DEEP CLUSTERING WITH GATED CONVOLUTIONAL NETWORKS

Li Li[1,2], *Hirokazu Kameoka*[1]

[1]NTT Communication Science Laboratories, NTT Corporation, Japan
[2]University of Tsukuba, Japan

lili@mmlab.cs.tsukuba.ac.jp, kameoka.hirokazu@lab.ntt.co.jp

ABSTRACT

Deep clustering is a recently introduced deep learning-based method for speech separation. The idea is to model and train the mapping from each time-frequency (TF) region of a spectrogram to an embedding space so that the embedding features of the TF regions dominated by the same source are forced to get close to each other and those dominated by different sources are forced to get separated from each other. This allows us to construct binary masks by applying a regular clustering algorithm to the mapped embedding vectors of a test mixture signal. The original deep clustering uses a bidirectional long short-term memory (BLSTM) recurrent neural network (RNN) to model the embedding process. Although RNN-based architectures are indeed a natural choice for modeling long-term dependencies of time series data, recent work has shown that convolutional networks (CNNs) with gating mechanisms also have an excellent potential for capturing long-term structures. In addition, they are less prone to overfitting and are suitable for parallel computations. Motivated by these facts, this paper proposes adopting CNN-based architectures for deep clustering. Specifically, we use a gated CNN architecture, which was introduced to model word sequences for language modeling and was shown to outperform LSTM language models trained in a similar setting. We tested various CNN architectures on a monaural source separation task. The results revealed that the proposed architectures achieved better performance than the BLSTM-based architecture under the same training condition and comparable performance even with a smaller amount of training data.

Index Terms— monaural source separation, speaker-independent, multi-speaker separation, deep clustering, gated convolutional networks

1. INTRODUCTION

Monaural multi-speaker separation is the challenging task of separating out all the individual speech signals from an observed mixture signal. Although human can easily focus on listening to one voice from multiple voices sounding simultaneously, this is an extremely difficult problem for machines, which is well known as a *cocktail party problem* [1]. Recently, inspired by the success of deep learning in different areas [2–5], many deep learning-based methods have been proposed to tackle this problem [6–10].

One impressive approach known as deep clustering [7] has shown great improvements in speaker-independent multi-speaker separation tasks. Deep clustering is a binary mask estimation framework, which is theoretically able to deal with arbitrary number of sources. One important feature as regards deep clustering involves permutation invariance. Namely, speaker labels do not need to be consistent over different utterances in training data. This particular feature makes this approach practically convenient. Deep clustering uses neural networks to learn a mapping from a feature vector obtained at each time-frequency (TF) region of an observed spectrogram to a high-dimensional embedding space such that embedding vectors that originate from the same source are forced to get close to each other and those that do not are forced to be separated from each other. At test time, we can thus obtain binary masks by first mapping the feature vector obtained at each TF point to the embedding space and then clustering the embedding vectors. In the original deep clustering paper, a bidirectional long short-term memory (BLSTM) recurrent neural network (RNN) was used to model the embedding process.

Although RNN-based architectures are indeed a natural choice for modeling long-term dependencies of time series data, recent work has shown that convolutional networks (CNNs) with gating mechanisms also have an excellent potential for capturing long-term structures. In addition, they are less prone to overfitting and are more suitable for parallel computations than RNNs. Motivated by this fact, we propose using CNN-based architectures to model the embedding process of deep clustering and investigate which architecture is best suited to source separation tasks. All the network architectures we investigated were built using the gated CNN [11], which was originally introduced to model word sequences for language modeling and was shown to outperform LSTM language models trained in a similar setting. Similar to LSTMs, the gating mechanism of gated CNNs allows the network to learn what information should be propagated through the hierarchy of layers. This mechanism is notable in that it can effectively prevent the network from suffering from the vanishing gradient problem. We also investigate the use of bottleneck architectures and dilated convolution [12,13]. Dilated convolution is similar to standard convolution, but is different in that the filters can be dilated or upsampled by inserting zeros between coefficients. This allows networks to model longer-term contextual dependencies with the same number of parameters. We also compare 2-dimensional convolution with 1-dimensional convolution which treats 2-dimensional

978-1-5386-4658-8/18/$31.00 ©2018 IEEE

inputs as sequences with multiple channels.

This paper is organized as follows. In sec. 2, we introduce the details of our proposed CNN-based architectures following with a review of the deep clustering method and the gated CNN architecture. We present the experimental results in sec. 3 and conclude in sec. 4.

2. DEEP CLUSTERING WITH CNN-BASED ARCHITECTURES

In this section, we first review the deep clustering method and the gated CNN architecture, which are the core components of the proposed method. We then show the details of the network architectures we investigated.

2.1. Deep clustering

Based on an assumption that the energy of each time-frequency (TF) region of a mixture signal is dominated by a single source, deep clustering [7] aims to find a set of TF points that are dominated by the same source. Given a mixture signal consisting of C sources, we denote its TF representation (e.g., log magnitude spectrogram) by $\mathbf{X} = \{X_n\} \in \mathbb{R}^{N\times 1}$, where n denotes a pair of the frequency and time indices (f,t) and so N is the number of TF points, $F \times T$. Deep clustering projects each TF region X_n into an unit D-dimensional embedding vector $\mathbf{V}_n = (V_{n,1},\ldots,V_{n,D})^T$ with a BLSTM network $\mathbf{V} = g_{\mathbf{\Theta}}(\mathbf{X})$, where $g(\cdot)$ denotes the nonlinear transformation operated by the network, $\mathbf{\Theta}$ denotes parameters of the network and $\mathbf{V} = \{\mathbf{V}_n\} \in \mathbb{R}^{N\times D}$. The BLSTM network can be trained by minimizing the objective function

$$\mathcal{J}(\mathbf{V}) = ||\mathbf{V}\mathbf{V}^T - \mathbf{Y}\mathbf{Y}^T||_F^2 \tag{1}$$

$$= ||\mathbf{V}^T\mathbf{V}||_F^2 - 2||\mathbf{V}^T\mathbf{Y}||_F^2 + ||\mathbf{Y}^T\mathbf{Y}||_F^2. \tag{2}$$

where $||\cdot||_F^2$ is the squared Frobenius norm. In (1), $\mathbf{Y} = \{\mathbf{Y}_{n,c}\} \in \mathbb{R}^{N\times C}$ is a source indicator matrix consisting of one-hot vectors in rows, indicating to which source among $1,\ldots,C$ the TF region n belongs. In this case, $\mathbf{Y}\mathbf{Y}^T$ is an $N \times N$ binary affinity matrix, where the element is given by $(\mathbf{Y}\mathbf{Y}^T)_{nn'} = 1$ if TF region n and n' are dominated by the same source, otherwise the element is given by $(\mathbf{Y}\mathbf{Y}^T)_{nn'} = 0$. This implies that this objective function encourages the mapped embedding vectors to become parallel if they are dominated by the same source and become orthogonal otherwise. Hence, the embedding vectors originating from the same source will be likely to form a single cluster. Here, although it may appear that $\mathbf{V}\mathbf{V}^T$ and $\mathbf{Y}\mathbf{Y}^T$ can be too huge to compute, we can use (2) to compute the gradients of $\mathbf{\Theta}$ with a reasonably small amount of computational effort. At test time, a clustering algorithm (e.g., K-means) is applied to the assigned embedding vectors of the observed mixture spectrogram to obtain binary mask for each source.

2.2. Proposed method and network architectures

RNNs, in particular LSTMs, are a natural choice for modeling time series data since the recurrent connection architectures allow the networks to make prediction with the entire input time series. However, the deeper the network architecture becomes, the more challenging its training becomes. Furthermore, it is difficult to employ parallel implementations for RNNs; thus, the training and prediction processing become computationally demanding. Motivated by the recent success achieved by CNNs in language modeling and the merits of CNNs that they are practically much easier to train and well suited to parallel implementation, in this paper, we propose using CNN-based neural networks to model the embedding process of deep clustering. Considering the fact that log magnitude spectrograms of speech signals have region dependency (i.e. they have different frequency structures in voiced and unvoiced segments), we use the gated CNN architecture [11] to design all the network architectures. We call the proposed method the gated convolutional deep clustering (GCDC).

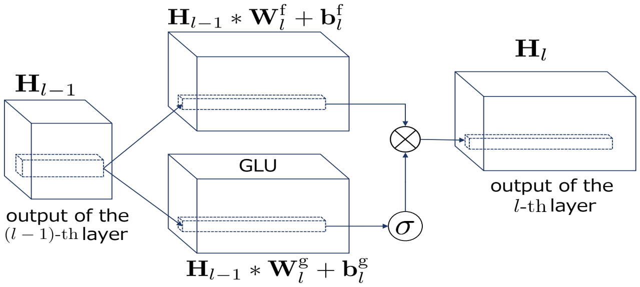
$\mathbf{H}_{l-1} * \mathbf{W}_l^{\mathrm{f}} + \mathbf{b}_l^{\mathrm{f}}$
$\mathbf{H}_l$
$\mathbf{H}_{l-1}$
GLU
$\otimes$
σ
output of the l-th layer
output of the $(l-1)$-th layer
$\mathbf{H}_{l-1} * \mathbf{W}_l^{\mathrm{g}} + \mathbf{b}_l^{\mathrm{g}}$

Fig. 1. Architecture of a gated CNN

2.2.1. *Gated convolutional networks*

By using $\mathbf{H}_{l-1}$ to denote the output of the $(l-1)$-th layer, the output of the l-th layer $\mathbf{H}_l$ of a gated CNN is given as a linear projection $\mathbf{H}_{l-1} * \mathbf{W}_l^{\mathrm{f}} + b_l^{\mathrm{f}}$ modulated by an output gate $\sigma(\mathbf{H}_{l-1} * \mathbf{W}_l^{\mathrm{g}} + b_l^{\mathrm{g}})$.

$$\mathbf{H}_l = (\mathbf{H}_{l-1} * \mathbf{W}_l^{\mathrm{f}} + b_l^{\mathrm{f}}) \otimes \sigma(\mathbf{H}_{l-1} * \mathbf{W}_l^{\mathrm{g}} + b_l^{\mathrm{g}}), \tag{3}$$

where $\mathbf{W}_l^{\mathrm{f}}$, $\mathbf{W}_l^{\mathrm{g}}$, b_l^{f} and b_l^{g} are weight and bias parameters of the l-th layer, $\otimes$ denotes the element-wise multiplication and σ is the sigmoid function. Fig. 1 shows the gated CNN architecture. The main difference between a gated CNN and a regular CNN layer is that a gated linear unit (GLU), namely the second term of (3), is used as an nonlinear activation function instead of tanh activation or regular rectified linear units (ReLUs) [14]. Similar to LSTMs, GLUs are data-driven gates, which play the role of controlling the information passed on in the hierarchy. This particular mechanism allows us to capture long-range context dependencies efficiently by deepening the layers without suffering from the vanishing gradient problem.

2.2.2. *Network architectures*

For network architecture designs, we focused on how to deal with the 2-dimensional inputs and how to efficiently capture long-term contextual dependencies.

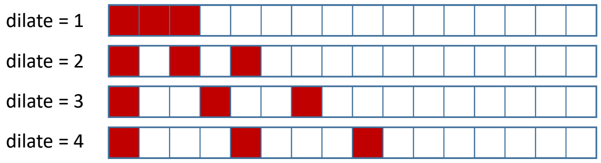

Fig. 2. Example of 1-dimensional dilated convolutions with various dilate numbers. Red blocks denote the origial filter.

A. 1D convolution or 2D convolution

For the first question, we investigate both 1-dimensional (1D) convolution and 2-dimensional (2D) convolution. With 1D convolution models, the frequency dimension is regarded as the channel dimension (just like the RGB channels of an image) and an input spectrogram is convolved with a $(1, k_T)$ filter, where k_T is the filter width in the time dimension. With 2D convolution models, an input spectrogram is convolved with a (k_F, k_T) filter, where k_F denotes the filter width in the frequency dimension.

B. Bottleneck or dilated convolution

To capture long-term contextual dependencies without increasing the parameters, we use bottleneck architectures and dilated convolution. With bottleneck architectures, 2-dimensional inputs are downsampled to 1/2 size at each layer by setting the stride at 2, and upsampled to the original size using deconvolutional networks [15]. We also use a skip architecture [16] to combine the final output layer with lower layer outputs. This allows the network to take account of both the higher-level and lower-level features when generating outputs.

Dilated convolution [12] is another effective approach allowing CNNs to capture wider receptive fields with a fewer parameters. Fig. 2 shows an image of 1-dimensional dilated convolutions with various dilate settings. Dilated convolution handles wider receptive fields without increasing model parameters by convolving a larger filter derived from the original filter with dilating zeros, namely the original filter is applied by skipping certain elements in the input.

C. Other settings

Tab. 1 details the network architectures. The symbols ↓ and ↑ denote downsampling and upsampling respectively. In practice, we use convolutional networks and deconvolutional networks with stride 2. Batch normalization [17] is applied to each layer. The number of layers and channels are set to different values depending on the scale of the networks and dataset. More specific, we use 64 channels for a sub training dataset and 128 channels for the total training dataset. All the models are designed to fit a single GPU memory.

3. EXPERIMENTS

3.1. Datasets and experimental settings

To comparatively evaluate the proposed method with BLSTM-based deep clustering [7], we created datasets using the utterances from the Wall Street Journal (WSJ0) corpus and data generation code provided in [18], which was also used for the evaluation of the previous deep clustering work [7–9]. It consisted in a 30h training data and 10h validation data generated by randomly mixing two different speakers selected from the WSJ10 training set `si_tr_s` with signal-to-noise ratios between 0 dB and 10 dB. A 5h test set was similarly generated using utterances from `si_dt_05` and `si_et_05`. The speakers were different from those in the training set and validation set. We created a sub dataset with 1/5 training data (roughly 5.5h) and 0.5h validation data to evaluate the effectiveness of our models when only a limited scale dataset is available.

We downsampled the data to 8 kHz to save the computational and memory cost. We used log magnitude spectrograms as inputs and calculated them using a short-time Fourier transform (STFT) with a 254-point long hanning window with 1/2 overlap to keep the input frequency size $F = 128$ being an even number. A mixture was separated into segments of 128 frames with 1/2 overlap to train the networks. But we could take utterances with arbitrary length as inputs at test time since all the architectures were designed as fully convolutional networks. We set embedding dimension D at 20 or 40. According to the results reported in [7], 20 had taken a good balance of the separation performance and model size, while 40 had achieved the best source separation performance. We trained the networks using Adam optimizer with a minibatch of size 16 or 8 depending on the model size. To save memory cost, 400 frames of each utterance were randomly chosen to calculate the backpropagation of the objective function (1). TF regions with magnitude under -40 dB, compared to the maximum of the magnitude, were omitted in calculating the loss function as being done in [7,8]. Signal-to-distortion ratio (SDR) [19] improvement was used as the performance evaluation criterion.

3.2. Results and discussions

As a baseline, we implemented the BLSTM architecture described in [7]. Although we would have liked to exactly replicate their implementation, we made our own design choices owing to missing details of hyperparameters. While the average SDR improvement with $D = 20$ presented in the original paper was 5.7 dB, that obtained with our implementation was 2.46 dB on similar training and test conditions. This implies that our current design choices may not be optimal. Our future work includes further investigation of the hyperparameter settings.

Tab. 2 lists the average SDR improvement obtained by the proposed CNN-based architectures trained using the sub training dataset and the total dataset. These results indicated that our proposed architectures achieved similar level performance comparing to the BLSTM-based architecture presented in [7]. Both of the two architectures built using dilated convolution outperformed the baseline and obtained a 1.08 dB improvement in terms of SDR improvement, showing that the dilated convolution is more effective than bottleneck architectures. Furthermore, the architecture combining 2D convolution and dilated convolution not only obtained the highest score with 30h training data but also showed the capability to

Table 1. Architectures of CNN-based networks. Details are expressed as "$k_F \times k_T, \alpha, \beta, \gamma,$ ", where $k_F \times k_T$ denotes filter size, and α, β and γ denote channel number, stride number and dilation respectively. $\uparrow$ and $\downarrow$ denote upsampling and downsampling respectively.

layer #	2D, B, w/o skip	2D, B, w/ skip	2D, DC	1D	1D, DC
1th	5×5, 64/128, 1, 1	5×5, 64/128, 1, 1	3×3, 64/128, 1, 1	1×11, 512, 1, 1,	1×3, 512, 1, 1
2th	4×4, 64/128, $\downarrow$ 2, 1	4×4, 64/128, $\downarrow$ 2, 1	3×3, 64/128, 1, 2	1×11, 1024, 1, 1	1×3, 1024, 1, 2
3th	3×3, 64/128, 1, 1	3×3, 64/128, 1, 1,	3×3, 64/128, 1, 3	1×11, 2048, 1, 1	1×3, 2048, 1, 3
4th	4×4, 64/128, $\downarrow$ 2, 1	4×4, 64/128, $\downarrow$ 2, 1	3×3, 64/128, 1, 4	1×11, 2048, 1, 1	1×3, 4096, 1, 4
5th	3×3, 64/128, 1, 1	3×3, 64/128, 1, 1	3×3, D, 1, 5	1×11, $F \times D$, 1, 1	1×3, 4096, 1, 4
6th	4×4, 64/128, $\uparrow$ 2, 1	4×4, D, $\uparrow$ 2, 1			1×3, 2048, 1, 4
7th	4×4, D, $\uparrow$ 2, 1	4×4, D, $\uparrow$ 2, 1			1×3, $F \times D$, 1, 4

Table 2. Average SDR improvement [dB] obtained by the proposed architectures, BLSTM-based deep clustering trained using the sub training dataset and the total dataset with D=20. Bold font indicates top scores.

model		5.5h	30h
GCDC	2D, B, w/o skip	3.90	5.49
	2D, B, w skip	3.78	5.23
	2D, DC	**5.78**	**6.78**
	1D	3.49	5.16
	1D, DC	3.94	6.36
DC	our implementation	1.57	2.46
	[7]	-	5.7

Table 3. Comparison of the average SDR improvement [dB] obtained by the proposed architectures with the best performance achieved by the original BLSTM-based deep clustering with D=40. Bold font indicates the top score.

model		SDRi [dB]
GCDC	2D, DC	6.71
	1D, DC	6.39
DC	[7]	6.0
	[8]	**9.4**

Table 4. Average SDR improvement [dB] of 3-speaker separation task. Bold font indicates the top score.

model		SDRi [dB]
GCDC	2D, DC	3.14
	1D, DC	2.48
DC	[7]	2.2
	[8]	**7.1**

perform well even only 1/5 amount of data being provided.

We also trained the two architectures using dilated convolution which obtained the high scores in Tab. 2 with setting embedding dimension D at 40 and compared them to the results reported in [7, 8]. The proposed architectures outperformed the result presented in [7], which also indicated the effectiveness of the CNN-based architectures. However, as shown above, the SDRs obtained with GCDC were about 3 dB lower than those obtained with the deeper and fine-tuned BLSTM-based model presented in [8]. To confirm how close we can get to these results with GCDC, we implemented a deeper version of GCDC with 2D dilated convolutions and trained it on multiple GPUs. Through this implementation, we were able to achieve 9.07 dB SDR improvement, which is comparable to the result in [8]. It is noteworthy that although the deeper architecture consists of 14 layers, the parameter number is 2/3 less than the BLSTM-based architecture presented in [8]. The proposed model can thus be trained more quickly. Our future work also includes investigating more deeper architectures and the effectiveness of regularziations such as dropout, L1, L2 weight regularization on our models since regularization has shown to play a crucial role in improving the separation performance [8].

For reference, we tested two models trained using 2-speaker mixture data on a 3-speaker separation task using the same test dataset presented in [7]. Tab. 4 shows the results that the proposed architectures outperformed the BLSTM-based deep clustering. In [8], the well-tuned BLSTM-based model improved the 3-speaker source separation performance from 2.2 dB to 7.1 dB using a curriculum training, which points out another direction to improve our models.

4. CONCLUSIONS

Deep clustering is a recently proposed promising approach to solve cocktail party problem. In this paper, we proposed using CNN-based architectures instead of BLSTM networks to model the embedding process of deep clustering. We investigated 5 different CNN-based architectures based on the gated CNN architecture. The results revealed that the proposed architectures using dilated convolution achieved better performance than the BLSTM-based architecture and the other CNN-based architecturesls on monaural speaker-independent multispeaker separation tasks. We also showed that 2D convolution with dilated convolution architecture obtained a comparable performance even with a smaller training dataset.

5. ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI 17H01763.

6. REFERENCES

[1] E Colin Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the acoustical society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[2] Jürgen Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[3] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[4] Yuchen Fan, Yao Qian, Feng-Long Xie, and Frank K Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[5] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[6] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis, "Deep learning for monaural speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1562–1566.

[7] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 31–35.

[8] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey, "Single-channel multi-speaker separation using deep clustering," *Interspeech 2016*, pp. 545–549, 2016.

[9] Zhuo Chen, Yi Luo, and Nima Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 246–250.

[10] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 241–245.

[11] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," *arXiv preprint arXiv:1612.08083*, 2016.

[12] Fisher Yu and Vladlen Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[13] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," in *9th ISCA Speech Synthesis Workshop*, pp. 125–125.

[14] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[15] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus, "Deconvolutional networks," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2528–2535.

[16] Evan Shelhamer, Jonathan Long, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[17] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[18] Avaliable online, "http://www.merl.com/demos/deep-clustering," .

[19] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.