

DEEP NEURAL NETWORKS FOR SINGLE CHANNEL SOURCE SEPARATION

Emad M. Grais, Mehmet Umut Sen, Hakan Erdogan

Faculty of Engineering and Natural Sciences,
Sabanci University, Orhanli Tuzla, 34956, Istanbul.
{grais, umutsen, haerdogan}@sabanciuniv.edu

ABSTRACT

In this paper, a novel approach for single channel source separation (SCSS) using a deep neural network (DNN) architecture is introduced. Unlike previous studies in which DNN and other classifiers were used for classifying time-frequency bins to obtain hard masks for each source, we use the DNN to classify estimated source spectra to check for their validity during separation. In the training stage, the training data for the source signals are used to train a DNN. In the separation stage, the trained DNN is utilized to aid in estimation of each source in the mixed signal. Single channel source separation problem is formulated as an energy minimization problem where each source spectra estimate is encouraged to fit the trained DNN model and the mixed signal spectrum is encouraged to be written as a weighted sum of the estimated source spectra. The proposed approach works regardless of the energy scale differences between the source signals in the training and separation stages. Nonnegative matrix factorization (NMF) is used to initialize the DNN estimate for each source. The experimental results show that using DNN initialized by NMF for source separation improves the quality of the separated signal compared with using NMF for source separation.

Index Terms— Single channel source separation, deep neural network, nonnegative matrix factorization.

1. INTRODUCTION

Single channel audio source separation is an important and challenging research problem and has received considerable interest in recent years. Since there is limited information in the mixed signal, usually one needs to use training data for each source to model each source and to improve the quality of separation. In this work, we introduce a new method for improved source separation using nonlinear models of sources trained using a deep neural network.

1.1. Related work

Approaches to solve the single channel source separation problem usually require training data for each source. The training data can be modeled using probabilistic models such as Gaussian mixture model (GMM) [1, 2, 3], hidden Markov model (HMM) or factorial HMM [4, 5, 6]. These models are learned from the training data and usually used in source separation under the assumption that the sources appear in the mixed signal with the same energy level as they appear in the training data. Fixing this limitation requires complicated computations as in [7, 8, 9, 10, 11, 12]. Another approach to model the training data is to train nonnegative dictionaries for the source signals [13, 14, 15]. This approach is more flexible with no limitation related to the energy differences between the

source signals in training and separation stages. The main problem in this approach is that any nonnegative linear combination of the trained dictionary vectors is a valid estimate for a source signal which may decrease the quality of separation. Modeling the training data with both nonnegative dictionary and cluster models like GMM and HMM was introduced in [16, 17, 18, 19] to fix the limitation related to the energy scaling between the source signals and training more powerful models that can fit the data properly. Another type of approach which is called classification-based speech separation aims to find hard masks where each time-frequency bin is classified as belonging to either of the sources. For example in [20], various classifiers based on GMM, support vector machines, conditional random fields, and deep neural networks were used for classification.

1.2. Contributions

In this paper, we model the training data for the source signals using a single joint deep neural network (DNN). The DNN is used as a classifier which can classify its input spectra into each possible source type. Unlike classification-based speech separation where the classifiers are used to segment time-frequency bins into sources, we can obtain soft masks using our approach. Single channel source separation problem is formulated as an energy minimization problem where each source spectral estimate is encouraged to fit the trained DNN model and the mixed signal spectrum is encouraged to be written as a weighted sum of the estimated source spectra. Basically, we can think of the DNN as checking whether the estimated source signals are lying in their corresponding nonlinear manifolds which are represented by the trained joint DNN. Using a DNN for modeling the sources and handling the energy differences in training and testing is considered to be the main novelty of this paper. Deep neural network (DNN) is a well known model for representing the detailed structures in complex real-world data [21, 22]. Another novelty of this paper is using nonnegative matrix factorization [23] to find initial estimates for the sources rather than using random initialization.

1.3. Organization of the paper

This paper is organized as follows: In Section 2 a mathematical formulation for the SCSS problem is given. Section 3 briefly describes the NMF method for source separation. In Section 4, we introduce our new method. We present our experimental results in Section 5. We conclude the paper in Section 6.

2. PROBLEM FORMULATION

In single channel source separation problems, the aim is to find estimates of source signals that are mixed on a single channel $y(t)$. For simplicity, in this paper we assume the number of sources is two.

This problem is usually solved in the short time Fourier transform (STFT) domain. Let $Y(t, f)$ be the STFT of $y(t)$, where t represents the frame index and f is the frequency-index. Due to the linearity of the STFT, we have:

$$Y(t, f) = S_1(t, f) + S_2(t, f), \quad (1)$$

where $S_1(t, f)$ and $S_2(t, f)$ are the unknown STFT of the first and second sources in the mixed signal. In this framework [13, 24], the phase angles of the STFT were usually ignored. Hence, we can approximate the magnitude spectrum of the measured signal as the sum of source signals' magnitude spectra as follows:

$$|Y(t, f)| \approx |S_1(t, f)| + |S_2(t, f)|. \quad (2)$$

We can write the magnitude spectrogram in matrix form as follows:

$$\mathbf{Y} \approx \mathbf{S}_1 + \mathbf{S}_2. \quad (3)$$

where $\mathbf{S}_1, \mathbf{S}_2$ are the unknown magnitude spectrograms of the source signals and need to be estimated using the observed mixed signal and the training data.

3. NMF FOR SUPERVISED SOURCE SEPARATION

In this section, we briefly describe the use of nonnegative matrix factorization (NMF) for supervised single channel source separation. We will relate our model to the NMF idea and we will use the source estimates obtained from using NMF as initialization for our method, so it is appropriate to introduce the use of NMF for source separation first.

To find a suitable initialization for the sources signals, we use nonnegative matrix factorization (NMF) as in [23]. NMF [25] factorizes any nonnegative matrix \mathbf{V} into a basis matrix (dictionary) \mathbf{B} and a gain matrix \mathbf{G} as

$$\mathbf{V} \approx \mathbf{B}\mathbf{G}. \quad (4)$$

The matrix \mathbf{B} contains the basis vectors that are optimized to allow the data in \mathbf{V} to be approximated as a linear combination of its constituent columns. The solution for \mathbf{B} and \mathbf{G} can be found by minimizing the following Itakura-Saito (IS) divergence cost function [26]:

$$\min_{\mathbf{B}, \mathbf{G}} D_{IS}(\mathbf{V} \parallel \mathbf{B}\mathbf{G}), \quad (5)$$

where

$$D_{IS}(\mathbf{V} \parallel \mathbf{B}\mathbf{G}) = \sum_{a,b} \left(\frac{\mathbf{V}_{a,b}}{(\mathbf{B}\mathbf{G})_{a,b}} - \log \frac{\mathbf{V}_{a,b}}{(\mathbf{B}\mathbf{G})_{a,b}} - 1 \right).$$

This divergence cost function is a good measurement for the perceptual difference between different audio signals [26]. The IS-NMF solution for equation (5) can be computed by alternating multiplicative updates of \mathbf{G} and \mathbf{B} as follows:

$$\mathbf{G} \leftarrow \mathbf{G} \otimes \frac{\mathbf{B}^T \left(\frac{\mathbf{V}}{(\mathbf{B}\mathbf{G})^2} \right)}{\mathbf{B}^T \left(\frac{\mathbf{1}}{\mathbf{B}\mathbf{G}} \right)}, \quad (6)$$

$$\mathbf{B} \leftarrow \mathbf{B} \otimes \frac{\left(\frac{\mathbf{V}}{(\mathbf{B}\mathbf{G})^2} \right) \mathbf{G}^T}{\left(\frac{\mathbf{1}}{\mathbf{B}\mathbf{G}} \right) \mathbf{G}^T}, \quad (7)$$

where $\mathbf{1}$ is a matrix of ones with the same size of \mathbf{V} , the operation \otimes is an element-wise multiplication, all divisions and $(\cdot)^2$ are

element-wise operations. The matrices \mathbf{B} and \mathbf{G} are usually initialized by positive random numbers and then updated iteratively using equations (6) and (7).

In the initialization stage, NMF is used to decompose the frames for each source i into a multiplication of a nonnegative dictionary \mathbf{B}_i and a gain matrix \mathbf{G}_i as follows:

$$\mathbf{S}_i^{train} \approx \mathbf{B}_i \mathbf{G}_i^{train}, \quad \forall i \in \{1, 2\}, \quad (8)$$

where \mathbf{S}_i^{train} is the nonnegative matrix that contains the spectrogram frames of the training data of source i . After observing the mixed signal, we calculate its spectrogram \mathbf{Y}_{psd} . NMF is used to decompose the mixed signal's spectrogram matrix \mathbf{Y}_{psd} with the trained dictionaries as follows:

$$\mathbf{Y}_{psd} \approx [\mathbf{B}_1, \mathbf{B}_2] \mathbf{G} \quad \text{or} \quad \mathbf{Y}_{psd} \approx [\mathbf{B}_1 \quad \mathbf{B}_2] \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \end{bmatrix}. \quad (9)$$

The only unknown here is the gains matrix \mathbf{G} since the dictionaries are fixed. The update rule in equation (6) is used to find \mathbf{G} . After finding the value of \mathbf{G} , the initial estimate for each source magnitude spectrogram is computed as follows:

$$\hat{\mathbf{S}}_{init1} = \frac{\mathbf{B}_1 \mathbf{G}_1}{\mathbf{B}_1 \mathbf{G}_1 + \mathbf{B}_2 \mathbf{G}_2} \otimes \mathbf{Y}, \quad \hat{\mathbf{S}}_{init2} = \frac{\mathbf{B}_2 \mathbf{G}_2}{\mathbf{B}_1 \mathbf{G}_1 + \mathbf{B}_2 \mathbf{G}_2} \otimes \mathbf{Y}, \quad (10)$$

where \otimes is an element-wise multiplication and the divisions are done element-wise. The magnitude spectrograms of the initial estimates of the source signals are used to initialize the sources in the separation stage of the DNN approach.

4. THE METHOD

In NMF, the basic idea is to model each source with a dictionary, so that source signals appear in the nonnegative span of this dictionary. In the separation stage, the mixed signal is expressed as a nonnegative linear combination of the source dictionaries and separation is performed by taking the parts corresponding to each source in the decomposition.

The basic problem in NMF is that each source is modeled to lie in a nonnegative cone defined by all the nonnegative *linear* combinations of its dictionary entries. This assumption may be a limiting assumption usually since the variability within each source indicates that nonlinear models may be more appropriate. This limitation led us to consider nonlinear models for each source. It is not trivial to use nonlinear models or classifiers in source separation. Since deep neural networks were recently used with increased success in speech recognition and other object recognition tasks, they can be considered as superior models of highly variable real-world signals. Other classifiers may also be used in our framework, however we conjecture that they would not perform better than deep neural networks. We leave further discussion of this point to our future work.

We first train a DNN to model each source in the training stage. We then use an energy minimization objective to estimate the sources and their gains during the separation stage. Each stage is explained below.

4.1. Training the DNN

We train a DNN that can classify sources present in the mixed signal. The input to the network is a frame of normalized magnitude spectrum, $\mathbf{x} \in \mathbb{R}^d$. The neural network architecture is illustrated in Figure 1. There are two outputs in the DNN, each corresponding to a source. The label of each training instance is a binary indicator

function, namely if the instance is from source one, the first output label $f_1(\mathbf{x}) = 1$ and the second output label $f_2(\mathbf{x}) = 0$. Let n_k be the number of hidden nodes in layer k for $k = 0, \dots, K$ where K is the number of layers. Note that $n_0 = d$ and $n_K = 2$. Let $\mathbf{W}_k \in \mathbb{R}^{n_k \times n_{k-1}}$ be the weights between layers $k-1$ and k , then the values of a hidden layer $\mathbf{h}_k \in \mathbb{R}^{n_k}$ are obtained as follows:

$$\mathbf{h}_k = g(\mathbf{W}_k \mathbf{h}_{k-1}), \quad (11)$$

where $g(x) = \frac{1}{1+\exp(-x)}$ is the elementwise sigmoid function. We skip the bias terms to avoid clutter in our notation. The input to the network is $\mathbf{h}_0 = \mathbf{x} \in \mathbb{R}^d$ and the output is $f(\mathbf{x}) = \mathbf{h}_K \in \mathbb{R}^2$.

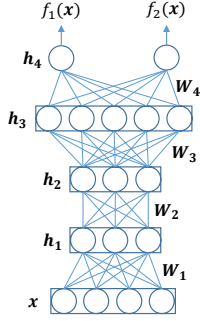


Fig. 1. Illustration of the DNN architecture.

Training a deep network necessitates a good initialization of the parameters. It is shown that layer-by-layer pretraining using unsupervised methods for initialization of the parameters results in superior performance as compared to using random initial values. We used Restricted Boltzmann Machines (RBM) for initialization. After initialization, supervised backpropagation algorithm is applied to fine-tune the parameters. The learning criteria we use is least-squares minimization. We are able to get the partial derivatives with respect to the inputs, and this derivative is also used in the source separation part. Let $f(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^2$ be the DNN, then $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are the scores that are proportional to the probabilities of source one and source two respectively for a given frame of normalized magnitude spectrum \mathbf{x} . We use these functions to measure how much the separated spectra carry the characteristics of each source as we elaborate more in the next section.

4.2. Source separation using DNN and energy minimization

In the separation stage, our algorithm works independently in each frame of the mixed audio signal. For each frame of the mixed signal spectrum, we calculate the normalized magnitude spectrum \mathbf{y} . We would like to express $\mathbf{y} = u\mathbf{x}_1 + v\mathbf{x}_2$ where u and v are the gains and \mathbf{x}_1 and \mathbf{x}_2 are normalized magnitude spectra of source one and two respectively.

We formulate the problem of finding the unknown parameters $\boldsymbol{\theta} = (\mathbf{x}_1, \mathbf{x}_2, u, v)$ as an energy minimization problem. We have a few different criteria that the source estimates need to satisfy. First, they must fit well to the DNN trained in the training stage. Second, their linear combination must sum to the mixed spectrum \mathbf{y} and third the source estimates must be nonnegative since they correspond to the magnitude spectra of each source.

The energy functions E_1 and E_2 below are least squares cost functions that quantify the fitness of a source estimate \mathbf{x} to each

corresponding source model in the DNN.

$$E_1(\mathbf{x}) = (1 - f_1(\mathbf{x}))^2 + (f_2(\mathbf{x}))^2, \quad (12)$$

$$E_2(\mathbf{x}) = (f_1(\mathbf{x}))^2 + (1 - f_2(\mathbf{x}))^2. \quad (13)$$

Basically, we expect to have $E_1(\mathbf{x}) \approx 0$ when \mathbf{x} comes from source one and vice versa. We also define the following energy function which quantifies the energy of error caused by the least squares difference between the mixed spectrum \mathbf{y} and its estimate found by linear combination of the two source estimates \mathbf{x}_1 and \mathbf{x}_2 :

$$E_{err}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, u, v) = \|u\mathbf{x}_1 + v\mathbf{x}_2 - \mathbf{y}\|^2. \quad (14)$$

Finally, we define an energy function that measures the negative energy of a variable, $R(\theta) = (\min(\theta, 0))^2$.

In order to estimate the unknowns in the model, we solve the following energy minimization problem.

$$(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \hat{u}, \hat{v}) = \underset{\{\mathbf{x}_1, \mathbf{x}_2, u, v\}}{\operatorname{argmin}} E(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, u, v), \quad (15)$$

where

$$E(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, u, v) = E_1(\mathbf{x}_1) + E_2(\mathbf{x}_2) + \lambda E_{err}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, u, v) + \beta \sum_i R(\theta_i)$$

is the joint energy function which we seek to minimize. λ and β are regularization parameters which are chosen experimentally. Here $\boldsymbol{\theta} = (\mathbf{x}_1, \mathbf{x}_2, u, v) = [\theta_1, \theta_2, \dots, \theta_n]$ is a vector containing all the unknowns which must all be nonnegative. Note that, the nonnegativity can be given as an optimization constraint as well, however we obtained faster solution of the optimization problem if we used the negative energy function penalty instead. If some of the parameters are found to be negative after the solution of the optimization problem (which rarely happens), we set them to zero. We used the LBFGS algorithm for solving the unconstrained optimization problem.

We need to calculate the gradient of the DNN outputs with respect to the input \mathbf{x} to be able to solve the optimization problem. The gradient of the input \mathbf{x} with respect to $f_i(\mathbf{x})$ is given as $\frac{\partial f_i(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{q}_{1,i}$ for $i = 1, 2$, where,

$$\mathbf{q}_{k,i} = \mathbf{W}_k^T (\mathbf{q}_{k+1,i} \otimes \mathbf{h}_k \otimes (1 - \mathbf{h}_k)), \quad (16)$$

and $\mathbf{q}_{K,i} = f_i(\mathbf{x})(1 - f_i(\mathbf{x}))\mathbf{w}_{K,i}^T$, where $\mathbf{w}_{K,i} \in \mathbb{R}^{n_K-1}$ contains the weights between i^{th} node of the output layer and the nodes at the previous layer, in other words the i^{th} row of \mathbf{W}_K .

The flowchart of the energy minimization setup is shown in Fig-ure 2. For illustration, we show the single DNN in two separate blocks in the flowchart. The fitness energies are measured using the DNN and the error energy is found from the summation requirement.

Note that, since there are many parameters to be estimated and the problem is clearly non-convex, the initialization of the parameters is very important. We initialize the estimates $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ from the NMF result after normalizing by their ℓ_2 -norms. \hat{u} is initialized by the ℓ_2 -norm of the initial NMF source estimate $\hat{\mathbf{s}}_1$ divided by the ℓ_2 -norm of the mixed signal \mathbf{y} . \hat{v} is initialized in a similar manner.

After we obtain $(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \hat{u}, \hat{v})$ as the result of the energy minimization problem, we use them as spectral estimates in a Wiener filter to reconstruct improved estimates of each source spectra, e.g. for source one we obtain the final estimate as follows:

$$\hat{\mathbf{s}}_1 = \frac{(\hat{u}\hat{\mathbf{x}}_1)^2}{(\hat{u}\hat{\mathbf{x}}_1)^2 + (\hat{v}\hat{\mathbf{x}}_2)^2} \otimes \mathbf{y}. \quad (17)$$

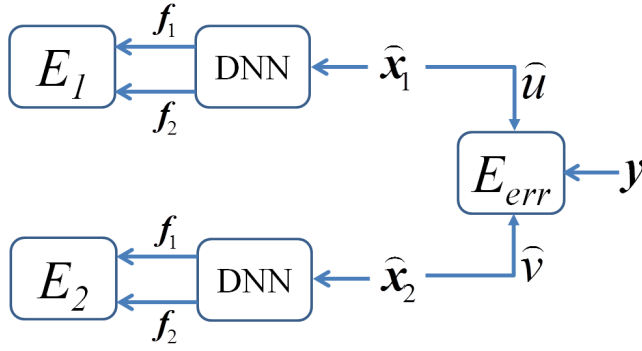


Fig. 2. Flowchart of the energy minimization setup. For illustration, we show the single DNN in two separate blocks in the flowchart.

5. EXPERIMENTS AND DISCUSSION

We applied the proposed algorithm to separate speech and music signals from their mixture. We simulated our algorithm on a collection of speech and piano data at 16kHz sampling rate. For speech data, we used the training and testing male speech data from the TIMIT database. For music data, we downloaded piano music data from piano society web site [27]. We used 39 pieces with approximate 185 minutes total duration from different composers but from a single artist for training and left out one piece for testing. The magnitude spectrograms for the speech and music data were calculated using the STFT: A Hamming window with 480 points length and 60% overlap was used and the FFT was taken at 512 points, the first 257 FFT points only were used since the conjugate of the remaining 255 points are involved in the first points.

The mixed data was formed by adding random portions of the test music file to 20 speech files from the test data of the TIMIT database at different speech to music ratio. The audio power levels of each file were found using the “speech voltmeter” program from the G.191 ITU-T STL software suite [28].

For the initialization of the source signals using nonnegative matrix factorization, we used a dictionary size of 128 for each source. For training the NMF dictionaries, we used 50 minutes of data for music and 30 minutes of the training data for speech. For training the DNN, we used a total 50 minute subset of music and speech training data for computational reasons.

For the DNN, the number of nodes in each hidden layer were 100-50-200 with three hidden layers. Sigmoid nonlinearity was used at each node including the output nodes. DNN was initialized with RBM training using contrastive divergence. We used 150 epochs for training each layer’s RBM. We used 500 epochs for backpropagation training. The first five epochs were used to optimize the output layer keeping the lower layer weights untouched.

In the energy minimization problem, the values for the regularization parameters were $\lambda = 5$ and $\beta = 3$. We used Mark Schmidt’s minFunc matlab LBFGS solver for solving the optimization problem [29].

Performance measurements of the separation algorithm were done using the signal to distortion ratio (SDR) and the signal to interference ratio (SIR) [30]. The average SDR and SIR over the 20 test utterances are reported. The source to distortion ratio (SDR) is defined as the ratio of the target energy to all errors in the re-

constructed signal. The target signal is defined as the projection of the predicted signal onto the original speech signal. Signal to interference ratio (SIR) is defined as the ratio of the target energy to the interference error due to the music signal only. The higher SDR and SIR we measure the better performance we achieve. We also use the output SNR as additional performance criteria.

The results are presented in Tables 1 and 2. We experimented with multi-frame DNN where the inputs to the DNN were taken from L neighbor spectral frames for both training and testing instead of using a single spectral frame. The approach here is similar to [15] where multiple reconstructions for each frame are averaged. We can see that using the DNN and the energy minimization idea, we can improve the source separation performance for all input speech-to-music ratio (SMR) values from -5 to 5 dB. In all cases, DNN is better than regular NMF and the improvement in SDR and SNR is usually around 1-1.5 dB. However, the improvement in SIR can be as high as 3 dB which indicates the fact that the introduced method can decrease remaining music portions in the reconstructed speech signal. We performed experiments with $L = 3$ neighboring frames which improved the results as compared to using a single frame input to the DNN. For $L = 3$, we used 500 nodes in the third layer of the DNN instead of 200. We conjecture that better results can be obtained if higher number of neighboring frames are used.

Table 1. SDR, SIR and SNR in dB for the estimated speech signal.

SMR dB	NMF			DNN					
	SDR	SIR	SNR	$L = 1$			$L = 3$		
-5	1.79	5.01	3.15	2.81	7.03	3.96	3.09	7.40	4.28
0	4.51	8.41	5.52	5.46	9.92	6.24	5.73	10.16	6.52
5	7.99	12.36	8.62	8.74	13.39	9.24	8.96	13.33	9.45

Table 2. SDR, SIR and SNR in dB for the estimated music signal.

SMR dB	NMF			DNN					
	SDR	SIR	SNR	$L = 1$			$L = 3$		
-5	5.52	15.75	6.30	6.31	18.48	7.11	6.67	18.30	7.43
0	3.51	12.65	4.88	4.23	16.03	5.60	4.45	15.90	5.88
5	0.93	9.03	3.35	1.79	12.94	3.96	1.97	13.09	4.17

6. CONCLUSION

In this work, we introduced a novel approach for single channel source separation (SCSS) using deep neural networks (DNN). The DNN was used in this paper as a helper to model each source signal. The training data for the source signals were used to train a DNN. The trained DNN was used in an energy minimization framework to separate the mixed signals while also estimating the scale for each source in the mixed signal. Many adjustments for the model parameters can be done to improve the proposed SCSS using the introduced approach. Different types of DNN such as deep autoencoders and deep recurrent neural networks which can handle the temporal structure of the source signals can be tested on the SCSS problem. We believe this idea is a novel idea and many improvements will be possible in the near future to improve its performance.

7. REFERENCES

- [1] T. Kristjansson, J. Hershey, and H. Attias, "Single microphone source separation using high resolution signal reconstruction," in *IEEE International Conference Acoustics, Speech and Signal Processing (ICASSP)*, 2004.
- [2] A. M. Reddy and B. Raj, "Soft Mask Methods for single-channel speaker separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, Aug. 2007.
- [3] A. M. Reddy and B. Raj, "A Minimum Mean squared error estimator for single channel speaker separation," in *International Conference on Spoken Language Processing (InterSpeech)*, 2004.
- [4] T. Virtanen, "Speech recognition using factorial hidden markov models for separation in the feature space," in *International Conference on Spoken Language Processing (InterSpeech)*, 2006.
- [5] S. T. Roweis, "One Microphone source separation," *Neural Information Processing Systems*, 13, pp. 793–799, 2000.
- [6] A. N. Deoras and A. H. Johnson, "A factorial hmm approach to simultaneous recognition of isolated digits spoken by multiple talkers on one audio channel," in *IEEE International Conference Acoustics, Speech and Signal Processing (ICASSP)*, 2004.
- [7] M. H. Radfar, W. Wong, W. Y. Chan, and R. M. Dansereau, "Gain estimation in model-based single channel speech separation," in *In proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP, Grenoble, France)*, Sept. 2009.
- [8] M. H. Radfar, R. M. Dansereau, and W. Y. Chan, "Monaural speech separation based on gain adapted minimum mean square error estimation," *Journal of Signal Processing Systems, Springer*, vol. 61, no. 1, pp. 21–37, 2008.
- [9] M. H. Radfar and R. M. Dansereau, "Long-term gain estimation in model-based single channel speech separation," in *In Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA, New Paltz, NY)*, 2007.
- [10] A. Ozerov, C. Fvotte, and M. Charbit, "Factorial scaled hidden markov model for polyphonic audio representation and source separation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA, Mohonk, NY)*, 2009.
- [11] M. H. Radfar, W. Wong, R. M. Dansereau, and W. Y. Chan, "Scaled factorial Hidden Markov Models: a new technique for compensating gain differences in model-based single channel speech separation," in *IEEE International Conference Acoustics, Speech and Signal Processing (ICASSP)*, 2010.
- [12] J. R. Hershey, S. J. Rennie, P. A. Olsen, and T. T. Kristjansson, "Super-human multi-talker speech recognition: A graphical modeling approach," *Computer Speech and Language*, vol. 24, no. 1, pp. 45–66, Jan. 2010.
- [13] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *International Conference on Spoken Language Processing (InterSpeech)*, 2006.
- [14] E. M. Grais and H. Erdogan, "Spectro-temporal post-smoothing in NMF based single-channel source separation," in *European Signal Processing Conference (EUSIPCO)*, 2012.
- [15] E. M. Grais and H. Erdogan, "Single channel speech music separation using nonnegative matrix factorization with sliding window and spectral masks," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, 2011.
- [16] E. M. Grais and H. Erdogan, "Regularized nonnegative matrix factorization using gaussian mixture priors for supervised single channel source separation," *Computer Speech and Language*, vol. 27, no. 3, pp. 746–762, May 2013.
- [17] E. M. Grais and H. Erdogan, "Gaussian mixture gain priors for regularized nonnegative matrix factorization in single-channel source separation," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, 2012.
- [18] E. M. Grais and H. Erdogan, "Hidden Markov Models as priors for regularized nonnegative matrix factorization in single-channel source separation," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, 2012.
- [19] E. M. Grais and H. Erdogan, "Spectro-temporal post-enhancement using MMSE estimation in NMF based single-channel source separation," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, 2013.
- [20] Y. Wang and D. L. Wang, "Cocktail party processing via structured prediction," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.
- [21] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, July 2006.
- [22] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *Signal Processing Magazine*, 2012.
- [23] E. M. Grais and H. Erdogan, "Single channel speech music separation using nonnegative matrix factorization and spectral masks," in *International Conference on Digital Signal Processing*, 2011.
- [24] T. Virtanen, "Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1066–1074, Mar. 2007.
- [25] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562, 2001.
- [26] 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, 2009.
- [27] URL, "<http://pianosociety.com>," 2009.
- [28] URL, "<http://www.itu.int/rec/T-REC-G.191/en>," 2009.
- [29] URL, "<http://www.di.ens.fr/~mschmidt/software/minfunc.html>," 2012.
- [30] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–69, July 2006.