View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Sabanci University Research Database

Single channel speech music separation using nonnegative matrix factorization with sliding windows and spectral masks

Emad M. Grais and Hakan Erdogan

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

Abstract

A single channel speech-music separation algorithm based on nonnegative matrix factorization (NMF) with sliding windows and spectral masks is proposed in this work. We train a set of basis vectors for each source signal using NMF in the magnitude spectral domain. Rather than forming the columns of the matrices to be decomposed by NMF of a single spectral frame, we build them with multiple spectral frames stacked in one column. After observing the mixed signal, NMF is used to decompose its magnitude spectra into a weighted linear combination of the trained basis vectors for both sources. An initial spectrogram estimate for each source is found, and a spectral mask is built using these initial estimates. This mask is used to weight the mixed signal spectrogram to find the contributions of each source signal in the mixed signal. The method is shown to perform better than the conventional NMF approach.

Index Terms: Single channel source separation, source separation, semi-blind source separation, speech music separation, speech processing, nonnegative matrix factorization, and Wiener filter.

1. Introduction

The problem of separating source signals from a mixture of multiple sources is encountered in many applications such as communication, medical, and multimedia. In many applications, the need to find an accurate estimate of the source signals is very urgent. In acoustic applications, the performance of the automatic speech recognition system (ASR) is very sensitive to the background component in the speech signal, and it may be desirable to separate the speech signal accurately from the background signal before applying ASR. The most complicated case of source separation is when only a single measurement of the mixed signal is available. Therefore, training data for each source signal in the mixed signal should be available separately. NMF has been an interesting algorithm for single channel source separation. It is usually used in the magnitude spectral domain to decompose the spectrogram of the mixed signal. In [1, 2, 3, 4], NMF was used with training data to train a set of basis vectors for each source, then these basis vectors were used with NMF to separate the mixed signal. The separation was done frame by frame without considering the smoothness transition and any other information between the consequent frames. In [5, 6], the continuity between the consequent frames was considered but the improvements in the results were small.

In this paper, NMF, sliding windows and spectral masks are used in magnitude spectral domain to accurately separate the speech signal from the background music signal. There are two stages in our algorithm. In the training phase, we use NMF with training data for each source to train a set of basis vectors for each source in the magnitude spectral domain. In the testing phase, after observing the mixed signal, NMF is used to decompose the magnitude spectra of the mixed signal into a weighted linear combination of the trained basis vectors of both sources. The weighted sum of the decomposition terms that include basis vectors for each source is used as an initial estimate of the magnitude spectra of each source. Then an initial spectrogram estimate for each source is obtained, and used to build a spectral mask which explains the contribution of every source in the mixed signal. Rather than using NMF to directly decompose the spectrogram of the signals as in the literature, we form the matrices to be decomposed as follows: We stack the spectrogram frames in one vector. We pass a window with length equal to multiple spectral frames size to select the first column of the matrix, then we shift or slide the window by one frame to choose the next column as shown in Figure 1. Therefore,

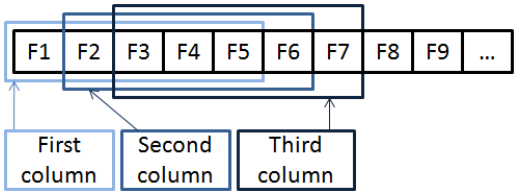

Figure 1: Columns construction and sliding windows with length five frames.

NMF is used in this work to decompose matrices with columns that contain multiple spectral frames in both training and separation stages. Thus, rather than decomposing every spectral frame in the spectrogram independently from each other, we decompose multiple frames at once in one column. Sliding the window by one frame each time to get the next column makes every frame decomposed multiple times with different neighbor frames. We take the average of the different decomposion results for each frame to find an accurate decomposion of the spectrograms. The novelty of this work is in using NMF with sliding windows and different types of spectral masks. The experiments' results show that using NMF, spectral masks, and sliding windows with multiple spectral frames improve the separation results compared to using NMF only.

The remainder of this paper is organized as follows: In section 2, a mathematical description of the single channel speech-music separation problem is given. In section 3, a brief explanation about NMF and how we use it to train the basis vectors for each source is given. In section 4, the separation process is pre-

sented. In the remaining sections, we present our observations and the results of our experiments.

2. Problem formulation

Single channel speech-music separation problem can be defined as follows: Assume we have a single observation signal $x(t)$, which is the mixture of two sources, speech $s(t)$ and music $m(t)$. The source separation problem aims to find estimates for $s(t)$ and $m(t)$ from $x(t)$. The framework of the algorithms that are presented here is in the short time Fourier transform (STFT) domain. Let $X(t,f)$ be the STFT of $x(t)$, where t represents the frame index and f is the frequency-index. Due to linearity of the STFT, we have:

$$X(t,f) = S(t,f) + M(t,f). \tag{1}$$

$$|X(t,f)|\, e^{j\phi_X(t,f)} = |S(t,f)|\, e^{j\phi_S(t,f)} + |M(t,f)|\, e^{j\phi_M(t,f)}. \tag{2}$$

In this work, we assume the sources have the same phase angle as the mixed signal for every frame, that is $\phi_S(t,f) = \phi_M(t,f) = \phi_X(t,f)$. This assumption was shown to yield good results in earlier work. So, we can write the magnitude spectrogram of the measured signal as the sum of source signals' magnitude spectrograms.

$$\boldsymbol{X} = \boldsymbol{S} + \boldsymbol{M}.^{1} \tag{3}$$

Here $\boldsymbol{S}$ and $\boldsymbol{M}$ are unknown magnitude spectrograms, and need to be estimated using observed data and training speech and music spectra. The magnitude spectrogram for the observed signal $x(t)$ is obtained by taking the magnitude of the DFT of the windowed signal for each column of the spectrogram.

3. Non-negative matrix factorization

Non-negative matrix factorization is a well known algorithm for matrix factorization with non-negativity constraints. It is used to decompose any nonnegative matrix $\boldsymbol{V}$ into a nonnegative basis vectors matrix $\boldsymbol{B}$ and a nonnegative weights matrix $\boldsymbol{W}$.

$$\boldsymbol{V} \approx \boldsymbol{BW}. \tag{4}$$

The columns in the matrix $\boldsymbol{V}$ are approximated by a weighted linear combination of the basis vectors in the columns of $\boldsymbol{B}$. The weights that every basis vector contributes in the columns of $\boldsymbol{V}$ appear in the corresponding columns of the matrix $\boldsymbol{W}$. The nonnegative basis vectors in matrix $\boldsymbol{B}$ are optimized to allow the data in $\boldsymbol{V}$ to be approximated as a nonnegative linear combination of its constituent vectors. The matrices $\boldsymbol{B}$ and $\boldsymbol{W}$ can be found by solving the following optimization problem:

$$\min_{\boldsymbol{B},\boldsymbol{W}} C\left(\boldsymbol{V}, \boldsymbol{BW}\right), \tag{5}$$

subject to elements of $\boldsymbol{B}, \boldsymbol{W} \geq 0$. Different cost functions C lead to different kinds of NMF, and the preference among them depends on the application. In [7], two different cost functions were represented. The first cost function is the Euclidean distance between $\boldsymbol{V}$ and $\boldsymbol{BW}$ given as follows:

$$\min_{\boldsymbol{B},\boldsymbol{W}} \left(\|\boldsymbol{V} - \boldsymbol{BW}\|_2^2\right), \tag{6}$$

[1]The notations here are as follows: bold capital letters are for matrices, bold small letters are for vectors others are for scalars.

where $\|\boldsymbol{V} - \boldsymbol{BW}\|_2^2 = \sum_{i,j} \left(\boldsymbol{V}_{i,j} - (\boldsymbol{BW})_{i,j}\right)^2$. The second cost function is the divergence of $\boldsymbol{V}$ from $\boldsymbol{BW}$ which yields the following optimization problem:

$$\min_{\boldsymbol{B},\boldsymbol{W}} D\left(\boldsymbol{V} \,||\, \boldsymbol{BW}\right), \tag{7}$$

where

$$D\left(\boldsymbol{V} \,||\, \boldsymbol{BW}\right) = \sum_{i,j} \left(\boldsymbol{V}_{i,j} \log \frac{\boldsymbol{V}_{i,j}}{(\boldsymbol{BW})_{i,j}} - \boldsymbol{V}_{i,j} + (\boldsymbol{BW})_{i,j}\right).$$

The second cost function is preferred to be used in audio source separation applications [2], thus we only consider it in this paper. The NMF solution for equation (7) can be computed by alternating updates of $\boldsymbol{B}$ and $\boldsymbol{W}$ as follows:

$$\boldsymbol{B} \leftarrow \boldsymbol{B} \otimes \frac{\frac{\boldsymbol{V}}{\boldsymbol{BW}}\boldsymbol{W}^T}{\mathbf{1}\boldsymbol{W}^T}, \tag{8}$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \otimes \frac{\boldsymbol{B}^T\frac{\boldsymbol{V}}{\boldsymbol{BW}}}{\boldsymbol{B}^T\mathbf{1}}, \tag{9}$$

where $\mathbf{1}$ is a matrix of ones with the same size of $\boldsymbol{V}$, the operations $\otimes$ and all divisions are element wise multiplication and division respectively.

3.1. Training the bases

Assume two sets of training data for speech and music signals are available. The STFT is computed and the magnitude spectrogram of speech and music are calculated. The NMF is used to model the training data as a set of basis vectors to represent the spectral characteristics for each source signal. Instead of using NMF directly to decompose the spectrograms, we build the matrices $\boldsymbol{S}_{\text{train}}$ and $\boldsymbol{M}_{\text{train}}$ with columns containing $2L+1$ frames of the speech and music spectrograms respectively as shown in Figure 1. Which means that, every column in these matrices contains $2L+1$ consequent frames from the spectrogram stacked in one column. For example, the column number l in the training speech matrix $\boldsymbol{S}_{\text{train}}$ is

$$\boldsymbol{s}(l) = \left[\boldsymbol{f}_s^T(l-L), ..., \boldsymbol{f}_s^T(l), ..., \boldsymbol{f}_s^T(l+L)\right]^T.$$

Where $\boldsymbol{f}_s(l)$ is the frame number l of the training speech signal spectrogram. A mirror imaging at the edges of the spectrograms is performed. After forming the two matrices $\boldsymbol{S}_{\text{train}}$ and $\boldsymbol{M}_{\text{train}}$ the NMF is used to decompose them into bases and weights matrices as follows:

$$\boldsymbol{S}_{\text{train}} \approx \boldsymbol{B}_{\text{speech}}\boldsymbol{W}_{\text{speech}}. \tag{10}$$

$$\boldsymbol{M}_{\text{train}} \approx \boldsymbol{B}_{\text{music}}\boldsymbol{W}_{\text{music}}. \tag{11}$$

We use the update rules in equations (8) and (9) to solve equations (10) and (11). The matrices $\boldsymbol{S}_{\text{train}}$ and $\boldsymbol{M}_{\text{train}}$ have normalized columns, and after each iteration, we normalize the columns of $\boldsymbol{B}_{\text{speech}}$ and $\boldsymbol{B}_{\text{music}}$. All the matrices $\boldsymbol{B}$ and $\boldsymbol{W}$ are initialized by positive random noise. The best number of basis vectors depends on the application, the signal type, and dimension. Since every column in the training matrices has $2L+1$ times the dimension of the spectrogram frames, more basis vectors than the single frame case will be used to be compatible with the dimension of the columns in $\boldsymbol{S}_{\text{train}}$ and $\boldsymbol{M}_{\text{train}}$.

4. Signal separation and masking

After observing the mixed signal $x(t)$, the magnitude spectrogram $\boldsymbol{X}$ of the mixed signal is computed using STFT. To find the contribution of every source in the mixture, we use NMF to decompose the spectrogram of the mixed signal into weighted linear combinations of the trained bases for both sources. Instead of using NMF directly to decompose the spectrogram of the mixed signal, we build a matrix $\boldsymbol{Y}$ with columns that contain $2L+1$ frames of the mixed signal spectrogram as shown in Figure 1. For example, the column number l in the mixed signal matrix $\boldsymbol{Y}$ is

$$\boldsymbol{y}(l) = \left[\boldsymbol{f}_x^T(l-L), ..., \boldsymbol{f}_x^T(l), ..., \boldsymbol{f}_x^T(l+L)\right]^T.$$

Where $\boldsymbol{f}_x(l)$ is the frame number l of the mixed signal spectrogram $\boldsymbol{X}$. A mirror imaging at the edges of the spectrogram is performed. The goal now is to decompose the matrix $\boldsymbol{Y}$ as a linear combination of the trained basis vectors in the columns of $\boldsymbol{B}_{\text{speech}}$ and $\boldsymbol{B}_{\text{music}}$ that were found from solving equations (10) and (11). Then the initial estimates of the underlying source signals in the mixed signal are found as described in section 4.1. We use the decomposition results to build different spectral masks. The mask weights every entry of the mixed signal spectrogram according to the amount of contributions of every source in the mixed signal. The final estimate for every entry for each source spectrogram is a scaled version of its corresponding entry of mixed signal spectrogram. This scale is defined by the spectral mask as we elaborate in section 4.2.

4.1. Decomposition of the mixed signal

The NMF is used again here to decompose the matrix $\boldsymbol{Y}$ but with a fixed concatenated bases matrix as follows:

$$\boldsymbol{Y} \approx \left[\boldsymbol{B}_{\text{speech}} \;\; \boldsymbol{B}_{\text{music}}\right] \boldsymbol{W}, \tag{12}$$

where $\boldsymbol{B}_{\text{speech}}$ and $\boldsymbol{B}_{\text{music}}$ are obtained from solving equations (10) and (11). Here only the update rule in equation (9) is used to solve equation (12), and the bases matrix is fixed. $\boldsymbol{W}$ is initialized by positive random noise. The matrix $\overline{\boldsymbol{S}}$ that contains rough estimates of the magnitude spectral frames of the speech signal in the mixture is found by multiplying the bases matrix $\boldsymbol{B}_{\text{speech}}$ with its corresponding weights in matrix $\boldsymbol{W}$ in equation (12). Also the matrix $\overline{\boldsymbol{M}}$ that contains rough estimates of the magnitude spectral frames of the music signal in the mixture is found by multiplying the bases matrix $\boldsymbol{B}_{\text{music}}$ with its corresponding weights in matrix $\boldsymbol{W}$ in equation (12). These matrices are calculated as follows:

$$\overline{\boldsymbol{S}} = \boldsymbol{B}_{\text{speech}} \boldsymbol{W}_S. \tag{13}$$

$$\overline{\boldsymbol{M}} = \boldsymbol{B}_{\text{music}} \boldsymbol{W}_M. \tag{14}$$

Where $\boldsymbol{W}_S$ and $\boldsymbol{W}_M$ are submatrices in matrix $\boldsymbol{W}$ that correspond to the speech and music components respectively in equation (12). In the matrix $\overline{\boldsymbol{S}}$ the estimated spectrogram frames of the estimated speech signal are estimated differently $2L+1$ times with different $2L+1$ neighbor frames. To find a smooth estimate of every spectral frame, we take the average of its corresponding $2L+1$ frames in the matrix $\overline{\boldsymbol{S}}$. After taking the average, we build the matrix $\tilde{\boldsymbol{S}}$ which is the initial estimate spectrogram of the estimated speech signal. We build the initial estimated spectrogram $\tilde{\boldsymbol{M}}$ of the music signal in a similar fashion.

4.2. Source signals reconstruction and masks.

We can directly use the initial estimate spectrograms $\tilde{\boldsymbol{S}}$ and $\tilde{\boldsymbol{M}}$ of the speech and music signals that are found in section 4.1 as the final estimate of every source, but the two estimated spectra $\tilde{\boldsymbol{S}}$ and $\tilde{\boldsymbol{M}}$ may not sum up to the mixed spectrogram $\boldsymbol{X}$. We usually get nonzero decomposion error. Thus, NMF gives us an approximation:

$$\boldsymbol{X} \approx \tilde{\boldsymbol{S}} + \tilde{\boldsymbol{M}}.$$

Assuming noise is negligible in our mixed signal, the component signals' sum should be directly equal to the mixed spectrogram. To make the error zero, we use the initial estimated spectrograms $\tilde{\boldsymbol{S}}$ and $\tilde{\boldsymbol{M}}$ to build a mask as follows:

$$\boldsymbol{H} = \frac{\tilde{\boldsymbol{S}}^p}{\tilde{\boldsymbol{S}}^p + \tilde{\boldsymbol{M}}^p}, \tag{15}$$

where $p > 0$ is a parameter, $(.)^p$, and the division are element wise operations. Notice that elements of $\boldsymbol{H} \in (0, 1)$ and using different p values lead to different kinds of masks. When $p = 2$ the mask $\boldsymbol{H}$ is a Wiener filter. The value of p controls the saturation level of the ratio in (15). When $p > 1$, the larger source component will dominate more in the mixture. At $p = \infty$, we achieve a binary mask (hard mask) which will choose the larger source component as the only component. These masks will scale every frequency component in the observed mixed spectrogram $\boldsymbol{X}$ with a ratio that explains how much each source contributes in the mixed signal such that:

$$\hat{\boldsymbol{S}} = \boldsymbol{H} \otimes \boldsymbol{X}, \tag{16}$$

$$\hat{\boldsymbol{M}} = (\boldsymbol{1} - \boldsymbol{H}) \otimes \boldsymbol{X}, \tag{17}$$

where $\hat{\boldsymbol{S}}$ and $\hat{\boldsymbol{M}}$ are the final estimates of the speech and music spectrograms, $\boldsymbol{1}$ is a matrix of ones, and $\otimes$ is element-wise multiplication. By using this idea we will make the approximation error zero, and we can make sure that the two estimated signals will add up to the mixed signal. After finding the contribution of the speech signal in the mixed signal, the estimated speech signal $\hat{s}(t)$ can be found by using inverse STFT to the estimated speech spectrogram $\hat{\boldsymbol{S}}$ with the phase angle of the mixed signal.

5. Experiments and Discussion

We simulated the proposed algorithms on a collection of speech and piano music data at 16kHz sampling rate. For training speech data, we used 540 short utterances from a single speaker. We used 20 utterances for testing. For music data, we downloaded piano music from piano society web site [8]. We used 38 pieces from different composers but from a single artist for training and left out one piece for the testing stage. The spectrograms for the training speech and music data were calculated by using the STFT, a Hamming window was used, and the FFT was taken at 512 points, the first 257 FFT points only were used since the remaining points are the conjugate of the first 257 points. Then we concatenated every five ($L = 2$) spectrogram frames in one column vector with size (5*257) as we have mentioned in section 3.1. Each vector in $\boldsymbol{S}_{\text{train}}$ and $\boldsymbol{M}_{\text{train}}$ is in 1285 dimensions (5*257). We trained different number of bases N_s for training speech signal and N_m for training music signal. N_s and N_m take values 1285, 642, 321, and 160 bases. The test data was formed by adding random portions of the test music file to the 20 speech utterance files at different speech to music ratio (SMR) values in dB. The audio power levels of each file were found using the "audio voltmeter" program from the

Table 1: Source/Distortion Ratio (SDR) in dB for the speech signal using NMF with sliding window and spectral mask with $p = 3$ for different numbers of bases.

SMR dB	$N_s = 1285$ $N_m = 1285$	$N_s = 1285$ $N_m = 642$	$N_s = 1285$ $N_m = 321$	$N_s = 642$ $N_m = 642$	$N_s = 642$ $N_m = 321$	$N_s = 642$ $N_m = 160$	$N_s = 321$ $N_m = 642$	$N_s = 321$ $N_m = 160$
-5	7.18	6.61	4.85	**7.40**	6.21	4.44	7.06	5.88
0	10.60	10.54	8.83	**11.12**	10.07	8.45	10.31	9.71
5	12.68	13.14	11.82	**13.34**	12.86	11.61	11.97	12.56
10	15.61	**17.03**	16.20	16.66	16.88	15.99	14.49	16.46
15	17.63	**19.64**	19.21	18.60	19.50	19.10	15.78	19.14
20	19.00	22.43	23.14	20.36	22.57	**23.43**	16.80	22.09

G.191 ITU-T STL software suite [9]. We obtained the spectrogram from the test signal with the same setup like the training signals. For each SMR value, we obtained 20 test utterances this way, then we averaged the 20 test utterances' results.

Performance measurement of the separation algorithms was done using the source distortion ratio metric that is introduced in [10]. Source distortion ratio (SDR) is defined as the ratio of the target energy to all errors in the reconstruction. The target signal is defined as the projection of the predicted signal onto the original speech signal.

We worked with training and testing matrices with columns that contain five spectral frames, because we got remarkable improvement in the separation results compared to work with columns which contain a single spectral frame [3].

Table 1 shows the separation performance of using NMF with a different number of bases N_s and N_m. We got these results by using the spectral mask with $p = 3$ in equation (15), sliding window with $L = 2$, and the maximum number of iterations in NMF is 1000. The NMF iterations were stopped when the rate of change in the cost function value to the initial cost function value is less than 10^{-3}. Table 2 shows the performance of using NMF and sliding window without masks and with different kinds of masks, which shows that, we got better results when $p = 3$ and $p = 4$ in equation (15).

To show the importance of using sliding windows with multiple frames, we repeated our experiments by using NMF with mask without using sliding windows [3]. NMF was used in this experiment to decompose matrices with columns containing a single spectral frame with length 257. Which means we used NMF to directly decompose the spectrograms of the signals. We used fewer numbers of bases since the dimension in this case was just 257. Table 3, shows the results of this experiment.

By comparing the results of using NMF only without using neither spectral mask nor sliding window as in the literature, which is shown in the first column in table 3 with the results of using NMF with $p = 3$ mask and sliding windows as in tables 1 and 2, we can see that our proposed algorithm gives remarkable improvements in the range of **2−6dB** in the performance of the separation. Audio demonstrations of our experiments are available at
http://students.sabanciuniv.edu/grais/speech/scsmsnmfsmsw/

6. CONCLUSION

In this work, we introduced single channel speech-music separation using nonnegative matrix factorization (NMF) with sliding windows and spectral masks. We used NMF to decompose matrices with columns contain multiple magnitude spectral frames. We built a spectral mask from the decomposition results to find the contribution of each source signal in the mixed signal. The proposed algorithm gave better results and more accurate speech music separation.

Table 2: Source/Distortion Ratio (SDR) in dB for the speech signal in case of using NMF with sliding window and different masks, with $N_s = N_m = 642$.

SMR dB	No mask	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$	Hard mask
-5	5.58	5.59	7.27	**7.40**	7.33	7.25	6.56
0	9.53	9.55	10.97	**11.12**	11.07	10.99	10.36
5	11.75	11.78	13.10	**13.34**	**13.34**	13.28	12.64
10	14.98	15.04	16.33	16.66	**16.69**	16.65	16.07
15	16.68	16.76	18.19	18.60	**18.66**	18.63	18.08
20	18.00	18.10	19.80	20.36	**20.47**	20.45	19.91

Table 3: Source/Distortion Ratio (SDR) in dB for the speech signal in case of using NMF with different masks, **without sliding window**, with $N_s = N_m = 128$.

SMR dB	No mask	$p = 1$	$p = 2$	$p = 3$	$p = 4$	Hard mask
-5	4.1	4.11	5.34	**5.41**	5.35	4.69
0	8.79	8.81	9.68	**9.72**	9.66	9.05
5	10.29	10.31	11.15	**11.22**	11.17	10.59
10	14.45	14.5	15.33	**15.52**	**15.52**	14.93
15	16.33	16.4	17.21	17.45	**17.48**	16.84
20	17.1	17.19	18.15	18.49	**18.56**	18.08

7. References

[1] Mikkel N. Schmidt and Rasmus K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *INTERSPEECH*, 2006.

[2] K. W. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors," in *in Proc. of ICASSP*, 2008.

[3] Emad M. Grais and Hakan Erdogan, "Single channel speech music separation using nonnegative matrix factorization and spectral masks," in *17th International Conference on Digital Signal Processing (DSP)*, 2011.

[4] B. Raj, T. Virtanen, S. Chaudhure, and R. Singh, "Non-negative matrix factorization based compensation of music for automatic speech recognition," in *Interspeech*, 2010.

[5] T. Virtanen, "Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria," *IEEE Trans. Audio, Speech, Lang. Process*, vol. 15, pp. 1066–1074, Mar. 2007.

[6] Hakan Erdogan and Emad M. Grais, "Semi-blind speech-music separation using sparsity and continuity priors," in *ICPR*, 2010.

[7] 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.

[8] URL, "http://pianosociety.com," 2009.

[9] URL, "http://www.itu.int/rec/T-REC-G.191/en," 2009.

[10] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Tr. Acoust. Sp. Sig. Proc.*, vol. 14, no. 4, pp. 1462–69, July 2006.