Non-Negative Matrix Factorization with Sparsity Learning for Single Channel Audio Source Separation

Bin Gao and W.L. Woo
School of Electrical and Electronic Engineering, Newcastle University, England, United Kingdom

1. Introduction

1.1 Single channel source separation (SCSS)

In this chapter, the special case of instantaneous underdetermined source separation problem termed as single channel source separation (SCSS) is focused. In general case and for many practical applications (e.g. audio processing) only one-channel recording is available and in such cases conventional source separation techniques are not appropriate. This leads to the SCSS research area where the problem can be simply treated as one observation instantaneous mixed with several unknown sources:

$$y(t)=\sum_{i=1}^{N_s} x_i(t) \tag{1}$$

where $i=1,\ldots,N_s$ denotes number of sources and the goal is to estimate the sources $x_i(t)$ when only the observation signal $y(t)$ is available. This is an underdetermined system of equation problem. Recently, new advances have been achieved in SCSS and this can be categorized either as *supervised* SCSS methods or *unsupervised* SCSS methods. For *supervised* SCSS methods, the probabilistic models of the source are trained as a prior knowledge by using some or the entire source signals. The mixture is first transformed into an appropriate representation, in which the source separation is performed. The source models are either constructed directly based on knowledge of the signal sources, or by learning from training data (e.g. using Gaussian mixture model construct source models either directly based on knowledge of signal sources, or by learning from isolated training data). In the inference stage, the models and data are combined to yield estimates of the sources. This category predominantly includes the frequency model-based SCSS methods [1, 2] where the prior bases are modeled in time-frequency domain (e.g. spectrogram or power spectrogram), and the underdetermined-ICA time model-based SCSS method [3] which the prior bases are modeled in time domain. For *unsupervised* SCSS methods, this denotes the separation of completely unknown sources without using additional training information. These methods typically rely on the assumption that the sources are non-redundant, and the methods are based on, for example, decorrelation, statistical independence, or the minimum description length principle. This category includes several widely used methods: Firstly, the CASA-based *unsupervised* SCSS methods [4] whose goal is to replicate the process of human auditory system by exploiting signal processing approaches (e.g. notes in music recordings) and grouping them into auditory streams using psycho-acoustical cues. Secondly, the subspace technique based *unsupervised* SCSS methods using NMF [5, 6] or independent subspace analysis (ISA) [7] which usually factorizes the spectrogram of the input signal into elementary components. Of special interest, EMD [8] based *unsupervised* SCSS methods which can separate audio mixed signal in time domain and recover sources by combing other data analysis tools, e.g. independent component analysis (ICA) [9] or principle component analysis (PCA).

1.2 Unsupervised SCSS using NMF

In this book chapter, we propose a new NMF method for solving *unsupervised* SCSS problem. In a conventional NMF, given a data matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_L] \in \Re_+^{K \times L}$ with $\mathbf{Y}_{k,l} > 0$, NMF factorizes this matrix into a product of two non-negative matrices:

$$\mathbf{Y} \approx \mathbf{DH} \tag{2}$$

where $\mathbf{D} \in \Re_+^{K \times \ell}$ and $\mathbf{H} \in \Re_+^{\ell \times L}$ where K and L represent the total number of rows and columns in matrix $\mathbf{Y}$, respectively. If ℓ is chosen to be $\ell = L$, no benefit is achieved at all. Thus the idea is to determine $\ell < L$ so that the matrix $\mathbf{D}$ can be compressed and reduced to its integral components such as $\mathbf{D}_{K \times \ell}$ is a matrix containing a set of dictionary vectors, and $\mathbf{H}_{\ell \times L}$ is an encoding matrix that describes the amplitude of each dictionary vector at each time point. A popular approach to solve the NMF optimization problem is the multiplicative update (MU) algorithm by Lee and Seung [10]. The MU update rule for Least square (LS) distance is given by:

$$\mathbf{D} \leftarrow \mathbf{D} \bullet \frac{\mathbf{YH}^{\mathrm{T}}}{\mathbf{DHH}^{\mathrm{T}}} \text{ and } \mathbf{H} \leftarrow \mathbf{H} \bullet \frac{\mathbf{D}^{\mathrm{T}}\mathbf{Y}}{\mathbf{D}^{\mathrm{T}}\mathbf{DH}} \tag{3}$$

Multiplicative update-based families of parameterized cost functions such as the Beta divergence [11], and Csiszar's divergences [12] have also been presented as well. A sparseness constraint [13, 14] can be added to the cost function, and this can be achieved by regularization using the L_1-norm. Here, 'sparseness' refers to a representational scheme where only a few units (out of a large population) are effectively used to represent typical data vectors [15]. In effect, this implies most units taking values close to zero while only few take significantly non-zero values. Several other types of prior over $\mathbf{D}$ and $\mathbf{H}$ can be defined e.g. in [16, 17], it is assumed that the prior of $\mathbf{D}$ and $\mathbf{H}$ satisfy the exponential density and the prior for the noise variance is chosen as an inverse gamma density. In [18], Gaussian distributions are chosen for both $\mathbf{D}$ and $\mathbf{H}$. The model parameters and hyperparameters are adapted by using the Markov chain Monte Carlo (MCMC) [19-21]. In all cases, a fully Bayesian treatment is applied to approximate inference for both model parameters and hyperparameters. While these approaches increase the accuracy of matrix factorization, it only works efficient when large sample dataset is available. Moreover, it consumes significantly high computational complexity at each iteration to adapt the parameters and its hyperparameters. Regardless of the cost function and sparseness constraint being used, the standard NMF or SNMF models [22] are only satisfactory for solving source separation provided that the spectral frequencies of the analyzed audio signal do not change over time. However, this is not the case for many realistic audio signals. As a result, the spectral dictionary obtained via the NMF or SNMF decomposition is not adequate to capture the temporal dependency of the frequency patterns within the signal. The recently developed two-dimensional sparse NMF deconvolution (SNMF2D) model [23, 24] extends the NMF model to be a two-dimensional convolution of $\mathbf{D}$ and $\mathbf{H}$ where the spectral dictionary and temporal code are optimized using the least square cost function with sparse penalty:

$$C_{LS}: \ \frac{1}{2}\sum_{k,l}(\mathbf{Y}_{k,l}-\tilde{\mathbf{Z}}_{k,l})^{2}+\lambda f(\mathbf{H}) \tag{4}$$

for $\forall k \in K, \forall l \in L$ where $\tilde{\mathbf{Z}} = \sum_{\tau,\phi} \overset{\downarrow\phi}{\tilde{\mathbf{D}}^{\tau}} \overset{\rightarrow\tau}{\mathbf{H}^{\phi}}$, $\tilde{\mathbf{D}}^{\tau}_{k,d} = \mathbf{D}^{\tau}_{k,d} \Big/ \sqrt{\sum_{\tau,k}(\mathbf{D}^{\tau}_{k,d})^{2}}$ and $f(\mathbf{H})$ can be any function with positive derivative such as $L_{\alpha} - norm\,(\alpha > 0)$ given by $f(\mathbf{H}) = \|\mathbf{H}\|_{\alpha} = \left(\sum_{\phi,d,l}\left|\mathbf{H}^{\phi}_{d,l}\right|^{\alpha}\right)^{1/\alpha}$. Here $\overset{\downarrow\phi}{\tilde{\mathbf{D}}^{\tau}}$ denotes the downward shift which moves each element in the matrix $\tilde{\mathbf{D}}^{\tau}$ down by ϕ rows, and $\overset{\rightarrow\tau}{\mathbf{H}^{\phi}}$ denotes the right shift which moves each element in the matrix $\mathbf{H}^{\phi}$ to the right by τ columns. The SNMF2D is effective in single channel audio source separation (SCASS) because it is able to capture both the temporal structure and the pitch change of an audio source. However, the drawbacks of SNMF2D originate from its lack of a generalized criterion for controlling the sparsity of $\mathbf{H}$. In practice, the sparsity parameter is set manually. When SNMF2D imposes uniform sparsity on all temporal codes, this is equivalent to enforcing each temporal code to be identical to a fixed distribution according to the selected sparsity parameter. In addition, by assigning the fixed distribution onto each individual code, this is equivalent to constraining all codes to be stationary. However, audio signals are non-stationary in the TF domain and have different temporal structure and sparsity. Hence, they cannot be realistically enforced by a fixed probability distribution. These characteristics are even more pronounced between different types of audio signals. In addition, since the SNMF2D introduces many temporal shifts, this will result in more temporal codes to deviate from the fixed distribution. In such situation, the obtained factorization will invariably suffer from either under- or over-sparseness which subsequently lead to ambiguity in separating the audio mixture. Thus, the above suggests that the present form of SNMF2D is still technically lacking and is not readily suited for SCASS especially mixtures involving different types of audio signals.

In this chapter, an adaptive sparsity two-dimensional non-negative matrix factorization is proposed. The proposed model allows: (*i*) overcomplete representation by allowing many spectral and temporal shifts which are not inherent in the NMF and SNMF models. Thus, imposing sparseness is necessary to give unique and realistic representations of the non-stationary audio signals. Unlike the SNMF2D, our model imposes sparseness on $\mathbf{H}$ element-wise so that *each individual code* has its own distribution. Therefore, the sparsity parameter can be individually optimized for each code. This overcomes the problem of under- and over-sparse factorization. (*ii*) Each sparsity parameter in our model is learned and adapted as part of the matrix factorization. This bypasses the need of manual selection as in the case of SNMF2D. The proposed method is tested on the application of single channel music separation and the results show that our proposed method can give superior separation performance.

The chapter is organized as follows: In Section II, the new model is derived. Experimental results coupled with a series of performance comparison with other NMF techniques are presented in Section III. Finally, Section IV concludes the paper.

2. Adaptive sparsity two-dimensional non-negative matrix factorization

In this section, we derive a new factorization method termed as the *adaptive sparsity* two-dimensional non-negative matrix factorization. The model is given by

$$\mathbf{Y} = \sum_{\tau=0}^{\tau_{\max}} \sum_{\phi=0}^{\phi_{\max}} \overset{\downarrow\phi}{\mathbf{D}^{\tau}} \overset{\rightarrow\tau}{\mathbf{H}^{\phi}} + \mathbf{V} = \sum_{d=1}^{d_{\max}} \sum_{\tau=0}^{\tau_{\max}} \sum_{\phi=0}^{\phi_{\max}} \overset{\downarrow\phi}{\mathbf{D}_d^{\tau}} \overset{\rightarrow\tau}{\mathbf{H}_d^{\phi}} + \mathbf{V} \tag{5}$$

where $\mathbf{H}^{\phi} \sim p\left(\mathbf{H}^{\phi} \mid \boldsymbol{\lambda}^{\phi}\right) = \prod_{d=1}^{d_{\max}} \prod_{l=1}^{l_{\max}} \lambda_{d,l}^{\phi} \exp\left(-\lambda_{d,l}^{\phi} \mathbf{H}_{d,l}^{\phi}\right)$. In (5), it is worth pointing out that *each individual element* in $\mathbf{H}^{\phi}$ is constrained to an exponential distribution with independent decay parameter $\lambda_{d,l}^{\phi}$. Here, $\mathbf{D}_d^{\tau}$ is the dth column of $\mathbf{D}^{\tau}$, $\mathbf{H}_d^{\phi}$ is the dth row of $\mathbf{H}^{\phi}$ and $\mathbf{V}$ is assumed to be independently and identically distributed (i.i.d.) as Gaussian distribution with noise having variance σ^2. The terms $d_{\max}$, $\tau_{\max}$, $\phi_{\max}$ and $l_{\max}$ are the maximum number of columns in $\mathbf{D}^{\tau}$, τ shifts, ϕ shifts and column length in $\mathbf{Y}$, respectively. This is in contrast with the conventional SNMF2D where $\lambda_{d,l}^{\phi}$ is simply set to a fixed constant i.e. $\lambda_{d,l}^{\phi} = \lambda$ for all d, l, ϕ. Such setting imposes uniform constant sparsity on all temporal codes $\mathbf{H}^{\phi}$ which enforces each temporal code to be identical to a fixed distribution according to the selected constant sparsity parameter. The consequence of this uniform constant sparsity has already been discussed in Section I. In Section III, we will present the details of the sparsity analysis for source separation and evaluate its performance against with other existing methods.

2.1 Formulation of the proposed adaptive sparsity NMF2D

To facilitate such spectral dictionaries with adaptive sparse coding, we first define $\mathbf{D} = \begin{bmatrix} \mathbf{D}^0 & \mathbf{D}^1 & \cdots & \mathbf{D}^{\tau_{\max}} \end{bmatrix}$, $\mathbf{H} = \begin{bmatrix} \mathbf{H}^0 & \mathbf{H}^1 & \cdots & \mathbf{H}^{\phi_{\max}} \end{bmatrix}$ and $\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}^1 \boldsymbol{\lambda}^2 \cdots \boldsymbol{\lambda}^{\phi_{\max}} \end{bmatrix}$, and then choose a prior distribution $p(\mathbf{D}, \mathbf{H})$ over the factors $\{\mathbf{D}, \mathbf{H}\}$ in the analysis equation. The posterior can be found by using Bayes' theorem as

$$p\left(\mathbf{D}, \mathbf{H} \middle| \mathbf{Y}, \sigma^2, \boldsymbol{\lambda}\right) = \frac{p\left(\mathbf{Y} \middle| \mathbf{D}, \mathbf{H}, \sigma^2\right) p\left(\mathbf{D}, \mathbf{H} \middle| \boldsymbol{\lambda}\right)}{P(\mathbf{Y})} \tag{6}$$

where the denominator is constant and therefore, the log-posterior can be expressed as

$$\log p\left(\mathbf{D},\mathbf{H}\middle|\mathbf{Y},\sigma^2,\boldsymbol{\lambda}\right)=\log p\left(\mathbf{Y}\,|\,\mathbf{D},\mathbf{H},\sigma^2\right)+\log p\left(\mathbf{D},\mathbf{H}\middle|\boldsymbol{\lambda}\right)+\text{const} \tag{7}$$

where $'\,\text{const}\,'$ denotes constant. The likelihood of the observations given $\mathbf{D}$ and $\mathbf{H}$ can be written[1] as:

$$p\left(\mathbf{Y}\,|\,\mathbf{D},\mathbf{H},\sigma^2\right)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\left\|\mathbf{Y}-\sum_{d}\sum_{\tau}\sum_{\phi}\overset{\downarrow\phi}{\mathbf{D}}{}^{\tau}_{d}\overset{\rightarrow\tau}{\mathbf{H}}{}^{\phi}_{d}\right\|_F^2\Big/2\sigma^2\right] \tag{8}$$

where $\|.\|_F$ denotes the Frobenius norm. The second term in (7) consists of the prior distribution of $\mathbf{D}$ and $\mathbf{H}$ where they are jointly independent. Each element of $\mathbf{H}$ is constrained to be exponential distributed with independent decay parameters, namely,

$$p(\mathbf{H}\,|\,\boldsymbol{\lambda})=\prod_{\phi}\prod_{d}\prod_{l}\lambda^{\phi}_{d,l}\exp\left(-\lambda^{\phi}_{d,l}\mathbf{H}^{\phi}_{d,l}\right)\ \text{so that}\ f(\mathbf{H})=\sum_{\phi,d,l}\lambda^{\phi}_{d,l}\mathbf{H}^{\phi}_{d,l} \tag{9}$$

Hence, the negative log likelihood serves as the cost function defined as:

$$\begin{aligned}L &\propto \frac{1}{2\sigma^2}\left\|\mathbf{Y}-\sum_{d}\sum_{\tau}\sum_{\phi}\overset{\downarrow\phi}{\mathbf{D}}{}^{\tau}_{d}\overset{\rightarrow\tau}{\mathbf{H}}{}^{\phi}_{d}\right\|_F^2+f(\mathbf{H})\\ &=\frac{1}{2\sigma^2}\left\|\mathbf{Y}-\sum_{d}\sum_{\tau}\sum_{\phi}\overset{\downarrow\phi}{\mathbf{D}}{}^{\tau}_{d}\overset{\rightarrow\tau}{\mathbf{H}}{}^{\phi}_{d}\right\|_F^2+\sum_{\phi,d,l}\lambda^{\phi}_{d,l}\mathbf{H}^{\phi}_{d,l}\end{aligned} \tag{10}$$

The sparsity term $f(\mathbf{H})$ forms the L_1-norm regularization which is used to resolve the ambiguity by forcing all structure in $\mathbf{H}$ onto $\mathbf{D}$. Therefore, the sparseness of the solution in (9) is highly dependent on the regularization parameter $\lambda^{\phi}_{d,l}$.

2.1.1 Estimation of the dictionary and temporal code

In (10), each spectral dictionary was constrained to unit length. This can be easily satisfied by normalizing each spectral dictionary according to $\tilde{\mathbf{D}}^{\tau}_{k,d}=\mathbf{D}^{\tau}_{k,d}\Big/\sqrt{\sum_{\tau,k}(\mathbf{D}^{\tau}_{k,d})^2}$ for all $d\in[1,\ldots,d_{\max}]$. With this normalization, the two-dimensional convolution of the spectral dictionary and temporal codes is now represented as $\tilde{\mathbf{Z}}=\sum_{d}\sum_{\tau}\sum_{\phi}\overset{\downarrow\phi}{\tilde{\mathbf{D}}}{}^{\tau}_{d}\overset{\rightarrow\tau}{\mathbf{H}}{}^{\phi}_{d}$. The derivatives of (10) corresponding to $\mathbf{D}^{\tau}$ and $\mathbf{H}^{\phi}$ of the adaptive sparsity factorization model are given by:

[1] To avoid cluttering the notation, we shall remove the upper limits from the summation terms. The upper limits can be inferred from (5).

$$\mathbf{H}^{\phi} \leftarrow \mathbf{H}^{\phi} \bullet \frac{\sum_{\tau} \overset{\downarrow \phi^{\mathrm{T}}}{\tilde{\mathbf{D}}^{\tau}} \overset{\leftarrow \tau}{\mathbf{Y}}}{\sum_{\tau} \overset{\downarrow \phi^{\mathrm{T}}}{\tilde{\mathbf{D}}^{\tau}} \overset{\leftarrow \tau}{\tilde{\mathbf{Z}}} + \boldsymbol{\lambda}^{\phi}} \tag{11}$$

$$\mathbf{D}^{\tau} \leftarrow \tilde{\mathbf{D}}^{\tau} \bullet \frac{\sum_{\phi} \overset{\uparrow \phi \to \tau^{\mathrm{T}}}{\mathbf{Y}\,\mathbf{H}^{\phi}} + \tilde{\mathbf{D}}^{\tau} diag\left(\sum_{\tau} \mathbf{1} \left(\left(\overset{\uparrow \phi \to \tau^{\mathrm{T}}}{\tilde{\mathbf{Z}}\,\mathbf{H}^{\phi}} \right) \bullet \tilde{\mathbf{D}}^{\tau} \right) \right)}{\sum_{\phi} \overset{\uparrow \phi \to \tau^{\mathrm{T}}}{\tilde{\mathbf{Z}}\,\mathbf{H}^{\phi}} + \tilde{\mathbf{D}}^{\tau} diag\left(\sum_{\tau} \mathbf{1} \left(\left(\overset{\uparrow \phi \to \tau^{\mathrm{T}}}{\mathbf{Y}\,\mathbf{H}^{\phi}} \right) \bullet \tilde{\mathbf{D}}^{\tau} \right) \right)} \quad \text{where} \quad \tilde{\mathbf{D}}_{k,d}^{\tau} = \frac{\mathbf{D}_{k,d}^{\tau}}{\sqrt{\sum_{\tau,k} \left(\mathbf{D}_{k,d}^{\tau} \right)^{2}}} \tag{12}$$

In (11), superscript $'\mathbf{T}'$ denotes matrix transpose, $'\bullet'$ is the element wise product and $diag(\cdot)$ denotes a matrix with the argument on the diagonal. The column vectors of $\mathbf{D}^{\tau}$ will be factor-wise normalized to unit length.

2.1.2 Estimation of the adaptive sparsity parameter

Since $\overset{\to \tau}{\mathbf{H}^{\phi}}$ is obtained directly from the original sparse code matrix $\overset{\to 0}{\mathbf{H}^{\phi}}$, it suffices to compute just for the regularization parameters associated with $\overset{\to 0}{\mathbf{H}^{\phi}}$. Therefore, we can set the cost function in (10) with $\tau_{\max} = 0$ as

$$F(\mathbf{H}) = \frac{1}{2\sigma^{2}} \left\| Vec(\mathbf{Y}) - \sum_{\phi=0}^{\phi_{\max}} \left(\mathbf{I} \otimes \overset{\downarrow \phi}{\mathbf{D}} \right) Vec\left(\mathbf{H}^{\phi} \right) \right\|_{F}^{2} + \sum_{\phi=0}^{\phi_{\max}} \left(\underline{\boldsymbol{\lambda}}^{\phi} \right)^{\mathbf{T}} Vec\left(\mathbf{H}^{\phi} \right) \tag{13}$$

with $Vec(\cdot)$ represents the column vectorization, $'\otimes'$ is the Kronecker product, and $\mathbf{I}$ is the identity matrix. Defining the following terms:

$$\begin{gathered} \underline{\mathbf{y}} = Vec(\mathbf{Y}) \quad , \quad \overline{\mathbf{D}} = \left[\mathbf{I} \otimes \overset{\downarrow 0}{\mathbf{D}} \; \mathbf{I} \otimes \overset{\downarrow 1}{\mathbf{D}} \cdots \mathbf{I} \otimes \overset{\downarrow \phi_{\max}}{\mathbf{D}} \right], \\ \underline{\mathbf{h}} = \begin{bmatrix} Vec(\mathbf{H}^{0}) \\ Vec(\mathbf{H}^{1}) \\ \vdots \\ Vec(\mathbf{H}^{\phi_{\max}}) \end{bmatrix} \quad , \quad \underline{\boldsymbol{\lambda}} = \begin{bmatrix} \underline{\boldsymbol{\lambda}}^{0} \\ \underline{\boldsymbol{\lambda}}^{1} \\ \vdots \\ \underline{\boldsymbol{\lambda}}^{\phi_{\max}} \end{bmatrix} \quad , \quad \underline{\boldsymbol{\lambda}}^{\phi} = \begin{bmatrix} \lambda_{1,1}^{\phi} \\ \lambda_{2,1}^{\phi} \\ \vdots \\ \lambda_{d_{\max},l_{\max}}^{\phi} \end{bmatrix} \end{gathered} \tag{14}$$

Thus, (13) can be rewritten in terms of $\underline{\mathbf{h}}$ as

$$F(\underline{\mathbf{h}}) = \frac{1}{2\sigma^{2}} \left\| \underline{\mathbf{y}} - \overline{\mathbf{D}} \underline{\mathbf{h}} \right\|_{F}^{2} + \underline{\boldsymbol{\lambda}}^{\mathbf{T}} \underline{\mathbf{h}} \tag{15}$$

Note that $\underline{\mathbf{h}}$ and $\underline{\boldsymbol{\lambda}}$ are vectors of dimension $R\times 1$ where $R = d_{\max}\times l_{\max}\times(\phi_{\max}+1)$. To determine $\underline{\boldsymbol{\lambda}}$, we use the Expectation-Maximization (EM) algorithm and treat $\underline{\mathbf{h}}$ as the hidden variable where the log-likelihood function can be optimized with respect to $\underline{\boldsymbol{\lambda}}$. Using the Jensen's inequality, it can be shown that for any distribution $Q(\underline{\mathbf{h}})$, the log-likelihood function satisfies the following [25-27]:

$$\ln p\left(\underline{\mathbf{y}}\,|\,\underline{\boldsymbol{\lambda}},\overline{\mathbf{D}},\sigma^2\right) \geq \int Q\left(\underline{\mathbf{h}}\right)\ln\left(\frac{p\left(\underline{\mathbf{y}},\underline{\mathbf{h}}\,|\,\underline{\boldsymbol{\lambda}},\overline{\mathbf{D}},\sigma^2\right)}{Q\left(\underline{\mathbf{h}}\right)}\right)d\underline{\mathbf{h}} \tag{16}$$

One can easily check that the distribution that maximizes the right hand side of (16) is given by $Q\left(\underline{\mathbf{h}}\right)=p\left(\underline{\mathbf{h}}\,|\,\underline{\mathbf{y}},\underline{\boldsymbol{\lambda}},\overline{\mathbf{D}},\sigma^2\right)$ which is the posterior distribution of $\underline{\mathbf{h}}$. In this paper, we represent the posterior distribution in the form of Gibbs distribution:

$$Q\left(\underline{\mathbf{h}}\right)=\frac{1}{Z_h}\exp\left[-F\left(\underline{\mathbf{h}}\right)\right] \quad \text{where} \quad Z_h=\int \exp\left[-F\left(\underline{\mathbf{h}}\right)\right]d\underline{\mathbf{h}} \tag{17}$$

The functional form of the Gibbs distribution in (17) is expressed in terms of $F\left(\underline{\mathbf{h}}\right)$ and this is crucial as it will enable us to simplify the variational optimization of $\underline{\boldsymbol{\lambda}}$. The maximum likelihood estimation of $\underline{\boldsymbol{\lambda}}$ can be expressed by

$$\begin{aligned}\underline{\boldsymbol{\lambda}}^{ML} &= \arg\max_{\underline{\boldsymbol{\lambda}}} \ln p\left(\underline{\mathbf{y}}\,|\,\underline{\boldsymbol{\lambda}},\overline{\mathbf{D}},\sigma^2\right)\\ &= \arg\max_{\underline{\boldsymbol{\lambda}}} \int Q\left(\underline{\mathbf{h}}\right)\ln p\left(\underline{\mathbf{h}}\,|\,\underline{\boldsymbol{\lambda}}\right)d\underline{\mathbf{h}}\end{aligned} \tag{18}$$

Similarly,

$$\begin{aligned}\sigma^2_{ML} &= \arg\max_{\sigma^2} \int Q\left(\underline{\mathbf{h}}\right)\left(\ln p\left(\underline{\mathbf{y}}\,|\,\underline{\mathbf{h}},\sigma^2,\overline{\mathbf{D}}\right)+\ln p\left(\underline{\mathbf{h}}\,|\,\underline{\boldsymbol{\lambda}}\right)\right)d\underline{\mathbf{h}}\\ &= \arg\max_{\sigma^2} \int Q\left(\underline{\mathbf{h}}\right)\ln p\left(\underline{\mathbf{y}}\,|\,\underline{\mathbf{h}},\sigma^2,\overline{\mathbf{D}}\right)d\underline{\mathbf{h}}\end{aligned} \tag{19}$$

Since each element of $\mathbf{H}$ is constrained to be exponential distributed with independent decay parameters, this gives $p\left(\underline{\mathbf{h}}\,|\,\underline{\boldsymbol{\lambda}}\right)=\prod_p \lambda_p \exp\left(-\lambda_p h_p\right)$ and therefore, (18) becomes:

$$\underline{\boldsymbol{\lambda}}^{ML} = \arg\max_{\underline{\boldsymbol{\lambda}}} \int Q\left(\underline{\mathbf{h}}\right)\left(\ln\lambda_p - \lambda_p h_p\right)d\underline{\mathbf{h}} \tag{20}$$

The Gibbs distribution $Q\left(\underline{\mathbf{h}}\right)$ treats $\underline{\mathbf{h}}$ as the dependent variable while assuming all other parameters to be constant. As such, the functional optimization of $\underline{\boldsymbol{\lambda}}$ in (20) is obtained by differentiating the terms within the integral with respect to λ_p and the end result is given by

$$\lambda_p = \frac{1}{\int h_p Q(\underline{\mathbf{h}})\, d\underline{\mathbf{h}}} \quad \text{for } p = 1,2,\ldots,R \tag{21}$$

where λ_p is the p^{th} element of $\underline{\boldsymbol{\lambda}}$. Since $p\left(\underline{\mathbf{y}} \mid \underline{\mathbf{h}}, \overline{\mathbf{D}}, \sigma^2\right) = \frac{1}{\left(2\pi\sigma^2\right)^{N_0/2}} \exp\left(-\frac{1}{2\sigma^2}\left\|\underline{\mathbf{y}} - \overline{\mathbf{D}}\underline{\mathbf{h}}\right\|^2\right)$ where $N_o = K \times L$, the iterative update rule for σ_{ML}^2 is given by

$$\begin{aligned}\sigma_{ML}^2 &= \arg\max_{\sigma^2} \int Q(\underline{\mathbf{h}})\left(-\frac{N_0}{2}\ln\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\left\|\underline{\mathbf{y}} - \overline{\mathbf{D}}\underline{\mathbf{h}}\right\|^2\right) d\underline{\mathbf{h}} \\ &= \frac{1}{N_0}\int Q(\underline{\mathbf{h}})\left(\left\|\underline{\mathbf{y}} - \overline{\mathbf{D}}\underline{\mathbf{h}}\right\|^2\right) d\underline{\mathbf{h}}\end{aligned} \tag{22}$$

Despite the simple form of (21) and (22), the integral is difficult to compute analytically and therefore, we seek an approximation to $Q(\underline{\mathbf{h}})$. We note that the solution $\underline{\mathbf{h}}$ naturally partition its elements into distinct subsets $\underline{\mathbf{h}}_P$ and $\underline{\mathbf{h}}_M$ consisting of components $\forall p \in P$ such that $h_p = 0$, and components $\forall m \in M$ such that $h_m > 0$. Thus, the $F(\underline{\mathbf{h}})$ can be expressed as following:

$$\begin{aligned}F(\underline{\mathbf{h}}) &= \frac{1}{2\sigma^2}\left\|\underline{\mathbf{y}} - \overline{\mathbf{D}}_P\underline{\mathbf{h}}_P - \overline{\mathbf{D}}_M\underline{\mathbf{h}}_M\right\|_F^2 + \underline{\boldsymbol{\lambda}}_P^{\mathrm{T}}\underline{\mathbf{h}}_P + \underline{\boldsymbol{\lambda}}_M^{\mathrm{T}}\underline{\mathbf{h}}_M \\ &= \underbrace{\frac{1}{2\sigma^2}\left\|\underline{\mathbf{y}} - \overline{\mathbf{D}}_M\underline{\mathbf{h}}_M\right\|_F^2 + \underline{\boldsymbol{\lambda}}_M^{\mathrm{T}}\underline{\mathbf{h}}_M}_{F(\underline{\mathbf{h}}_M)} + \underbrace{\frac{1}{2\sigma^2}\left\|\underline{\mathbf{y}} - \overline{\mathbf{D}}_P\underline{\mathbf{h}}_P\right\|_F^2 + \underline{\boldsymbol{\lambda}}_P^{\mathrm{T}}\underline{\mathbf{h}}_P}_{F(\underline{\mathbf{h}}_P)} + \underbrace{\frac{1}{2\sigma^2}\left[2\left(\overline{\mathbf{D}}_M\underline{\mathbf{h}}_M\right)^{\mathrm{T}}\left(\overline{\mathbf{D}}_P\underline{\mathbf{h}}_P\right) - \left\|\underline{\mathbf{y}}\right\|^2\right]}_{G} \\ &= F(\underline{\mathbf{h}}_M) + F(\underline{\mathbf{h}}_P) + G\end{aligned} \tag{23}$$

In (23), the term $\left\|\underline{\mathbf{y}}\right\|^2$ in G is a constant and the cross-term $\left(\overline{\mathbf{D}}_M\underline{\mathbf{h}}_M\right)^{\mathrm{T}}\left(\overline{\mathbf{D}}_P\underline{\mathbf{h}}_P\right)$ measures the orthogonality between $\overline{\mathbf{D}}_M\underline{\mathbf{h}}_M$ and $\overline{\mathbf{D}}_P\underline{\mathbf{h}}_P$. where $\overline{\mathbf{D}}_P$ is the sub-matrix of $\overline{\mathbf{D}}$ that corresponds to $\underline{\mathbf{h}}_P$, $\overline{\mathbf{D}}_M$ is the sub-matrix of $\overline{\mathbf{D}}$ that corresponds to $\underline{\mathbf{h}}_M$. In this work, we intend to simply the expression in (23) by discounting the contribution from these terms and let $F(\underline{\mathbf{h}})$ be approximated as $F(\underline{\mathbf{h}}) \approx F(\underline{\mathbf{h}}_M) + F(\underline{\mathbf{h}}_P)$. Given this approximation, $Q(\underline{\mathbf{h}})$ can be decomposed as

$$\begin{aligned}Q(\underline{\mathbf{h}}) &= \frac{1}{Z_h}\exp\left[-F(\underline{\mathbf{h}})\right] \\ &\approx \frac{1}{Z_h}\exp\left[-\left(F(\underline{\mathbf{h}}_P) + F(\underline{\mathbf{h}}_M)\right)\right] \\ &= \frac{1}{Z_P}\exp\left[-F(\underline{\mathbf{h}}_P)\right]\frac{1}{Z_M}\exp\left[-F(\underline{\mathbf{h}}_M)\right] \\ &= Q_P(\underline{\mathbf{h}}_P)Q_M(\underline{\mathbf{h}}_M)\end{aligned} \tag{24}$$

with $Z_P = \int \exp\left[-F\left(\underline{\mathbf{h}}_P\right)\right] d\underline{\mathbf{h}}_P$ and $Z_M = \int \exp\left[-F\left(\underline{\mathbf{h}}_M\right)\right] d\underline{\mathbf{h}}_M$. Since $\underline{\mathbf{h}}_P = \underline{\mathbf{0}}$ is on the boundary of the distribution, this distribution is represented by using the Taylor expansion about the MAP estimate, $\underline{\mathbf{h}}^{MAP}$:

$$\begin{aligned} Q_P\left(\underline{\mathbf{h}}_P \ge 0\right) &\propto \exp\left\{-\left[\left(\frac{\partial F}{\partial \underline{\mathbf{h}}}\right)\Bigg|_{\underline{\mathbf{h}}^{MAP}}\right]_P^{\mathbf{T}} \underline{\mathbf{h}}_P - \frac{1}{2}\underline{\mathbf{h}}_P^{\mathbf{T}}\overline{\mathbf{\Lambda}}_P\underline{\mathbf{h}}_P\right\} \\ &= \exp\left[-\left(\overline{\mathbf{\Lambda}}\underline{\mathbf{h}}^{MAP} - \frac{1}{\sigma^2}\overline{\mathbf{D}}^{\mathbf{T}}\underline{\mathbf{y}} + \underline{\boldsymbol{\lambda}}\right)_P^{\mathbf{T}} \underline{\mathbf{h}}_P - \frac{1}{2}\underline{\mathbf{h}}_P^{\mathbf{T}}\overline{\mathbf{\Lambda}}_P\underline{\mathbf{h}}_P\right] \end{aligned} \tag{25}$$

where $\overline{\mathbf{\Lambda}}_P = \frac{1}{\sigma^2}\overline{\mathbf{D}}_P^{\mathbf{T}}\overline{\mathbf{D}}_P$, $\overline{\mathbf{\Lambda}} = \frac{1}{\sigma^2}\overline{\mathbf{D}}^{\mathbf{T}}\overline{\mathbf{D}}$. We perform variational approximation to $Q_P(\underline{\mathbf{h}}_P)$ by using the exponential distribution:

$$\hat{Q}_P\left(\underline{\mathbf{h}}_P \ge 0\right) = \prod_{p \in P} \frac{1}{u_p}\exp\left(-h_p / u_p\right) \tag{26}$$

The variational parameters $\underline{\mathbf{u}} = \left\{u_p\right\}$ for $\forall p \in P$ are obtained by minimizing the Kullback-Leibler divergence between Q_P and $\hat{Q}_P$:

$$\begin{aligned} \underline{\mathbf{u}} &= \arg\min_{\underline{\mathbf{u}}} \int \hat{Q}_P\left(\underline{\mathbf{h}}_P\right)\ln\frac{\hat{Q}_P\left(\underline{\mathbf{h}}_P\right)}{Q_P\left(\underline{\mathbf{h}}_P\right)}d\underline{\mathbf{h}}_P \\ &= \arg\min_{\underline{\mathbf{u}}} \int \hat{Q}_P\left(\underline{\mathbf{h}}_P\right)\left[\ln\hat{Q}_P\left(\underline{\mathbf{h}}_P\right) - \ln Q_P\left(\underline{\mathbf{h}}_P\right)\right]d\underline{\mathbf{h}}_P \end{aligned} \tag{27}$$

In Eqn. (27).

$$\begin{aligned} &\int \hat{Q}_P\left(\underline{\mathbf{h}}_P\right)\ln\left[\hat{Q}_P\left(\underline{\mathbf{h}}_P\right)\right]d\underline{\mathbf{h}}_P = \sum_{p \in P}\int \hat{Q}_P\left(h_p\right)\ln\left[\hat{Q}_P\left(h_p\right)\right]dh_p \\ &= \sum_{p \in P}\int_0^\infty dh_p \frac{1}{u_p}\exp\left(-h_p / u_p\right)\left(-\ln u_p - h_p / u_p\right) \\ &= -\sum_{p \in P}\ln u_p \int_0^\infty d\left(\frac{h_p}{u_p}\right)\exp\left(-h_p / u_p\right) - \sum_{p \in P}\int_0^\infty d\left(\frac{h_p}{u_p}\right)\frac{h_p}{u_p}\exp\left(-h_p / u_p\right) \\ &= -\sum_{p \in P}\ln u_p + 1 \end{aligned} \tag{28}$$

and

$$
\begin{aligned}
&\int \hat{Q}_P(\underline{\mathbf{h}}_P)\ln\left[Q_P(\underline{\mathbf{h}}_P)\right]d\underline{\mathbf{h}}_P \\
&= -\int d\underline{\mathbf{h}}_P\left[\left(\overline{\mathbf{\Lambda}}\underline{\mathbf{h}}^{MAP} - \frac{1}{\sigma^2}\overline{\mathbf{D}}^{\mathbf{T}}\underline{\mathbf{y}} + \underline{\boldsymbol{\lambda}}\right)_P^{\mathbf{T}}\underline{\mathbf{h}}_P + \frac{1}{2}\underline{\mathbf{h}}_P^{\mathbf{T}}\overline{\mathbf{\Lambda}}_P\underline{\mathbf{h}}_P\right]\hat{Q}_P(\underline{\mathbf{h}}_P) \qquad (29)\\
&= -\sum_{p\in P, m\in M}\frac{1}{2}\left(\overline{\mathbf{\Lambda}}\right)_{pm}\left\langle h_p h_m\right\rangle - \sum_{p\in P}\left(\overline{\mathbf{\Lambda}}\underline{\mathbf{h}}^{MAP} - \frac{1}{\sigma^2}\overline{\mathbf{D}}^{\mathbf{T}}\underline{\mathbf{y}} + \underline{\boldsymbol{\lambda}}\right)_p\left\langle h_p\right\rangle
\end{aligned}
$$

with $\langle\bullet\rangle$ denotes the expectation under $\hat{Q}_P(\underline{\mathbf{h}}_P)$ distribution [28] such that $\left\langle h_p h_m\right\rangle = u_p u_m$ and $\left\langle h_p\right\rangle = u_p$ which leads to:

$$\min_{u_p} \hat{\underline{\mathbf{b}}}_P^{\mathbf{T}}\underline{\mathbf{u}} + \frac{1}{2}\underline{\mathbf{u}}^{\mathbf{T}}\hat{\mathbf{\Lambda}}\underline{\mathbf{u}} - \sum_{p\in P}\ln u_p \qquad (30)$$

where $\hat{\underline{\mathbf{b}}}_P = \left(\overline{\mathbf{\Lambda}}\underline{\mathbf{h}}^{MAP} - \frac{1}{\sigma^2}\overline{\mathbf{D}}^{\mathbf{T}}\underline{\mathbf{y}} + \underline{\boldsymbol{\lambda}}\right)_P$ and $\hat{\mathbf{\Lambda}} = \overline{\mathbf{\Lambda}}_P + diag\left(\overline{\mathbf{\Lambda}}_P\right)$. The optimization of (30) can be accomplished be expanding (30) as follows:

$$G(\underline{\mathbf{u}},\tilde{\underline{\mathbf{u}}}) = \hat{\underline{\mathbf{b}}}_P^{\mathbf{T}}\underline{\mathbf{u}} + \frac{1}{2}\sum_{p\in P}\frac{\left(\hat{\mathbf{\Lambda}}\tilde{\underline{\mathbf{u}}}\right)_p}{\tilde{u}_p}u_p^2 - \sum_{p\in P}\ln u_p \qquad (31)$$

Taking the derivative of $G(\underline{\mathbf{u}},\tilde{\underline{\mathbf{u}}})$ in (31) with respect to $\underline{\mathbf{u}}$ and setting it to be zero, we have:

$$\frac{\left(\hat{\mathbf{\Lambda}}\tilde{\underline{\mathbf{u}}}\right)_p}{\tilde{u}_p}u_p + \hat{b}_p - \frac{1}{u_p} = 0 \qquad (32)$$

The above equation is equivalent to the following quadratic equations:

$$\frac{\left(\hat{\mathbf{\Lambda}}\tilde{\underline{\mathbf{u}}}\right)_p}{\tilde{u}_p}u_p^2 + \hat{b}_p u_p - 1 = 0 \qquad (33)$$

Solving (33) for u_p leads to the following update:

$$u_p \leftarrow u_p\frac{-\hat{b}_p + \sqrt{\hat{b}_p^2 + 4\frac{\left(\hat{\mathbf{\Lambda}}\underline{\mathbf{u}}\right)_p}{\tilde{u}_p}}}{2\left(\hat{\mathbf{\Lambda}}\underline{\mathbf{u}}\right)_p} \qquad (34)$$

As for components $\underline{\mathbf{h}}_M$, since none of the non-negative constraints are active, we approximate $Q_M(\underline{\mathbf{h}}_M)$ as unconstrained Gaussian with mean $\underline{\mathbf{h}}_M^{MAP}$. Thus using the factorized approximation $Q(\underline{\mathbf{h}}) = \hat{Q}_P(\underline{\mathbf{h}}_P)Q_M(\underline{\mathbf{h}}_M)$ in (21), we obtain the following:

$$\lambda_p = \begin{cases} \dfrac{1}{\int h_p Q_M\left(\underline{\mathbf{h}}_M\right) d\underline{\mathbf{h}}_M} = \dfrac{1}{h_p^{MAP}} & \text{if } p \in M \\ \dfrac{1}{\int h_p \hat{Q}_P\left(\underline{\mathbf{h}}_P\right) d\underline{\mathbf{h}}_P} = \dfrac{1}{u_p} & \text{if } p \in P \end{cases} \tag{35}$$

for $p = 1, 2, \ldots, R$ and h_p^{MAP} is the p^{th} element of sparse code $\underline{\mathbf{h}}_P$ computed from (11) and its covariance $\mathbf{C}$ is given by

$$C_{pm} = \begin{cases} \left(\overline{\mathbf{\Lambda}}_P^{-1}\right)_{pm} & \text{if } p, m \in M \\ u_p^2 \delta_{pm} & \text{Otherwise} \end{cases} \tag{36}$$

Thus, the update rule for σ^2 computed from (22) can be obtained as

$$\sigma^2 = \frac{1}{N_0}\left[\left(\underline{\mathbf{y}} - \overline{\mathbf{D}}\hat{\underline{\mathbf{h}}}\right)^{\mathbf{T}}\left(\underline{\mathbf{y}} - \overline{\mathbf{D}}\hat{\underline{\mathbf{h}}}\right) + \mathrm{Tr}\left(\overline{\mathbf{D}}^{\mathbf{T}}\overline{\mathbf{D}}\mathbf{C}\right)\right] \text{ where } \hat{h}_p = \begin{cases} h_p^{MAP} & \text{if } p \in M \\ u_p & \text{if } p \in P \end{cases} \tag{37}$$

The specific steps of the proposed method can be summarized as the following table:

1. Initialize $\mathbf{D}^\tau$ and $\mathbf{H}^\phi$ with nonnegative random values.
2. Define $\tilde{\mathbf{D}}_{k,d}^\tau = \mathbf{D}_{k,d}^\tau \Big/ \sqrt{\sum_{\tau,k}(\mathbf{D}_{k,d}^\tau)^2}$ and Compute $\tilde{\mathbf{Z}} = \sum_d \sum_\tau \sum_\phi \overset{\downarrow\phi}{\tilde{\mathbf{D}}_d^\tau} \overset{\rightarrow\tau}{\mathbf{H}_d^\phi}$.
3. Assign $\lambda_p = \begin{cases} \dfrac{1}{h_p^{MAP}} & \text{if } p \in M \\ \dfrac{1}{u_p} & \text{if } p \in P \end{cases}$.
4. Assign $\sigma^2 = \dfrac{1}{N_0}\left[\left(\underline{\mathbf{y}} - \overline{\mathbf{D}}\hat{\underline{\mathbf{h}}}\right)^{\mathbf{T}}\left(\underline{\mathbf{y}} - \overline{\mathbf{D}}\hat{\underline{\mathbf{h}}}\right) + \mathrm{Tr}\left(\overline{\mathbf{D}}^{\mathbf{T}}\overline{\mathbf{D}}\mathbf{C}\right)\right]$.
5. Update $\mathbf{H}^\phi \leftarrow \mathbf{H}^\phi \bullet \dfrac{\sum_\tau \overset{\downarrow\phi^{\mathbf{T}}}{\tilde{\mathbf{D}}^\tau} \overset{\leftarrow\tau}{\mathbf{Y}}}{\sum_\tau \overset{\downarrow\phi^{\mathbf{T}}}{\tilde{\mathbf{D}}^\tau} \overset{\leftarrow\tau}{\tilde{\mathbf{Z}}} + \boldsymbol{\lambda}_p^\phi}$ and compute $\tilde{\mathbf{Z}} = \sum_d \sum_\tau \sum_\phi \overset{\downarrow\phi}{\tilde{\mathbf{D}}_d^\tau} \overset{\rightarrow\tau}{\mathbf{H}_d^\phi}$.
6. Update $\mathbf{D}^\tau \leftarrow \tilde{\mathbf{D}}^\tau \bullet \dfrac{\sum_\phi \overset{\uparrow\phi}{\mathbf{Y}} \overset{\rightarrow\tau^{\mathbf{T}}}{\mathbf{H}^\phi} + \tilde{\mathbf{D}}^\tau diag\left(\sum_\tau \mathbf{1}\left(\left(\overset{\uparrow\phi}{\tilde{\mathbf{Z}}} \overset{\rightarrow\tau^{\mathbf{T}}}{\mathbf{H}^\phi}\right) \bullet \tilde{\mathbf{D}}^\tau\right)\right)}{\sum_\phi \overset{\uparrow\phi}{\tilde{\mathbf{Z}}} \overset{\rightarrow\tau^{\mathbf{T}}}{\mathbf{H}^\phi} + \tilde{\mathbf{D}}^\tau diag\left(\sum_\tau \mathbf{1}\left(\left(\overset{\uparrow\phi}{\mathbf{Y}} \overset{\rightarrow\tau^{\mathbf{T}}}{\mathbf{H}^\phi}\right) \bullet \tilde{\mathbf{D}}^\tau\right)\right)}$.
7. Repeat steps 2 to 6 until convergence.

Table 1. Proposed Adaptive Sparsity NMF2D

3. Single channel audio source separation

3.1 TF representation

The classic spectrogram decomposes signals to components of linearly spaced frequencies. However, in western music, the typically used frequencies are geometrically spaced. Thus, obtaining an acceptable low-frequency resolution is absolutely necessary, while a resolution that is geometrically related to the frequency is desirable, although not critical. The constant Q transform as introduced in [29], tries to solve both issues. In general, the twelve-tone equal tempered scale which forms the basis of modern western music divides each octave into twelve half notes where the frequency ratio between each successive half note is equal [23]. The fundamental frequency of the note which is k_Q half note above can be expressed as $f_{k_Q}^{Q} = f_{\text{fund}} \cdot 2^{k_Q/24}$. Taking the logarithmic, this gives $\log f_{k_Q}^{Q} = \log f_{\text{fund}} + \frac{k_Q}{24}\log 2$. Thus, in a log-frequency representation the notes are linearly spaced. In our method, the frequency axis of the obtained spectrogram is logarithmically scaled and grouped into 175 frequency bins in the range of 50Hz to 8kHz (given $f_s = 16\text{kHz}$) with 24 bins per octave and the bandwidth follows the constant-Q rule. Figure 1 shows an example of the estimated spectral dictionary $\mathbf{D}$ and temporal code $\mathbf{H}$ based on SNMF2D method on the log-frequency spectrogram.

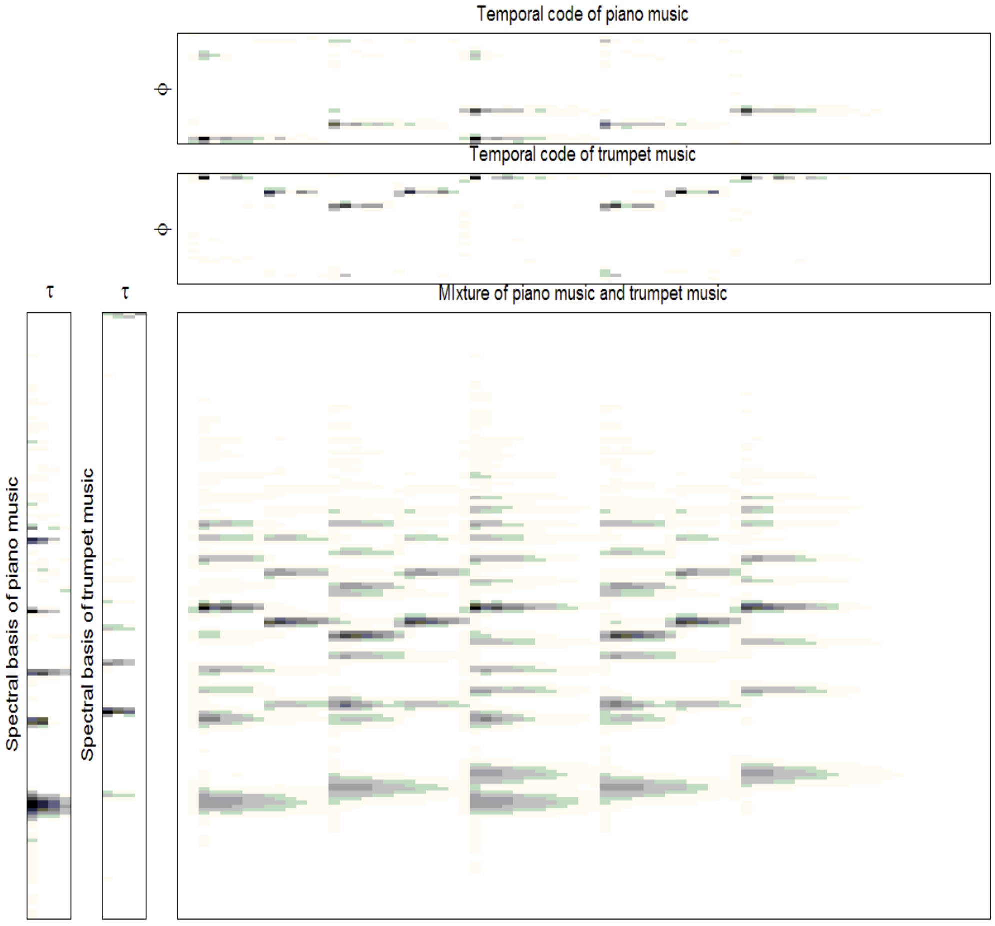

Fig. 1. The estimated spectral dictionary and temporal code of piano and trumpet mixture log-frequency spectrum using SNMF2D.

3.2 Source reconstruction

The Figure 2 shows the framework of the proposed *unsupervised* SCSS methods. The single channel audio mixture is constructed by several unknown sources, namely $y(t)=\sum_{d=1}^{d_{\max}} x_d(t)$. where $d=1,\ldots,d_{\max}$ denotes the sources number and $t=1,2,\ldots,T$ denotes the time index. The goal is to estimate the sources $x_d(t)$ when only the observation signal $y(t)$ is available. The mixture is then transformed into a suitable representation e.g. Time-Frequency (TF) representation. Thus the mixture $y(t)$ is given by $Y(f,t_s)=\sum_{d=1}^{d_{\max}} X_d(f,t_s)$ where $Y(f,t_s)$ and $X_d(f,t_s)$ denote the TF components obtained by applying the short time Fourier transform (STFT) on $y(t)$ and $x_d(t)$, respectively, e.g. $Y(f,t_s)=STFT\big(y(t)\big)$. The time slots are given by $t_s=1,2,\ldots,T_s$ while frequency bins by $f=1,2,\ldots,F$. Since each component is a function of t_s and f, we represent this as $\mathbf{Y}=\left[Y(f,t_s)\right]_{t_s=1,2,\ldots,T_s}^{f=1,2,\ldots,F}$ and $\mathbf{X}_d=\left[X_d(f,t_s)\right]_{t_s=1,2,\ldots,T_s}^{f=1,2,\ldots,F}$. The power spectrogram is defined as the squared magnitude STFT and hence, its matrix representation is given by $\left|\mathbf{Y}\right|^{.2}\approx\sum_{d=1}^{d_{\max}}\left|\mathbf{X}_d\right|^{.2}$ where the superscript '$\cdot$' represents element wise operation. The frequencies scale of power spectrogram $\left|\mathbf{Y}\right|^{.2}$ can be mapped into log-frequency scale which described in Section III A and this will result log-frequency power spectrogram $\left|\widehat{\mathbf{Y}}\right|^{.2}=\sum_{d=1}^{N_s}\left|\widehat{\mathbf{X}}_d\right|^{.2}$. The matrices we seek to determine are $\left\{\left|\widehat{\mathbf{X}}_d\right|^{.2}\right\}_{d=1}^{N_s}$ which will be obtained during the feature extraction process by using the proposed matrix factorization as $\left|\tilde{\mathbf{X}}_d\right|^{.2}=\sum_{\tau}\sum_{\phi}\overset{\downarrow\phi}{\mathbf{D}}{}_d^{\tau}\overset{\rightarrow\tau}{\mathbf{H}}{}_d^{\phi}$ where $\mathbf{D}_d^{\tau}$ and $\mathbf{H}_d^{\phi}$ are estimated using (11) and (12). Once these matrices are estimated, we form the dth binary mask according to $W_d(f,t_s)=1$ if $\left|\tilde{X}_d(f,t_s)\right|^{.2}>\left|\tilde{X}_j(f,t_s)\right|^{.2}$ $d\neq j$ and zero otherwise to approach source separation. Finally, the estimated time-domain sources are obtained as $\tilde{\mathbf{x}}_d=\xi^{-1}\big(\mathbf{W}_d\bullet\widehat{\mathbf{Y}}\big)$ where $\xi^{-1}(\bullet)$ denotes the inverse mapping of the log-frequency axis to the original frequency axis and followed by the inverse STFT back to the time domain. $\tilde{\mathbf{x}}_d=\left[\tilde{x}_d(1),\ldots,\tilde{x}_d(T)\right]^{\mathbf{T}}$ denotes the dth estimated audio sources in the time-domain.

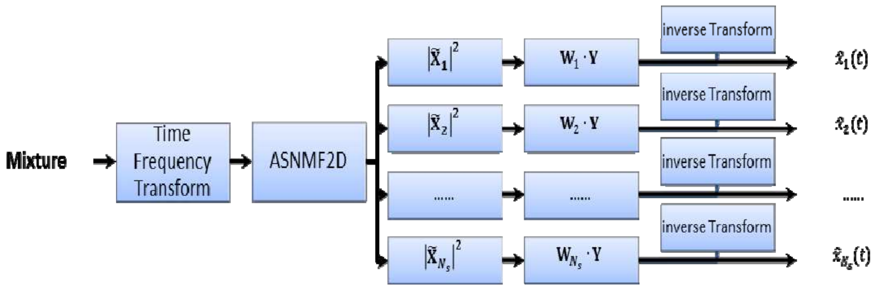
Mixture
Time Frequency Transform
ASNMF2D
$|\tilde{\mathbf{X}}_1|^2$
$|\tilde{\mathbf{X}}_2|^2$
......
$|\tilde{\mathbf{X}}_{N_s}|^2$
$\mathbf{W}_1\cdot\mathbf{Y}$
$\mathbf{W}_2\cdot\mathbf{Y}$
......
$\mathbf{W}_{N_s}\cdot\mathbf{Y}$
inverse Transform
inverse Transform
inverse Transform
inverse Transform
$\hat{x}_1(t)$
$\hat{x}_2(t)$
......
$\hat{x}_{N_s}(t)$

Fig. 2. A framework for the proposed *unsupervised* SCSS methods.

3.3 Efficiency of source extraction in TF domain

In this sub-section, we will analyze how different sparsity factorization methods impact on the source extraction performance in TF domain for SCASS. For separation, one generates the TF mask corresponding to each source and applies the generated mask to the mixture to obtain the estimated source TF representation. In particular, when the sources have no overlap in the TF domain, an optimum mask $W_d^{opt}(f,t_s)$ (optimal source extractor) exists which allows one to extract the dth original source from the mixture as

$$X_d(f,t_s) = W_d^{opt}(f,t_s)Y(f,t_s) \tag{38}$$

Given any TF mask $W_d(f,t_s)$ (source extractor) such that $0 \le W_d(f,t_s) \le 1$ for all (f,t_s), we define the efficiency of source extraction (ESE) in the TF domain for target source $x_d(t)$ in the presence of the interfering sources $\beta_d(t) = \sum_{j=1,j\neq d}^{d_{\max}} x_j(t)$ as

$$\psi(W_d) \triangleq \frac{\left\|W_d(f,t_s)X_d(f,t_s)\right\|_F^2}{\left\|X_d(f,t_s)\right\|_F^2} - \frac{\left\|W_d(f,t_s)B_d(f,t_s)\right\|_F^2}{\left\|X_d(f,t_s)\right\|_F^2} \tag{39}$$

where $X_d(f,t_s)$ and $B_d(f,t_s)$ are the TF representations of $x_d(t)$ and $\beta_d(t)$, respectively. The above represents the normalized energy difference between the extracted source and interferences. We also define the ESE of the mixture with respect to all the $d_{\max}$ sources as

$$\Omega = \frac{1}{d_{\max}} \sum_{d=1}^{d_{\max}} \psi(W_i) \tag{40}$$

Eqn. (39) is equivalent to measuring the ability of extracting the dth source $X_d(f,t_s)$ from the mixture $Y(f,t_s)$ given the TF mask $W_d(f,t_s)$. Eqn. (40) measures the ability of extracting all the $d_{\max}$ sources simultaneously from the mixture. To further study the ESE, we use the following two criteria [30]: (i) preserved signal ratio (PSR) which determines how well the mask preserves the source of interest and (ii) signal-to-interference ratio (SIR) which indicates how well the mask suppresses the interfering sources:

$$PSR_{W_d}^{X_d} \triangleq \frac{\left\|W_d(f,t_s)X_d(f,t_s)\right\|_F^2}{\left\|X_d(f,t_s)\right\|_F^2} \quad \text{and} \quad SIR_{W_d}^{X_d} \triangleq \frac{\left\|W_d(f,t_s)X_d(f,t_s)\right\|_F^2}{\left\|W_d(f,t_s)B_d(f,t_s)\right\|_F^2} \tag{41}$$

Using (41), (39) can be expressed as $\psi(W_d) = PSR_{W_d}^{X_d} - PSR_{W_d}^{X_d} \big/ SIR_{W_d}^{X_d}$. Analyzing the terms in (39), we have

$$
\begin{aligned}
PSR_{W_d}^{X_d} &:= \begin{cases} 1 & , \textit{if } \text{supp } W_d^{opt} = \text{supp } W_d \\ <1 & , \textit{if } \text{supp } W_d^{opt} \subset \text{supp } W_d \end{cases} \\
SIR_{W_d}^{X_d} &:= \begin{cases} \infty & , \textit{if } \text{supp}[W_d X_d] \cap \text{supp} B_d = \varnothing \\ \textit{finite} & , \textit{if } \text{supp}[W_d X_d] \cap \text{supp} B_d \neq \varnothing \end{cases}
\end{aligned} \tag{42}
$$

where 'supp' denotes the support. When $\psi(W_d) = 1$ (i.e. $PSR_{W_d}^{X_d} = 1$ and $SIR_{W_d}^{X_d} = \infty$), this indicates that the mixture $y(t)$ is separable with respect to the dth source $x_d(t)$. In other words, $X_d(f,t_s)$ does not overlap with $B_d(f,t_s)$ and the TF mask $W_d(f,t_s)$ has perfectly separated the dth source $X_d(f,t_s)$ from the mixture $Y(f,t_s)$. This corresponds to $W_d(f,t_s) = W_d^{opt}(f,t_s)$ in (38). Hence, this is the maximum attainable $\psi(W_d)$ value. For other cases of $PSR_{W_d}^{X_d}$ and $SIR_{W_d}^{X_d}$, we have $\psi(W_d) < 1$. Using the above concept, we can extend the analysis for the case of separating $d_{\max}$ sources. A mixture $y(t)$ is fully separable to all the N sources if and only if $\Omega = 1$ in (40). For the case $\Omega < 1$, this implies that some of the sources overlap with each other in the TF domain and therefore, they cannot be fully separated. Thus, Ω provides the quantitative performance measure to evaluate how separable the mixture is in the TF domain. In the following, we show the analysis of how different sparsity factorization methods affect the ESE of the mixture.

4. Results and analysis

4.1 Experiment set-up

The proposed method is tested by separating music sources. Several experimental simulations under different conditions have been designed to investigate the efficacy of the proposed method. All simulations and analyses are performed using a PC with Intel Core 2 CPU 6600 @ 2.4GHz and 2GB RAM. MATLAB is used as the programming platform. We have tested the proposed method in the wider types of music mixtures. All mixed signals are sampled at 16 kHz sampling rate. 30 music signals including 10 jazz, 10 piano and 10 trumpet signals are selected from the RWC [31] database. Three types of mixture have been generated: (i) jazz mixed with piano, (ii) jazz mixed with trumpet, (iii) piano mixed with trumpet. The sources are randomly chosen from the database and the mixed signal is generated by adding the chosen sources. In all cases, the sources are mixed with equal average power over the duration of the signals. The TF representation is computed by normalizing the time-domain signal to unit power and computing the STFT using 2048 point Hanning window FFT with 50% overlap. The frequency axis of the obtained spectrogram is then logarithmically scaled and grouped into 175 frequency bins in the range of 50Hz to 8kHz with 24 bins per octave. This corresponds to twice the resolution of the equal tempered musical scale. For the proposed adaptive sparsity factorization model, the convolutive components in time and frequency are selected to be (i) For piano and trumpet mixture $\tau = \{0,\ldots,3\}$ and $\phi = \{0,\ldots,31\}$, respectively; (ii) For piano and jazz mixture $\tau = \{0,\ldots,6\}$ and $\phi = \{0,\ldots,9\}$, respectively; (iii) For trumpet and jazz mixture $\tau = \{0,\ldots,6\}$ and $\phi = \{0,\ldots,9\}$, respectively. The corresponding sparse factor was determined by (35). We have evaluated our separation performance in terms of the signal-to-distortion ratio (SDR) which is one form of perceptual measure. This is a global measure that unifies source-to-interference ratio (SIR), source-to-artifacts ratio (SAR) and source-to-noise ratio (SNR). MATLAB routines for computing these criteria are obtained from the SiSEC'08 webpage [32, 33].

4.2 Impact of adaptive and fixed sparsity

In this implementation, we have conducted several experiments to compare the performance of the proposed method with SNMF2D under different sparsity regularization. In particular, Figures 3 and 4 show the separated sources by using the proposed method in terms of spectrogram and time-domain representation, respectively.

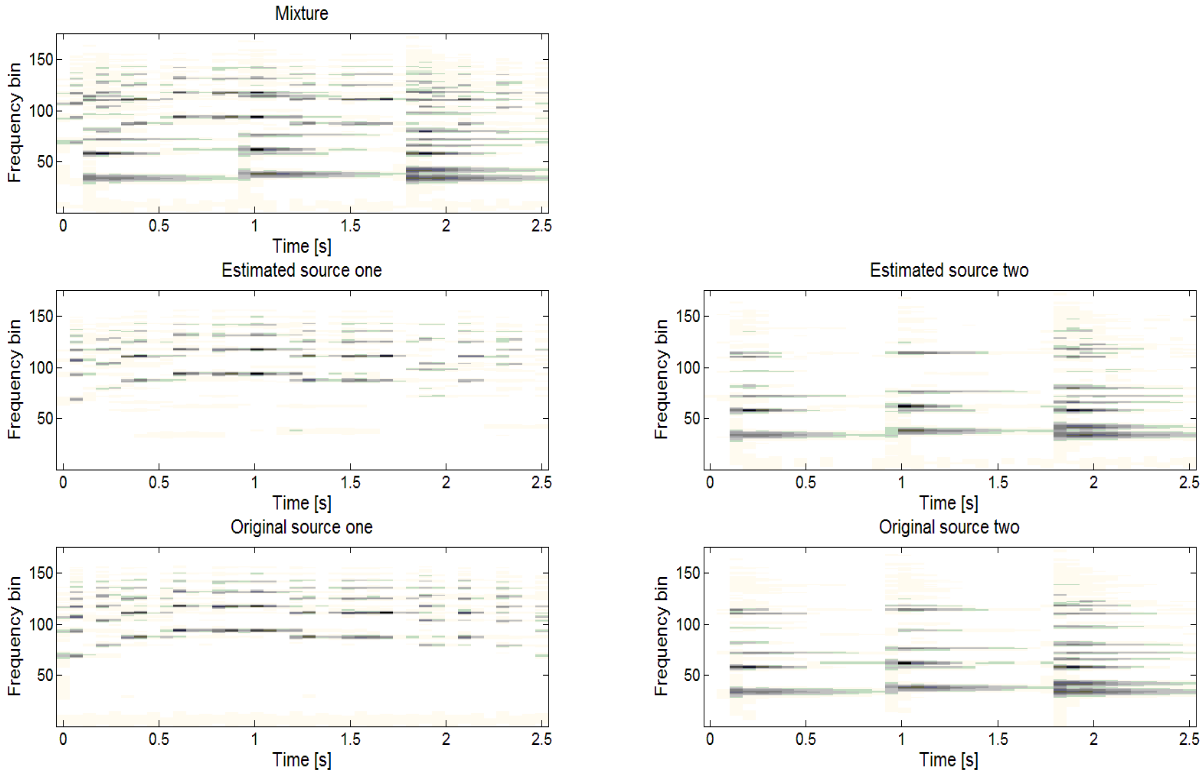

Fig. 3. Spectrogram of the mixed signal (top panel), the recovered trumpet music and piano music (middle panels) and original trumpet music and piano music (bottom panels).

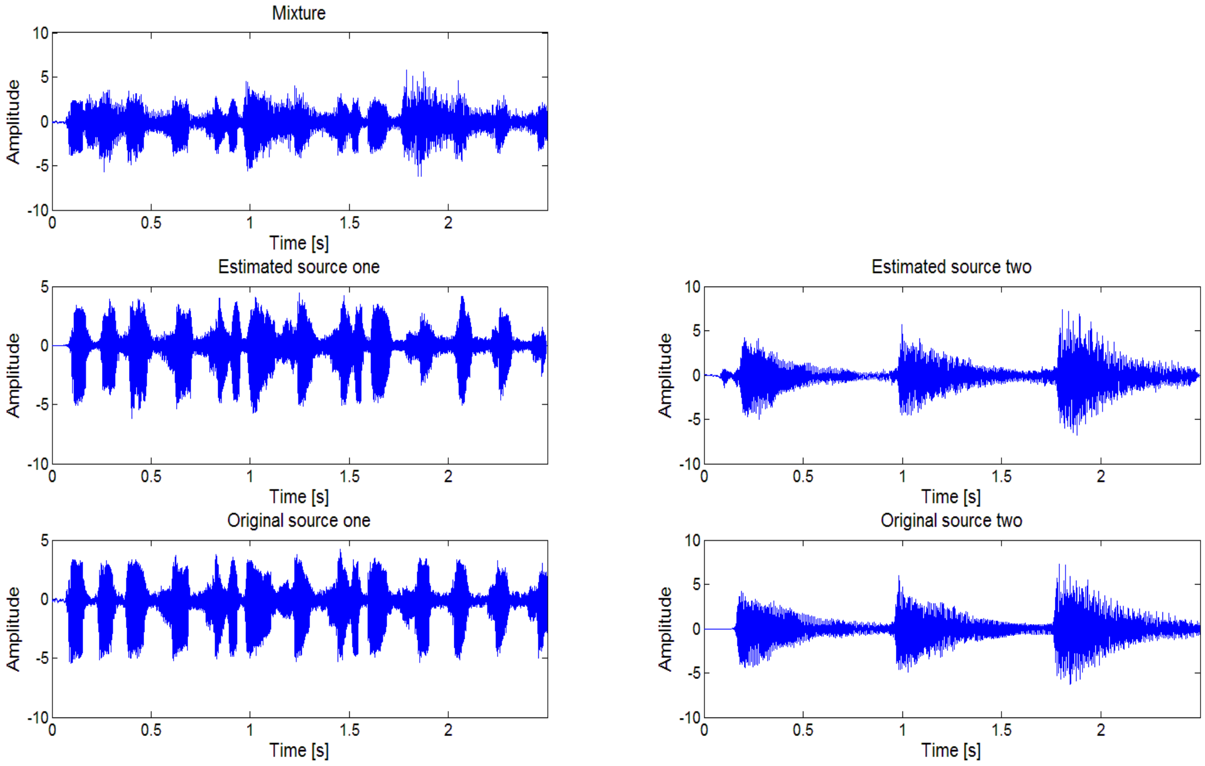

Fig. 4. Time domain of the mixed signal (top panel), the recovered trumpet music and piano music (middle panels) and original trumpet music and piano music (bottom panels).

To investigate this further, the impact of sparsity regularization on the separation results in terms of the SDR under different uniform regularization has been undertaken and the results are plotted in Figure 4. In this implementation, the uniform regularization is chosen as $c = 0, 0.5, \ldots, 10$ for all sparsity parameters i.e. $\lambda_{d,l}^{\phi} = \lambda = c$. The best result is retained and tabulated in Table I. In the case of the proposed method, it assigns a regularization parameter to each temporal code which is individually and adaptively tuned to yield the optimal number of times the spectral dictionary of a source recurs in the spectrogram. The sparsity on $\mathbf{H}_d^{\phi}$ is imposed *element-wise* in the proposed model so that each individual code in $\mathbf{H}_d^{\phi}$ is optimally sparse in the L_1-norm. In the conventional SNMF2D method, the sparsity is not fully controlled but is imposed uniformly on all the codes. The ensuing consequence is that the temporal codes are no longer optimal and this leads to 'under-sparse' or 'over-sparse' factorization which eventually results in inferior separation performance.

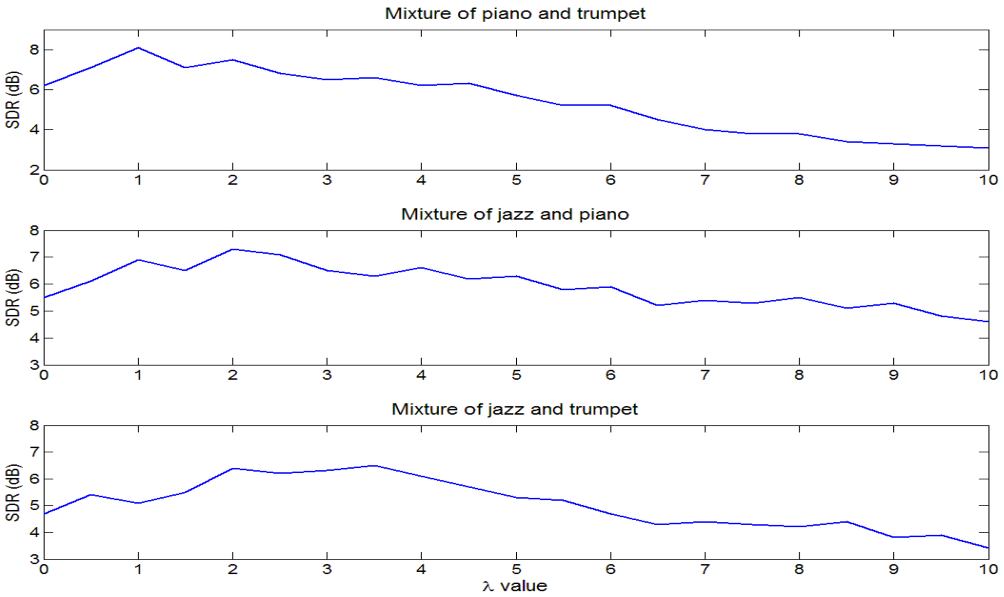

Fig. 5. Separation results of SNMF2D by using different uniform regularization.

In Figure 5, the results have clearly indicated that there are certain values of λ where the SNMF2D performs with exceptionally good results. In the case of piano and trumpet mixtures, the best performance is obtained when λ ranges from 0.5 to 2 where the highest SDR is 8.1dB. As for jazz and piano mixtures, the best performance is obtained when λ ranges from 1.0 to 2.5 where the highest SDR is 7.2dB and for jazz and trumpet mixtures, the best performance is obtained when λ ranges from 2 to 3.5 where the highest SDR is 8.6dB. On the contrary, when λ is set too high, the separation performance tends to degrade. It is also worth pointing out that the separation results are coarse when the factorization is non-regularized. Here, we see that (i) for piano and trumpet mixtures, the SDR is only 6.2dB, (ii) for jazz and piano mixtures, the SDR is only 5.6dB, (iii) for jazz and trumpet mixtures, the SDR is only 4.7dB. From above, it is evident that uniform sparsity scheme gives varying performance depending on the value of λ which in turn depends on the type of mixture. Hence, this poses a practical difficulty in selecting the appropriate level sparseness necessary for matrix factorization to resolve the ambiguity between the sources in the TF domain.

The overall comparison results between the adaptive and uniform sparsity methods have been summarized in Figure 6. According to the table, SNMF2D with adaptive sparsity tends to yield better result than the uniform sparsity-based methods. We may summarize the average performance improvement of our method against the uniform constant sparsity method: (i) For the piano and trumpet music, the improvement per source in terms of the SDR is 2dB (ii) For the piano and jazz music, the improvement per source in terms of SDR is 1.3dB. (iii) For the trumpet and jazz music, the improvement per source in terms of SDR is 1.1dB.

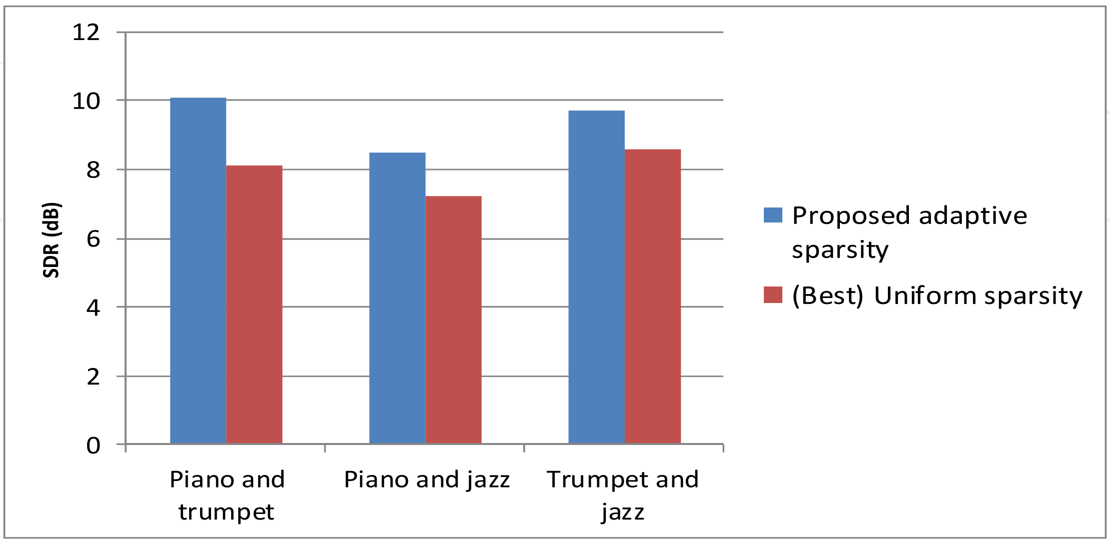

Fig. 6. SDR results comparison between adaptive and uniform sparsity methods.

4.2.1 Adaptive behavior of sparsity parameter

In this sub-section, the adaptive behavior of the sparsity parameters by using the proposed method will be demonstrated. Several sparsity parameters have been selected to illustrate its adaptive behavior. In the experiment, all sparsity parameters are initialized as $\lambda_{d,l}^{\phi}=5$ for all d,l,ϕ and are subsequently adapted according to (35). After 300 iterations, the sparsity parameters converge to their steady-states. We have plotted the histogram of the converged adaptive sparsity parameters in Figure 7. The figure suggests that the histogram can be represented as a bimodal distribution that each element code has its own sparseness. In addition, it is worth pointing out that in the case of piano and trumpet mixture the SDR result rises to 10dB when $\lambda_{d,l}^{\phi}$ is adaptive. This represents a 2dB per source improvement over the case of uniform constant sparsity (which is only 8.1dB in Figure 6). On the separate hand, when no sparsity is imposed onto the codes the SDR result immediately deteriorates to approximately 6dB. This represents a 4dB per source depreciation compared with the proposed adaptive sparsity method. From above, the results are ready to suggest that the performances of source separation have been undermined when the uniform constant sparsity scheme is used. On the contrary, improved performances can be obtained by allowing the sparsity parameters to be individually adapted for each element code. This is evident based on source separation performance as indicated in Figure 6.

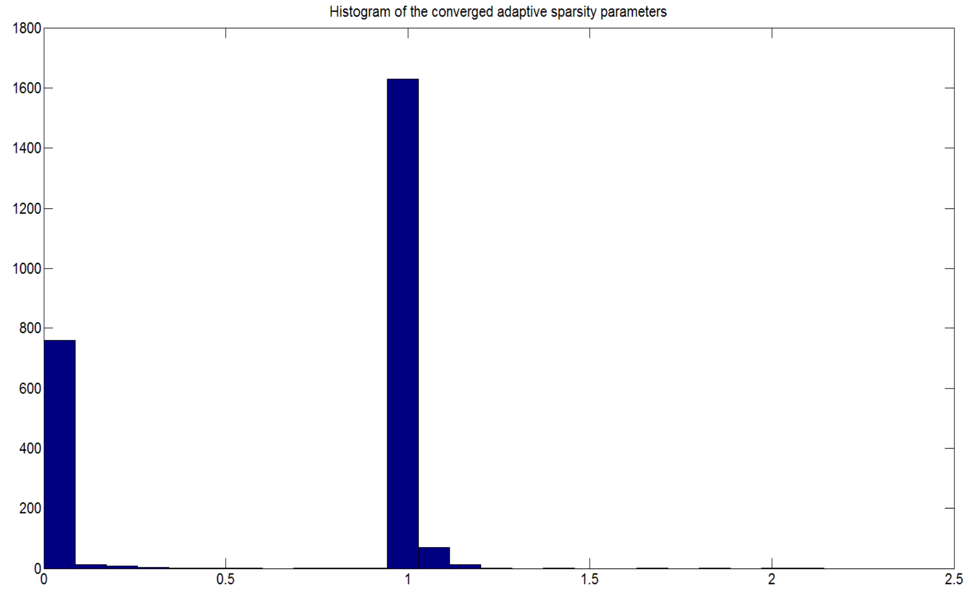

Fig. 7. The histogram of the converged adaptive sparsity parameter.

4.2.2 Efficiency of source extraction in TF domain

In this sub-section, we will analyze the efficiency of source extraction based on SNMF2D and the proposed method. Binary masks are constructed using the approach discussed in Section III B for each of the both methods. To ensure fair comparison, we generate the ideal binary mask (IBM) [34] from the original source which is used as a reference for comparison. The IBM for a target source is found for each TF unit by comparing the energy of the target source to the energy of all the interfering sources. Hence, the ideal binary mask produces the optimal signal-to-distortion ratio (SDR) gain of all binary masks and thus, it can be considered as an optimal source extractor in TF domain. The comparison results between IBM, uniform sparsity and proposed adaptive sparsity are tabulated in Table II.

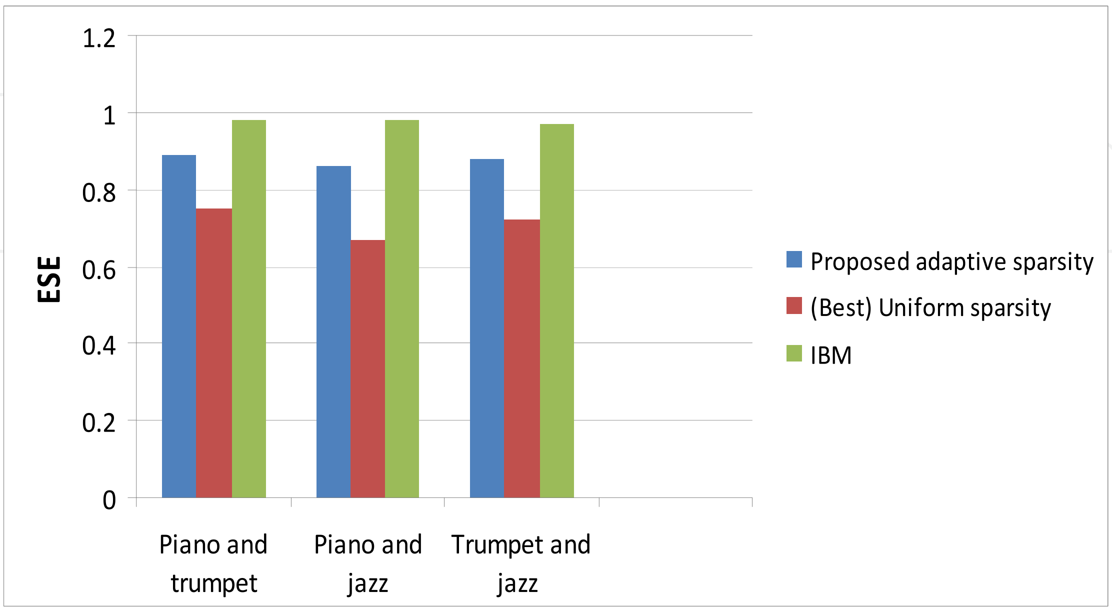

Fig. 8. Overall ESE performance

In Figure 8, the results of ESE for each mixture type are obtained by averaging over 100 realizations. From listening performance test, any $\psi(W_d) > 0.8$ indicates acceptable quality of source extraction performance in TF domain. Therefore, it is noted from the results in Figure 7 that both IBM and the proposed method satisfy this condition. In addition, the proposed method yields better ESE improvement against the uniform sparsity method. The average improvement results have been summarized as follows: (i) For the piano and trumpet music, 18.4%. (ii) For the piano and jazz music 26.5%. (iii) For the trumpet and jazz music, 20.6%. In addition, the average SIR of the proposed method exhibits much a higher value than the uniform sparsity SNMF2D. This clearly shows that the amount of interference between any two sources is lesser for the proposed method. Therefore, the above results unanimously indicate that the proposed adaptive sparsity method leads to higher ESE results than the uniform constant sparsity method.

4.3 Comparison with other sparse NMF-based SCASS methods

In Section IV B, analysis has been carried out to investigate effects between adaptive sparsity and uniform constant sparsity on source separation. In this evaluation, we compare the proposed method with other sparse NMF-based source separation methods. These consist of the followings:

- SNMF [13]. The uniform constant sparsity parameter is progressively varied from 0 to 10 with every increment of 0.1 (i.e. $\lambda = 0, 0.1, 0.2, \ldots, 10$) and the best result is retained for comparison.
- Automatic Relevance Determination NMF (NMF-ARD) [35] exploits a hierarchical Bayesian framework SNMF that amounts to imposing an exponential prior for pruning and thereby enables estimation of the NMF model order. The NMF-ARD assumes prior on $\mathbf{H}$, namely, $p(\mathbf{H}|\lambda) = \prod_d \lambda_d^{l_{\max}} \exp-\left(\lambda_d \sum_l \mathbf{H}_{d,l}\right)$ and uses Automatic Relevance Determination (ARD) approach to determine the desirable number of components in $\mathbf{D}$.
- NMF with Temporal Continuity and Sparseness Criteria [14] (NMF-TCS) is based on factorizing the magnitude spectrogram of the mixed signal into a sum of components, which include the temporal continuity and sparseness criteria into the separation framework. In [14], the temporal continuity α is chosen as $[0,1,10,100,1000]$, sparseness weight β is chosen as $[0,1,10,100,1000]$. The best separation result is retained for comparison.

Figure 9 summarizes the SDR comparison results between our proposed method and the above three sparse NMF methods. From the results, it can be seen that the above methods fail to take into account the relative position of each spectrum and thereby discarding the temporal information. Better separation results will require a proper model that can represent both temporal structure and the pitch change which occurs when an instrument plays different notes simultaneously. If the temporal structure and the pitch change are not considered in the model, the mixing ambiguity is still contained in each separated source.

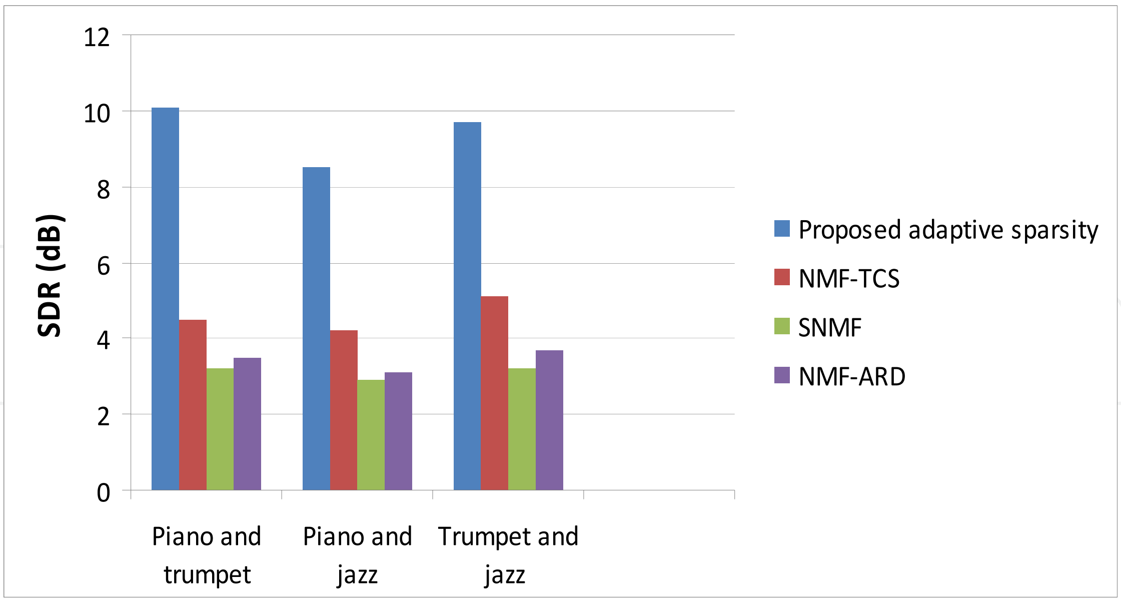

Fig. 9. Performance comparison between other NMF based SCASS methods and proposed method

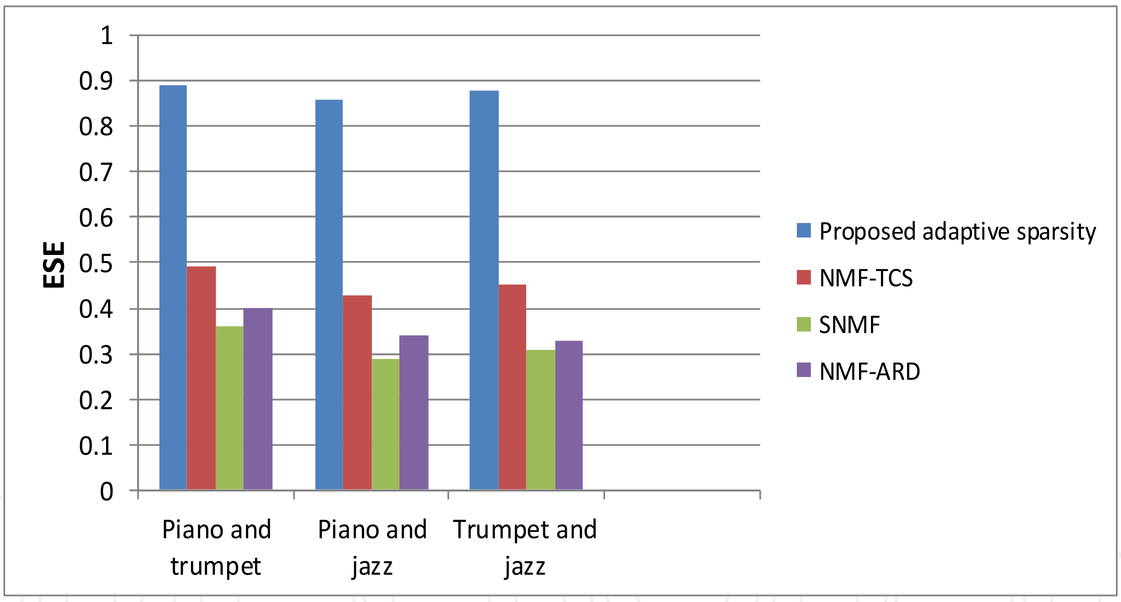

Fig. 10. ESE comparison between other NMF based SCASS methods and the proposed method

The improvement of our method compared with NMF-TCS, SNMF and NMF-ARD can be summarized as follows: (i) For the piano and trumpet music, the average improvement per source in terms of the SDR is 6.3dB. (ii) For the piano and jazz music, the average improvement per source in terms of SDR is 5dB. (iii) For the trumpet and jazz music, the average improvement per source in terms of SDR is 5.4dB. In the case of ESE (Figure 10), the proposed method exhibits much better average ESE of approximately 106.9%, 138.8% and 114.6% improvement with NMF-TCS, SNMF and NMF-ARD, respectively. Analyzing the separation results and ESE performance, the proposed method leads to the best separation performance for both recovered sources. The SNMF method performs with poorer results whereas the separation performance by the NMF-TCS method is slightly better than the NMF-ARD and SNMF methods. Our proposed method gives significantly better performance than the NMF-TCS, SNMF and NMF-ARD methods. The spectral dictionary obtained via NMF-TCS, SNMF and NMF-ARD methods are not adequate to capture the temporal dependency of the frequency patterns within the audio signal. In addition, the NMF-TCS, SNMF and NMF-ARD do not model notes but rather unique events only. Thus if two notes are always played simultaneously they will be modeled as one component. Also, some components might not correspond to notes but rather to the model e.g. background noise.

4.4 Comparison with underdetermined-based ICA SCSS method

In the underdetermined-ICA SCSS method [3], the key point is to exploit the prior knowledge of the sources such as the basis functions to generate the sparse codes. In this work, these basis functions are obtained in two stages: (i) Training stage: the basis functions are obtained by performing ICA on each concatenated sources. In this experiment, we derive a set of 64 basis functions for each type of source. These training data exclude the target sources which have been exclusively used to generate the mixture signals. (ii) Adaptation stage: the obtained ICA basis functions from the training stage are further adapted based on the current estimated sources during the separation process. In this method, both the estimated sources and the ICA basis functions are jointly optimized by maximizing the log-likelihood of the current mixture signal until it converges to the steady-state solution. If two sets of basis functions overlap significantly with each other, the underdetermined-ICA SCSS method is less efficient in resolving the mixing ambiguity between sources. The improvement of proposed method compared with underdetermined-ICA SCSS method can be summarized as follows: (i) For the piano and trumpet music, the average improvement per source in terms of the SDR is 4.3dB. (ii) For the piano and jazz music, the average improvement per source in terms of SDR is 4dB. (iii) For the trumpet and jazz music, the average improvement per source in terms of SDR is 4.2dB.

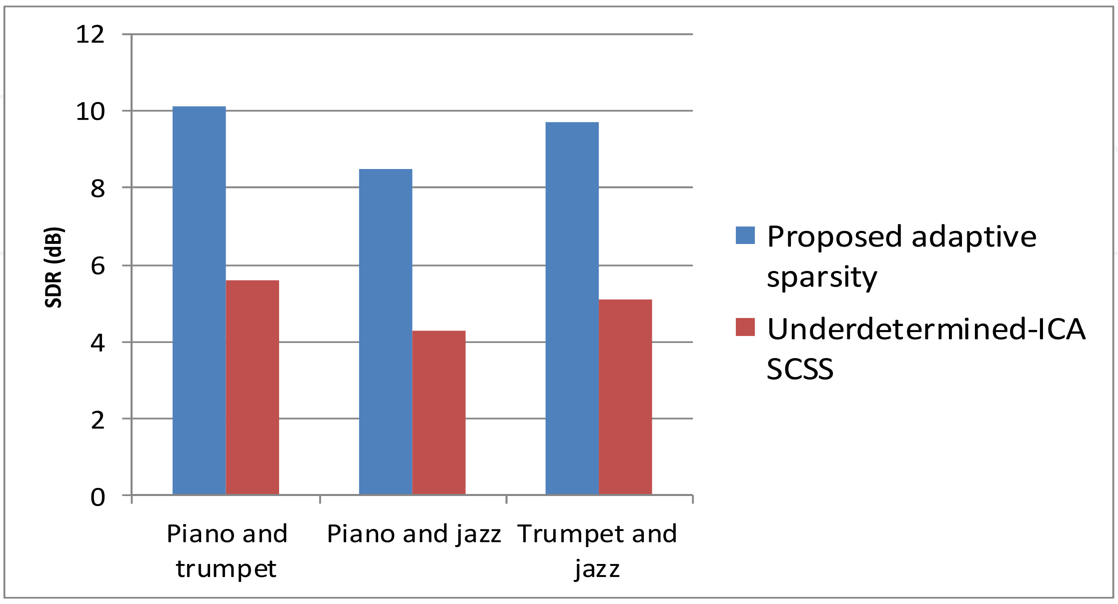

Fig. 11. Performance underdetermined-ICA SCSS method and proposed method

The performance of the underdetermined-ICA SCSS method relies on the ICA-derived time domain basis functions. High level performance can be achieved only when the basis functions of each source are sufficiently distinct. However, the result becomes considerably less robust in separating mixture where the original sources are of the same type e.g. mixture of music with music.

5. Conclusion

The chapter has presented an adaptive strategy to sparsifying the non-negative matrix factorization. The impetus behind this work is that the sparsity achieved by conventional SNMF and SNMF2D is not enough; in such situations it might be useful to control the degree of sparseness explicitly. In the proposed method, the regularization term is adaptively tuned using a variational Bayesian approach to yield desired sparse decomposition, thus enabling the spectral dictionary and temporal codes of non-stationary audio signals to be estimated more efficiently. This has been verified concretely based on our simulation results. In addition, the proposed method has yielded significant improvements in single channel music separation when compared with other sparse NMF-based source separation methods. Future work could investigate the extension of the proposed method to separate non-stationary (here non-stationary refers to the sources not located in the fixed places, e.g. the speakers are talking while on the move) and reverberant mixing model. For non-stationary reverberant mixing model, this gives

$$y(t) = \sum_{i=1}^{N_s} \sum_{\tau_r=0}^{L_r-1} m_i(\tau_r, t) x_i(t - \tau_r) + n(t)$$

where $m_i(\tau_r, t)$ is the finite impulse response of causal filter at t time and τ_r is the time delay. The expanded adaptive sparsity non-negative matrix factorization can then be developed to estimate mixing $\tilde{m}_i$ and sources $\tilde{x}_i$, respectively.

6. References

[1] Radfa M.H, Dansereau R.M. Single-channel speech separation using soft mask filtering. IEEE Trans. on Audio, Speech and Language Processing. 2007; 15: 2299-2310.

[2] Ellis D. Model-based scene analysis, in Computational Auditory Scene Analysis: Principles, Algorithms, and Applications, D. Wang and G. Brown, Eds. New York: Wiley/IEEE Press; 2006

[3] Jang G.J, Lee T.W. A maximum likelihood approach to single channel source separation. Journal of Machine Learning Research. 2003; 4: 1365–1392.

[4] Li P, Guan Y, Xu B, Liu W. Monaural speech separation based on computational auditory scene analysis and objective quality assessment of speech. IEEE Trans. on Audio, Speech and Language Processing. 2006; 14: 2014–2023.

[5] Paatero P, Tapper U. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics. 1994; 5: 111–126.

[6] Ozerov A, Févotte C. Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. IEEE Transactions on Audio, Speech, and Language Processing 2010; 18: 550-563.

[7] Casey M. A, Westner A. Separation of mixed audio sources by independent subspace analysis. proceeding of. Int. Comput. Music Conf, 2000. 154–161; 2000

[8] Molla Md. K. I, Hirose K. Single-Mixture Audio Source Separation by Subspace Decomposition of Hilbert Spectrum. IEEE Trans. on Audio, Speech and Language Processing. 2007; 15: 893–900.

[9] Hyvarinen A, Karhunen J, Oja E, Independent component analysis and blind source separation, John Wiley & Sons 2005. p.20–60.

[10] Lee D, Seung H. Learning the parts of objects by nonnegative matrix factorisation. Nature. 1999; 401: 788–791.

[11] Kompass R. A generalized divergence measure for nonnegative matrix factorization. Neural Computation. 2007; 19: 780-791.

[12] Cichocki A, Zdunek R, Amari S.I. Csisz´ar's divergences for non-negative matrix factorization: family of new algorithms. Proceeding of. Intl. Conf. on Independent Component Analysis and Blind Signal Separation (ICABSS'06), Charleston, USA, March 2006, 3889: 32–39. 2006.

[13] Hoyer P. O. Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research. 2004; 5: 1457–1469.

[14] Virtanen T (2007) Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria. IEEE Transactions on Audio, Speech, and Language Processing. 2007; 15: 1066–1074.

[15] Vincent E (2006) Musical source separation using time-frequency source priors. IEEE Trans. Audio, Speech and Language Processing. 2006; 14: 91–98.

[16] Ozerov A, Févotte C. Multichannel nonnegative matrix factorization in convolutive mixtures. With application to blind audio source separation. Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'09), 3137-3140. 2009.

[17] Mysore G, Smaragdis P, Raj B. Non-negative hidden Markov modeling of audio with application to source separation. Proceeding of 9th international conference on Latent Variable Analysis and Signal Separation (LCA/ICA). 2010.

[18] Nakano *M*, et al. Nonnegative Matrix Factorization with Markov-chained Bases for Modeling Time-varying in Music Spectrograms. Proceeding of 9th international conference on Latent Variable Analysis and Signal Separation (LCA/ICA). 2010

[19] Salakhutdinov R, Mnih A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. Proceedings of the 25th international conference on Machine learning. 880-887. 2008.

[20] Cemgil A. T. Bayesian inference for nonnegative matrix factorisation models. Computational Intelligence and Neuroscience. 2009; doi: 10.1155/2009/785152.

[21] Moussaoui S, Brie D, Mohammad-Djafari A, Carteret C, Separation of non-negative mixture of non-negative sources using a Bayesian approach and MCMC sampling. IEEE Trans. on Signal Processing. 2006; 54: 4133–4145.

[22] Schmidt M. N, Winther O, Hansen L.K. Bayesian non-negative matrix factorization. Proceeding of Independent Component Analysis and Signal Separation, International Conference. 2009

[23] Morup M, Schmidt M.N. Sparse non-negative matrix factor 2-D deconvolution. Technical University of Denmark, Copenhagen, Denmark. 2006.

[24] Schmidt M.N, Morup M. Nonnegative matrix factor 2-D deconvolution for blind single channel source separation. Proceeding of Intl. Conf. Independent Component Analysis and Blind Signal Separation (ICABSS'06), Charleston, USA. 3889: 700–707. 2006.

[25] Lin Y. Q. l_1-norm sparse Bayesian learning: theory and applications. *Ph.D. Thesis, University of Pennsylvania*. 2008.

[26] Gao Bin, Woo W.L, Dlay S.S. Single Channel Source Separation Using EMD-Subband Variable Regularised Sparse Features. IEEE Trans. on Audio, Speech, and Language Processing. 2011; 19: 961–976.

[27] Gao Bin, Woo W.L, Dlay S.S. Adaptive Sparsity Non-negative Matrix Factorization for Single Channel Source Separation. IEEE the Journal of Selected Topics in Signal Processing. 2011; 5: 1932-4553.

[28] Sha F, Saul L.K, Lee D.D. Multiplicative updates for nonnegative quadratic programming in support vector machines. Proceeding of Advances in Neural Information Process. Systems. 15: 1041–1048. 2002

[29] Brown J. C. Calculation of a constant Q spectral transform. *J. Acoust. Soc. Am.* 1991; 89: 425–434.

[30] Yilmaz O, Rickard S. Blind separation of speech mixtures via time-frequency masking. IEEE Trans. Signal Processing. 2004; 52: 1830–1847.

[31] Goto M, Hashiguchi H, Nishimura T, Oka R. RWC music database: Music genre database and musical instrument sound database. in Proc. of Intl. Symp. on Music Information Retrieval (ISMIR), Baltimore, Maryland, USA. 229–230. 2003.

[32] Signal Separation Evaluation Campaign (SiSEC 2008), (2008). [Online]. Available: http://sisec.wiki.irisa.fr

[33] Vincent E, Gribonval R, Fevotte C. Performance measurement in blind audio source separation. IEEE Trans. Speech Audio Process. 2006; 14: 1462–1469.

[34] Wang D. L. On ideal binary mask as the computational goal of auditory scene analysis. in Speech Separation by Humans and Machines, P. Divenyi, Ed. Norwell, MA: Kluwer, pp. 181–197. 2005.

[35] Mørup M, Hansen K.L. Tuning Pruning in Sparse Non-negative Matrix Factorization. Proceeding of 17th European Signal Processing Conference (EUSIPCO'2009), Glasgow, Scotland. 2009.