

Gaussian Mixture Gain Priors for Regularized Nonnegative Matrix Factorization in Single-Channel Source Separation

Emad M. Grais and Hakan Erdogan

Faculty of Engineering and Natural Sciences,
Sabanci University, Orhanli Tuzla, 34956, Istanbul.

{grais,haerdogan}@sabanciuniv.edu

Abstract

We propose a new method to incorporate statistical priors on the solution of the nonnegative matrix factorization (NMF) for single-channel source separation (SCSS) applications. The Gaussian mixture model (GMM) is used as a log-normalized gain prior model for the NMF solution. The normalization makes the prior models energy independent. In NMF based SCSS, NMF is used to decompose the spectra of the observed mixed signal as a weighted linear combination of a set of trained basis vectors. In this work, the NMF decomposition weights are enforced to consider statistical prior information on the weight combination patterns that the trained basis vectors can jointly receive for each source in the observed mixed signal. The NMF solutions for the weights are encouraged to increase the log-likelihood with the trained gain prior GMMs while reducing the NMF reconstruction error at the same time.

Index Terms: Nonnegative matrix factorization, single-channel source separation, and Gaussian mixture models.

1. Introduction

Nonnegative matrix factorization [1], is extensively used in source separation applications, especially when only one observation of the mixed signal is available [2]. In NMF based single-channel source separation, NMF uses the training data for each source to train a set of basis vectors. After observing the mixed signal, NMF is used to decompose the mixed signal as a weighted linear combination of the trained basis vectors. The estimate for each source is found by summing the decomposition terms that include its corresponding trained basis vectors. To improve the performance of NMF, there have been many works that aim to enforce the NMF decomposition weights to satisfy certain characteristics of the source signals. In [3], temporal continuity and sparsity priors were enforced in the decomposition weights. In [2, 4] temporal smoothness was enforced on the NMF decomposition weights.

In this work, we propose a method that makes better use of the available training data to improve the separation process. In the training stage, NMF is used to decompose the power spectral density of the training data of each source into a basis matrix and a gains matrix. The gains matrix which was usually ignored in previous works is used here to build a GMM prior model for each source. The columns of the gains matrices are normalized and their logarithm is taken and used to train the prior GMMs. After observing the mixed signal, NMF is used to decompose

the power spectra of the mixed signal with the trained basis matrices. The decomposition solutions of the NMF are encouraged to increase the log-likelihood with the trained GMM model for each source. In [5] a single Gaussian model for the training gain matrix was used. In that work, the training and testing signals for all sources must have the same energy level. Our proposed algorithm uses GMM, which better represents the data than a single Gaussian model. We also do not put any restriction on the energy level of the testing data compared to the training data. Moreover, the source signals can have different energy levels in the mixed signal without any restriction. Furthermore, our update rules for the regularized NMF handle the nonnegativity constrains better than the ones that were used in [5]. In addition, we show the importance of the regularization parameters which control the trade-off between the NMF cost function and the prior likelihood.

The remainder of this paper is organized as follows: In section 2, a mathematical description of the SCSS problem is given. In section 3, we give a brief explanation about NMF. In section 4, we show the training processes of the NMF basis models and the GMM prior gain models for the source signals. In section 5, the separation process is presented. In the remaining sections, we present our observations and the results of our experiments.

2. Problem formulation

In SCSS problems, the aim is to find estimates of source signals that are mixed on a single observation channel $y(t)$. 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:

$$Y(t, f) = \sum_{i=1}^Z S^{(i)}(t, f), \quad (1)$$

where $S^{(i)}(t, f)$ is the unknown STFT of source i in the mixed signal, and Z is the number of sources in the mixed signal. Assuming independence of the sources, we can write the power spectral density (PSD) of the measured signal as the sum of source signal PSDs as follows:

$$\sigma_y^2(t, f) = \sum_{i=1}^Z \sigma_i^2(t, f), \quad (2)$$

This research is partially supported by Turk-Telekom group research and development, project entitled "Single-channel source separation", project year 2012.

where $\sigma_y^2(t, f) = E(|Y(t, f)|^2)$. We can write the PSDs in matrix form as follows:

$$\mathbf{Y} = \sum_{i=1}^Z \mathbf{S}^{(i)}, \quad (3)$$

where $\mathbf{S} = \{\mathbf{S}^{(1)}, \dots, \mathbf{S}^{(i)}, \dots, \mathbf{S}^{(Z)}\}$ are the unknown PSDs of the source signals, and they need to be estimated using the observed mixed signal and training data for each source. The PSD for the measured signal $y(t)$ is calculated by taking the squared magnitude of the DFT of the windowed signal.

The main idea to solve for \mathbf{S} is to decompose each PSD frame of the mixed signal \mathbf{y} as a nonnegative weighted linear combination of the trained set of nonnegative basis vectors \mathbf{b} for all sources as follows:

$$\mathbf{y} \approx \underbrace{\sum_{q=1}^{Q^{(1)}} g_q^{(1)} \mathbf{b}_q^{(1)}}_{\tilde{\mathbf{s}}^{(1)}} + \dots + \underbrace{\sum_{q=1}^{Q^{(i)}} g_q^{(i)} \mathbf{b}_q^{(i)}}_{\tilde{\mathbf{s}}^{(i)}} + \dots + \underbrace{\sum_{q=1}^{Q^{(Z)}} g_q^{(Z)} \mathbf{b}_q^{(Z)}}_{\tilde{\mathbf{s}}^{(Z)}}. \quad (4)$$

Where $\tilde{\mathbf{s}}^{(i)}$ is the estimated PSD frame of source i that is corresponding to the PSD frame of the mixed signal \mathbf{y} , $\mathbf{b}_q^{(i)}$ is the trained basis vector number q for source i , $g_q^{(i)}$ is the gain that basis $\mathbf{b}_q^{(i)}$ gets in the mixed signal, and $Q^{(i)}$ is the number of trained basis vectors for source i . In this work, the set of gain values $g^{(i)}$ for the set of basis vectors $\mathbf{b}^{(i)}$ for each source i are jointly encouraged to increase the log-likelihood with its corresponding trained gain prior GMM.

3. Nonnegative matrix factorization

Nonnegative matrix factorization is a matrix factorization algorithm based on nonnegativity constraints. The nonnegative matrix \mathbf{V} can be decomposed into a nonnegative basis matrix \mathbf{B} and a nonnegative gains matrix \mathbf{G} as follows:

$$\mathbf{V} \approx \mathbf{B}\mathbf{G}. \quad (5)$$

The matrix \mathbf{B} contains nonnegative basis vectors that are optimized to allow the data in \mathbf{V} to be approximated as a nonnegative 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 [4]:

$$\min_{\mathbf{B}, \mathbf{G}} D_{IS}(\mathbf{V} \parallel \mathbf{B}\mathbf{G}), \quad (6)$$

where

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

This divergence cost function is a good measurement for the perceptual difference between different signals [4, 6]. The IS-NMF solution for equation (6) can be computed by alternating multiplicative updates of \mathbf{B} and \mathbf{G} as follows [6]:

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

$$\mathbf{G} \leftarrow \mathbf{G} \otimes \frac{\mathbf{B}^T \left(\frac{\mathbf{V}}{(\mathbf{B}\mathbf{G})^2} \right)}{\mathbf{B}^T \left(\frac{\mathbf{1}}{\mathbf{B}\mathbf{G}} \right)}, \quad (8)$$

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.

4. Training the source models

Given a set of training data for each source signal, the power spectrogram $\mathbf{S}_{\text{train}}^{(i)}$ for each source i is calculated. NMF is used to decompose the power spectrogram into bases and gains matrices as follows:

$$\mathbf{S}_{\text{train}}^{(i)} \approx \mathbf{B}^{(i)} \mathbf{G}_{\text{train}}^{(i)}, \quad (9)$$

the multiplicative update rules in equations (7) and (8) are used to solve equation (9). Within each iteration, we normalize the columns of $\mathbf{B}^{(i)}$. All the matrices \mathbf{B} and $\mathbf{G}_{\text{train}}$ are initialized by positive random noise. After finding matrices \mathbf{B} and $\mathbf{G}_{\text{train}}$ for all sources, all matrices \mathbf{B} are used in the mixed signal decomposition as shown in section 5. We use the matrices $\mathbf{G}_{\text{train}}$ to train prior models for the gain patterns that each source signal can possibly receive in the gains matrix. For each matrix $\mathbf{G}_{\text{train}}^{(i)}$, we take the logarithm of its normalized columns, and use them to build its gain prior GMM. The reason for using logarithm is because GMM is usually a better fit to the logarithm of the values between 0 and 1. Logarithm of values between 0 and 1 have wider support, so GMM fits better. The reason for normalization is to make the models insensitive to the energy level of the signals, which leads to an energy independent prior model. The multivariate Gaussian mixture model is defined as:

$$p(\mathbf{x}) = \sum_{k=1}^K \frac{w_k}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\}, \quad (10)$$

where K is the number of Gaussian mixture components, w_k is the mixture weight, d is the vector dimension, $\boldsymbol{\mu}_k$ is the mean vector and Σ_k is the diagonal covariance matrix of the k^{th} Gaussian model.

5. Signal separation

After observing the mixed signal $y(t)$, the power spectral density \mathbf{Y} of the mixed signal is computed using STFT. To find the contribution of every source in the mixed signal PSD, we use NMF to decompose the power spectrogram \mathbf{Y} with the trained basis matrices \mathbf{B} from equation (9) as follows:

$$\mathbf{Y} \approx \left[\mathbf{B}^{(1)}, \dots, \mathbf{B}^{(i)}, \dots, \mathbf{B}^{(Z)} \right] \mathbf{G}. \quad (11)$$

Let $\mathbf{B} = \left[\mathbf{B}^{(1)}, \dots, \mathbf{B}^{(i)}, \dots, \mathbf{B}^{(Z)} \right]$, we need only to solve for the gain matrix \mathbf{G} since the bases matrix \mathbf{B} is fixed. The matrix \mathbf{G} is a combination of submatrices, and every column n of \mathbf{G} is a concatenation of subcolumns as follows:

$$\begin{bmatrix} \mathbf{G}^{(1)} \\ \vdots \\ \mathbf{G}^{(i)} \\ \vdots \\ \mathbf{G}^{(Z)} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_1^{(1)} & \dots & \mathbf{g}_n^{(1)} & \dots & \mathbf{g}_N^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{g}_1^{(i)} & \dots & \mathbf{g}_n^{(i)} & \dots & \mathbf{g}_N^{(i)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{g}_1^{(Z)} & \dots & \mathbf{g}_n^{(Z)} & \dots & \mathbf{g}_N^{(Z)} \end{bmatrix}, \quad (12)$$

where N is the maximum number of columns in matrix \mathbf{G} , and $\mathbf{g}_n^{(i)}$ is the column number n in the gain submatrix $\mathbf{G}^{(i)}$. Each submatrix represents the set of gains that its corresponding trained basis matrix has in the mixed signal. For the log-normalized columns of the submatrix $\mathbf{G}^{(i)}$ there is a corresponding trained gain prior GMM. We need the solution of \mathbf{G} in equation (11) to minimize the IS-divergence cost function in equation (6), and the corresponding log-normalized columns of each submatrix $\mathbf{G}^{(i)}$ to maximize the log-likelihood with its

corresponding trained gain prior GMM. Combining these two objectives, the solution of \mathbf{G} should minimize the following regularized IS-divergence cost function:

$$C(\mathbf{G}) = D_{IS}(\mathbf{Y} \parallel \mathbf{B}\mathbf{G}) - R(\mathbf{G}). \quad (13)$$

Where $D_{IS}(\mathbf{Y} \parallel \mathbf{B}\mathbf{G})$ is the regular IS-divergence cost function, and $R(\mathbf{G})$ is the weighted sum of the log-likelihoods of the log-normalized columns that correspond to the gain submatrices in \mathbf{G} under the GMMs trained using the columns of $\mathbf{G}_{\text{train}}$ in equation (9). For each log-likelihood of the gain submatrix $\mathbf{G}^{(i)}$ there is a corresponding regularization parameter $\alpha^{(i)}$. $R(\mathbf{G})$ can be written as follows:

$$R(\mathbf{G}) = \sum_{i=1}^Z \alpha^{(i)} L(\mathbf{G}^{(i)}), \quad (14)$$

where $\alpha^{(i)}$ is a regularization parameter of the log-likelihood of source i . The regularization parameters play an important role in the separation performance as we show later. The log-likelihood for the submatrix $\mathbf{G}^{(i)}$ for source i can be written as follows:

$$L(\mathbf{G}^{(i)}) = \sum_{n=1}^N \log \sum_{k=1}^K A_{k,n}^{(i)}, \quad (15)$$

where

$$A_{k,n}^{(i)} = \frac{w_k^{(i)}}{(2\pi)^{(d^{(i)}/2)} |\boldsymbol{\Sigma}_k^{(i)}|^{1/2}} \exp \left\{ -\frac{1}{2} \left(\log \frac{\mathbf{g}_n^{(i)}}{\|\mathbf{g}_n^{(i)}\|_2} - \boldsymbol{\mu}_k^{(i)} \right)^T \left(\boldsymbol{\Sigma}_k^{(i)} \right)^{-1} \left(\log \frac{\mathbf{g}_n^{(i)}}{\|\mathbf{g}_n^{(i)}\|_2} - \boldsymbol{\mu}_k^{(i)} \right) \right\}. \quad (16)$$

Each source subcolumns $[\mathbf{g}_1^{(i)}, \dots, \mathbf{g}_n^{(i)}, \dots, \mathbf{g}_N^{(i)}]$ in matrix \mathbf{G} in equation (12) are normalized and treated separately than other subcolumns sets, and each set of subcolumns is associated with its corresponding trained gain prior GMM.

To find the multiplicative update rule solution for \mathbf{G} in equation (13), we follow the same procedures as in [3, 2]. We express the gradient with respect to \mathbf{G} of the cost function $\nabla_{\mathbf{G}} C$ as the difference of two positive terms $\nabla_{\mathbf{G}}^+ C$ and $\nabla_{\mathbf{G}}^- C$:

$$\nabla_{\mathbf{G}} C = \nabla_{\mathbf{G}}^+ C - \nabla_{\mathbf{G}}^- C. \quad (17)$$

The cost function is shown to be nonincreasing under the update rule [3, 2]:

$$\mathbf{G} \leftarrow \mathbf{G} \otimes \frac{\nabla_{\mathbf{G}}^- C}{\nabla_{\mathbf{G}}^+ C}, \quad (18)$$

where the operations \otimes and division are element-wise as in equation (8). We can write the gradients as

$$\nabla_{\mathbf{G}} C = \nabla_{\mathbf{G}} D_{IS} - \nabla R(\mathbf{G}), \quad (19)$$

where $\nabla R(\mathbf{G})$ is a matrix with the same size of \mathbf{G} and it is a combination of submatrices as follows:

$$\nabla R(\mathbf{G}) = \begin{bmatrix} \alpha^{(1)} \nabla L(\mathbf{G}^{(1)}) \\ \vdots \\ \alpha^{(i)} \nabla L(\mathbf{G}^{(i)}) \\ \vdots \\ \alpha^{(Z)} \nabla L(\mathbf{G}^{(Z)}) \end{bmatrix}. \quad (20)$$

The gradient for the IS-cost function and the log-likelihood can also be written as the difference of two positive terms as follows:

$$\nabla_{\mathbf{G}} D_{IS} = \nabla_{\mathbf{G}}^+ D_{IS} - \nabla_{\mathbf{G}}^- D_{IS}, \quad (21)$$

and

$$\nabla R(\mathbf{G}) = \nabla^+ R(\mathbf{G}) - \nabla^- R(\mathbf{G}). \quad (22)$$

We can rewrite equations (17, 19) as:

$$\nabla_{\mathbf{G}} C = (\nabla_{\mathbf{G}}^+ D_{IS} + \nabla^- R(\mathbf{G})) - (\nabla_{\mathbf{G}}^- D_{IS} + \nabla^+ R(\mathbf{G})). \quad (23)$$

The final update rule in equation (18) can be written as follows:

$$\mathbf{G} \leftarrow \mathbf{G} \otimes \frac{\nabla_{\mathbf{G}}^- D_{IS} + \nabla^+ R(\mathbf{G})}{\nabla_{\mathbf{G}}^+ D_{IS} + \nabla^- R(\mathbf{G})}, \quad (24)$$

where

$$\nabla_{\mathbf{G}} D_{IS} = \mathbf{B}^T \frac{\mathbf{1}}{\mathbf{B}\mathbf{G}} - \mathbf{B}^T \frac{\mathbf{Y}}{(\mathbf{B}\mathbf{G})^2}, \quad (25)$$

$$\nabla_{\mathbf{G}}^- D_{IS} = \mathbf{B}^T \frac{\mathbf{Y}}{(\mathbf{B}\mathbf{G})^2}, \quad (26)$$

and

$$\nabla_{\mathbf{G}}^+ D_{IS} = \mathbf{B}^T \frac{\mathbf{1}}{\mathbf{B}\mathbf{G}}. \quad (27)$$

The row j and column n component of the gradient of the log-likelihood in equation (15) can also be written as the difference of two positive terms as

$$\left(\nabla L(\mathbf{G}^{(i)}) \right)_{jn} = \left(\nabla^+ L(\mathbf{G}^{(i)}) \right)_{jn} - \left(\nabla^- L(\mathbf{G}^{(i)}) \right)_{jn}, \quad (28)$$

where

$$\left(\nabla^- L(\mathbf{G}^{(i)}) \right)_{jn} = \sum_{k=1}^K \gamma_{k,n}^{(i)} \left(\boldsymbol{\Sigma}_{k,jj}^{(i)} \right)^{-1} \left(\frac{\boldsymbol{\mu}_{k,j}^{(i)}}{\mathbf{g}_{jn}^{(i)}} + \frac{\mathbf{g}_{jn}^{(i)}}{\|\mathbf{g}_n^{(i)}\|_2} \log \frac{\mathbf{g}_{jn}^{(i)}}{\|\mathbf{g}_n^{(i)}\|_2} \right), \quad (29)$$

$$\left(\nabla^+ L(\mathbf{G}^{(i)}) \right)_{jn} = \sum_{k=1}^K \gamma_{k,n}^{(i)} \left(\boldsymbol{\Sigma}_{k,jj}^{(i)} \right)^{-1} \left(\frac{\boldsymbol{\mu}_{k,j}^{(i)} \mathbf{g}_{jn}^{(i)}}{\|\mathbf{g}_n^{(i)}\|_2^2} + \frac{1}{\mathbf{g}_{jn}^{(i)}} \log \frac{\mathbf{g}_{jn}^{(i)}}{\|\mathbf{g}_n^{(i)}\|_2} \right), \quad (30)$$

and

$$\gamma_{k,n}^{(i)} = \frac{-A_{k,n}^{(i)}}{\sum_{k=1}^K A_{k,n}^{(i)}}.$$

Since the GMMs are trained by log-normalized columns, we know that the values of the mean vectors $\boldsymbol{\mu}$ are always negative. The values of the vectors \mathbf{g} are always positive, so the values from equations (29) and (30) will be always positive. Equations (26, 27, 29, 30, 20) are used to find the total gradients in equation (23) and then to derive the update rules for \mathbf{G} in equation (24). The initialization of the matrix \mathbf{G} is done by running one regular NMF iteration without any prior.

Normalizing vectors in the prior model slightly increases the complexity of the gradients computation, but it is beneficial in situations where the source signals occur with varying energy levels. Normalizing the training and testing gain matrices gives the prior models a chance to work with any energy level that the source signals can take in the mixed signal regardless of the energy levels of the training signals. It is important to note that, normalization during the separation process is done only for maximizing the log-likelihood with the prior models only. The general solution for the cost function in equation (13) is not normalized. The normalization is done for the prior to match

the energy level of the training signals that are used to train the GMMs.

After finding the suitable solution for the matrix \mathbf{G} , the initial power spectrogram estimate of any source i is found as follows:

$$\tilde{\mathbf{S}}^{(i)} = \mathbf{B}^{(i)} \mathbf{G}^{(i)}. \quad (31)$$

The final STFT estimate $\hat{\mathbf{S}}^{(i)}(t, f)$ of source i can be found as in [4] by scaling each entry of the mixed signal STFT according to the contribution of the source i in the mixed signal as follows:

$$\hat{\mathbf{S}}^{(i)}(t, f) = \mathbf{H}^{(i)}(t, f) Y(t, f), \quad (32)$$

where

$$\mathbf{H}^{(i)} = \frac{\mathbf{B}^{(i)} \mathbf{G}^{(i)}}{\sum_{j=1}^Z (\mathbf{B}^{(j)} \mathbf{G}^{(j)})}, \quad (33)$$

and the division is done element-wise. Since the multiplication $(\mathbf{B}^{(i)} \mathbf{G}^{(i)})$ represents a PSD, $\mathbf{H}^{(i)}$ can be seen as a Wiener filter. After finding the contribution of each source signal i in the mixed signal, the estimated source signal $\hat{s}^{(i)}(t)$ can be found by using inverse STFT of its corresponding STFT $\hat{\mathbf{S}}^{(i)}(t, f)$.

6. Experiments and Discussion

We applied the proposed algorithm to separate a speech signal from a background piano music signal. Our main goal was to get a clean speech signal from a mixture of speech and piano signals. We simulated our algorithm on a collection of speech and piano data at 16kHz sampling rate. For training speech data, we used 540 short utterances from a single speaker, we used other 20 utterances for testing. For music data, we downloaded piano music data from piano society web site [7]. We used 12 pieces from different composers but from a single artist for training and left out one piece for testing. The PSD for the training speech and music data were calculated by using the STFT: A Hamming window with 480 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 first FFT points. We trained 128 basis vectors for each source, which makes the size of each \mathbf{B} matrix to be 257×128 , hence, the vector dimension $d = 128$ in equation (16) for both sources. The suitable number of the GMM component K always depends on the size and the type of the training data. In this work, we fixed the number of GMM components to be 32 for each source.

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 G.191 ITU-T STL software suite [8]. For each SMR value, we obtained 20 test utterances this way.

Performance measurement of the separation algorithm was done using the signal to noise ratio (SNR). The average SNR over the 20 test utterances for each SMR case are reported.

Table 1 shows the signal to noise ratio of the separated speech signal using NMF with different values of the regularization parameters $\alpha^{(\text{speech})}$ and $\alpha^{(\text{music})}$. First column of this table shows the separation results of using NMF without using the GMM gain prior models " $\alpha^{(\text{speech})} = 0, \alpha^{(\text{music})} = 0$ ". In the second column, we show small values for the regularization parameters that improve the separation results comparing to using NMF without any prior information for all SMR cases. If we know some information about SMR of the mixed signal, we

can choose different values for the regularization parameters for each SMR case, that can lead to better results as we can see in the last column of the table.

Table 1: Signal to Noise Ratio in dB for the speech signal using regularized NMF with different values of the regularization parameters $\alpha^{(\text{speech})}$ and $\alpha^{(\text{music})}$.

SMR dB	$\alpha^{(\text{speech})} = 0$ $\alpha^{(\text{music})} = 0$	$\alpha^{(\text{speech})} = 0.1$ $\alpha^{(\text{music})} = 0.1$	better choices		
				$\alpha^{(\text{speech})}$	$\alpha^{(\text{music})}$
-5	3.32	3.45	4.55	0.5	0.005
0	7.19	7.43	7.68	0.1	0.01
5	10.58	10.64	10.67	0.1	0.01
10	12.91	13.02	13.06	0.01	0.05
15	15.62	15.92	16.42	0.01	0.5
20	17.14	17.69	20.69	0.01	10

As we can see from the last column in the table, at low SMR we get better results when the values of $\alpha^{(\text{speech})}$ is slightly higher than the values of $\alpha^{(\text{music})}$. This means, when the speech signal has less energy in the mixed signal, we rely more on the prior model for the speech signal. As the energy level of the speech signal increases, the values of $\alpha^{(\text{speech})}$ decreases and the value of $\alpha^{(\text{music})}$ increases since the energy level of the music signal is decreasing. We can also see that, comparing with no prior case, we can get better separation results by choosing suitable values for the regularization parameters.

We applied our proposed algorithm using IS-NMF divergence cost function. It also can be easily used with any other NMF cost function. The gradients of the log-likelihood of the gain GMMs will be the same. The only differences will be related to the gradient of the chosen NMF cost function.

7. Conclusion

In this work, we introduced a new regularized NMF algorithm for single channel source separation. The energy independent GMM prior models were incorporated with NMF solutions to improve the separation performance.

8. References

- [1] 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.
- [2] Nancy Bertin, Roland Badeau, and Emmanuel Vincent, "Enforcing harmonicity and smoothness in bayesian nonnegative matrix factorization applied to polyphonic music transcription," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 3, pp. 538–549, 2010.
- [3] 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.
- [4] 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.
- [5] Kevin W. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors," in *Proc. of ICASSP*, 2008.
- [6] Xabier Jaureguiberry, Pierre Leveau, Simon Maller, and Juan Jose Burred, "Adaptation of source-specific dictionaries in non-negative matrix factorization for source separation," in *ICASSP*, 2011.
- [7] URL, "http://pianosociety.com," 2009.
- [8] URL, "http://www.itu.int/rec/T-REC-G.191/en," 2009.