Source separation using regularized NMF with MMSE estimates under GMM priors with online learning for the uncertainties

Emad M. Grais *, Hakan Erdogan

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

1. Introduction

Nonnegative matrix factorization [1] is an important tool for source separation applications, especially when only one observation of the mixed signal is available. NMF is used to decompose a nonnegative matrix into a multiplication of two nonnegative matrices, a basis/dictionary matrix and a gains/weights/activities matrix. The basis matrix contains a set of basis vectors and the gains matrix contains the weights corresponding to the basis vectors in the basis matrix. The NMF solutions are found by solving an optimization problem based on minimizing a predefined cost function. As most optimization problems, the main goal in NMF is to find the solutions that minimize the cost function without considering any prior information rather than the nonnegativity constraint. There have been many works which tried to enforce prior information related to the nature of the application on the NMF decomposition results. For audio source separation applications, the continuity and sparsity priors were enforced in the NMF decomposition weights [2]. In [3,4], smoothness and harmonicity priors were enforced on the NMF solution in Bayesian framework applied to music transcription. In [5,6] the regularized NMF was used to increase the NMF decomposition weights matrix likelihood under a prior Gaussian distribution. In [7], Markov chain prior model for smoothness was used within a Bayesian framework in regularized NMF with Itakura–Saito (IS-NMF) divergence. In [8], the conjugate prior distributions on the NMF weights and basis matrices solutions with the Poisson observation model within Bayesian framework was introduced. The Gamma distribution and the Gamma Markov chain [9] were used as priors for the basis and weights/gains matrices respectively in [8]. A mixture of Gamma...
prior model was used as a prior for the basis matrix in [10]. The regularized NMF with smoothness and spatial decorrelation constraints was used in [11] for EEG applications. In [11,12], a variety of constrained NMF algorithms were used for different applications.

In supervised single channel source separation (SCSS), NMF is used in two main stages, the training stage and the separation stage [13–17]. In the training stage, NMF is used to decompose the spectrogram of clean training data for the source signals into a multiplication of trained basis and weights/gains matrices for each source. The trained basis matrix is used as a representative model for the training data of each source and the trained gains matrices are usually ignored. In the separation stage, NMF is used to decompose the mixed signal spectrogram as a nonnegative weighted linear combination of the columns in the trained basis matrices. The spectrogram estimate for each source in the mixed signal can be found by summing its corresponding trained basis terms from the NMF decomposition during the separation stage.

One of the main problems of this framework is that, the estimate for each source spectrogram is affected by other sources in the mixed signal. More prior information about the source signals besides their trained basis matrices is usually needed during the separation stage. The trained gains/weights matrix can also be used to incorporate more information about each source signal during the separation stage. The trained weights matrix contains valid weight combinations that their corresponding bases in the basis matrix can jointly receive for a certain source signal. The information in the trained weights matrix can be used to favor solutions with valid weight combinations during the separation stage. The trained weights matrix can be used to train a prior model that captures the statistics of the valid weight combinations for each source signal. The trained prior models can be used to guide the NMF solution of the weights matrix during the separation stage to find a better estimate for the source signals.

In this work, we use the trained weights matrix columns to train a prior multivariate Gaussian mixture model (GMM) for each source signal. The GMM is a rich model which captures the statistics and the correlations of the valid weight combinations for the trained basis vectors for each source signal. GMMs are extensively used in speech processing applications like speech recognition and speaker verification. GMMs are used to model the multi-modal nature in speech feature vectors due to phonetic differences, speaking styles, gender, accents [18]. We are conjecturing that the weight vectors of the NMF gains matrix can be considered as a feature extracted from the signal in a frame so that it can be modeled well with a GMM.

The trained basis matrix and its corresponding trained prior GMM for the weights matrix are used as representative models for the training data for each source signal. In the separation stage, the prior GMMs are used to guide the NMF to find a weights matrix that is more compatible with its corresponding trained weights matrix for each source. Since the prior GMMs are trained using training data for the source signals, we can see these GMMs as source dependent prior models. Using trained prior models is different than incorporating other intuitive priors such as continuity, smoothness, and sparsity [2–4,11], which are valid prior assumptions for a wide range of source signals. Using a powerful model like GMM which can capture the multi-modal structure of training data is better than using single modal prior models [5–8] that cannot capture the multi-modal nature in some source signals such as speech signals due to phonetic differences, speaking styles, gender, and accents.

In [19], the prior GMM that models the valid weight combinations for each source was also used to guide the NMF solution for the gains/weights matrix during the separation stage. The priors in [19] are enforced by maximizing the log-likelihood of the NMF solution with the trained prior GMMs. The priors in [19] are enforced without evaluating how good the NMF solution is without using the priors. For example, if the NMF solution without prior is not satisfactory, we would like to rely more on the priors and vice versa.

In this paper, we introduce a new strategy of applying the priors on the NMF solutions of the weights matrix during the separation stage. The new strategy is based on evaluating how much the solution of the NMF gains matrix needs to lean on the prior GMMs. The prior information about the gains matrix is incorporated on the NMF solution of the gains matrix during the separation stage in three main stages. In the first stage, NMF is used to decompose the mixed signal spectrogram as weighted linear combinations of the trained basis matrices without considering any prior information about the weight combinations in the gains matrix. In the second stage, the NMF solution of the gains matrix is then evaluated by measuring how far the weight combinations in the gains matrix is from the trained prior GMM for each source. The NMF solutions without using priors for the weights matrix for each source during the separation stage can be seen as a deformed image. The deformation operator parameters which measure the uncertainty of the NMF solution of the weights matrix are learned directly from the observed mixed signal. The uncertainty in this work is a measurement of how far the NMF solution of the weights matrix during the separation stage is from being a valid weight pattern that is modeled in the prior GMM. In the third stage, the learned uncertainties are used with the minimum mean square error (MMSE) estimator to find an estimate of the weights matrix. The estimated weights matrix should also consider the minimization of the NMF cost function. To achieve these two goals, a regularized NMF cost function is used. The uncertainties within MMSE estimates of the valid weight combinations are embedded in the regularized NMF cost function. The uncertainty measurements play a very important role in this work as we show in the following sections. If the uncertainty of the NMF solution of the weights matrix is high, that means the regularized NMF needs more support from the prior term. In case of low uncertainty, the regularized NMF needs less support from the prior term. Including the uncertainty measurements in the regularization term using the MMSE estimate makes the proposed regularized NMF algorithm decide automatically how much the solution should rely on the prior GMM term. This is the main advantage of the proposed regularized NMF compared to the regularization using the log-likelihood of the GMM prior or other prior distributions [19–21]. Incorporation of the uncertainties that measure the extent of distortion in the NMF weights in the regularization term is a main novelty of this work, which has not been seen before in the regularization literature.

The remainder of this paper is organized as follows: In Section 2 we give a brief explanation about the SCSS problem, NMF, the conventional use of NMF in the SCSS problem, and our previous work of using GMM as a prior model for regularizing the NMF cost function. In Section 3, we introduce the new proposed regularized NMF and how it is used in the SCSS problem, which is the main contribution of this paper. In the remaining sections, we present our observations and the results of our experiments.

2. Background and related work

In this paper, we apply the proposed MMSE based regularization for the single channel source separation (SCSS) problem using NMF.
2.1. Problem formulation for SCSS

In single channel source separation (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 \( Y(f, t) \) be the STFT of \( y(t) \), where \( t \) represents the frame index and \( f \) is the frequency-index. Assuming the number of sources is two and due to the linearity of the STFT, we have:

\[
Y(f, t) = S_1(f, t) + S_2(f, t),
\]

where \( S_1(f, t) \) and \( S_2(f, t) \) are the unknown STFT of the first and second 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 PSDs as follows:

\[
\sigma^2_y(f, t) = \sigma^2_{S_1}(f, t) + \sigma^2_{S_2}(f, t),
\]

where \( \sigma^2_{S_1}(f, t) = E[|Y(f, t)|^2] \). We can write the PSDs for all frames as a spectrogram matrix as follows:

\[
Y = S_1 + S_2,
\]

where \( S_1 \) and \( S_2 \) are the unknown spectrograms of the source signals, and they need to be estimated using the observed mixed signal and training data for each source. The spectrogram of the measured signal \( Y \) is calculated by taking the squared magnitude of the STFT of the measured signal \( y(t) \) and the rows of the matrix correspond to frequencies and the columns correspond to frames.

2.2. Nonnegative matrix factorization

Nonnegative matrix factorization is used to decompose any nonnegative matrix \( V \) into a multiplication of a nonnegative basis matrix \( B \) and a nonnegative gains or weights matrix \( G \) as follows:

\[
V \approx BG.
\]

The columns of matrix \( B \) contain nonnegative basis or dictionary vectors that are optimized to allow the data in \( V \) to be approximated as a nonnegative linear combination of its constituent vectors. Each column in the gains/weights matrix \( G \) contains the set of weight combinations that the basis vectors in the basis matrix have for its corresponding column in the \( V \) matrix. To solve for matrix \( B \) and \( G \), different NMF cost functions can be used. For audio source separation applications, the Itakura–Saito (IS-NMF) divergence cost function [7] is usually used. This cost function is found to be a good measurement for the perceptual differences between different audio signals [7,22,23]. The IS-NMF cost function is defined as:

\[
\min_{B,G} D_{IS}(V \| BG),
\]

where

\[
D_{IS}(V \| BG) = \sum_{m,n} \left( \frac{V_{m,n}}{(BG)_{m,n}} - \log \frac{V_{m,n}}{(BG)_{m,n}} - 1 \right).
\]

The IS-NMF solutions for Eq. (5) can be computed by alternating multiplicative updates of \( B \) and \( G \) [7,22] as:

\[
B \leftarrow B \odot \frac{V(BG)^T}{(BG)G^T},
\]

\[
G \leftarrow G \odot \frac{B^T V}{(BG)^T},
\]

where the operation \( \odot \) is an element-wise multiplication, all divisions and \( (.)^2 \) are element-wise operations. In source separation applications, IS-NMF is used with matrices of power spectral densities of the source signals [7,22].

2.3. Conventional NMF for SCSS

In conventional single channel source separation using NMF without regularization [15], there are two main stages to find estimates for \( S_1 \) and \( S_2 \) in Eq. (3). The first stage is the training stage and the second stage is the separation/testing stage. In the training stage, the spectrogram \( S^{\text{train}} \) for each source is calculated by computing the squared magnitude of the STFT of each source training signal. NMF is used to decompose the spectrograms into basis and gains matrices as follows:

\[
S^{\text{train}}_1 \approx B_1G^{\text{train}}_1, \quad S^{\text{train}}_2 \approx B_2G^{\text{train}}_2.
\]

The multiplicative update rules in Eqs. (6) and (7) are used to solve for \( B_1, B_2, G^{\text{train}}_1 \) and \( G^{\text{train}}_2 \) for both sources. Within each iteration, the columns of \( B_1 \) and \( B_2 \) are normalized and the matrices \( G^{\text{train}}_1 \) and \( G^{\text{train}}_2 \) are computed accordingly. The initialization of all matrices \( B_1, B_2, G^{\text{train}}_1 \) and \( G^{\text{train}}_2 \) is done using positive random noise. After finding basis and gains matrices for each source training data, the basis matrices are used in the mixed signal decomposition as shown in the following sections. All the basis matrices \( B_1 \) and \( B_2 \) are kept fixed in the remaining sections in this paper.

In the separation stage after observing the mixed signal \( y(t) \), NMF is used to decompose the mixed signal spectrogram \( Y \) with the trained bases matrices \( B_1 \) and \( B_2 \) that were found from solving Eq. (8) as follows:

\[
Y \approx [B_1, B_2]G, \quad \text{or} \quad Y \approx \begin{bmatrix} B_1 & B_2 \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}.
\]

Let \( B^{\text{train}} = [B_1, B_2] \). The only unknown here is the gains matrix \( G \) since the matrix \( B^{\text{train}} \) was found during the training stage and it is fixed in the separation stage. The matrix \( G \) is a combination of two submatrices as shown in Eq. (9). NMF is used to solve for \( G \) in (9) using the update rule in Eq. (7) and \( G \) is initialized with positive random numbers.

After computing the gains matrix, the initial spectrogram estimate for each source can be found as:

\[
\tilde{S}_1 = B_1G_1, \quad \tilde{S}_2 = B_2G_2.
\]

2.4. Motivation for regularized NMF

The solution of the gains submatrix \( G_1 \) in (9) is usually affected by the existence of the second source in the mixed signal. Also, \( G_2 \) is affected by the first source in the mixed signal. The effect of one source into the gains matrix solution of the other source strongly depends on the energy level of each source in the mixed signal. Therefore, the estimated spectrograms \( \tilde{S}_1 \) and \( \tilde{S}_2 \) in Eq. (10) that are found from solving \( G \) using the update rule in (7) may contain residual contribution from each other and other distortions. To fix this problem, more discriminative constraints must be added to the solution of each gains submatrix. The columns of the gains submatrix \( G_1 \) (or \( G_2 \)) should form a valid/expected weight combination for its corresponding basis matrix of its corresponding source signal. The information about the valid weight combinations that can exist in the gains matrix for a source signal can be found in the gains matrix that was computed from the clean training data of the same source. For example, the information about valid weight combinations that can exist in the gains matrix \( G_1 \) in Eq. (9) can be found in its training gains matrix \( G^{\text{train}}_1 \) in Eq. (8).
weight combinations that the basis matrix $B_1$ can receive for the first source. Note that, the basis matrix $B_1$ is common in the training and separation stages. The solution of the gains submatrix $G_1$ in Eq. (9) should consider the prior about the valid combination that is present in its corresponding trained gains matrix $G_{1\text{train}}$ in Eq. (8) for the same source.

2.5. GMM-based regularization

In our previous work [19], data in the training gains matrix $G_{i\text{train}}$ for source $i$ was modeled using a GMM. The NMF solution of the gains matrix during the separation stage was guided by the prior GMM. The GMM was learned using the logarithm of the normalized columns of the training gains matrix.

2.5.1. Training the GMM prior models

The gains matrices $G_{1\text{train}}$ and $G_{2\text{train}}$ in Eq. (8) were used in our previous work [19] to train prior models for the expected/valid weight patterns in the gains matrix for each source. For each matrix $G_{1\text{train}}$ and $G_{2\text{train}}$, we normalized their columns and then calculated their logarithm. The normalization is usually done using the Euclidean norm. The log-normalized columns were then used to train a prior GMM for each source. The GMM for a random variable $x$ is defined as:

$$p(x) = \sum_{k=1}^{K} \frac{o_k}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \times \exp \left\{ -\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) \right\},$$

where $K$ is the number of Gaussian mixture components, $o_k$ is the mixture weight, $d$ is the vector dimension, $\mu_k$ is the mean vector and $\Sigma_k$ is the diagonal covariance matrix of the $k$th Gaussian model. In training GMM, the expectation maximization (EM) algorithm [24] is used to learn the GMM parameters ($o_k, \mu_k, \Sigma_k, \forall k = \{1, 2, \ldots, K\}$) for each source given its trained gain matrix $G_{\text{train}}$. The suitable value for $K$ usually depends on the nature, dimension and the size of the available training data. We used the logarithm because it has been shown that the logarithm of a variable taking values between 0 and 1 can be modeled well by a GMM [25]. Since the main goal of the prior model is to capture the statistics of the patterns in the trained gains matrix, we used normalization to make the prior models insensitive to the energy level of the training data. The normalization makes the same prior models applicable for a wide range of energy levels and avoids the need to train a different prior model for different energy levels. By normalization we are modeling the ratio and correlation between the combination of the weights that the bases can jointly receive.

2.5.2. NMF-log-likelihood based regularization

The NMF solution for the gains matrix during the separation stage was enforced in [19] to increase its log-likelihood with the trained GMM prior using regularized NMF as follows:

$$C_{\text{old}} = D_{BL}(Y \parallel B_{\text{train}}G) - R_{\text{old}}(G),$$

where $R_{\text{old}}(G)$ is the weighted sum of the log-likelihoods of the log-normalized columns of the gains matrix $G$. $R_{\text{old}}(G)$ was defined as follows:

$$R_{\text{old}}(G) = \sum_{i=1}^{2} \eta_i I_{\text{old}}(G_i),$$

where $I_{\text{old}}(G_i)$ is the log-likelihood for the submatrix $G_i$, and $\eta_i$ is the regularization parameter for source $i$. The regularization parameter in [19] was playing two important roles. The first role was to match the scale of the IS-NMF divergence term with the scale of the log-likelihood prior term. The second role was to decide how much the regularized NMF cost function needs to rely on the prior term. The results in [19] show that, when the source $i$ has higher energy level than the other sources, the value of its corresponding regularization parameter $\eta_i$ becomes smaller than the values of other regularization parameters for the other sources. This observation can be stated as follows: when the source has high energy level, the gains matrix solution of the regularized NMF in (12) relies less on the prior model and vice versa. The values of the regularization parameters in [19] were chosen manually for every energy level for each source.

2.6. The conjugate prior based regularization

The conjugate prior models usually enforced on NMF solutions using a Bayesian framework [7,8,21]. The conjugate prior models are usually used to make the probabilistic inference tractable. There is no need to learn the parameters (hyper-parameters) of the conjugate prior models during the training stage. The hyper-parameters are usually chosen during the separation/testing stage. These hyper-parameters are usually multi-dimensional parameters and trying all possible combinations for their entries to find the suitable values for the given task is impossible. In the cases when the conjugate prior models of the NMF solutions were used [8,21], the hyper-parameters of the prior models were also chosen manually. In [19], it was also shown that, the hyper-parameter choices for the conjugate prior models can also depend on the energy level differences of the source signals in the mixed signal.

3. The proposed MMSE estimate based regularization

In the following sections, we give the motivation of the proposed MMSE-based regularization. Then, we give more details about our proposed regularized NMF using MMSE estimate to find better solution for the gains matrix in (9). In Section 3.2, we present our proposed regularized NMF in a general fashion. In Section 3.2, we assume we have a trained basis matrix $B$, a trained prior GMM for a clean gains matrix, and a gains matrix $G$ that inherited some distortion from the original matrix $V$ from solving Eq. (5). We introduce our proposed regularized NMF in a general fashion in Section 3.2 to make the idea clearer for different NMF applications like: dimensionality reduction, denoising, and other applications. The update rules that solve the proposed regularized NMF are also derived in Section 3.2 in a general fashion regardless of the application. The GMM in Section 3.2 is the trained prior GMM that captures the statistics of the valid weights combinations that should have been existed in the gains matrix $G$. In Section 3.3, we show how we use the proposed regularized NMF to find better solutions for the gain submatrices in Eq. (9) for our single channel source separation problem.

3.1. Motivation for the proposed regularized NMF

In this work, we use prior GMMs to guide the solution of the gains matrix during the separation stage using regularized NMF as in [19] but following a totally different regularization strategy. We try to find a way to estimate how much the solution of the regularized NMF needs to rely on the prior GMMs automatically, not manually, as in [19]. The way of finding how much the regularized NMF solution of the gains matrix needs to rely on the prior GMM is by measuring how far the statistics of the solution of the gains matrix $G_i$ in (9) is from the statistics of the solution of the valid gains matrix solution $G_{i\text{train}}$ in (8) for source $i$. Recall that, the matrix $G_{i\text{train}}$ in (8) contains the valid weight combinations that the columns in the basis matrix $B_i$ can jointly receive for the clean
data of source $i$. The data in $G_{i}^{train}$ can be used as a prior information for what kind of weight combinations that should exist in $G_{i}$ in (9) since the matrix $B_{i}$ is the same in (8) and (9). The matrix $G_{i}^{train}$ in (8) is used to train a prior GMM for the expected (valid) weight combinations that can exist in the gains matrix for source $i$ as in [19]. The solution of the gains submatrix $G_{i}$ in (9) can be seen as a deformed observation that needs to be restored using MMSE estimate under its corresponding GMM prior for source $i$. How far the statistics of the solution of the gains matrix $G_{i}$ is from the statistics of the solution of the valid gains matrix solution $G_{i}^{train}$ can be seen as how much the gains submatrix $G_{i}$ is deformed. How much deformation exists in the gains matrix $G_{i}$ can be learned directly and the logarithm of this deformation is modeled using a Gaussian distribution with zero mean and a diagonal covariance matrix $Ψ_{i}$. When the deformation or the uncertainty measurement $Ψ_{i}$ of the gain submatrix $G_{i}$ is high, we expect our target regularized NMF cost function to rely more on the prior GMM for source $i$ and vice versa. Based on the measurement $Ψ_{i}$, the proposed NMF cost function decides automatically how much the solution of the regularized NMF needs to rely on the prior GMMs, which is a main advantage of the proposed regularized NMF over our previous work [19]. Applying the prior information on the gains matrix $G_{i}$ in (9) using MMSE estimate under a GMM prior using regularized NMF is the new strategy that we introduce in this paper.

3.2. Formulation for the proposed regularized NMF

In this work, we need the solution of the gains matrix $G_{i}$ to minimize the IS-divergence cost function in Eq. (5), and the columns of the gains matrix $G_{i}$ should form valid weight combinations under a prior GMM model. Gaussian mixture model is a very general prior model where we can see the means of the GMM mixture components as “valid templates” that were observed in the training data. Even, Parzen density priors [26] can be seen under the same framework. In Parzen density prior estimation, training examples are seen as “valid templates” and a fixed variance is assigned to each example. In GMM priors, we learn the templates as cluster means from training data and we can also estimate the cluster variances from the data. We can think of the GMM prior as a way to encourage the use of valid templates or cluster means in the NMF solution during the separation stage. This view of the GMM prior will be helpful in understanding the MMSE estimate method we introduce in this paper.

We can see the solution of the conventional NMF (NMF without prior) as distorted observations of a true/valid template. We can find a way of measuring how far the conventional NMF solution is from the trained templates in the prior GMM and call this the error term. We can learn a probability distribution model for the distortion that captures how far the observations in the conventional gains matrix is from the prior GMM. The distortion or the error model can be seen as a summary of the distortion that exists in all columns in the gains matrix of the NMF solution. Based on this distortion evaluation, the proposed regularized NMF can decide automatically how much the solution of the NMF needs help from the prior model. By deciding automatically how much the regularized NMF needs to rely on the prior we conjecture that, we do not need to manually change the values for the regularization parameter based on the energy differences of the sources in the mixed signal\(^1\) to improve the performance of NMF as in [19].

Based on the prior GMM and the trained distortion model, we can find a better estimate for the desired observation for each column in the distorted gains matrix. We can mathematically formulate this by seeing the solution matrix $G$ that only minimizes the cost function in Eq. (5) as a distorted image where its restored image needs to be estimated. The columns of the matrix $G$ are normalized using the $l^{2}$ norm and their logarithm is then calculated. Let the log-normalized column $n$ namely $(\log \frac{g_{n}}{\Psi_{n}})$ of the gains matrix be $q_{n}$. The vector $q_{n}$ is treated as a distorted observation as:

$$q_{n} = x_{n} + e,$$

where $x_{n}$ is the unknown desired pattern that corresponds to the observation $q_{n}$, and needs to be estimated under a prior GMM, $e$ is the logarithm of the deformation operator, which is modeled by a Gaussian distribution with zero mean and diagonal covariance matrix $Ψ$ as $\mathcal{N}(e|0,Ψ)$. The GMM prior model for the gains matrix is trained using log-normalized columns of the trained gains matrix from training data as shown for example in Section 2.5.2. The uncertainty $Ψ$ is trained using maximum likelihood directly from all the log-normalized columns of the gains matrix $q_{i} = \{q_{n}, \ldots, q_{n}, \ldots, q_{N}\}$, where $N$ is the number of columns in the matrix $G$. The uncertainty $Ψ$ can be iteratively learned using the expectation maximization (EM) algorithm. Given the prior GMM parameters which are considered fixed here, the update of $Ψ$ is found in the M-step based on the sufficient statistics $\hat{z}_{n}$ and $\hat{R}_{n}$ as follows [27–29] [Appendix A]:

$$Ψ = \text{diag}\left\{\frac{1}{N} \sum_{n=1}^{N}(q_{n}q_{n}^{T} - q_{n}\hat{z}_{n} - \hat{z}_{n}q_{n}^{T} + \hat{R}_{n})\right\},$$

where the “diag” operator sets all the off-diagonal elements of a matrix to zero and the sufficient statistics $\hat{z}_{n}$ and $\hat{R}_{n}$ can be updated in the E-step using $Ψ$ from the previous iteration as follows:

$$\hat{z}_{n} = \sum_{k=1}^{K} \gamma_{kn}\hat{z}_{kn},$$

and

$$\hat{R}_{n} = \sum_{k=1}^{K} \gamma_{kn}\hat{R}_{kn},$$

where

$$\gamma_{kn} = \frac{\omega_{k}N(q_{n}|\mu_{k},\Sigma_{k} + Ψ)}{\sum_{j=1}^{K} \omega_{j}N(q_{n}|\mu_{j},\Sigma_{j} + Ψ)};$$

$$\hat{R}_{kn} = \Sigma_{k} - \Sigma_{k}(\Sigma_{k} + Ψ)^{-1}\Sigma_{k} + \hat{z}_{kn}\hat{z}_{kn}^{T};$$

and

$$\hat{z}_{kn} = \mu_{k} + \Sigma_{k}(\Sigma_{k} + Ψ)^{-1}(q_{n} - \mu_{k}).$$

$Ψ$ is considered as a general uncertainty measurement over all the observations in matrix $G$.

Given the GMM prior parameters and the uncertainty measurement $Ψ$, the MMSE estimate of each pattern $x_{n}$ given its observation $q_{n}$ under the observation model in Eq. (14) can be found similar to [27–29] as in Appendix A as follows:

$$\hat{x}_{n} = f(q_{n}) = \sum_{k=1}^{K} \gamma_{kn}[\mu_{k} + \Sigma_{k}(\Sigma_{k} + Ψ)^{-1}(q_{n} - \mu_{k})].$$

where

$$\gamma_{kn} = \frac{\omega_{k}N(q_{n}|\mu_{k},\Sigma_{k} + Ψ)}{\sum_{j=1}^{K} \omega_{j}N(q_{n}|\mu_{j},\Sigma_{j} + Ψ)}.$$
The value of $\Psi$ in the term $\Sigma_k(\Sigma_k + \Psi)^{-1}$ in Eq. (21) plays an important role in this framework. When the entries of the uncertainty $\Psi$ are very small compared to their corresponding entries in $\Sigma_k$ for a certain active GMM component $k$, the term $\Sigma_k(\Sigma_k + \Psi)^{-1}$ tends to be the identity matrix, and MMSE estimate in (21) will be the observation $q_n$. When the entries of the uncertainty $\Psi$ are very high compared to their corresponding entries in $\Sigma_k$ for a certain active GMM component $k$, the term $\Sigma_k(\Sigma_k + \Psi)^{-1}$ tends to be a zeros matrix, and MMSE estimate will be the weighted sum of prior templates $\sum_{k=1}^{K} y_{kn} \mu_k$. In most cases $y_{kn}$ tends to be close to one for one Gaussian component, and close to zero for the other components in a large dimensional space. This makes the MMSE estimate in the case of high $\Psi$ to be one of the mean vectors in the prior GMM, which is considered as a template pattern for the valid observation. We can rephrase this as follows: When the uncertainty of the observations $q$ is high, the MMSE estimate of $x$ relies more on the prior GMM of $x$. When the uncertainty of the observations $q$ is low, the MMSE estimate of $x$ relies more on the observation $q$. In general, the MMSE solution of $x$ lies between the observation $q$ and one of the templates in the prior GMM. The term $\Sigma_k(\Sigma_k + \Psi)^{-1}$ controls the distance between $\hat{x}_n$ and $q_n$ and also the distance between $\hat{x}_n$ and one of the template $\mu_k$ assuming that $y_{kn} \approx 1$ for a Gaussian component $k$.

The model in Eq. (14) expresses the normalized columns of the gains matrix as a distorted image with a diagonal multiplicative deformation matrix. For the normalized gain columns $\mathbf{E}(\mathbf{G})$ there is a deformation matrix $\mathbf{E}$ with log-normal distribution that applies to the correct pattern to obtain the observation. We need to estimate $\hat{g}_n$ which satisfies the following equation:

$$\frac{\hat{g}_n}{\|\hat{g}_n\|_2} = \mathbf{E}\hat{g}_n,$$  \hspace{1cm} (23)

The uncertainty for $\mathbf{E}$ is represented in the covariance matrix $\Psi$. For the distorted matrix $\mathbf{G}$ we find the corresponding MMSE estimate for its normalized columns $\mathbf{G}$. Another reason for working in logarithm domain is that, the gains are constrained to be nonnegative and the MMSE estimate can be negative so the logarithm of the normalized gains is an unconstrained variable that we can work with. The estimated weight patterns in $\mathbf{G}$ that are corresponding to the MMSE estimates for the correct patterns do not consider minimizing the NMF cost function in Eq. (5), which is still the main goal. We need the solution of $\mathbf{G}$ to consider the pattern shape priors on the solution of the gains matrix, and also consider the reconstruction error of the NMF cost function. To consider the combination of the two objectives, we use regularized NMF. We add a penalty term to the NMF cost function. The penalty term tries to minimize the distance between the solution of log-normalized columns of $\hat{g}_n$ with its corresponding MMSE estimate $f(\mathbf{g}_n)$ as follows:

$$\log \frac{\mathbf{g}_n}{\|\mathbf{g}_n\|_2} \approx f\left( \log \frac{\mathbf{g}_n}{\|\mathbf{g}_n\|_2} \right)$$

$$\frac{\|\mathbf{g}_n\|_2}{\|\mathbf{\hat{g}}_n\|_2} \approx \exp \left( f\left( \log \frac{\mathbf{g}_n}{\|\mathbf{g}_n\|_2} \right) \right),$$ \hspace{1cm} (24)

The regularized IS-NMF cost function is defined as follows:

$$C = D_{G}(\mathbf{V} \parallel \mathbf{BG}) + \alpha L(\mathbf{G}),$$ \hspace{1cm} (25)

where

$$L(\mathbf{G}) = \sum_{n=1}^{N} \frac{\|\mathbf{g}_n\|_2}{\|\mathbf{\hat{g}}_n\|_2} - \exp \left( f\left( \log \frac{\mathbf{g}_n}{\|\mathbf{g}_n\|_2} \right) \right),$$ \hspace{1cm} (26)

$f(\log \mathbf{g}/\|\mathbf{g}\|_2)$ is the MMSE estimate defined in Eq. (21), and $\alpha$ is a regularization parameter. The regularized NMF can be rewritten in more details as

$$C = \sum_{m,n} \left( \frac{V_{m,n}}{(BG)_{m,n}} - \log \frac{V_{m,n}}{(BG)_{m,n}} - 1 \right)$$

$$+ \alpha \sum_{n=1}^{N} \frac{\|\mathbf{g}_n\|_2}{\|\mathbf{\hat{g}}_n\|_2} - \exp \left( \sum_{k=1}^{K} y_{kn} \left( \mu_k + \Sigma_k(\Sigma_k + \Psi)^{-1} \right) \right)$$

$$\times \left( \log \frac{\mathbf{g}_n}{\|\mathbf{\hat{g}}_n\|_2} - \mu_k \right)^2.$$  \hspace{1cm} (27)

In Eq. (27), the MMSE estimate of the desired patterns of the gains matrix is embedded in the regularized NMF cost function. The first term in (27), decreases the reconstruction error between $\mathbf{V}$ and $\mathbf{BG}$. Given $\mathbf{V}$, we can forget for a while the MMSE estimate concept that led us to our target regularized NMF cost function in (27) and see Eq. (27) as an optimization problem. We can see from (27) that, if the distortion measurement parameter $\Psi$ is high, the regularized nonnegative matrix factorization solution for the gains matrix will rely more on the prior GMM for the gains matrix. If the distortion parameter $\Psi$ is low, the regularized nonnegative matrix factorization solution for the gains matrix will be close to the ordinary NMF solution for the gains matrix without considering any prior. The second term in Eq. (27) is ignored in the case of zero uncertainty $\Psi$. In case of high values of $\Psi$, the second term encourages to decrease the distance between each normal column $\mathbf{E}(\mathbf{G})$ in $\mathbf{G}$ with a corresponding prior template $\exp(\mu_k)$ assuming that $y_{kn} \approx 1$ for a certain Gaussian component $k$. For different values $\Psi$, the penalty term decreases the distance between each $\mathbf{E}(\mathbf{G})$ and an estimated pattern that lies between a prior template and $\mathbf{E}(\mathbf{G})$. Here has the effect of a multi-dimension regularization parameter which regularizes each row in the gains matrix differently. This multi-dimensional regularization parameter makes the idea of using the MMSE estimate based regularization more powerful than the log-likelihood based regularization in [19] where we have only scalar regularization parameters. The term $(\log \frac{\mathbf{E}(\mathbf{G})}{\|\mathbf{\hat{g}}_n\|_2} - \mu_k)$ in (27) measures how far each log-normalized column in the gains matrix is from a valid template $\mu_k$. Under the assumption $y_{kn} \approx 1$ for a certain Gaussian component $k$, the second term in (27) is also ignored when the observation log-normalized column in gains matrix is from a valid template $\mu_k$. How far each log-normalized column in the gains matrix is from a valid template decides how much influence the MMSE estimate prior term has to the solution of (27) for each observation.

The multiplicative update rule for $\mathbf{B}$ in (27) is still the same as in Eq. (6). The multiplicative update rule for $\mathbf{G}$ can be found by following the same procedures as in [2,4,19]. The gradient with respect to $\mathbf{G}$ of the cost function $\nabla_{\mathbf{G}} C$ can be expressed as a difference of two positive terms $\nabla^+_C \mathbf{G}$ and $\nabla^-_C \mathbf{G}$ as follows:

$$\nabla_{\mathbf{G}} C = \nabla^+_C \mathbf{G} - \nabla^-_C \mathbf{G},$$ \hspace{1cm} (28)

We can apply diagonally weighted gradient descent on the cost function where the diagonal step sizes are chosen such that the result is a multiplicative update of the matrix entries:

$$\mathbf{G} \leftarrow \mathbf{G} \odot \frac{\nabla^+_C \mathbf{G}}{\nabla^+_C \mathbf{G}}.$$ \hspace{1cm} (29)

where the operations $\odot$ and division are element-wise as in Eq. (7). We can write the gradients as:

$$\nabla_{\mathbf{G}} C = \nabla^+_C \mathbf{D}_B + \alpha \nabla_{\mathbf{L}(\mathbf{G})},$$ \hspace{1cm} (30)

where $\nabla_{\mathbf{L}(\mathbf{G})}$ is a matrix with the same size of $\mathbf{G}$. The gradient for the IS-NMF and the gradient of the prior term can also be expressed as a difference of two positive terms as follows:
\[ \nabla_G D_{IS} = \nabla^+_G D_{IS} - \nabla^-_G D_{IS}. \]  
(31)

and

\[ \nabla_G L(G) = \nabla^+_G L(G) - \nabla^-_G L(G). \]  
(32)

We can rewrite Eqs. (28), (30) as:

\[ \nabla_C C = \left( \nabla^+_G D_{IS} + \alpha \nabla^+_G L(G) \right) - \left( \nabla^-_G D_{IS} + \alpha \nabla^-_G L(G) \right). \]  
(33)

The final update rule in Eq. (29) can be written as follows:

\[ G \leftarrow G \otimes \frac{\nabla^-_G D_{IS} + \alpha \nabla^-_G L(G)}{\nabla^+_G D_{IS} + \alpha \nabla^+_G L(G)}. \]  
(34)

where

\[ \nabla^-_G D_{IS} = B^T \begin{bmatrix} 1 & \mathbf{B} \end{bmatrix} - B^T \frac{\mathbf{V}}{\mathbf{BG}} \]  
(35)

\[ \nabla^+_G D_{IS} = B^T \frac{\mathbf{V}}{\mathbf{BG}^2}. \]  
(36)

Note that, in calculating the gradients \( \nabla^+_G L(G) \) and \( \nabla^-_G L(G) \), the term \( \gamma \) is also a function of \( G \). The gradients \( \nabla^+_G L(G) \) and \( \nabla^-_G L(G) \) are calculated in Appendix B. Since all the terms in Eq. (34) are nonnegative, then the values of \( G \) of the update rule (34) are nonnegative.

3.3. The proposed regularized NMF for SCSS

In this section, we are back to the single channel source separation problem to find a better solution to Eq. (9). Fig. 1 shows the flow chart that summarizes all stages of applying our proposed regularized NMF method for SCSS problems. Given the trained basis matrices \( \mathbf{B}_1, \mathbf{B}_2 \) that were computed from solving (8), and the trained gain prior GMM for each source from Section 2.5.2, we try to apply the proposed regularized NMF cost function in Section 3.2 to find a better solution for the gain submatrices in Eq. (9). The bases matrix \( \mathbf{B}_{\text{train}} = [\mathbf{B}_1, \mathbf{B}_2] \) is still fixed here, we just need to update the gains matrix \( G \) in (9). The normalized columns of the submatrices \( G_1 \) and \( G_2 \) in (9) can be seen as deformed images as in Eq. (23) and their restored images need to be estimated. First, we need to learn the uncertainty parameters \( \Psi_1 \) and \( \Psi_2 \) for the deformation operators \( \mathbf{E}_1 \) and \( \mathbf{E}_2 \) respectively for each image as shown in learning the uncertainties stage in Fig. 1. The columns of the submatrix \( G_1 \) are normalized and their logarithm are calculated and used with the trained GMM prior parameters for the first source to estimate \( \Psi_1 \) iteratively using the EM algorithm in Eqs. (15) to (20). The log-normalized columns “\( \log \frac{\text{prior}}{\text{data}} \)” of \( G_1 \) can be seen as \( q_n \) in Eqs. (15) to (20). We repeat the same procedures to calculate \( \Psi_2 \), using the log-normalized columns of \( G_2 \) and the prior GMM for the second source. The uncertainties \( \Psi_1 \) and \( \Psi_2 \) can also be seen as measurements of the remaining distortion from one source into another source, which also depends on the mixing ratio between the two sources. For example, if the first source has higher energy than the second source in the mixed signal, we expect the values of \( \Psi_2 \) to be higher than the values in \( \Psi_1 \) and vice versa. After calculating the uncertainty parameters for both sources \( \Psi_1 \) and \( \Psi_2 \), we use the regularized NMF in (25) to solve for \( G \) with the prior GMMs for both sources and the estimated uncertainties \( \Psi_1 \) and \( \Psi_2 \) as follows:

\[ C = D_{IS} (\mathbf{Y} \parallel \mathbf{B}_{\text{train}} G) + R(G), \]  
(37)

where

\[ R(G) = \alpha_1 L_1(G_1) + \alpha_2 L_2(G_2). \]  
(38)

\( L_1(G_1) \) is defined as in Eq. (26) for the first source, \( L_2(G_2) \) is for the second source, \( \alpha_1 \), and \( \alpha_2 \) are their corresponding regularization parameters. The update rule in Eq. (34) can be used to solve for \( G \) after modifying it as follows:

\[ G \leftarrow G \otimes \frac{\nabla^-_G D_{IS} + \alpha \nabla^-_G L(G)}{\nabla^+_G D_{IS} + \alpha \nabla^+_G L(G)}. \]  
(39)

where \( \nabla^+_G R(G) \) and \( \nabla^-_G R(G) \) are nonnegative matrices with the same size of \( G \) and they are combinations of two submatrices as follows:

\[ \nabla^+_G R(G) = \begin{bmatrix} \alpha_1 \nabla^+_G L(G_1) \\ \alpha_2 \nabla^+_G L(G_2) \end{bmatrix}, \]

\[ \nabla^-_G R(G) = \begin{bmatrix} \alpha_1 \nabla^-_G L(G_1) \\ \alpha_2 \nabla^-_G L(G_2) \end{bmatrix}, \]  
(40)

where \( \nabla^+_G L(G_1), \nabla^-_G L(G_1), \nabla^+_G L(G_2), \) and \( \nabla^-_G L(G_2) \) are calculated as in Section 3.2 for each source.

The normalization of the columns of the gain matrices are used in the prior term \( R(G) \) and its gradient terms only. The general solution for the gains matrix of Eq. (37) at each iteration is not normalized. The normalization is done only in the prior term since the prior models have been trained by normalized data before. Normalization is also useful in cases where the source signals occur with different energy levels from each other in the mixed signal. 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.

The regularization parameters in (38) have only one role. They are chosen to match the scale between the NMF divergence term and the MMSE estimate prior term in the regularized NMF cost function in (37). There is no need to change the values of the regularization parameters according to the energy differences of the source signals in the mixed signal as in [19]. Reasonable values for the regularization parameters are chosen manually and kept fixed in this work. Another main difference between the regularized NMF in [19] that is shown in Eq. (12) and the proposed regularized NMF in this paper is related to the training procedures for the source models. In both works, the main aim of the training stage is to train the basis matrices and the prior GMMs for the gains for the source signals.

In [19], to match between the way the trained models were used during training with the way they were used during separation, the basis matrices and the prior GMM parameters were learned jointly using the regularized NMF cost function in (12). The joint training for the source models was introduced in [19] to improve the separation performance. In joint training, after updating the gains matrix at each NMF iteration using the gain update rule for the regularized NMF in (12), the GMM parameters were then updated (retrained). Since, we needed to update (retrain) the GMM parameters at each NMF iteration, joint training slowed down the training of the source models in [19]. Another problem of using joint training is that, we had other regularization parameters during the training stage that needed to be chosen. Using joint training duplicates the number of the regularization parameters that need to be chosen. Choosing the regularization parameters in [19] was done using validation data. That means, in [19] we had to train many source models (basis matrix and prior GMM) for different regularization parameter values. Then, we chose the best combination for the regularization parameter values in training and separation stages that gave the best results during the separation stage. In the case of using MMSE estimate regularization for NMF, we do not need to use joint training. In this paper, we do not need to consider solving the regularized NMF in (27) during
the training stage to solve \( (8) \). In the training stage, the training data for each source is assumed to be clean data. Since the spectrogram of each source training data represents clean source data, the NMF solution for the gains matrix cannot be seen as a distorted image. Therefore, the deformation measurement parameter \( \Psi_{\text{train}} \) is a matrix of zeros. When \( \Psi_{\text{train}} = 0 \), the MMSE estimates prior term in \( (27) \) will disappear because \( \sum_{k=1}^{K} \gamma_{kn} = 1 \). Then, the regularized NMF \( (27) \) becomes just NMF. That means, we do not need to use the regularized NMF during the training stage which is not the case in \([19]\). Here in the training stage, we just need to use IS-NMF to decompose the spectrogram of the training data into trained basis and gains matrices. After the trained gains matrix is computed, it is used to train the prior GMM as shown in Sections 2.3 and 2.5.2.

3.4. Source signals reconstruction

After finding the suitable solution for the gains matrix \( G \) in Section 3.3, the initial estimated spectrograms \( \tilde{S}_1 \) and \( \tilde{S}_2 \) can be calculated from \( (10) \) and then used to build spectral masks as follows:

\[
H_1 = \frac{\tilde{S}_1}{\tilde{S}_1 + \tilde{S}_2}, \quad H_2 = \frac{\tilde{S}_2}{\tilde{S}_1 + \tilde{S}_2},
\]

where the divisions are done element-wise. The final estimate of each source STFT can be obtained as follows:

\[
\hat{S}_1(t, f) = H_1(t, f)Y(t, f), \quad \hat{S}_2(t, f) = H_2(t, f)Y(t, f),
\]

where \( Y(t, f) \) is the STFT of the observed mixed signal in \( (1) \), \( H_1(t, f) \) and \( H_2(t, f) \) are the entries at row \( f \) and column \( t \) of the spectral masks \( H_1 \) and \( H_2 \) respectively. The spectral mask entries scale the observed mixed signal STFT entries according to the contribution of each source in the mixed signal. The spectral masks can be seen as the Wiener filter as in \([7]\). The estimated source signals \( \hat{S}_1(t) \) and \( \hat{S}_2(t) \) can be found by using inverse STFT of their corresponding STFTs \( \hat{S}_1(t, f) \) and \( \hat{S}_2(t, f) \).

4. Experiments and discussion

We applied the proposed algorithm to separate speech signals from background music signals. We used two different experimental setups. In the first experiment we separate speech signals of male speakers from TIMIT database from background piano signals. In the second experiment, we separate male and female speech signals from TIMIT dataset from general background music signals where different musical instruments are used.

4.1. First experiment

In this experiment, 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 16 kHz 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 the piano society web site \([30]\). We used 12 pieces with approximate 50 minutes total duration from different composers but from a single artist for training and left out one piece for testing. The PSD for the speech and music data were calculated by 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. We trained 128 basis vectors for each source, which makes the size of \( B_{\text{speech}} \) and \( B_{\text{music}} \) matrices to be \( 257 \times 128 \), hence, the vector dimension \( d = 128 \) in \( (11) \) for both sources. 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
(SNR) values in dB. The audio power levels of each file were found using the “speech voltmeter” program from the G.191 ITU-T STL software suite [31]. For each SNR value, we obtained 20 mixed utterances this way. We used the first 10 utterances as a validation set to choose reasonable values for the regularization parameters $\alpha_{\text{speech}}$ and $\alpha_{\text{music}}$. The other 10 mixed utterances were used for testing.

Performance measurement of the separation algorithm was done using the signal to noise ratio (SNR). The average SNR over the 10 test utterances for each SNR case are reported. We also used signal to interference ratio (SIR), which is defined as the ratio of the target energy to the interference error due to the music signal only [32].

Table 1 shows SNR and SIR of the separated speech signal using NMF with different values of the number of Gaussian mixture components $K$ and fixed regularization parameters $\alpha_{\text{speech}} = \alpha_{\text{music}} = 1$. The first column of the table shows the separation results of using just NMF without any prior.

As we can see from the table, the proposed regularized NMF algorithm improves the separation performance for challenging SMR cases compared with using just NMF without priors. Increasing the number of Gaussian mixture components $K$ improves the separation performance until $K = 16$. From the shown results, $K = 16$ seems to be a good choice for the given data sets. The best choice for $K$ usually depends on the nature and the size of the training data. For example, for speech signal in general there are variety of phonetic differences, gender, speaking styles, accents, which raises the necessity for using many Gaussian components.

4.1.1. Comparison with other priors

In this section we compared our proposed method of using MMSE estimates under GMM prior on the solution of NMF with two other prior methods. The first prior is the sparsity prior and the second prior is enforced by maximizing the loglikelihood under GMM prior distribution.

In the sparsity prior, the NMF solution of the gains matrix was enforced to be sparse [10,13]. The sparse NMF is defined as

$$C(G) = D_{\text{IS}}(Y \parallel BG) + \lambda \sum_{m,n} G_{m,n}. \quad (43)$$

where $\lambda$ is the regularization parameter. The gain update rule of $G$ can be found as follows:

$$G \leftarrow G \odot \frac{B^T Y + \lambda}{B^T B + \lambda}. \quad (44)$$

Enforcing sparsity on the NMF solution of the gains matrix is equivalent to model the prior of the gains matrix using exponential distribution with parameter $\lambda$ [10]. The update rule in Eq. (44) is found based on maximizing the likelihood of the gains matrix under the exponential prior distribution.

The second method of enforcing prior on the NMF solution is by using GMM gain prior [19,20]. The NMF solution for the gains matrix is enforced to increase its log-likelihood with the trained GMM prior as follows:

$$C = D_{\text{IS}}(Y \parallel BG) - R_2(G). \quad (45)$$

where $R_2(G)$ is the weighted sum of the log-likelihoods of the log-normalized columns of the gains matrix $G$. $R_2(G)$ can be written as follows:

$$R_2(G) = \sum_{i=1}^{2} \eta_i \Gamma(G_i), \quad (46)$$

where $\Gamma(G_i)$ is the log-likelihood for the submatrix $G_i$.

In sparsity and GMM based log-likelihood prior methods, to match between the used update rule for the gains matrix during training and separation, the priors were enforced during both training and separation stages. In sparse NMF we used sparsity constraints during training and separation stages. In regularized NMF with GMM based log-likelihood prior we trained the NMF bases and the prior GMM parameters jointly as argued in [19].

In the sparse NMF case, we obtained best results when $\lambda = 0.0001$ for both sources in the training and separation stages. In the case of enforcing the gains matrix to increase the log-likelihood under GMM prior [19] we obtained the best results when $\eta = 1$ in the training and $\eta = 0.1$ in the separation stage. The number of Gaussian components was $K = 4$ for both sources. It is important to note that, in the case of using MMSE under GMM prior there is no need to enforce prior during training since the uncertainty measurements during training are assumed to be zeros since the training data are clean signals. When the uncertainty is zero, then the regularized NMF in case of MMSE under GMM prior is the same as the NMF cost function, then the update rule for the gains matrix in the training stage is the same as the update rule in regular NMF.

Figs. 2 and 3 show the SNR and SIR for the different type of prior models. The black line shows the separation performance when no prior is used. The red line shows the performance for the case of using sparse NMF. The green line shows the performance when enforcing the gains matrix to increase its likelihood with the prior GMM. The blue line shows the separation performance in the case of using MMSE estimate under GMM prior that is proposed in this paper. As we can see, the proposed method of enforcing prior on the gains matrix using MMSE estimate under GMM prior gives the best performance compared with other methods. The used MMSE estimates prior in this work gives better results than the GMM likelihood method [19] because of the measurements of the uncertainties in the MMSE under GMM case. The uncertainties work as feedback measurements that adjust the need to use the prior based on the amount of distortion in the gains matrix during the separation stage.

Comparing the relative improvements in dB that we obtained in this experiment with the achieved improvements in other works [2,3,6,10] we can see that the improvements in this work can be considered to be high.

4.2. Second experiment

In this experiment, we used general speech data (male and female speakers) from the TIMIT database. We also used general background music signals where different musical instruments are used. We downloaded these music signals from the website [33].

<p>| Table 1 | SNR and SIR in dB for the estimated speech signal with regularization parameters $\alpha_{\text{speech}} = \alpha_{\text{music}} = 1$ and different number of Gaussian mixture components $K$. |
|---------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|</p>
<table>
<thead>
<tr>
<th>SMR dB</th>
<th>No prior</th>
<th>$K = 1$</th>
<th>$K = 4$</th>
<th>$K = 8$</th>
<th>$K = 16$</th>
<th>$K = 32$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-5$</td>
<td>2.88</td>
<td>4.86</td>
<td>3.31</td>
<td>5.71</td>
<td>3.61</td>
<td>6.58</td>
</tr>
<tr>
<td>$0$</td>
<td>5.50</td>
<td>8.70</td>
<td>5.74</td>
<td>9.31</td>
<td>5.90</td>
<td>9.99</td>
</tr>
</tbody>
</table>
the dataset which was collected for the purposes of music/speech discrimination. We used 20 minutes of these music files which are available at [34] for this experiment. For speech data, we used TIMIT dataset for male and female speakers to train general models for the speech signal. We trained a single basis matrix and a single GMM prior model for male and female speakers of the TIMIT database as general models for the speech signals. For the music signals, we used all the available music files for training. We trained the basis matrix of the speech and music data with 128 basis vectors which makes the size of $B_{\text{speech}}$ and $B_{\text{music}}$ matrices to be $257 \times 128$. For training the prior GMMs, we tried with different values for the number of Gaussian mixture components $K$ and we achieved reasonable performance at $K = 256$. The higher value for $K$ in this experiment is because of the variety of the used speech and music signals.

The mixed data was formed by adding random portions of the available 20 minutes music files to 20 speech files from the test data of the TIMIT database at different speech-to-music ratio (SMR) values in dB. The 20 speech files are 10 files from each gender. The audio power levels of each file were found similar to the first experiment. For each SMR value, we obtained 20 mixed utterances. In this experiment, we chose $\alpha_{\text{speech}} = \alpha_{\text{music}} = 1$. We also compared with using different prior models similar to the first experiment.

Figs. 4 and 5 show the SNR and SIR for the different type of prior models. The first observation is that in general the results are much lower than the first experiment which indicates the hardness of the problem in the general speech + music case. As can be seen, the proposed method of enforcing prior on the gains matrix using MMSE estimate under GMM prior gives the best SIR performance for all SMR cases compared with the other methods. For SNR values, at low SMR the proposed MMSE regularization gives better results but at high SMR it seems that just using NMF without any regularization works slightly better than using regularized...
NMF. However, we can see that the output SNR is between 4 and 4.5 dB for all methods when the input SMR is 5 dB in this case. This indicates that all the methods are decreasing the SNR a little bit. However, the SIR results indicate that the MMSE method decreases the contribution of the music in the reconstructed signal much more than the other methods. We believe that SIR is a more important performance measure than SNR when the SMR of the input signal is higher than 5 dB.

5. Conclusion

In this work, we introduced a new regularized NMF algorithm. The NMF solution for the gains matrix was guided by the MMSE estimate under a GMM prior where the uncertainty of the observed mixed signal was learned online from the observed data. The proposed MMSE estimate based regularization method was evaluated on speech–music separation and the achieved results outperform other older regularization methods. The proposed algorithm can be extended for better measurements of the distortion in the observed signal by embedding more parameters in Eq. (14) that can be learned online from the observed signal. The proposed method can also be improved by using dynamic models like hidden Markov model to train the prior information about the gains matrix.

Acknowledgments

This research is partially supported by Turk Telekom Group Research and Development, project entitled “Single Channel Source Separation”, grant number 3014-06, support year 2012.

Appendix A

In this appendix, we derive the MMSE estimate formula and the learning algorithm for the parameter $\Psi$ similar to [27–29]. Assume we have a noisy observation $y$ as shown in the graphical model in Fig. 6, which can be formulated as follows:

$$y = x + e,$$

(47)

where $e$ is the noise term, and $x$ is the unknown underlying correct signal which needs to be estimated under a GMM prior distribution:

$$p(x) = \sum_{k=1}^{K} \omega_k \mathcal{N}(x|\mu_k, \Sigma_k),$$

(48)

the error term $e$ has a Gaussian distribution with zero mean and diagonal covariance matrix $\Psi$:

$$p(e) = \mathcal{N}(e|0, \Psi).$$

(49)

The conditional distribution of $y$ is a Gaussian with mean $x$ and diagonal covariance matrix $\Psi$:

$$p(y|x, k) = p(y|x) = \mathcal{N}(y|x, \Psi).$$

(50)

The distribution of $y$ given the Gaussian component $k$ is a Gaussian with mean $\mu_k$ and diagonal covariance matrix $\Sigma_k + \Psi$:

$$p(y|k) = \mathcal{N}(y|\mu_k, \Sigma_k + \Psi).$$

(51)

The marginal probability distribution of $y$ is a GMM:

$$p(y) = \sum_{k=1}^{K} \omega_k \mathcal{N}(y|\mu_k, \Sigma_k + \Psi),$$

(52)

where the expectations $E(x) = E(y)$, and $E(e) = 0$. Note that, this observation model has some mathematical similarities but different concepts with factor analysis models assuming the load matrix is the identity matrix [27–29].

The MMSE estimate of $x$ can be found by calculating the conditional expectation of $x$ given the observation $y$. Given the Gaussian component $k$, the joint distribution of $x$ and $y$ is a multivariate Gaussian distribution with conditional expectation and conditional variance as follows [27,35]:

$$E(x|y, k) = \mu_k + \Sigma_k \Sigma_k^{-1} (y - \mu_k),$$

(53)

$$\text{var}(x|y, k) = \Sigma_k - \Sigma_k \Sigma_k^{-1} \Sigma_k^T,$$

(54)

we know that

$$\Sigma_k y = \Sigma_k + \Psi,$$

(55)

and

$$\Sigma_{k y} = \text{cov}(x, \Psi) = E(xy^T) - E(x)E(y^T)$$

$$= E[(x(x^T + e^T)] - E(x)E(y^T)$$

$$= E(xx^T) + E(x)e^T - E(x)E(y^T)$$

$$= \text{var}(x) + E(x)e^T - E(x)E(y^T)$$

$$= \text{var}(x) = \Sigma_k.$$

(56)

The conditional expectation given the Gaussian component $k$ of the prior model is

$$E(x|y, k) = \mu_k + \Sigma_k (\Sigma_k + \Psi)^{-1} (y - \mu_k) = \hat{x}_k.$$

(57)

We also can find the following conditional expectation given only the observation $y$ as follows:

$$E(x|y) = \sum_{k=1}^{K} p(k|y) E(x|y, k)$$

$$= \sum_{k=1}^{K} \gamma_k E(x|y, k)$$

$$= \hat{x},$$

(58)

where

$$p(k|y) = \frac{\omega_k p(y|k)}{\sum_{j=1}^{K} \omega_j p(y|j)} = \gamma_k.$$

(59)

From Eqs. (57), (58), (59) we can write the final MMSE estimate of $x$ given the model parameters as follows:

$$\hat{x} = \sum_{k=1}^{K} \gamma_k [\mu_k + \Sigma_k (\Sigma_k + \Psi)^{-1} (y - \mu_k)].$$

(60)

We also need to find the following sufficient statistics to be used in estimating the model parameters:

$$\text{var}(x|y, k) = \Sigma_k - \Sigma_k (\Sigma_k + \Psi)^{-1} \Sigma_k^T,$$

(61)
\[ E(\mathbf{xx}^T | \mathbf{y}, k) = \text{var}(\mathbf{x} | \mathbf{y}, k) + E(\mathbf{x} | \mathbf{y}, k)E(\mathbf{x} | \mathbf{y}, k)^T \]
\[ = \mathbf{S}_k - \mathbf{S}_k(\mathbf{S}_k + \mathbf{\Psi})^{-1}\mathbf{S}_k^T + \hat{\mathbf{x}}_k\mathbf{x}_k^T \]
\[ = \hat{\mathbf{R}}_k, \tag{62} \]

and
\[ E(\mathbf{xx}^T | \mathbf{y}) = \sum_{k=1}^{K} \gamma_k E(\mathbf{xx}^T | \mathbf{y}, k) \]
\[ = \sum_{k=1}^{K} \gamma_k \hat{\mathbf{R}}_k \]
\[ = \hat{\mathbf{R}}. \tag{63} \]

A.1. Parameters learning using the EM algorithm

In the training stage, we assume we have clean data with \( \mathbf{e} = \mathbf{0} \). The prior GMM parameters \( \omega, \mathbf{\mu}, \mathbf{\Sigma} \) are learned as regular GMM models. The only parameter that needs to be estimated is \( \mathbf{\Psi} \), which is learned from the deformed signal \( \Psi_n \) in Section 3.2. The parameter \( \mathbf{\Psi} \) is learned iteratively using maximum likelihood estimation. Given the data points \( q = (q_1, q_2, \ldots, q_n) \), and the GMM parameters, we need to find an estimate for \( \mathbf{\Psi} \). We follow the same procedures as in [27–29].

Let us rewrite the sufficient statistics in Eqs. (59), (57), (60), (62), (63) after replacing \( \mathbf{x} \) with \( \mathbf{z} \) (to avoid confusion between calculating the MMSE estimate and training the model parameters) as follows:
\[ \gamma_{kn} = \frac{\omega_k \mathcal{N}(q_n | \mathbf{\mu}_k, \mathbf{\Sigma}_k + \mathbf{\Psi})}{\sum_{j=1}^{K} \omega_j \mathcal{N}(q_n | \mathbf{\mu}_j, \mathbf{\Sigma}_j + \mathbf{\Psi})}, \tag{64} \]
\[ \hat{z}_{kn} = E_{z_n | q_n, k}(z_n | q_n, k) = \mu_k + \mathbf{\Sigma}_k(\mathbf{\Sigma}_k + \mathbf{\Psi})^{-1}(q_n - \mu_k), \tag{65} \]
\[ \hat{z}_n = E_{z_n | q_n}(z_n | q_n) = \sum_{k=1}^{K} \gamma_{kn} \hat{z}_{kn}, \tag{66} \]
\[ \hat{R}_{kn} = E_{z_n | q_n, k}(\mathbf{z}_n \mathbf{z}_n^T | q_n, k) \]
\[ = \mathbf{S}_k - \mathbf{S}_k(\mathbf{\Sigma}_k + \mathbf{\Psi})^{-1}\mathbf{S}_k^T + \hat{\mathbf{x}}_k\hat{\mathbf{x}}_k^T, \tag{67} \]

and
\[ \hat{R}_n = E_{z_n | q_n}(\mathbf{z}_n \mathbf{z}_n^T | q_n) = \sum_{k=1}^{K} \gamma_{kn} \hat{R}_{kn}. \tag{68} \]

We define \( q = (q_1, q_2, \ldots, q_n), \mathbf{z} = (z_1, z_2, \ldots, z_n), \) and \( \mathcal{K} = \{k_1, k_2, \ldots, k_n\}. \) Given the generative model in Fig. 7, the complete log-likelihood can be written in a product form as follows:
\[ I(\mathbf{q}, \mathbf{z}, \mathcal{K}|\theta) = \log \prod_{n=1}^{N} \prod_{k=1}^{K} p(k_n) p(z_n | k_n) p(q_n | z_n, k_n) \]
\[ = \log \prod_{n=1}^{N} \prod_{k=1}^{K} [p(k)p(z_n | k)p(q_n | z_n, k)]^{\delta_{k_1,k_2}} \]
\[ = \log \prod_{n=1}^{N} \prod_{k=1}^{K} [\omega_k \mathcal{N}(q_n | \mathbf{\mu}_k, \mathbf{\Sigma}_k) \mathcal{N}(z_n | \mathbf{\mu}_k, \mathbf{\Sigma}_k) \mathcal{N}(q_n | z_n, \mathbf{\Psi})]^{\delta_{k_1,k_2}}, \tag{69} \]

where \( \theta = (\mathbf{\mu}, \mathbf{\Sigma}, \omega, \mathbf{\Psi}), \delta_{k_1,k_2} = \left\{ \begin{array}{ll} 1 & k_1 = k_2 \\ 0 & \text{else} \end{array} \right\}. \]
\[ C(G) = D_{\text{KL}}(V \mid\mid B G) + \alpha L(G), \] 

where

\[ L(G) = \sum_{n=1}^{N} \left( \frac{\|g_n - \exp(f(x_n))\|^2}{\|g_n\|^2} \right), \]

\[ f(g_n) = \sum_{k=1}^{K} \gamma_k \left[ \mu_k + \Sigma_k (\Sigma + \Psi)^{-1} \left( 2 \log \frac{g_n}{\|g_n\|^2} - \mu_k \right) \right], \]

and

\[ \gamma_k = \left( \frac{\omega_k \mathcal{N}(\log \frac{g_n}{\|g_n\|^2} | \mu_k, \Sigma_k + \Psi)}{\sum_{j=1}^{K} \omega_j \mathcal{N}(\log \frac{g_n}{\|g_n\|^2} | \mu_j, \Sigma_j + \Psi)} \right). \]

Since the training data for the GMM models are the logarithm of the normalized vectors, then the mean vectors of the GMM are always not positive, also the values of \( \log \frac{g_n}{\|g_n\|^2} \) are always not positive, and \( g_n \) is always nonnegative.

Let \( g_n = \mathbf{x} \), and its component \( a \) is \( g_{na} = x_n \), and \( f(g_n) = f(\mathbf{x}) \). We can write the constraint in Eq. (80) as:

\[ L(\mathbf{x}) = \left( x - \exp(f(\mathbf{x})) \right)^2. \]

The \( a \) component of the gradient of \( L(\mathbf{x}) \) is

\[ \nabla L(\mathbf{x}) = 2 \left( \frac{x}{\|x\|^2} - \exp(f(x)) \right) \times \left( 1 - \frac{x^2}{\|x\|^2} - \nabla f(x) \exp(f(x)) \right) = \nabla L(x_a), \]

which can be written as a difference of two positive terms

\[ \nabla L(x_a) = \nabla^+ L(x_a) - \nabla^- L(x_a). \]

The component \( a \) of the gradient of \( f(\mathbf{x}) \) can be written as a difference of two positive terms:

\[ \frac{\partial f(\mathbf{x})}{\partial x_a} = \nabla^+ f(x_a) - \nabla^- f(x_a). \]

The component \( a \) of the gradient of \( L(\mathbf{x}) \) in Eq. (85) can be written as:

\[ \nabla^+ L(x_a) = 2 \left\{ \frac{x_a}{\|x\|^2} \left( \frac{1}{\|x\|^2} + \exp(f(x_a)) \nabla^- f(x_a) \right) + \exp(f(x_a)) \left( \frac{x_a^2}{\|x\|^2} + \exp(f(x_a)) \nabla^+ f(x_a) \right) \right\}. \]

and

\[ \nabla^- L(x_a) = 2 \left\{ \frac{x_a}{\|x\|^2} \left( \frac{x_a^2}{\|x\|^2} + \exp(f(x_a)) \nabla^- f(x_a) \right) + \exp(f(x_a)) \left( \frac{1}{\|x\|^2} + \exp(f(x_a)) \nabla^+ f(x_a) \right) \right\}. \]

We need to find the values of \( \nabla^+ f(x_a) \) and \( \nabla^- f(x_a) \). Note that, the term \( \Sigma_k (\Sigma + \Psi)^{-1} \) forms a diagonal matrix.

Let

\[ H(x_a) = \mu_{ka} + \Sigma_{ka} (\Sigma + \Psi)^{-1} \left( \log \frac{x_a}{\|x\|^2} - \mu_k \right). \]

then \( f(\mathbf{x}) \) in Eq. (81) can be written as:
\[ f(x) = \sum_{k=1}^{K} \gamma_k(x) H(x). \]  

The gradient of \( f(x) \) in Eq. (90) can be written as:

\[ \nabla f(x) = \sum_{k=1}^{K} [\gamma_k(x) \nabla H(x) + H(x) \nabla \gamma_k(x)]. \]  

where

\[ \gamma_k(x) = \left[ \frac{\omega_k N(\log \frac{x}{\|x\|_2}, \mu_k, \Sigma_k + \Psi)}{\sum_{j=1}^{K} \pi_j N(\log \frac{x}{\|x\|_2}, \mu_j, \Sigma_j + \Psi)} \right] = \frac{M_k(x)}{N_k(x)}. \]  

We can also write the gradient components of \( H(x) \) and \( \gamma_k(x) \) as a difference of two positive terms:

\[ \nabla H(x) = \nabla^+ H(x) - \nabla^- H(x), \]  

and

\[ \nabla \gamma_k(x) = \nabla^+ \gamma_k(x) - \nabla^- \gamma_k(x). \]  

The gradient of \( f(x) \) in Eqs. (86), (91) can be written as:

\[ \nabla^+ f(x) = \sum_{k=1}^{K} \left[ \gamma_k(x) \nabla^+ H(x) + H(x) \nabla^+ \gamma_k(x) \right] \]  

\[ + H(x) \nabla^- \gamma_k(x), \]  

\[ \nabla^- f(x) = \sum_{k=1}^{K} \left[ \gamma_k(x) \nabla^- H(x) + H(x) \nabla^+ \gamma_k(x) \right] \]  

\[ + H(x) \nabla^- \gamma_k(x), \]  

where

\[ \nabla^+ H(x) = \Sigma_{kaa} (\Sigma_{kaa} + \Psi_{aa})^{-1} \frac{1}{x_a}, \]  

\[ \nabla^- H(x) = \Sigma_{kaa} (\Sigma_{kaa} + \Psi_{aa})^{-1} \frac{x_a}{\|x\|_2^2}, \]  

and \( H(x) \) can be written as a difference of two positive terms:

\[ H(x) = H^+(x) - H^-(x), \]  

where

\[ H^+(x) = -\Sigma_{kaa} (\Sigma_{kaa} + \Psi_{aa})^{-1} \mu_{ka}, \]  

and

\[ H^-(x) = -\left[ \mu_{ka} + \Sigma_{kaa} (\Sigma_{kaa} + \Psi_{aa})^{-1} \log \frac{x_a}{\|x\|_2} \right]. \]  

We can rewrite \( \gamma_k(x) \) in Eq. (92) as:

\[ \gamma_k(x) = \frac{M_k(x)}{N_k(x)}, \]  

note that \( \gamma_k(x), M_k(x), N_k(x) \geq 0 \).

The component \( a \) of the gradient of \( \gamma_k(x) \) can be written as:

\[ \nabla \gamma_k(x) = \frac{N_k(x) \nabla M_k(x) - M_k(x) \nabla N_k(x)}{N_k(x)}. \]  

We can write the gradients of \( M_k(x) \) and \( N_k(x) \) as a difference of two positive terms:

\[ \nabla M_k(x) = \nabla^+ M_k(x) - \nabla^- M_k(x), \]  

and

\[ \nabla N_k(x) = \sum_{k=1}^{K} \nabla^+ M_k(x) - \sum_{k=1}^{K} \nabla^- M_k(x). \]  

The gradient of \( \gamma_k(x) \) in Eq. (94) can be written as:

\[ \nabla^+ \gamma_k(x) = \frac{N_k(x) \nabla M^+_k(x) + M_k(x) \sum_{k=1}^{K} \nabla^- M_k(x)}{N_k(x)}, \]  

\[ \nabla^- \gamma_k(x) = \frac{N_k(x) \nabla M^-_k(x) + M_k(x) \sum_{k=1}^{K} \nabla^+ M_k(x)}{N_k(x)}, \]  

where

\[ \nabla^+ M_k(x) = \frac{M_k(x) (\Sigma_{kaa} + \Psi_{aa})^{-1} \left[ -1 \frac{\log x_a}{\|x\|_2} - \frac{\mu_{ka} x_a}{\|x\|_2^2} \right]}{M_k(x)}. \]  

\[ \nabla^- M_k(x) = \frac{M_k(x) (\Sigma_{kaa} + \Psi_{aa})^{-1} \left[ -\mu_{ka} \frac{x_a}{\|x\|_2^2} \right]}{M_k(x)}. \]  

After finding \( \nabla^+ \gamma_k(x) \), and \( \nabla^- \gamma_k(x) \) from Eqs. (106), (107), and \( \nabla^+ H(x) \), and \( \nabla^- H(x) \) from Eqs. (97), (98), we can find the gradients \( \nabla^+ f(x) \), and \( \nabla f(x) \) in Eqs. (95), (96), which complete our solution for \( \nabla^+ L(x) \), and \( \nabla^- L(x) \) in Eqs. (87), (88).

References


Emad M. Grais is a faculty member at Helwan University. Previously he was a postdoctoral fellow at Sabanci University in Istanbul, Turkey. He received his B.S. degree in Communication and Electronics Engineering from Helwan University, Egypt, in 2004, and Ph.D. degree in Electronics Engineering from Sabanci University, Turkey in 2013. His research interests are single channel source separation, speech enhancement, and machine learning for signal processing.

Hakan Erdogan is an associate professor at Sabanci University in Istanbul, Turkey. He received his B.S. degree in Communication and Electronics Engineering in 1993 from METU, Ankara and his M.S. and Ph.D. degrees in Electrical Engineering from the University of Michigan, Ann Arbor in 1995 and 1999 respectively. He was with the Human Language Technologies group at IBM T.J. Watson Research Center, NY between 1999 and 2002. He has been with Sabanci University since 2002. His research interests are in developing and applying probabilistic methods and algorithms for multimedia information extraction.