Learning the Statistical Model of the NMF Using the Deep Multiplicative Update Algorithm with Applications

Hiroki Tanji †, Takahiro Murakami ‡
Department of Electronics and Bioinformatics, Meiji University, Kanagawa, Japan
{†htanji, ‡tmrkm}@meiji.ac.jp

***Abstract*—Nonnegative matrix factorization (NMF) is widely used in the audio applications. In these applications, recently, the cost functions based on the complex probability distribution and the optimization algorithms have been developed. However, the flexibility of these methods is limited because the cost functions have only one or no hyper-parameter. Thus, in this paper, we utilize a neural network to construct a generalized cost function of the NMF. Moreover, we propose a novel neural network architecture to estimate the NMF parameters. The proposed neural network solves the NMF optimization problem of which cost function is parameterized by the part of itself. We show the proposed neural network can learn a statistical model by applying it to the denoising and the signal separation tasks.**

I. Introduction

Nonnegative matrix factorization (NMF) [1] is a popular method for single-channel signal separation and audio denoising [2]–[7]. In audio signal processing, the NMF is typically applied to the power or amplitude spectrogram of the observed signal. The spectrogram $\boldsymbol{Y} = [y_{mn}] \in \mathbb{R}_+^{M\times N}$ ($\mathbb{R}_+ = [0, \infty)$) is approximated as the product $\boldsymbol{Y} \simeq \boldsymbol{WH}$, where $\boldsymbol{W} = [w_{mk}] \in \mathbb{R}_+^{M\times K}$ is a basis matrix, which includes frequently appearing spectral patterns, and $\boldsymbol{H} = [h_{kn}] \in \mathbb{R}_+^{K\times N}$ is an activation matrix.

$\boldsymbol{W}$ and $\boldsymbol{H}$ can be obtained by minimizing a divergence function, which measures reconstruction error between $\boldsymbol{Y}$ and $\boldsymbol{WH}$. To find a statistical interpretation of the divergence function, recently, a number of the statistical models of the NMF have been developed [8]–[14]. In these works, the divergence function is derived from the complex probability density function (PDF). In particular, some of the statistical models improve the model flexibility by using the hyper-parameter [10]–[12], [14]. However, their model flexibility is still limited because they has only one hyper-parameter. Therefore, constructing a statistical model of the NMF based on a generalized PDF which includes the PDFs of the existing models is an open problem.

In this paper, to construct a generalized PDF, we focus on neural networks (NNs), which are well known as the approximations of various nonlinear functions. If a statistical model based on a PDF expressed by a NN can be constructed, it should help the comprehensive evaluation of the NMF in the audio applications. Thus, in this paper, we tackle two questions: (a) *using a NN, can we design a statistical model of the NMF and an algorithm which optimizes the cost function*; (b) *in a particular audio application, can we train the NN so that the performance measure can be maximized.*

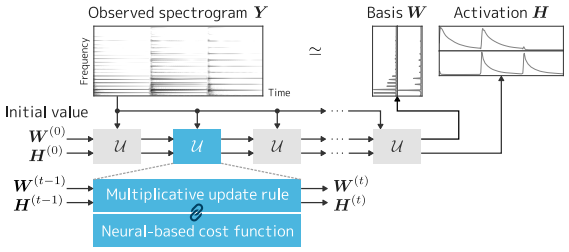

Fig. 1. The block diagram of the proposed network architecture (top) to estimate the NMF parameters (basis and activation matrices) from a given observed spectrogram. Its sub-block (bottom) acts as the update rules of the NMF which minimize the cost function parameterized by the NN.

To answer these questions, in this paper, we propose a network architecture which represents the algorithm for the NMF, named *deep multiplicative update algorithm* (DeMUA) for the NMF. The overview of the DeMUA architecture is illustrated in Fig. 1. In this architecture, the algorithm for the NMF is regarded as a NN which has a recursive structure. Because the algorithm depends on the statistical model, interestingly, the DeMUA can not only estimate the NMF parameters from the observed spectrogram, but also give the neural-based cost function of the NMF. Moreover, we propose the training procedures for the DeMUA. While the DeMUA estimates the NMF parameters so that the neural-based cost function can be minimized, it is trained so that a performance measure (e.g., source-to-distortion ratio (SDR) [15], [16]) can be maximized.

The main contributions of this paper are: (a) a novel network architecture which represents both the generalized statistical model of the NMF and the optimization algorithm for the NMF (Sect. III-B); (b) training procedures for the proposed network architecture in the denoising and the separation tasks (Sects. III-C and III-D, respectively); (c) experimental results in audio denoising and supervised signal separation (Sect. IV).

II. Related work

Our work follows the existing statistical models of the NMF based on the complex PDF [8]–[14]. The concept of the complex PDF-based NMF has been introduced in [8].

The statistical model in [8], called Gaussian-NMF, exploits the complex Gaussian distribution as the PDF of the complex observed spectrogram to justify the additivity of the power spectra. In Gaussian-NMF, $\boldsymbol{W}$ and $\boldsymbol{H}$ are estimated by using the multiplicative update algorithm so that the cost function based on the Gaussian distribution can be minimized.

In order to find a better alternative to Gaussian-NMF and to improve model flexibility, the complex non-Gaussian PDFs have been exploited in [9]–[14]. In particular, the statistical models based on the heavy-tailed distribution (e.g., the NMF based on the complex Student's t distribution (t-NMF [10]) and the complex generalized Gaussian distribution (GGD-NMF [11])) have shown favorable ability in audio denoising [11].

In [12], [14] as well as t-NMF and GGD-NMF, the model flexibility is improved compared to Gaussian-NMF by introducing the hyper-parameter. In this paper, we present a generalized model including these statistical models and propose a framework for tuning the complex PDF of the NMF.

There are several works on the NNs to obtain the NMF parameters (e.g., [17], [18]). Whereas these works focus on estimating the NMF parameters using the fixed cost function (or update rules) of the NMF, we focus on how to design an architecture which can be interpreted as both the update rules and the cost function.

Since the main purpose of this paper is constructing the generalized statistical model of the NMF, the comparison with the state-of-the-art network architectures for the denoising and the separation tasks (e.g., [19] and [20]) is not essential to evaluate the proposed method. The proposed method is compared to the conventional NMF-based methods in Sect. IV.

III. Deep multiplicative update algorithm for NMF

In this section, we explain the proposed network architecture which represents the statistical model and the multiplicative update algorithm of the NMF. First, in Sect. III-A, we reformulate the statistical model of the NMF using the complex PDF. Second, in Sects. III-B, we introduce the architecture of the DeMUA for the NMF. Third, we describe how to train the architecture in the denoising and the separation tasks in Sects. III-C and III-D, respectively.

A. NMF based on the complex PDF

Let $y^{\mathbb{C}}_{mn}$ be the complex spectrogram, and $y_{mn} = |y^{\mathbb{C}}_{mn}|^2$ be the power spectrogram at the mth frequency bin and the nth frame. The NMF approximates y_{mn} using $\hat{y}_{mn} = \sum_{k=1}^{K} w_{mk}h_{kn}$. In [8]–[14], the divergence function, which measures the reconstruction error between y_{mn} and $\hat{y}_{mn}$, is derived using the complex PDF of $y^{\mathbb{C}}_{mn}$ which has the form

$$p_{\mathbb{C}}(y^{\mathbb{C}}_{mn}; \hat{y}_{mn}) \propto \hat{y}^{-1}_{mn} f(y_{mn}\hat{y}^{-1}_{mn}), \tag{1}$$

where $f(r)$ is nonnegative and non-increasing in $(0, \infty)$. From (1), the negative log-PDF is given by

$$-\log p_{\mathbb{C}}(y^{\mathbb{C}}_{mn}; \hat{y}_{mn}) \stackrel{\mathrm{c}}{=} \log \hat{y}_{mn} - \log f(y_{mn}\hat{y}^{-1}_{mn}), \tag{2}$$

where $\stackrel{\mathrm{c}}{=}$ is equality up to constant terms. For example, $f(r)$ is set to $\exp(-r)$ for Gaussian-NMF [8]. In this case, the divergence function is reduced to the Itakura-Saito divergence [8].

The cost function of the complex PDF-based NMF is given by

$$\mathcal{J}(\boldsymbol{W}, \boldsymbol{H}) = \sum_{m,n} \left(\log \hat{y}_{mn} - \log f(y_{mn}\hat{y}^{-1}_{mn})\right). \tag{3}$$

We design the multiplicative update algorithm to minimize $\mathcal{J}(\boldsymbol{W}, \boldsymbol{H})$ under nonnegativity constraints on the parameters. The heuristic approach [9], [17] derives the update rules by splitting the partial derivative of $\mathcal{J}(\boldsymbol{W}, \boldsymbol{H})$ with respect to w_{mk} (or h_{kn}) into the positive and the negative parts. To derive the update rule of any parameter $\vartheta \in \{w_{mk}, h_{kn}\}$, we express $\frac{\partial \mathcal{J}(\boldsymbol{W}, \boldsymbol{H})}{\partial \vartheta}$ using two nonnegative terms: $\nabla[\mathcal{J}]^{+}_{\vartheta}$ and $\nabla[\mathcal{J}]^{-}_{\vartheta}$ as

$$\frac{\partial \mathcal{J}(\boldsymbol{W}, \boldsymbol{H})}{\partial \vartheta} = \nabla[\mathcal{J}]^{+}_{\vartheta} - \nabla[\mathcal{J}]^{-}_{\vartheta}. \tag{4}$$

The update rule of ϑ is given by

$$\vartheta \leftarrow \vartheta \, \frac{\nabla[\mathcal{J}]^{-}_{\vartheta}}{\nabla[\mathcal{J}]^{+}_{\vartheta}}. \tag{5}$$

From (5), the multiplicative update algorithm is given by iterating the update rules:

$$w_{mk} \leftarrow w_{mk} \, \frac{\sum_n \frac{\zeta_{mn} y_{mn}}{\hat{y}^2_{mn}} h_{kn}}{\sum_n h_{kn}/\hat{y}_{mn}} \tag{6}$$

$$h_{kn} \leftarrow h_{kn} \, \frac{\sum_m \frac{\zeta_{mn} y_{mn}}{\hat{y}^2_{mn}} w_{mk}}{\sum_m w_{mk}/\hat{y}_{mn}}, \tag{7}$$

where

$$\zeta_{mn} = -\frac{f'(y_{mn}\hat{y}^{-1}_{mn})}{f(y_{mn}\hat{y}^{-1}_{mn})}, \tag{8}$$

and f' is the derivative of f. At each iteration, after (6), $\boldsymbol{W}$ is normalized using

$$w_{mk} \leftarrow \frac{w_{mk}}{\sum_{m=1}^{M} w_{mk}}. \tag{9}$$

Equation (8) can be interpreted as the expectation of the latent variable in the Gaussian scale mixture (GSM). The complex PDFs in [8]–[10], [12]–[14] and the super-Gaussian case of the GGD [11] are in the class of the GSM written as

$$p_{\mathrm{GSM}}(y^{\mathbb{C}}_{mn}; \hat{y}_{mn}) = \int_0^{\infty} \mathcal{N}_{\mathbb{C}}(y^{\mathbb{C}}_{mn}; 0, z^{-1}_{mn}\hat{y}_{mn}) p(dz_{mn}), \tag{10}$$

where $\mathcal{N}_{\mathbb{C}}(\cdot; \mu, \sigma^2)$ is the complex Gaussian distribution with mean μ and variance σ^2, and $p(z_{mn})$ is a PDF on $(0, \infty)$. In the statistical model using (10), ζ_{mn} indicates the following expectation:

$$\mathbb{E}_{p(z_{mn}|y^{\mathbb{C}}_{mn})}[z_{mn}] = \int_0^{\infty} z_{mn} p(dz_{mn}|y^{\mathbb{C}}_{mn}). \tag{11}$$

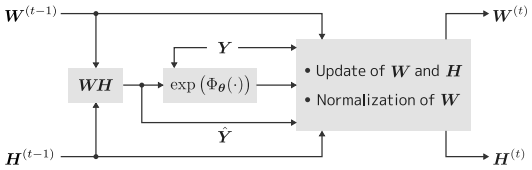

Fig. 2. The block diagram of the update rules based on the NN. Given $\boldsymbol{Y}$ and $\hat{\boldsymbol{Y}}$, $\exp\left(\Phi_{\boldsymbol{\theta}}(\cdot)\right)$ is calculated for each element. At the tth iteration, the update of $\boldsymbol{W}$ and $\boldsymbol{H}$ is done by (6) and (7). In addition, $\boldsymbol{W}$ is normalized using (9).

B. Network architecture

As described in the previous subsection, (8) represents the relation between the statistical model and the update rules. We exploit (8) to design the NN which acts as the PDF and the algorithm of the NMF. The basic idea of the proposed architecture is to replace ζ_{mn} with the output of a NN. Specifically, using a function $\Phi_{\boldsymbol{\theta}} : (0, \infty) \to \mathbb{R}$ based on a NN with a vector of trainable parameters $\boldsymbol{\theta}$, we calculate ζ_{mn} as

$$\tilde{\zeta}_{mn}^{\boldsymbol{\theta}} = \exp\left(\Phi_{\boldsymbol{\theta}}(y_{mn}\hat{y}_{mn}^{-1})\right). \tag{12}$$

This gives the interpretation of the update rules as the NN illustrated in Fig. 2. This network acts as the DeMUA sub-block in Fig. 1. At the tth iteration, the DeMUA sub-block calculates the new basis and activation matrices (denoted by $\boldsymbol{W}^{(t)}$ and $\boldsymbol{H}^{(t)}$, respectively) using (12) in addition to (6), (7), and (9). To obtain the final estimate of the NMF parameters, the sub-block in Fig. 2 is stacked as shown in Fig. 1 while sharing $\boldsymbol{\theta}$.

The NN $\Phi_{\boldsymbol{\theta}}$ can be interpreted as the PDF in addition to a part of the update rules. Given $\Phi_{\boldsymbol{\theta}}$, by solving (8) with respect to f, we can obtain f as

$$\tilde{f}_{\boldsymbol{\theta}}(r) \propto \exp\left(-\int \exp\left(\Phi_{\boldsymbol{\theta}}(r)\right) dr\right). \tag{13}$$

From (1), $\tilde{f}_{\boldsymbol{\theta}}(|y_{mn}^{\mathbb{C}}|^2)$ is regarded as a complex PDF of $y_{mn}^{\mathbb{C}}$. The integral in (13) cannot always be evaluated analytically. However, we have the heuristic update rules stated in (6) and (7) which minimize the neural-based cost function:

$$\tilde{\mathcal{J}}_{\boldsymbol{\theta}}(\boldsymbol{W}, \boldsymbol{H}) = \sum_{m,n} \left(\log \hat{y}_{mn} - \log \tilde{f}_{\boldsymbol{\theta}}(y_{mn}\hat{y}_{mn}^{-1})\right). \tag{14}$$

C. Training procedure in audio denoising

In the denoising task [11], [21], the noise is reduced by applying the NMF to the noisy spectrogram. Since the DeMUA computes the basis and the activation matrices in a similar way to the conventional NMF algorithms, the DeMUA can be straightforwardly applied to this task. To obtain the network output as an estimate of the NMF parameters, the DeMUA takes a random input as the initial value of the NMF parameters and the spectrogram of the observed signal. Using the estimate of the NMF parameters, an estimate of the clean spectrogram $\hat{s}_{mn}^{\mathbb{C}}$ is computed as

$$\hat{s}_{mn}^{\mathbb{C}} = \frac{\sqrt{\hat{y}_{mn}}}{|y_{mn}^{\mathbb{C}}|} y_{mn}^{\mathbb{C}}. \tag{15}$$

The waveform of $\hat{s}_{mn}^{\mathbb{C}}$ is obtained using the inverse short-time Fourier transform (STFT). The forward propagation is shown in Algorithm 1. The DeMUA can be trained so that a loss function between the estimate and the corresponding target waveform can be minimized. To obtain a PDF which improves the performance, in this paper, we train the DeMUA using the loss function based on the performance measure in the denoising tasks. In particular, on the basis of the scale-independent SDR (SI-SDR) [16], we minimize the following loss function:

$$\mathcal{M}_{\mathrm{a}}(\boldsymbol{s}, \hat{\boldsymbol{s}}_{\boldsymbol{\theta}}) = \frac{\|\boldsymbol{s}\|^2 \|\hat{\boldsymbol{s}}_{\boldsymbol{\theta}}\|^2}{(\boldsymbol{s}^\top \hat{\boldsymbol{s}}_{\boldsymbol{\theta}})^2} - 1, \tag{16}$$

where $\boldsymbol{s}$ and $\hat{\boldsymbol{s}}_{\boldsymbol{\theta}}$ are the vectors of the target waveform and its estimate, respectively, and $\|\cdot\|$ is the l_2 norm.

Algorithm 1 The forward propagation of audio denoising

Obtain the complex-valued spectrogram $\boldsymbol{Y}^{\mathbb{C}} = [y_{mn}^{\mathbb{C}}]$ and its power spectrogram $\boldsymbol{Y} = [|y_{mn}^{\mathbb{C}}|^2]$ of the observed signal.
Initialize the NMF parameters $\boldsymbol{W}^{(0)}$ and $\boldsymbol{H}^{(0)}$.
for $t = 1, \ldots, T$ **do**
  Update $\boldsymbol{W}^{(t)}$ and $\boldsymbol{H}^{(t)}$ using (6) and (7), respectively.
  Normalize $\boldsymbol{W}^{(t)}$ using (9).
end for
Calculate the estimate of the clean spectrogram using (15).
Calculate the waveform of the estimated clean signal.

Algorithm 2 The forward propagation of supervised signal separation

`# Training stage`
Obtain the power spectrogram $\boldsymbol{Y}_l$ of the training signal for each speaker l ($l = 1, \ldots, L$).
for $l = 1, \ldots, L$ **do**
  Initialize the NMF parameters $\boldsymbol{W}_l^{(0)}$ and $\boldsymbol{H}_l^{(0)}$.
  for $t = 1, \ldots, T_1$ **do**
    Update $\boldsymbol{W}_l^{(t)}$ and $\boldsymbol{H}_l^{(t)}$ using (6) and (7), respectively.
    Normalize $\boldsymbol{W}_l^{(t)}$ using (9).
  end for
end for
Construct the trained basis matrix as $\tilde{\boldsymbol{W}} = [\boldsymbol{W}_1^{(T)}, \ldots, \boldsymbol{W}_L^{(T)}]$.
`# Separation stage`
Obtain the complex-valued spectrogram $\boldsymbol{X}^{\mathbb{C}} = [x_{mn}^{\mathbb{C}}]$ and its power spectrogram $\boldsymbol{X} = [|x_{mn}^{\mathbb{C}}|^2]$ of the mixture signal.
Initialize the weight matrix $\tilde{\boldsymbol{U}}^{(0)^\top} = [\boldsymbol{U}_1^{(0)^\top}, \ldots, \boldsymbol{U}_L^{(0)^\top}]$.
for $t = 1, \ldots, T_2$ **do**
  Given the power spectrogram $\boldsymbol{X}$ and the basis matrix $\tilde{\boldsymbol{W}}$, update $\tilde{\boldsymbol{U}}^{(t)}$ using (7).
end for
for $l = 1, \ldots, L$ **do**
  Calculate the estimate of the spectrogram $\hat{\boldsymbol{S}}_l^{\mathbb{C}}$ of the lth speaker using (17).
  Calculate the waveform of $\hat{\boldsymbol{S}}_l^{\mathbb{C}}$.
end for

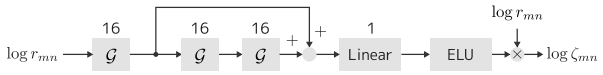

Fig. 3. The architecture of $\Phi_{\boldsymbol{\theta}}$ used in the experiment. The number above each layer indicates its output size. $\mathcal{G}$ represents the gated linear unit (GLU). "ELU" indicates the exponential linear unit activation function.

D. Training procedure in supervised signal separation

The DeMUA can also be applied to the supervised signal separation based on the NMF [2]. The forward propagation in this application is shown in Algorithm 2. The separation procedure consits of the training and the separation stages. In the training stage, we train the basis matrix $\boldsymbol{W}_l$ for each speaker l ($l = 1, \ldots, L$) using the training data. To separate the mixture signal, in the separation stage, the weight matrix $\tilde{\boldsymbol{U}}^\top = [\boldsymbol{U}_1^\top, \ldots, \boldsymbol{U}_L^\top]$ is learned on the mixture spectrogram $\boldsymbol{X}^{\mathbb{C}}$ while the set of the trained basis matrices $\tilde{\boldsymbol{W}} = [\boldsymbol{W}_1, \ldots, \boldsymbol{W}_L]$ is fixed. We reconstruct the complex-valued spectrogram $\hat{\boldsymbol{S}}_l^{\mathbb{C}}$ of the lth speaker using the Wiener filter:

$$\hat{\boldsymbol{S}}_l^{\mathbb{C}} = (\boldsymbol{W}_l \boldsymbol{U}_l) \oslash (\tilde{\boldsymbol{W}} \tilde{\boldsymbol{U}}) \odot \boldsymbol{X}^{\mathbb{C}}, \tag{17}$$

where $\oslash$ and $\odot$ are element-wise division and product, respectively. We train the DeMUA by minimizing the average SI-SDR-based loss function as follows:

$$\mathcal{M}_{\mathrm{b}}(\{\boldsymbol{s}_l, \hat{\boldsymbol{s}}_{l\boldsymbol{\theta}}\}) = \frac{1}{L} \sum_{l=1}^{L} \frac{\|\boldsymbol{s}_l\|^2 \|\hat{\boldsymbol{s}}_{l\boldsymbol{\theta}}\|^2}{(\boldsymbol{s}_l^\top \hat{\boldsymbol{s}}_{l\boldsymbol{\theta}})^2} - 1, \tag{18}$$

where $\boldsymbol{s}_l$ and $\hat{\boldsymbol{s}}_{l\boldsymbol{\theta}}$ are the target waveform and its estimate of the lth speaker, respectively.

IV. Experiments

We evaluated the DeMUA in audio denoising [11], [21] and supervised signal separation [2]. In our experimetns, we used three architectures for $\Phi_{\boldsymbol{\theta}}$. The two of them regarded as the multiplicative update algorithms for t-NMF [10] and GGD-NMF [11]. Specifically, for these method, we constructed Φ as follows:

$$t\text{-NMF: } \Phi_{\theta_1}^t(r_{mn}) = -\log\left(\frac{e^{\theta_1} + 2r_{mn}}{e^{\theta_1} + 2}\right) \tag{19}$$

$$\text{GGD-NMF: } \Phi_{\theta_2}^{\mathrm{GGD}}(r_{mn}) = \frac{e^{\theta_2} - 2}{2} \log r_{mn} \tag{20}$$

where $r_{mn} = y_{mn} \hat{y}_{mn}^{-1}$ and $\theta_1, \theta_2 \in \mathbb{R}$ are the trainable parameters. Note that the hyper-parameters of t-NMF and GGD-NMF are $\nu = e^{\theta_1}$ and $\beta = e^{\theta_2}$, respectively. As a generalized statistical model, we designed the network $\Phi_{\boldsymbol{\theta}}$ using the gated linear unit (GLU) [22]. The $\Phi_{\boldsymbol{\theta}}$ used in our experiments is illustrated in Fig. 3.

A. Audio denoising

In the audio denoising task, the spectrogram of the source is corrupted by an impulsive noise. The impulsive noise simulates a musical noise, which is a common artifact caused by spectral subtraction algorithms [11]. The power and the phase spectrograms of the impulsive noise were drawn from the log-normal distribution with scale 4 and the uniform distribution, respectively. The waveform of the synthetic noise was added to the source signal at signal-to-noise ratio (SNR) of -10 dB. For source signals, we extracted 12 recordings played on electric guitar from the IDMT-SMT-GUITAR database [23]. We divided the recordings into two sets: SID1-1 and SID1-2, which consist of `AR_Lick[1-6]_KN.wav` and `AR_Lick[7-12]_KN.wav`, respectively. Each of the signals was resampled at 8 kHz. We trained the DeMUA using the 6 sources in SID1-1 and 5 i.i.d. random initial values for $\boldsymbol{W}$ and $\boldsymbol{H}$. The noise signal was drawn at each iteration. The size of the dataset is similar to that used in [11], [21]. Our dataset is small but enough to evaluate the NMF-based denoising method.

To obtain the power spectrogram, we performed the STFT with 128 ms hamming window and 32 ms hop size. In this experiment, the number of bases was set to $K = 30$.

The estimates of $\boldsymbol{W}$ and $\boldsymbol{H}$ were obtained from the output of the 300th stacks. The network parameters were updated with the Adam optimizer [24] and a batch size of 10. To train $\Phi_{\theta_1}^t$ and $\Phi_{\theta_2}^{\mathrm{GGD}}$, the learning rate 10^{-2} was used. We initialized t-NMF and GGD-NMF using $e^{\theta_1} = 4$ and $e^{\theta_2} = 1.2$, respectively. To calculate the gradient of the parameters, the backpropagation was truncated by the last 100 stacks. The gradient clipping [25] with threshold 1 was used for the training. To train $\Phi_{\boldsymbol{\theta}}$, we used the learning rate 10^{-4} and the truncation level 30.

At the test time, we obtained the estimated sources using SID1-1 and SID1-2. Each NMF algorithm was performed on two datasets: one consists of SID1-1, 5 i.i.d. random initial values for $\boldsymbol{W}$ and $\boldsymbol{H}$, and 10 i.i.d. noise signals, another contains SID1-2 instead of SID1-1. The initial values of the NMF and the noise signals were shared between the datasets but were different from those used at the training time. To evaluate the denoising performance, we used the SI-SDR improvement.

As the results of the denoising experiment, the evolution of the SI-SDR improvement depending on the iteration number of the multiplicative update algorithm is shown in Fig. 4. In this figure, the SI-SDR improvements obtained by using each architecture were averaged within each test set. Moreover, the numerical results are shown in Table I. In Fig. 4 and Table I, the evaluation results at initialization time are also shown. From Fig. 4, we can see the effective update rules were obtained by training the DeMUA. In addition, the denoising performance can be improved by generalizing the statistical model using the NN-based architecture $\Phi_{\boldsymbol{\theta}}$.

In this experiment, we obtained $\nu \simeq 0.73$ and $\beta \simeq 0.14$ as the estimates of the hyper-parameters of t-NMF and GGD-NMF, respectively. This suggests that heavier-tailed distribution shows higher SI-SDR improvements because $\hat{y}_{mn}$ can be prevented from fitting to the noise by using the algorithm based on the heavy-tailed distribution. At initialization time, the SI-SDRs decrease after about 50th iteration in Fig. 4. On the other hand, this performance degradation can be prevented

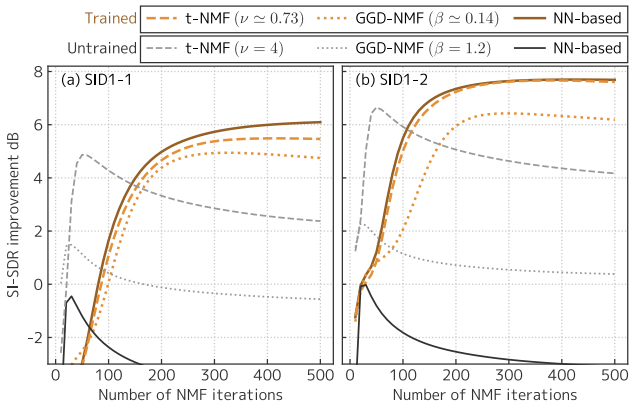

Fig. 4. Evolution of the average SI-SDR improvement depending on the iteration number of the multiplicative update algorithm in audio denoising. "NN-based" is the result using Fig. 3. "SID1-1" was used for training. The black and the gray lines show the results at initialization time.

TABLE I
THE NUMERICAL RESULTS OF AUDIO DENOISING IN TERMS OF THE AVERAGE SI-SDR IMPROVEMENT DB AT $T = 300, 500$.

Trained				
Source	SID1-1		SID1-2	
Number of iterations T	300	500	300	500
t-NMF (Eq. (19), $\nu \simeq 0.73$)	5.37	5.46	7.62	7.61
GGD-NMF (Eq. (20), $\beta \simeq 0.14$)	4.93	4.75	6.42	6.19
NN-based (Fig. 3)	**5.73**	**6.09**	**7.64**	**7.69**
Untrained				
Source	SID1-1		SID1-2	
Number of iterations T	300	500	300	500
t-NMF (Eq. (19), $\nu \simeq 4$)	2.86	2.37	4.63	4.17
GGD-NMF (Eq. (20), $\beta \simeq 1.2$)	−0.34	−0.56	0.55	0.38
NN-based (Fig. 3)	−3.67	−4.04	−2.83	−3.09

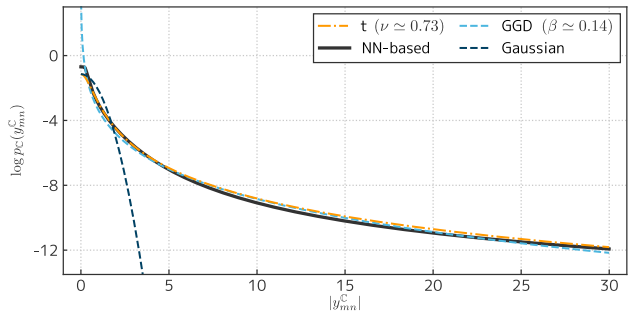

Fig. 5. The log-PDFs with respect to $|y^{\mathbb{C}}_{mn}|$ estimated in audio denoising. The log-PDF of the Gaussian distribution is also shown.

by training the DeMUA while considering the denoising performance.

In Fig. 5, we show the log-PDF of $\Phi_{\boldsymbol{\theta}}$ which is calculated from the trained network using (13). The estimated log-PDF is more close to the t distribution than the GGD. This result is reasonable because $\Phi_{\boldsymbol{\theta}}$ shows similar performance to t-NMF with $\nu \simeq 0.73$ in Fig. 4.

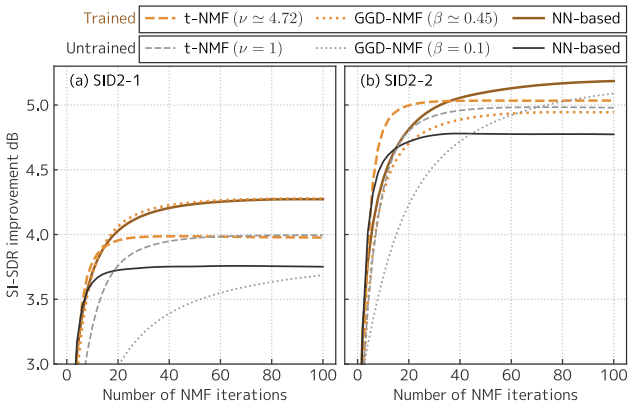

Fig. 6. Evolution of the average SI-SDR improvement depending on the iteration number of the multiplicative update algorithm in supervised signal separation. "NN-based" is the result using Fig. 3. "SID2-1" was used for training. The black and the gray lines show the results at initialization time.

TABLE II
THE NUMERICAL RESULTS OF SUPERVISED SIGNAL SEPARATION IN TERMS OF THE AVERAGE SI-SDR IMPROVEMENT DB AT $T_2 = 50, 100$.

Trained				
Source	SID2-1		SID2-2	
Number of iterations T_2	50	100	50	100
t-NMF (Eq. (19), $\nu \simeq 4.72$)	3.99	3.98	5.03	5.03
GGD-NMF (Eq. (20), $\beta \simeq 0.45$)	**4.25**	**4.28**	4.92	4.94
NN-based (Fig. 3)	4.24	4.27	**5.10**	**5.19**
Untrained				
Source	SID2-1		SID2-2	
Number of iterations T_2	50	100	50	100
t-NMF (Eq. (19), $\nu \simeq 1$)	3.97	4.00	4.97	4.98
GGD-NMF (Eq. (20), $\beta \simeq 0.1$)	3.48	3.69	4.84	5.09
NN-based (Fig. 3)	3.75	3.75	4.78	4.77

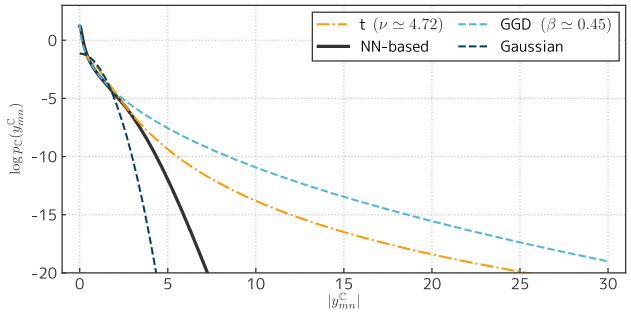

Fig. 7. The log-PDFs with respect to $|y^{\mathbb{C}}_{mn}|$ estimated in supervised signal separation. The log-PDF of the Gaussian distribution is also shown.

B. Supervised signal separation

In this experiment, we separated a mixture signal into two source signals. We used 20 pairs of the female and the male speakers resampled at 8 kHz from the Librispeech database [26]. We constructed SID2-1 and SID2-2, which consit of 10 speaker pairs, respectively. The number of pairs of each dataset is similar to that in [2]. At the training and the separation stages in Algorithm 2, the same speaker was used

to obtain $\boldsymbol{Y}_l$ and $\boldsymbol{X}$. The lengths of the signals at the training and the separation stages were 20 s and 5 s, respectively. To obtain $\tilde{\boldsymbol{W}}$, we used the number of bases $K = 30$ per speaker and $T_1 = 200$. At the separation stage, T_2 was set to $T_2 = 100$.

We trained the networks using SID2-1 and 5 i.i.d. random initial values for $\boldsymbol{W}_l$, $\boldsymbol{H}_l$, and $\boldsymbol{U}_l$. The configuration of the optimizer was same as the previous subsection. For t-NMF and GGD-NMF, we initialized using $e^{\theta_1} = 1$ and $e^{\theta_2} = 0.1$. To train $\Phi^t_{\theta_1}$ and $\Phi^{\mathrm{GGD}}_{\theta_2}$, the truncation levels of the backpropagation were 100 and 50 for the training and the separation stages, respectively. On the other hand, to train $\Phi_{\boldsymbol{\theta}}$, we set those to 10 for both two stages. The threshold of the gradient clipping was set to 0.01. We evaluated the separation performance using SID2-1, SID2-2, and 5 i.i.d. random initial values for the NMF parameters.

In Fig. 6 and Table II, we show the evolution of the SI-SDR improvement and the numerical results, respectively. Table II shows that GGD-NMF overfits on SID2-1, and the performance of t-NMF degrades compared to the initial condition although the parameter ϑ_1 was trained. However, the effective update rules were obtained by training the NN-based architecture as shown in Fig. 6.

Remarkably, even though the statistical model was randomly generated, the NN-based architecture shows about 4.77 dB for SID2-2 as shown in the black line of Fig. 6(b). This suggests that the separation algorithm depends less on the statistical model, and more on the low-rank approximation model of the NMF.

We show the log-PDF of $\Phi_{\boldsymbol{\theta}}$ in Fig. 7. $\Phi_{\boldsymbol{\theta}}$ represents the PDF with a peak as sharp as the GGD with $\beta \simeq 0.45$ and a thinner tail than the t distribution with $\nu \simeq 4.72$. Whereas fitting to the observed matrix is prevented by the heavy tail in the denoising task, we guess $\Phi_{\boldsymbol{\theta}}$ in this task reconstructs the observed matrix accurately using its thinner tail to enhance the low-rank approximation ability.

V. Conclusions

In this paper, we proposed the network architecture, called DeMUA for the NMF, which can not only solve the optimization problem of the NMF, but also represent the PDF used in the statistical model of the NMF. Moreover, we evaluated the proposed framework on the denoising and the separation tasks. The experiment results showed that the effective statistical models are obtained by training the proposed architecture using the performance-based cost function, and the generalized statistical models represented by the NN improve the denoising and the separation performances. Future work includes investigation of a network architecture appropriate to represent the PDF.

Acknowledgment

This work was partially supported by Grant-in-Aid for Young Scientists (KAKENHI 21K17769) and Grant-in-Aid for Scientific Research (C) (KAKENHI 20K12745).

References

[1] D. D. Lee and H. S. Seung, "Learning the parts of objects with nonnegative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[2] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 1–12, Jan. 2007.

[3] F. Weninger, J. Le Roux, J. R. Hershey, and S. Watanabe, "Discriminative NMF and its application to single-channel source separation," in *Proc. 15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Singapore, Sept. 2014, pp. 865–869.

[4] F.-J. Canadas-Quesada, P. Vera-Candeas, N. Ruiz-Reyes, J. Carabias-Orti, and P. Cabanas-Molero, "Percussive/harmonic sound separation by non-negative matrix factorization with smoothness/sparseness constraints," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2014, no. 1, pp. 26–42, July 2014.

[5] K. W. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors," in *Proc. 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, USA, Mar. 2008, pp. 4029–4032.

[6] N. Lyubimov and M. Kotov, "Non-negative matrix factorization with linear constraints for single-channel speech enhancement," in *Proc. 14th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Lyon, France, Aug. 2013, pp. 446–450.

[7] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2140–2151, Oct. 2013.

[8] C. Fevotte, 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, Sept. 2008.

[9] A. Liutkus, D. FitzGerald, and R. Badeau, "Cauchy nonnegative matrix factorization," in *Proc. 2015 IEEE International Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New York, USA, Oct. 2015.

[10] K. Yoshii, K. Itoyama, and M. Goto, "Student's t nonnegative matrix factorization and positive semidefinite tensor factorization for single-channel audio source separation," in *Proc. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Mar. 2016, pp. 51–55.

[11] D. Kitamura, "Nonnegative matrix factorization based on complex generative model," *Acoustical Science and Technology*, vol. 40, no. 3, pp. 155–161, May 2019.

[12] U. Simsekli, A. Liutkus, and A. T. Cemgil, "Alpha-stable matrix factorization," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2289–2293, Dec. 2015.

[13] H. Tanji, T. Murakami, and H. Kamata, "Laplace nonnegative matrix factorization with application to semi-supervised audio denoising," in *Proc. 27th European Signal Processing Conference (EUSIPCO)*, A Corunã, Spain, Sept. 2019.

[14] H. Tanji, T. Murakami, and H. Kamata, "A generalization of Laplace nonnegative matrix factorization and its multichannel extension," in *Proc. 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Lanzhou, China, Nov. 2019, pp. 1694–1699.

[15] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, July 2006.

[16] J. Le Roux, S. Wisdom, D. Erdogan, and J. R. Hershey, "SDR - half-baked or well done?," in *Proc. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, May 2019, pp. 626–630.

[17] J. Le Roux, J. R. Hershey, and F. Weninger, "Deep NMF for speech separation," in *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, Australia, Apr. 2015, pp. 66–70.

[18] P. Smaragdis and S. Venkataramani, "A neural network alternative to non-negative audio models," in *Proc. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA, Mar. 2017, pp. 86–90.

[19] Q. Zhang, A. Nicolson, M. Wang, K. K. Paliwal, and C. Wang, "DeepMMSE: a deep learning approach to MMSE-based noise power

spectral density estimation," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 28, pp. 1404–1415, 2020.

[20] Y. Luo and N. Mesgarani, "Conv-tasnet: surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.

[21] P. Magron, R. Badeau, and A. Liutkus, "Lévy NMF for robust nonnegative source separation," in *Proc. 2017 IEEE International Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New York, USA, Oct. 2017, pp. 259–263.

[22] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. 34th International Conference on Machine Learning (ICML)*, Sydney, Australia, Aug. 2017, pp. 933–941.

[23] C. Kehling, J. Abesser, C. Dittmar, and G. Schuller, "Automatic tablature transcription of electric guitar recordings by estimation of score- and instrument-related parameters," in *Proc. 17th International Conference on Digital Audio Effects (DAFx)*, Erlangen, Germany, Sept. 2014.

[24] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proc. 3rd International Conference on Learning Representations (ICLR)*, San Diego, Dec. 2015.

[25] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. 30th International Conference on Machine Learning (ICML)*, Atlanta, USA, June 2013, pp. 1310–1318.

[26] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, Australia, Apr. 2015, pp. 5206–5210.