Hidden Markov Models as 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 rich statistical priors, modeling temporal gain sequences in the solutions of nonnegative matrix factorization (NMF). The proposed method can be used for single-channel source separation (SCSS) applications. 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 and temporal prior information on the weight combination patterns that the trained basis vectors can jointly receive for each source in the observed mixed signal. The Hidden Markov Model (HMM) is used as a log-normalized gains (weights) prior model for the NMF solution. The normalization makes the prior models energy independent. HMM is used as a rich model that characterizes the statistics of sequential data. The NMF solutions for the weights are encouraged to increase the log-likelihood with the trained gain prior HMMs while reducing the NMF reconstruction error at the same time.

Index Terms: Nonnegative matrix factorization, single-channel source separation, and Hidden Markov Models.

1. Introduction

Nonnegative matrix factorization [1], is an important tool that is used often in source separation problems, especially when only one observation of the mixed signal is available [2]. In single-channel source separation, NMF uses the training data to train a set of basis vectors for each source. Then NMF is used to decompose the spectrogram of the observed mixed signal as a weighted linear combination of the trained basis vectors for all sources that are involved in the mixed signal. The spectrogram estimate for each source is found by summing the decomposition terms that include its corresponding trained basis vectors. Prior information about the NMF decomposition results is usually considered to improve the separation performance of NMF. This prior information can be harmonicity and temporal smoothness of the source signals [2], or sparsity and temporal continuity [3].

In this work, we try to make better use of the available training data. NMF is usually used to decompose the spectrogram of training data of each source into a trained basis matrix and a trained gains matrix. In separation stage, the trained basis matrices for all sources are only used and the trained gain matrices are usually ignored. The columns of the trained gains matrix represents the valid gain combination sequences for a certain type of source signal. This gains matrix can be used to train a prior model for the valid weight pattern sequence for each source. The prior models can guide the NMF decomposition weights during the separation stage to find the solution that can be considered as valid weight pattern sequences for the underlying source signal and also minimizing the NMF reconstruction error. The trained gain matrix is used here to build a HMM prior model for each source. The columns of the trained gain matrices are normalized and their logarithm is taken and used to train the prior HMM for each source. After observing the mixed signal, NMF is used to decompose the spectrogram of the mixed signal as a weighted linear combination of the columns of the trained basis matrices. The decomposition weights are jointly encouraged to increase the log-likelihood with their corresponding trained prior HMMs. The proposed algorithm uses HMM, which is a rich model to represent the statistical distribution of any sequential training data. Temporal relations between frames are also modeled in the HMM. Since the HMMs are trained using normalized data, there is no 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 limitations.

The remainder of this paper is organized as follows: In section 2, a mathematical formulation of the SCSS problem is given. In sections 3 and 4, we give a brief explanation about NMF and show the training processes of the NMF bases models and the HMM 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

The main aim of SCSS is to find estimates of source signals that are mixed on a single observation channel $y(t)$. This problem is usually formed in the short time Fourier transform (STFT) domain as follows:

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

where $Y(t,f)$ is the STFT of $y(t)$, t represents the frame index, f is the frequency-index, $S^{(z)}(t,f)$ is the unknown STFT of source z 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 $\sigma_y^2(t,f) = \sum_{z=1}^{Z} \sigma_z^2(t,f)$ where $\sigma_y^2(t,f) = E(|Y(t,f)|^2)$. We can ap-

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

proximately write the PSDs in matrix form as follows:

$$\boldsymbol{Y} = \sum_{z=1}^{Z} \boldsymbol{S}^{(z)}, \tag{2}$$

where $\boldsymbol{S}^{(z)}, z \in \{1, .., Z\}$ 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.

3. Nonnegative matrix factorization

Nonnegative matrix factorization is used to decompose any nonnegative matrix $\boldsymbol{V}$ into a nonnegative bases matrix $\boldsymbol{B}$ and a nonnegative gains matrix $\boldsymbol{G}$ as $\boldsymbol{V} \approx \boldsymbol{BG}$. The solutions for $\boldsymbol{B}$ and $\boldsymbol{G}$ can be found by minimizing the following Itakura-Saito (IS) divergence cost function [4]:

$$\min_{\boldsymbol{B},\boldsymbol{G}} D_{IS}\left(\boldsymbol{V} \,||\, \boldsymbol{BG}\right), \tag{3}$$

where

$$D_{IS}\left(\boldsymbol{V} \,||\, \boldsymbol{BG}\right) = \sum_{a,b} \left(\frac{\boldsymbol{V}_{a,b}}{\left(\boldsymbol{BG}\right)_{a,b}} - \log \frac{\boldsymbol{V}_{a,b}}{\left(\boldsymbol{BG}\right)_{a,b}} - 1 \right).$$

This divergence cost function is a good measurement for the perceptual difference between different signals [4]. The IS-NMF solution for equation (3) can be iteratively computed by using the following multiplicative update rules of $\boldsymbol{B}$ and $\boldsymbol{G}$ as follows [4]:

$$\boldsymbol{B} \leftarrow \boldsymbol{B} \otimes \frac{\left(\frac{\boldsymbol{V}}{(\boldsymbol{BG})^2}\right) \boldsymbol{G}^T}{\left(\frac{\boldsymbol{1}}{\boldsymbol{BG}}\right) \boldsymbol{G}^T}, \tag{4}$$

$$\boldsymbol{G} \leftarrow \boldsymbol{G} \otimes \frac{\boldsymbol{B}^T \left(\frac{\boldsymbol{V}}{(\boldsymbol{BG})^2}\right)}{\boldsymbol{B}^T \left(\frac{\boldsymbol{1}}{\boldsymbol{BG}}\right)}, \tag{5}$$

where $\boldsymbol{1}$ is a matrix of ones with the same size of $\boldsymbol{V}$, the operation $\otimes$ is an element-wise multiplication, all divisions and $(.)^2$ are element-wise operations. The matrices $\boldsymbol{B}$ and $\boldsymbol{G}$ are initialized by positive random noise.

4. Training the source models

The power spectrogram of the training data for each source $\boldsymbol{S}^{(z)}_{\text{train}}$ is calculated. The multiplicative update rules in equations (4) and (5) are used to decompose the power spectrogram for each source into trained basis matrix and trained gains matrix as follows:

$$\boldsymbol{S}^{(z)}_{\text{train}} \approx \boldsymbol{B}^{(z)} \boldsymbol{G}^{(z)}_{\text{train}}, \tag{6}$$

within each iteration, we normalize the columns of $\boldsymbol{B}^{(z)}$ and find $\boldsymbol{G}^{(z)}_{\text{train}}$ accordingly. After computing the basis and gains matrices for each source training data, all the basis matrices are used in the mixed signal decomposition as shown in equation (7). We use the gains matrices to train prior models for the possible pattern sequences that each source signal can possibly have in the gains matrix. For each gains matrix $\boldsymbol{G}^{(z)}_{\text{train}}$ for each source, we normalize its columns and compute the logarithm of the normalized columns, and use them to train its gain prior HMM with Gaussian mixture GMM as the emission distribution. Using the Baum-Welch algorithm [5], we train a fully connected HMM for each source in an unsupervised fashion. We hope that the HMM learns phonetic classes or musical sound clusters as its states, when we train in this fashion. The reason for normalization is to make the prior models insensitive to the energy level of the signals, which leads to an energy independent prior model. Normalization is done using the L_2 norm.

5. Signal separation

After observing the mixed signal $y(t)$, the power spectral density $\boldsymbol{Y}$ of the mixed signal is computed using STFT. NMF decomposes the power spectrogram $\boldsymbol{Y}$ with the trained basis matrices that were found from solving equation (6) as follows:

$$\boldsymbol{Y} \approx \left[\boldsymbol{B}^{(1)}, ..., \boldsymbol{B}^{(z)}, ..., \boldsymbol{B}^{(Z)}\right] \boldsymbol{G} \text{ or } \boldsymbol{Y} \approx \boldsymbol{BG}. \tag{7}$$

Then the initial spectrogram estimate of each source can be calculated as

$$\widetilde{\boldsymbol{S}}^{(z)} = \boldsymbol{B}^{(z)} \boldsymbol{G}^{(z)} \quad \text{for any } z. \tag{8}$$

The only unknown that we need to find is the gains matrix $\boldsymbol{G}$ since the bases matrix $\boldsymbol{B}$ is fixed. The matrix $\boldsymbol{G}$ with N columns is a combination of submatrices, and each column $\boldsymbol{g}_n$ of $\boldsymbol{G}$ is a concatenation of subcolumns $\boldsymbol{g}^{(z)}_n$. Each submatrix $\boldsymbol{G}^{(z)}$ represents the gain combination that its corresponding basis matrix $\boldsymbol{B}^{(z)}$ contributes in the PSD of the observed mixed signal. For each submatrix $\boldsymbol{G}^{(z)}$ there is a corresponding trained prior HMM for its corresponding log-normalized columns. We need the solution of $\boldsymbol{G}$ in equation (7) to minimize the IS-divergence cost function in equation (3), and the corresponding log-normalized columns of each submatrix $\boldsymbol{G}^{(z)}$ in $\boldsymbol{G}$ to maximize the log-likelihood with its corresponding trained gain prior HMM. Combining these two objectives, the solution of $\boldsymbol{G}$ should minimize the following regularized IS-divergence cost function:

$$C\left(G\right) = D_{IS}\left(\boldsymbol{Y} \,||\, \boldsymbol{BG}\right) - R(\boldsymbol{G}). \tag{9}$$

Where $D_{IS}\left(\boldsymbol{Y} \,||\, \boldsymbol{BG}\right)$ is the regular IS-divergence cost function, and $R(\boldsymbol{G})$ is the weighted sum of the log-likelihoods of the log-normalized columns of the gain submatrices under the trained HMMs. For each log-likelihood of the gain submatrix $\boldsymbol{G}^{(z)}$ there is a corresponding regularization parameter $\lambda^{(z)}$. $R(\boldsymbol{G})$ can be written as follows:

$$R(\boldsymbol{G}) = \sum_{z=1}^{Z} \lambda^{(z)} L(\boldsymbol{G}^{(z)}), \tag{10}$$

where $\lambda^{(z)}$ is the regularization parameter of the log-likelihood of source z. The log-likelihood for the sequence of the log-normalized columns that corresponding to the submatrix $\boldsymbol{G}^{(z)}$ for source z can be written as follows:

$$L(\boldsymbol{G}^{(z)}) = \log p \left(\log \frac{\boldsymbol{g}^{(z)}_1}{\left\|\boldsymbol{g}^{(z)}_1\right\|_2}, .., \log \frac{\boldsymbol{g}^{(z)}_n}{\left\|\boldsymbol{g}^{(z)}_n\right\|_2}, .., \log \frac{\boldsymbol{g}^{(z)}_N}{\left\|\boldsymbol{g}^{(z)}_N\right\|_2} \right). \tag{11}$$

To find the multiplicative update rule solution for $\boldsymbol{G}$ in equation (9), we follow the same procedures as in [3, 2]. We express the gradient with respect to $\boldsymbol{G}$ of the cost function in equation (9) $\nabla_G C$ as the difference of two positive terms $\nabla^+_G C$ and $\nabla^-_G C$ as follow:

$$\nabla_G C = \nabla^+_G C - \nabla^-_G C. \tag{12}$$

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

$$\boldsymbol{G} \leftarrow \boldsymbol{G} \otimes \frac{\nabla^-_G C}{\nabla^+_G C}, \tag{13}$$

where the operations $\otimes$ and division are element-wise as in equation (5). We can write the gradients as

$$\nabla_G C = \nabla_G D_{IS} - \nabla R(\boldsymbol{G}), \tag{14}$$

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

$$\nabla R(\boldsymbol{G}) = \begin{bmatrix} \lambda^{(1)} \nabla L(\boldsymbol{G}^{(1)}) \\ \cdot \\ \cdot \\ \lambda^{(z)} \nabla L(\boldsymbol{G}^{(z)}) \\ \cdot \\ \cdot \\ \lambda^{(Z)} \nabla L(\boldsymbol{G}^{(Z)}) \end{bmatrix}. \tag{15}$$

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

$$\nabla_G D_{IS} = \nabla_G^+ D_{IS} - \nabla_G^- D_{IS}, \tag{16}$$

and

$$\nabla R(\boldsymbol{G}) = \nabla^+ R(\boldsymbol{G}) - \nabla^- R(\boldsymbol{G}). \tag{17}$$

We can rewrite equations (12, 14) as:

$$\nabla_G C = \left(\nabla_G^+ D_{IS} + \nabla^- R(\boldsymbol{G}) \right) - \left(\nabla_G^- D_{IS} + \nabla^+ R(\boldsymbol{G}) \right). \tag{18}$$

The final update rule in equation (13) can be written as follows:

$$\boldsymbol{G} \leftarrow \boldsymbol{G} \otimes \frac{\nabla_G^- D_{IS} + \nabla^+ R(\boldsymbol{G})}{\nabla_G^+ D_{IS} + \nabla^- R(\boldsymbol{G})}, \tag{19}$$

where

$$\nabla_G D_{IS} = \boldsymbol{B}^T \frac{\boldsymbol{1}}{\boldsymbol{BG}} - \boldsymbol{B}^T \frac{\boldsymbol{Y}}{(\boldsymbol{BG})^2}, \tag{20}$$

$$\nabla_G^- D_{IS} = \boldsymbol{B}^T \frac{\boldsymbol{Y}}{(\boldsymbol{BG})^2}, \quad \text{and} \quad \nabla_G^+ D_{IS} = \boldsymbol{B}^T \frac{\boldsymbol{1}}{\boldsymbol{BG}}. \tag{21}$$

To find the gradients for the log-likelihood in equations (10, 11), let $\log \frac{\boldsymbol{g}_n^{(z)}}{\left\| \boldsymbol{g}_n^{(z)} \right\|_2} = \boldsymbol{x}_n$, given a set of data $\boldsymbol{x} = \{\boldsymbol{x}_1, .., \boldsymbol{x}_n .., \boldsymbol{x}_N\}$, a state sequence $q_1, .., q_n .., q_N \in |Q|$, and the trained HMM parameters $\Lambda = \{\boldsymbol{A}, \boldsymbol{E}, \pi\}$, where $\boldsymbol{A}$ is the transition matrix with entries $a_{ij} = p(q_{n+1} = j | q_n = i)$, $\boldsymbol{E}$ is the set of weights, means and covariances parameters of the GMM emission probabilities, and $\pi = p(q_1 = i)$ is the initial state probabilities, the likelihood can be calculated as follows:

$$p(\boldsymbol{x}_{1:N} | \Lambda) = \sum_{q_{1:N}} p\left(\boldsymbol{x}_{1:N} | q_{1:N}, \Lambda\right) p\left(q_{1:N} | \Lambda\right), \tag{22}$$

where $p\left(q_{1:N} | \Lambda\right) = \prod_n p\left(q_n | q_{n-1}, \Lambda\right)$ is the multiplication of transition probabilities, and $p\left(\boldsymbol{x}_{1:N} | q_{1:N}, \Lambda\right) = \prod_n p\left(\boldsymbol{x}_n | q_n, \Lambda\right)$ is the multiplication of the GMM emission probabilities which are defined as:

$$p(\boldsymbol{x}_n | q_n = j, \Lambda) = \sum_{k=1}^{K} \gamma_{jkn}, \tag{23}$$

$$\gamma_{jkn} = \frac{w_{jk}}{(2\pi)^{d/2} \left| \boldsymbol{\Sigma}_{jk} \right|^{1/2}} \exp \left\{ -\frac{1}{2} \left(\boldsymbol{x}_n - \boldsymbol{\mu}_{jk} \right)^T \boldsymbol{\Sigma}_{jk}^{-1} \left(\boldsymbol{x}_n - \boldsymbol{\mu}_{jk} \right) \right\},$$

where K is the number of Gaussian mixture components, w_{jk} is the mixture weight, d is the vector dimension, $\boldsymbol{\mu}_{jk}$ is the mean vector and $\boldsymbol{\Sigma}_{jk}$ is the diagonal covariance matrix of the k^{th} Gaussian model for state j. The likelihood in equation (22) can be calculated using the forward-backward algorithm [5] as follows:

$$p(\boldsymbol{x}_{1:N} | \Lambda) = \sum_{j=1}^{|Q|} \alpha_n(j) \beta_n(j) \ \text{ for any } n, \tag{24}$$

where

$$\begin{aligned} &\alpha_n(j) = \sum_{i=1}^{|Q|} \alpha_{n-1}(j) a_{ij} p\left(\boldsymbol{x}_n | j\right) \quad \forall j = 1, ..., Q, \\ &\alpha_1(j) = \pi_j p\left(\boldsymbol{x}_1 | j\right) \quad \forall j = 1, ..., Q, \end{aligned} \tag{25}$$

$$\begin{aligned} &\beta_n(j) = \sum_{i=1}^{|Q|} a_{ij} p\left(\boldsymbol{x}_{n+1} | j\right) \beta_{n+1}(j) \quad \forall j = 1, ..., Q, \ \text{ and} \\ &\beta_N(j) = 1, \quad \forall j = 1, ..., Q. \end{aligned} \tag{26}$$

The gradient of the log-likelihood in equation (11) can be found using (24). The gradient with respect to the data point $\boldsymbol{g}_n$ of the log-likelihood in equation (24) can be found as follows:

$$\nabla_{\boldsymbol{g}_n} \left[\log p(\boldsymbol{x}_{1:N}) \right] = \frac{\sum_{j=1}^{|Q|} \beta_n(j) \nabla_{\boldsymbol{g}_n} \left[\alpha_n(j) \right]}{\sum_{j=1}^{|Q|} \alpha_n(j) \beta_n(j)}, \tag{27}$$

where

$$\nabla_{\boldsymbol{g}_n} \left[\alpha_n(j) \right] = \sum_{i=1}^{|Q|} \alpha_{n-1}(j) a_{ij} \nabla_{\boldsymbol{g}_n} \left[p\left(\boldsymbol{x}_n | j\right) \right]. \tag{28}$$

Note that $\beta_n(j)$ and $\alpha_{n-1}(j)$ are not functions of $\boldsymbol{g}_n$. The gradient $\nabla_{\boldsymbol{g}_n} \left[p\left(\boldsymbol{x}_n | j\right) \right]$ can also be written as the difference of two positive terms

$$\nabla_{\boldsymbol{g}_n} \left[p\left(\boldsymbol{x}_n | j\right) \right] = \nabla_{\boldsymbol{g}_n}^+ \left[p\left(\boldsymbol{x}_n | j\right) \right] - \nabla_{\boldsymbol{g}_n}^- \left[p\left(\boldsymbol{x}_n | j\right) \right], \tag{29}$$

these gradients can be calculated after replacing $\boldsymbol{x}_n$ with $\log \frac{\boldsymbol{g}_n^{(z)}}{\left\| \boldsymbol{g}_n^{(z)} \right\|_2}$ in equation (23). The component a of these gradient vectors can be calculated as follows:

$$\nabla_{\boldsymbol{g}_n}^- \left[p\left(\boldsymbol{x}_n | j\right) \right]_a = \sum_{k=1}^{K} -\gamma_{jkn} \left(\boldsymbol{\Sigma}_{jk_{aa}} \right)^{-1} \left(\frac{\boldsymbol{\mu}_{jk_a}}{\boldsymbol{g}_{an}^{(z)}} + \frac{\boldsymbol{g}_{an}^{(z)}}{\left\| \boldsymbol{g}_n^{(z)} \right\|_2^2} \log \frac{\boldsymbol{g}_{an}^{(z)}}{\left\| \boldsymbol{g}_n^{(z)} \right\|_2} \right), \tag{30}$$

$$\nabla_{\boldsymbol{g}_n}^+ \left[p\left(\boldsymbol{x}_n | j\right) \right]_a = \sum_{k=1}^{K} -\gamma_{jkn} \left(\boldsymbol{\Sigma}_{k_{aa}} \right)^{-1} \left(\frac{\boldsymbol{\mu}_{jk_a} \boldsymbol{g}_{an}^{(z)}}{\left\| \boldsymbol{g}_n^{(z)} \right\|_2^2} + \frac{1}{\boldsymbol{g}_{an}^{(z)}} \log \frac{\boldsymbol{g}_{an}^{(z)}}{\left\| \boldsymbol{g}_n^{(z)} \right\|_2} \right). \tag{31}$$

Since the HMMs 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 $\boldsymbol{g}$ are always positive, so the values from equations (30) and (31) will be always positive. To calculate the gradients for each submatrix in equations (15,17): first, we calculate all values of α and β using equations (25, 26) for all HMM states and all observations after replacing each $\boldsymbol{x}_n$ with $\log \frac{\boldsymbol{g}_n^{(z)}}{\left\| \boldsymbol{g}_n^{(z)} \right\|_2}$. Second, equations (27) to (31) are used to calculate the gradient of each column in the submatrix. We repeat these procedures for each submatrix and construct the prior gradients matrix in (15,17). We calculate the gradients in equation (21) and use them to derive the update rules for $\boldsymbol{G}$ in equation (19). The initialization of the matrix $\boldsymbol{G}$ is done by running one regular NMF iteration without any prior. Calculating the gradient of the log-likelihood in equation (27) gives us the chance to scale the values of α and β as shown in [5] to avoid any numerical problem. Since the same scale will appear in both numerator and denominator of equation (27), then this scale will not affect the values of the gradients of the log-likelihood.

Normalizing vectors in the prior model in training and testing is beneficial in situations where the source signals occur with varying energy levels. Normalization 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 (9) 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 HMMs.

After finding the suitable solution for the matrix $\boldsymbol{G}$, the initial power spectrogram estimate $\widetilde{\boldsymbol{S}}^{(z)}$ of each source z is found using equation (8). Given the initial estimated power spectral density $\widetilde{\boldsymbol{S}}^{(z)}$, the final minimum mean square error estimates of each source STFT can be obtained through Wiener filtering [4] as follows:

$$\hat{S}^{(z)}\left(t,f\right) = \boldsymbol{H}^{(z)}\left(t,f\right) Y\left(t,f\right), \tag{32}$$

where

$$\boldsymbol{H}^{(z)} = \frac{\widetilde{\boldsymbol{S}}^{(z)}}{\sum_{r=1}^{Z} \widetilde{\boldsymbol{S}}^{(r)}}, \tag{33}$$

and the division is done element-wise. The estimated source signal $\hat{s}^{(z)}(t)$ can be found by using inverse STFT of its corresponding STFT $\hat{S}^{(z)}(t,f)$.

6. Experiments and Discussion

We applied the proposed algorithm to separate a speech signal from a background piano music signal. The main aim was to get a clean speech signal from a single mixture of speech and piano signals. The proposed algorithm was simulated on a collection of speech and piano data at 16kHz sampling rate. For training speech data, 540 short utterances from a single speaker were used, we used other 20 utterances for testing. For music data, piano music data from piano society web site [6] were downloaded. 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 the first FFT points. We trained 128 basis vectors for each source, which makes the size of each trained basis matrix to be 257 × 128, hence, the vector dimension $d = 128$ in equation (23) for both sources. For the HMM models, the suitable number of state Q and number of GMM components K are always dependent on the size and the type of the training data. In this work, we fixed the number of states to be $Q = 4$ with fully connected topology and GMM components to be $K = 8$ for each state for each source signal. 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 [7]. 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 $\lambda^{(\text{speech})}$ and $\lambda^{(\text{music})}$. First column of this table shows the separation results of using NMF without using the HMM gain prior models "$\lambda^{(\text{speech})} = 0$, $\lambda^{(\text{music})} = 0$". In the second column, we show the case where the same values for the regularization parameters improve the separation results for all SMR cases comparing to using NMF without any prior information. Assuming we have some information about the SMR of the mixed signal, we can make better choices 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: SNR in dB for the speech signal using regularized NMF with different values of the regularization parameters $\lambda^{(\text{speech})}$ and $\lambda^{(\text{music})}$.

SMR	$\lambda^{(\text{speech})} = 0$	$\lambda^{(\text{speech})} = 0.1$	better choices		
dB	$\lambda^{(\text{music})} = 0$	$\lambda^{(\text{music})} = 0.1$		$\lambda^{(\text{speech})}$	$\lambda^{(\text{music})}$
-5	3.69	4.21	**4.54**	0.1	0.01
0	7.41	7.81	**7.92**	0.1	0.01
5	10.75	10.90	**10.90**	0.1	0.1
10	13.02	13.43	**13.43**	0.1	0.1
15	15.75	16.06	**16.51**	0.01	0.5
20	17.26	17.80	**21.87**	0.01	100

As we can see from the last column of the table, at low SMR we get better results when the values of $\lambda^{(\text{speech})}$ is slightly higher compared with high SMR. 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 $\lambda^{(\text{speech})}$ decreases and the value of $\lambda^{(\text{music})}$ 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.

7. Conclusion

In this work, we introduced a new regularized NMF algorithm for single channel source separation. The energy independent HMM 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 Emmanual 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] Lawrence R Rabiner, "A tutorial on hidden Markov models and selected application in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–285, Feb. 1989.

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

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