameter will be shown later. The first term of $\mathcal{J}(\boldsymbol{\theta})$ becomes 0 when the magnitude spectrograms of all the $\boldsymbol{S}$ have exactly rank-1 structures. It is important to note that as with complex NMF, this model allows the components to cancel each other out, and therefore some constraint is needed to induce the sparsity of $\boldsymbol{U}$. The second term of $\mathcal{J}(\boldsymbol{\theta})$ is introduced for this purpose, which we define as the ℓ_p norm, where $\lambda > 0$ weighs the importance of the sparsity cost relative to the fitting cost. For $0 < p < 2$, it promotes sparsity if the norm of $\boldsymbol{U}$ is bounded. To bound $\boldsymbol{U}$, we assume

$$\sum_k H_{k,l}^2 = 1. \tag{3}$$

2.2. Majorization-minimization algorithm

Let $F(\boldsymbol{\theta})$ be a cost function that we wish to minimize with respect to $\boldsymbol{\theta}$. $F^+(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}})$ is defined as an auxiliary function for $F(\boldsymbol{\theta})$ if it satisfies

$$F(\boldsymbol{\theta}) = \min_{\bar{\boldsymbol{\theta}}} F^+(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}}). \tag{4}$$

Here, we call $\bar{\boldsymbol{\theta}}$ an auxiliary variable.

Theorem 1 ([13]). *$F(\boldsymbol{\theta})$ is non-increasing under the following updates:*

$$\bar{\boldsymbol{\theta}} \leftarrow \operatorname*{argmin}_{\bar{\boldsymbol{\theta}}} F^+(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}}), \tag{5}$$

$$\boldsymbol{\theta} \leftarrow \operatorname*{argmin}_{\boldsymbol{\theta}} F^+(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}}). \tag{6}$$

Note that the convergence of the algorithm is still guaranteed as long as F^+ is decreased with respect to $\boldsymbol{\theta}$ at each iteration. Thus, F^+ does not need to be exactly minimized.

2.3. Auxiliary function of $\mathcal{J}$

One difficulty as regards the current optimization problem comes from the nonsmoothness of $|\boldsymbol{\psi}^{\mathsf{H}}_{k,m}\boldsymbol{s}_l|$ and $|U_{l,m}|^p$ for $0 < p < 1$ in $\mathcal{J}$. We can design an auxiliary function for $\mathcal{J}$ with a convenient form

$$\mathcal{J}^+(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}}) = \sum_{l,k,m} \frac{1}{\beta_{l,k,m}} \left|\boldsymbol{\psi}^{\mathsf{H}}_{k,m}\boldsymbol{s}_l - H_{k,l}U_{l,m}c_{l,k,m}\right|^2$$
$$+ \lambda \sum_{l,m} \left[p|V_{l,m}|^{p-2}U_{l,m}^2 + (2-p)|V_{l,m}|^p\right], \tag{7}$$

by using the inequalities (A.30) and (A.31) given in the appendix, where $\bar{\boldsymbol{\theta}} = \{\boldsymbol{c}, \boldsymbol{V}\}$ with $\boldsymbol{c} = \{c_{l,k,m}\}$ and $\boldsymbol{V} = \{V_{l,m}\}$. A minimizer of the auxiliary function with respect to $\boldsymbol{C}$ and $\boldsymbol{V}$ has the following closed-form expression:

$$c_{l,k,m} = \boldsymbol{\psi}^{\mathsf{H}}_{k,m}\boldsymbol{s}_l / |\boldsymbol{\psi}^{\mathsf{H}}_{k,m}\boldsymbol{s}_l|, \tag{8}$$

$$V_{l,m} = U_{l,m}. \tag{9}$$

3. OPTIMIZATION WITH PROJECTED GRADIENT

By using the method of Lagrange multipliers, we obtain the update rule for $\boldsymbol{s}_l$ as

$$\boldsymbol{s}_l = \boldsymbol{\Psi}_l^{-1}(\boldsymbol{d}_l - \boldsymbol{\mu}), \tag{10}$$

where

$$\boldsymbol{\Psi}_l = \sum_{k,m} \frac{2\mathrm{Re}[\boldsymbol{\psi}_{k,m}\boldsymbol{\psi}_{k,m}^{\mathsf{H}}]}{\beta_{l,k,m}}, \tag{11}$$

$$\boldsymbol{d}_l = \sum_{k,m} \frac{2\mathrm{Re}[c_{l,k,m}\boldsymbol{\psi}_{k,m}]H_{k,l}U_{l,m}}{\beta_{l,k,m}}, \tag{12}$$

$$\boldsymbol{\mu} = \Big(\sum_l \boldsymbol{\Psi}_l\Big)^{-1}\Big(\sum_l \boldsymbol{\Psi}_l^{-1}\boldsymbol{d}_l - \boldsymbol{y}\Big). \tag{13}$$

The conventional TSF algorithm uses (10) as the update rule for $\boldsymbol{s}_l$. As can be seen from the above, one problem with the conventional algorithm is that it involves large matrix inversions, resulting in a very slow algorithm. In [12], we showed that in a particular case where $\beta_{1,k,m} = \cdots = \beta_{L,k,m}$ (i.e., $\forall l, \beta_{l,k,m} = 1/L$), $\boldsymbol{\Psi}_l$ becomes an identity matrix, which means that the update rule of $\boldsymbol{s}_l$ can be computed without matrix inversions. However, it transpired that this β setting results in a poor separation accuracy. We show in this section that using the projected gradient method to update $\boldsymbol{s}_l$ allows us to sidestep these computations and leads to a reasonably efficient optimization algorithm.

3.1. Update rule for $\boldsymbol{s}_l$

In this subsection, we solely consider the update for $\boldsymbol{s}_l$ and fix all the other variables included in $\boldsymbol{\theta}$ and $\bar{\boldsymbol{\theta}}$. We therefore regard $\mathcal{J}^+$ as a function of $\boldsymbol{s}_l$, and denote it by $\mathcal{J}^+(\boldsymbol{s}_l)$. The partial derivative of $\mathcal{J}^+$ with respect to $\boldsymbol{s}_l$ is given as

$$\nabla_{\boldsymbol{s}_l}\mathcal{J}^+(\boldsymbol{s}_l) = \boldsymbol{\Psi}_l\boldsymbol{s}_l - \boldsymbol{d}_l. \tag{14}$$

Each term of (14) can be computed efficiently. Indeed, when $\boldsymbol{\psi}_{k,m}$ is defined as the STFT basis function, $\boldsymbol{\psi}_{k,m}^{\mathsf{H}}\boldsymbol{s}_l$ represents the (k,m) element of the STFT of $\boldsymbol{s}_l$. Since the operator $\sum_{k,m}\mathrm{Re}[\boldsymbol{\psi}_{k,m}\cdot]$ in (11) corresponds to the inverse STFT process, $\boldsymbol{\Psi}_l\boldsymbol{s}_l$ can be obtained by computing the inverse STFT of a vector containing $2\boldsymbol{\psi}_{k,m}^{\mathsf{H}}\boldsymbol{s}_l/\beta_{l,k,m}$. Analogously, $\boldsymbol{d}_l$ in (12) can be obtained by computing the inverse STFT of a vector containing $2H_{k,l}U_{l,m}c_{l,k,m}$ $/\beta_{l,k,m}$. We can thus efficiently update $\boldsymbol{s}_l$ using the gradient-based update rule

$$\boldsymbol{s}_l \leftarrow \boldsymbol{s}_l - \gamma\nabla_{\boldsymbol{s}_l}\mathcal{J}^+(\boldsymbol{s}_l), \tag{15}$$

where $\gamma \in (0, 2/\kappa)$ is step size. Here, $\kappa > 0$ is the Lipschitz constant of the gradient operator $\nabla_{\boldsymbol{s}_l}\mathcal{J}^+$, and is given by the largest eigenvalue of the matrix $\boldsymbol{\Psi}_l$. For the sake of computational efficiency, we use the following step size:

$$\gamma := \frac{2}{\rho} \in (0, 2/\kappa), \tag{16}$$

$$\rho := \sum_l \mathrm{tr}\,(\boldsymbol{\Psi}_l) = \sum_{l,n}\sum_{k,m}\frac{2\{w(n-\alpha_m)\}^2}{\beta_{l,k,m}}, \tag{17}$$

where $w(n)$ is the window function and α_m is the hop size of the STFT.

Define

$$\tilde{\boldsymbol{s}} := \begin{bmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_L \end{bmatrix} \in \mathbb{R}^{LN}, \tag{18}$$

$$\boldsymbol{G} := [\boldsymbol{I}_N \ \cdots \ \boldsymbol{I}_N] \in \mathbb{R}^{N\times LN}, \tag{19}$$

where $\boldsymbol{I}_N$ is the $N \times N$ identity matrix. The linear constraint $\boldsymbol{y} = \sum_l \boldsymbol{s}_l$ can then be written equivalently as $\boldsymbol{y} = \boldsymbol{G}\tilde{\boldsymbol{s}}$. Also define

$$\mathcal{V} := \left\{\tilde{\boldsymbol{s}} \in \mathbb{R}^{LN} \mid \boldsymbol{G}\tilde{\boldsymbol{s}} = \boldsymbol{y}\right\}. \tag{20}$$

Then, the projection onto the affine subspace $\mathcal{V}$ is given by [14]

$$P_{\mathcal{V}}(\tilde{\boldsymbol{s}}) = \tilde{\boldsymbol{s}} - \boldsymbol{G}^{\mathsf{T}}(\boldsymbol{G}\boldsymbol{G}^{\mathsf{T}})^{-1}(\boldsymbol{G}\tilde{\boldsymbol{s}} - \boldsymbol{y}). \tag{21}$$

By considering $\boldsymbol{G}\boldsymbol{G}^{\mathsf{T}} = L\boldsymbol{I}_N$ and

$$\boldsymbol{G}^{\mathsf{T}}\boldsymbol{G} = \begin{bmatrix} \boldsymbol{I}_N \ \boldsymbol{I}_N \ \ldots \ \boldsymbol{I}_N \\ \boldsymbol{I}_N \ \boldsymbol{I}_N \ \ldots \ \boldsymbol{I}_N \\ \vdots \\ \boldsymbol{I}_N \ \boldsymbol{I}_N \ \ldots \ \boldsymbol{I}_N \end{bmatrix} \in \mathbb{R}^{LN\times LN}, \tag{22}$$

we obtain

$$[P_{\mathcal{V}}(\tilde{\boldsymbol{s}})]_l = \boldsymbol{s}_l - \frac{1}{L}\left(\sum_{l'=1}^{L}\boldsymbol{s}_{l'} - \boldsymbol{y}\right). \tag{23}$$

Accordingly, we seek to minimize $\mathcal{J}^+$ with respect to $\boldsymbol{s}_l$ by alternately performing (15) and (23). In fact, ρ gives the upper bound of κ, and this ensures $\gamma \in (0, 2/\kappa)$ so that the projected gradient algorithm converge to a minimizer of $\mathcal{J}^+$ under the linear constraint [15].

3.2. Summary of proposed algorithm

By setting the partial derivative of $\mathcal{J}^+$ with respect to each element of $\boldsymbol{U}$ at zero, the update rule for $\boldsymbol{U}$ is obtained as

$$U_{l,m} = \frac{\sum_k H_{k,l}|\boldsymbol{\psi}_{k,m}^{\mathsf{H}}\boldsymbol{s}_l|/\beta_{l,k,m}}{\sum_k H_{k,l}^2/\beta_{l,k,m} + \lambda p|V_{l,m}|^{p-2}}. \tag{24}$$

Similarly, we obtain the update rules for $\boldsymbol{H}$ and $\boldsymbol{\beta}$ by using the method of Lagrange multipliers as follows:

$$H_{k,l} = \frac{\sum_m U_{l,m}|\boldsymbol{\psi}_{k,m}^{\mathsf{H}}\boldsymbol{s}_l|/\beta_{l,k,m}}{\sqrt{\sum_k(\sum_m U_{l,m}|\boldsymbol{\psi}_{k,m}^{\mathsf{H}}\boldsymbol{s}_l|/\beta_{l,k,m})^2}}, \tag{25}$$

$$\beta_{l,k,m} = \frac{\big||\boldsymbol{\psi}_{k,m}^{\mathsf{H}}\boldsymbol{s}_l| - H_{k,l}U_{l,m}\big|}{\sum_l \big||\boldsymbol{\psi}_{k,m}^{\mathsf{H}}\boldsymbol{s}_l| - H_{k,l}U_{l,m}\big|}. \tag{26}$$

Overall, the proposed iterative algorithm is summarized as follows.

```
1. Initialize $\boldsymbol{H}$, $\boldsymbol{U}$ and $\boldsymbol{S}$.
2. Update $\boldsymbol{c}$ and $\boldsymbol{V}$ using (8), (9).
3. Update $\boldsymbol{S}$ and $\gamma$ using (14), (16)–(17), (23).
     for $t = 1$ : iteration
       $\boldsymbol{s}_l^{(t+1)} = P_{\mathcal{V}}\left(\boldsymbol{s}_l^{(t)} - \gamma\nabla_{\boldsymbol{s}_l}\mathcal{J}^+(\boldsymbol{s}_l^{(t)})\right)$
     end
4. Update $\boldsymbol{U}$, $\boldsymbol{H}$ and $\boldsymbol{\beta}$ using (24)–(26).
```

4. RELATION TO COMPLEX NMF

It is interesting to note that the auxiliary function $\mathcal{J}^+$ has a similar form to that of complex NMF [7]. The aim of complex NMF is to approximate an observed complex spectrogram $Y_{k,m}$ using the following model:

$$Y_{k,m} \approx \sum_l H_{k,l}U_{l,m}e^{j\phi_{l,k,m}}, \tag{27}$$

where $\phi_{l,k,m}$ denotes the phase spectrogram of the l-th signal component. In [7], the cost function that must be minimized with respect

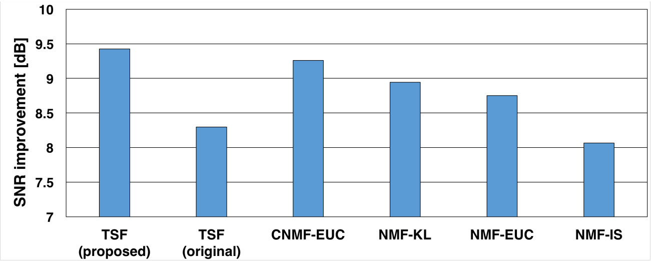

Fig. 2. SNR improvements.

to $H_{k,l}$, $U_{l,m}$ and $\phi_{l,k,m}$ is defined as

$$\mathcal{I}(\boldsymbol{\theta}) = \sum_{k,m} \left| Y_{k,m} - \sum_{l} H_{k,l} U_{l,m} e^{j\phi_{l,k,m}} \right|^2 + 2\lambda \sum_{l,m} |U_{l,m}|^p , \tag{28}$$

with $\boldsymbol{\theta} = \{\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\phi}\}$ where $\boldsymbol{H} = \{H_{k,l}\}, \boldsymbol{U} = \{U_{l,m}\}$ and $\boldsymbol{\phi} = \{\phi_{l,k,m}\}$. It can be shown that

$$\mathcal{I}^{+}(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}}) = \sum_{l,k,m} \frac{1}{\beta_{l,k,m}} \left| X_{l,k,m} - H_{k,l} U_{l,m} e^{j\phi_{l,k,m}} \right|^2 + \lambda \sum_{l,m} \left\{ p|V_{l,m}|^{p-2} U_{l,m}^2 + (2-p)|V_{l,m}|^p \right\}, \tag{29}$$

is an auxiliary function for $\mathcal{I}$, where $\bar{\boldsymbol{\theta}} = \{\boldsymbol{\beta}, \boldsymbol{X}\}$, $\boldsymbol{\beta} = \{\beta_{l,k,m}\}$ and $\boldsymbol{X} = \{X_{l,m}\}$ are the auxiliary variables. Here, $\beta_{l,k,m}$ can be any positive number satisfying $\sum_l \beta_{l,k,m} = 1$, and $X_{l,k,m}$ must satisfy $Y_{k,m} = \sum_l X_{l,k,m}$. $X_{l,k,m}$ can be viewed as an estimate of the complex spectrogram of the l-th signal component.

By comparing (7) and (29), we see that $X_{l,k,m}$ and $e^{j\phi_{l,k,m}}$ in the auxiliary function of the complex NMF objective are analogous to $\boldsymbol{s}_l$ and $c_{l,k,m}$ in that of the present TSF objective, respectively. One drawback with complex NMF is that $X_{l,k,m}$ is not guaranteed to satisfy the explicit condition that complex spectrograms must satisfy. This implies that the complex NMF algorithm is designed to search for optimal parameters in an unnecessarily large solution space. By contrast, the present algorithm always ensures that the estimate of the complex spectrogram of the l-th latent component $\boldsymbol{\psi}_{k,m}^{\mathsf{H}} \boldsymbol{s}_l$ is associated with a time-domain signal $\boldsymbol{s}_l$, keeping the search within an appropriate solution space.

5. EXPERIMENTAL RESULTS

We quantitatively compared the source separation performance of the proposed time-domain spectrogram factorization (TSF) by conducting supervised source separation experiments, original TSF [12], complex NMF (CNMF) [7], and NMF using the Euclidean distance (EUC), Kullback-Leibler (KL) divergence and Itakura-Saito (IS) divergence by conducting supervised source separation experiments. The original TSF minimizes (2) in a particular case where $\beta_{1,k,m} = \cdots = \beta_{L,k,m}$ (i.e., $\forall l, \beta_{l,k,m} = 1/L$). We used professionally produced music recordings from the SiSEC 2013 database, available at https://sisec.wiki.irisa.fr/, as the experimental data. Each recording was a mixture of multiple tracks, each of which was produced by a single instrument or singer. The tracks were also available separately. We performed 3-fold cross validation. We partitioned each recording into three segments, used one segment as the test data and the other two segments as the training data, repeated signal-to-noise (SNR) evaluations three times with different test segments, and take the average SNR improvement obtained with the three repeated rounds. With all these methods, the basis spectra were pretrained using the individual tracks of the training data, and then source separation was performed on the test data. All the audio samples were monaural and sampled at 22.05kHz. The STFT was computed using a square-root Hanning window that was 32 ms long with a 16 ms overlap. With both methods, 6 basis spectra were assigned to each track. Thus, for 5-track recordings, a total of 30 basis spectra were used for the separation. Fig. 2 shows the SNR improvements after the separations with the six methods. These results show that proposed TSF performed better than NMF, CNMF and original TSF.

6. CONCLUSION

This paper presented an efficient TSF algorithm that performs source separation in the time domain. A noteworthy advantage of TSF is that, while preserving the desirable property of NMF, it exploits (i) the additive nature of time-domain signals and (ii) the interdependence of the phase/amplitude components at different time-frequency points. The proposed iterative algorithm was built based on a majorization-minimization scheme leveraging the projected gradient method. Simulation results showed that the proposed TSF algorithm yielded better source separation performance than NMF.

Appendix A. INEQUALITIES

We summarize the inequalities that are used to design the auxiliary function.

Lemma 1. For any complex number z and any complex number c satisfying $|c| = 1$, it holds that

$$-|z| \le -\mathrm{Re}[c^* z]. \tag{A.30}$$

Equality holds when $c = z/|z|$.

Lemma 2. When $0 < p < 2$, for any real or complex number x, it holds that

$$2|x|^2 \le p|v|^{p-2}|x|^2 + 2 - p|v|^p . \tag{A.31}$$

Equality holds when $v = x$.

7. REFERENCES

[1] S. Roweis, "One microphone source separation," in *NIPS*, 2000, vol. 13, pp. 793–799.

[2] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 7, pp. 1381–1390, 2013.

[3] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP 2016*, 2016, pp. 31–35.

[4] P. Smaragdis, B. Raj, and M. Shashanka, "Supervised and semi-supervised separation of sounds from single-channel mixtures," in *Proc. LVA/ICA 2010*, 2010, pp. 140–148.

[5] Y. Morikawa and M. Yukawa, "A sparse optimization approach to supervised NMF based on convex analytic method," in *Proc. ICASSP*, May 2013, pp. 6078–6082.

[6] H. Kagami and M. Yukawa, "Supervised nonnegative matrix factorization with Dual-Itakura-Saito and Kullback-Leibler divergences for music transcription," in *Proc. EUSIPCO*, Hungary, Aug. 2016, pp. 1138–1142.

[7] H. Kameoka, N. Ono, K. Kashino, and S. Sagayama, “Complex NMF: A new sparse representation for acoustic signals,” in *Proc. IEEE ICASSP*, April 2009, pp. 3437–3440.

[8] R.M. Parry and I. Essa, “Phase-aware non-negative spectrogram factorization,” *Independent Component Analysis and Signal Separation*, pp. 536–543, 2007.

[9] 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, 2009.

[10] A. Liutkus, D. Fitzgerald, and R. Badeau, “Cauchy nonnegative matrix factorization,” in *Proc. WASPAA 2015*, 2015.

[11] J. Le Roux, H. Kameoka, N. Ono, and S. Sagayama, “Fast signal reconstruction from magnitude STFT spectrogram based on spectrogram consistency,” in *Proc. DAFx-10*, 2010, pp. 397–403.

[12] H. Kameoka, “Multi-resolution signal decomposition with time-domain spectrogram factorization,” in *Proc. ICASSP 2015*, 2015, pp. 86–90.

[13] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems 13*, pp. 556–562. 2001.

[14] D. G. Luenberger, *Optimization by Vector Space Methods*, New York: Wiley, 1969.

[15] A. A. Goldstein, “Convex programming in Hilbert space,” *Bull. Amer. Math. Soc. 70*, pp. 709–710, 1964.