A diagonal plus low-rank covariance model for computationally efficient source separation

Antoine Liutkus, Kazuyoshi Yoshii

2017 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 25–28, 2017, TOKYO, JAPAN

A DIAGONAL PLUS LOW-RANK COVARIANCE MODEL FOR COMPUTATIONALLY EFFICIENT SOURCE SEPARATION

Antoine Liutkus *

Inria, Speech Processing Team, France
antoine.liutkus@inria.fr

Kazuyoshi Yoshii †

Kyoto University/RIKEN AIP, Japan
yoshii@kuis.kyoto-u.ac.jp

ABSTRACT

This paper presents an accelerated version of positive semidefinite tensor factorization (PSDTF) for blind source separation. PSDTF works better than nonnegative matrix factorization (NMF) by dropping the arguable assumption that audio signals can be whitened in the frequency domain by using short-term Fourier transform (STFT). Indeed, this assumption only holds true in an ideal situation where each frame is infinitely long and the target signal is completely stationary in each frame. PSDTF thus deals with full covariance matrices over frequency bins instead of forcing them to be diagonal as in NMF. Although PSDTF significantly outperforms NMF in terms of separation performance, it suffers from a heavy computational cost due to the repeated inversion of big covariance matrices. To solve this problem, we propose an intermediate model based on diagonal plus low-rank covariance matrices and derive the expectation-maximization (EM) algorithm for efficiently updating the parameters of PSDTF. Experimental results showed that our method can dramatically reduce the complexity of PSDTF by several orders of magnitude without a significant decrease in separation performance.

Index Terms— Blind source separation, nonnegative matrix factorization, positive semidefinite tensor factorization, low-rank approximation.

1. INTRODUCTION

A major approach to blind source separation of single-channel audio signals in the last decade is to use nonnegative matrix factorization (NMF) and Wiener filtering by assuming that all the time-frequency bins in the short-time Fourier transform (STFT) domain are independent from each other [1–3]. In NMF, given a set of nonnegative vectors as input data, each vector is approximated by the weighted sum of nonnegative basis vectors. In audio source separation, a mixture spectrum (magnitude, power, or α-fractional power spectrum [4, 5]) is approximated as the weighted sum of source spectra at each frame. Many variants of NMF have been proposed by designing various cost functions that evaluate the approximation error at each time-frequency bin [6]. Among these, NMF based on the Kullback-Leibler (KL) [7] or β-divergence [8, 9] have been empirically known to work well.

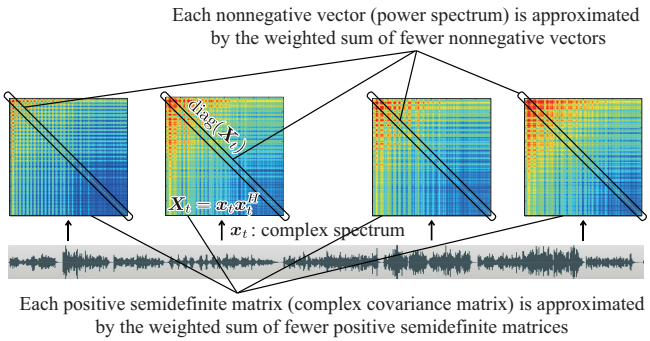

Fig. 1. Comparion between LD-PSDTF and IS-NMF.

NMF based on the Itakura-Saito (IS) divergence is known to be theoretically justified for audio source separation under the assumption that all the coefficients of the sources STFT are independent and isotropic Gaussian [8]. In this context, it can indeed be interpreted as the maximum likelihood estimation of the sources variances, that are called power spectral densities. Separation through posterior inference of the Gaussian sources conditionally on the mixture then coincides with a frame-wise Wiener filter. A serious limitation of this approach is that, while the magnitude of the input mixture STFT is partitioned to the different sources, all sources get the same original phase information. It is thus difficult to resynthesize high-quality time-domain source signals.

To overcome this limitation, positive semidefinite tensor factorization (PSDTF) based on the log-det (LD) divergence has recently been proposed [10] by assuming that the source complex spectrum at each frame is multivariate complex Gaussian distributed. In PSDTF, given a set of positive semidefinite (PSD) matrices as input data, each PSD matrix is approximated by the weighted sum of PSD basis matrices (Fig. 1). Focusing on the fact that the positive semidefiniteness of matrices is an extended concept of the nonnegativity of vectors,

*This work was partly supported by the research programmes KAMoulox (ANR-15-CE38-0003-01) funded by ANR, the French State agency for research.

†This work was partly supported ACCEL No. JPMJAC1602 and KAKENHI Nos. 26700020 and 16H01744 funded by JST, Japan.

LD-PSDTF is a natural extension of IS-NMF. More specifically, each PSD matrix of input data is given by calculating the product of the complex spectrum and its conjugate transpose. NMF, on the other hand, focuses on only the diagonal elements of the PSD matrix, i.e., nonnegative vector (power spectrum), by ignoring the correlations between frequency bins. PSDTF can estimate the complex spectrum of each source with phase information by processing jointly all the frequencies in an interdependent manner. Since such phase-aware frequency-domain decomposition was shown to correspond to time-domain decomposition, PSDTF can recover high-quality time-domain source signals. Note that recently-proposed phase-aware or time-domain decomposition methods based on additivity of complex spectra or audio signals do not take into account inter-frequency correlations [11–13] (high-resolution NMF [11] deals with inter-frame dependency based on autoregressive modeling).

A critical problem of LD-PSDTF, however, lies in extremely large computational cost, which hinders its use in practice. When a mixture spectrogram with T frames and F frequency bins is analyzed, the time complexity of LD-PSDTF with K basis matrices is $\mathcal{O}(KTF^3)$ while that of IS-NMF with K basis vectors is only $\mathcal{O}(KTF)$. LD-PSDTF is performed with a convergence-guaranteed iterative optimization method that needs to repeatedly calculate the inversion of very large matrices of size $F \times F$. The matrix inversion costs $\mathcal{O}(F^3)$ and is numerically unstable because F (window size) is usually several thousands. If all the input and basis matrices are restricted to diagonal matrices by discarding the correlations between frequency bins, i.e., LD-PSDTF reduces to IS-NMF, the matrix inversion costs only $\mathcal{O}(F)$ but the separation performance is deteriorated accordingly.

To solve this problem, we propose a constrained version of LD-PSDTF that represents each basis matrix as the sum of a diagonal matrix and a low-rank matrix (Fig. 2). This contribution is inspired by the factor analysis model presented in [14], but goes further by adopting such a structure for each source. If the rank of the low-rank matrix is $N \ll F$, the time complexity of our model is $\mathcal{O}(KTF^2N)$. If a basis matrix represents a harmonic musical instrument sound, the frequency bins corresponding to the harmonic partials are highly correlated to each other. This implies that the rank N can be reduced to around the number of harmonic partials without sacrificing separation performance too much. As either an expectation-maximization (EM) algorithm or an auxiliary-function-based (minorization-maximization) method can be used for IS-NMF, in this paper we derive an EM-based method with fast inversion of diagonal plus low-rank basis matrices. The main contribution of this paper is to propose an efficient and accurate approximation to full LD-PSDTF by leveraging the characteristics of basis matrices. This appears important because PSDTF has a great potential for fundamentally raising the performance of any audio analysis methods using NMF if the problem of computational cost is solved.

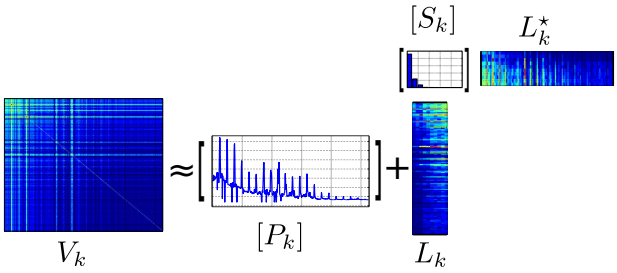

Fig. 2. Diagonal plus low-rank approximation of a full covariance matrix over frequency bins. $\boldsymbol{L}_k$ is $F \times N$ with $N \ll F$. $[v]$ is a diagonal matrix with diagonal v.

2. LOG-DET POSITIVE SEMIDEFINITE TENSOR FACTORIZATION (LD-PSDTF)

In this section, we review log-det positive semidefinite tensor factorization (LD-PSDTF) for audio source separation from the viewpoint of probabilistic modeling and newly derive the expectation-maximization (EM) algorithm for PSDTF.

2.1. Model formulation

All the signals are represented in the STFT domain, with F non-redundant frequency bins and T frames. In that domain, for one given frame t, the K source signals are $F \times 1$ complex vectors written $\boldsymbol{x}_{kt}$. We take them as all independent and distributed according to the LD-PSDTF model as follows:

$$\boldsymbol{x}_{kt} \sim \mathcal{N}_c\left(0, \boldsymbol{Y}_{kt}\right), \tag{1}$$

where $\mathcal{N}_c$ indicates the multivariate complex Gaussian distribution and $\boldsymbol{Y}_{kt}$ is the covariance matrix. $\boldsymbol{Y}_{kt}$ is assumed to the time-varying scaled version of a time-invariant $F \times F$ *covariance template* $\boldsymbol{V}_k \succeq 0$ as follows:

$$\boldsymbol{Y}_{kt} = h_{kt}\boldsymbol{V}_k,$$

with $h_{kt} \geq 0$ being an activation gain for source k at frame t. As demonstrated in [10], LD-PSDTF generalizes IS-NMF [8] for which all entries of $\boldsymbol{x}_{kt}$ are assumed to be independent, resulting in diagonal $\boldsymbol{V}_k$. The major advantage of LD-PSDTF is to overcome the limitation of this independence assumption. Since the perfect stationarity of the waveforms does not hold true and finite-size frames are used in practice, inter-frequency covariances cannot be avoided.

Now, we model the $F \times 1$ mixture $\boldsymbol{x}_t$ at frame t as the sum of the sources as follows:

$$\boldsymbol{x}_t = \sum_{k=1}^{K} \boldsymbol{x}_{kt}.$$

As a sum of independent Gaussian random vectors, $\boldsymbol{x}_t$ also has a Gaussian distribution as follows:

$$\boldsymbol{x}_t \sim \mathcal{N}_c\left(0, \boldsymbol{Y}_t = \sum_{k=1}^{K} \boldsymbol{Y}_{kt}\right).$$

This is the likelihood function of LD-PSDTF. Its maximization is known to be equivalent to the minimization of the log-

det divergence between $\boldsymbol{X}_t = \boldsymbol{x}_t \boldsymbol{x}_t^H$ and $\boldsymbol{Y}_t$ [10]. Since a complex spectrum $\boldsymbol{x}_t$ is the linear transform of a time-domain signal, such frequency-domain Gaussian modeling can be represented in the time domain.

2.2. Parameter estimation

To estimate the covariance templates $\boldsymbol{V}_k$ as well as the activation gains h_{kt}, we propose an generalized EM algorithm whose E-step and M-step are iterated until convergence. Note that we could use the auxiliary function approach presented in [10]. Let $\boldsymbol{\Theta}$ denote the set of all parameters. The whole procedure is summarized in Algorithm 1 and detailed below.

2.2.1. E-step

The posterior distribution of the sources $\boldsymbol{x}_{kt}$ is computed. The posterior of Gaussian random variables is given by

$$\boldsymbol{x}_{kt} \mid \boldsymbol{\Theta}, x \sim \mathcal{N}_c\left(\hat{\boldsymbol{x}}_{kt}, \boldsymbol{C}_{kt}\right), \tag{2}$$

where the posterior mean $\hat{\boldsymbol{x}}_{kt}$ and covariance $\boldsymbol{C}_{kt}$ are of dimension $F \times 1$ and $F \times F$, respectively, and given by

$$\hat{\boldsymbol{x}}_{kt} = \boldsymbol{W}_{kt} \boldsymbol{x}_t, \tag{3}$$

$$\boldsymbol{C}_{kt} = \boldsymbol{Y}_{kt} - \boldsymbol{W}_{kt} \boldsymbol{Y}_{kt}, \tag{4}$$

with $\boldsymbol{W}_{kt}$ being the $F \times F$ Wiener gain for source k at frame t. In full generality, it is given by

$$\boldsymbol{W}_{kt} = \boldsymbol{Y}_{kt} \boldsymbol{Y}_t^{-1}. \tag{5}$$

The total posterior covariance of $\boldsymbol{x}_{kt}$ is thus

$$\mathbb{E}\left[\boldsymbol{x}_{kt} \boldsymbol{x}_{kt}^{\star} \mid \boldsymbol{\Theta}, \boldsymbol{x}\right] \triangleq \boldsymbol{\Sigma}_{kt} = \hat{\boldsymbol{x}}_{kt} \hat{\boldsymbol{x}}_{kt}^{\star} + \boldsymbol{C}_{kt}. \tag{6}$$

2.2.2. M-step

Given the total posterior covariances $\boldsymbol{\Sigma}_{kt}$ for the sources computed in the E-step, h_{kt} and $\boldsymbol{V}_k$ are alternatively updated. A straightforward option to estimate the activations for source k is to exploit the posterior distribution of $\boldsymbol{x}_{kt} \mid \boldsymbol{\Theta}, \boldsymbol{x}$ as achieved in (4). We then estimate h_{kt} as:

$$h_{kt} \leftarrow \frac{\operatorname{tr}\left(\boldsymbol{V}_k^{-1} \boldsymbol{\Sigma}_{kt}\right)}{F}. \tag{7}$$

Note that this requires the inversion of the $F \times F$ matrix $\boldsymbol{V}_k$. This inversion needs to be done once per iteration though, and not for all frames. The update for the covariance template $\boldsymbol{V}_k$ can also be derived when the posterior distribution for the sources has been computed as follows:

$$\boldsymbol{V}_k \leftarrow \frac{1}{T} \sum_{t=1}^{T} \frac{\boldsymbol{\Sigma}_{kt}}{h_{kt}}. \tag{8}$$

Note that the likelihood function is not directly maximized in the M-step. Using the interdependent steps (7) and (8), the likelihood function is incrementally maximized.

2.3. Computational complexity

We now roughly estimate the time complexity of the EM algorithm. The iterative algorithm proposed here comprises several demanding operations, in particular:

- Inversion of all $F \times F$ covariance matrices $\boldsymbol{Y}_t$: $\mathcal{O}\left(TF^3\right)$.
- Computation of all $\boldsymbol{W}_{kt}$ and $\boldsymbol{W}_{kt} \boldsymbol{Y}_{kt}$: $\mathcal{O}\left(KTF^3\right)$.

Accounting only for these most demanding operations, the total computational complexity $\boldsymbol{C}_b$ for each iteration of this *baseline* algorithm is thus[1]:

$$\boldsymbol{C}_b = \mathcal{O}\left((K+1)TF^3\right). \tag{9}$$

In this paper, we propose a refinement of LD-PSDTF that permits to dramatically reduce this computational load.

Algorithm 1 EM algorithm for LD-PSDTF.

Input: x, K, initializations for $\boldsymbol{V}_k$ and h_{kt}
Iterate until convergence:

1. **E-step**: compute all Σ_{kt} as in (6)
2. **M-step**: update h_{kt} with (7) and $\boldsymbol{V}_k$ with (8).

Separate: compute the sources $\hat{\boldsymbol{x}}_{kt}$ as in (4)

3. FAST LD-PSDTF

This section explain the proposed refinement of LD-PSDTF based on diagonal plus low-rank covariance matrices.

3.1. Diagonal plus low-rank approximation

Instead of leaving the covariance templates $\boldsymbol{V}_k$ totally unconstrained as done in [10], we assume that they can be approximated as the sum of low-rank and diagonal matrices (Fig. 2):

$$\boldsymbol{V}_k = \left[\boldsymbol{P}_k\right] + \boldsymbol{L}_k \left[\boldsymbol{S}_k\right] \boldsymbol{L}_k^{\star}, \tag{10}$$

where we write $[v]$ as the diagonal matrix with vector v as its diagonal. $\boldsymbol{L}_k$ is a $F \times N$ matrix, while $\boldsymbol{P}_k$ and $\boldsymbol{S}_k$ are $F \times 1$ and $N \times 1$ vectors, respectively, with $N \ll F$. Note that we introduced $\boldsymbol{S}_k$ to make parameter estimation easy (see Section 3.2.2).

We call this model (10) *structured*. It is for instance already reviewed in [14] under the name of *factor analysis*. Its rationale is to allow for the dependency structure between frequencies to be correctly explained by a limited amount of correlations. In other words, while $\boldsymbol{P}_k$ may be roughly understood as the power spectral density (PSD) of the stochastic part of source k as in IS-NMF, $\boldsymbol{L}_k \left[\boldsymbol{S}_k\right] \boldsymbol{L}_k^{\star}$ rather stands for its sinusoidal part, which introduces deterministic relations between the entries of $\boldsymbol{x}_{kt}$. Then, these two components are coupled because they share the activation gains h_{kt}. Interestingly, if we take $N = 0$, this model reduces to IS-NMF and coincides with PSDTF for $N = F$.

Now, taking (10) as the covariance templates, the mixture covariance matrices $\boldsymbol{Y}_t$ write:

$$\boldsymbol{Y}_t = \left[\sum_{k=1}^{K} h_{kt} \boldsymbol{P}_k\right] + \sum_{k=1}^{K} h_{kt} \boldsymbol{L}_k \left[\boldsymbol{S}_k\right] \boldsymbol{L}_k^{\star}.$$

[1] Note that the computational complexity of the method proposed in [10] is comparable to that of the proposed EM algorithm. Even if it doesn't require computation of $\boldsymbol{\Sigma}_{kt}$, it requires $\boldsymbol{Y}_t^{-1} \boldsymbol{V}_k$, also scaling as $\mathcal{O}\left(KTF^3\right)$.

The purpose of this paper is to study the consequences of model (10) over both computational load and performance in audio separation. For notational convenience below, and for a given $p = 1, \ldots, K$, we define the $F \times F$ matrix $\boldsymbol{M}_{p,t}$ as:

$$\boldsymbol{M}_{p,t} = \left[\sum_{k=1}^{K} h_{kt}\boldsymbol{P}_k\right] + \sum_{k=1}^{p} h_{kt}\boldsymbol{L}_k\left[\boldsymbol{S}_k\right]\boldsymbol{L}_k^{\star}.$$

With the structured model, the parameters $\boldsymbol{P}_k$, $\boldsymbol{S}_k$, and $\boldsymbol{L}_k$ replace the unconstrained $\boldsymbol{V}_k$, leading to $K\left(F+N+FN\right)$ parameters for the templates instead of $KF\left(F+1\right)/2$. The special structure (10) chosen for the covariance templates $\boldsymbol{V}_k$ may be understood as stating each source itself is the sum of a *stochastic* component $\boldsymbol{x}_{kt}^{(s)}$ with diagonal covariance $\boldsymbol{Y}_{kt}^{(s)} = [h_{kt}\boldsymbol{P}_k]$ and an independent *deterministic* component $\boldsymbol{x}_{kt}^{(d)}$ with low-rank covariance $\boldsymbol{Y}_{kt}^{(d)} = \boldsymbol{L}_k\left[h_{kt}\boldsymbol{S}_k\right]\boldsymbol{L}_k^{\star}$. For the re-estimation of these parameters, we modify the EM algorithm in the following way.

3.2. Paramter estimation

We explain an acceralated version of the generalized EM algorithm proposed in Section 2.2.

3.2.1. Modified E-step

We need to compute the total posterior covariances of $\boldsymbol{x}_{kt}^{(s)}$ and $\boldsymbol{x}_{kt}^{(d)}$. For this purpose, and for each component $c = s$ or d, we need the respective Wiener gains $\boldsymbol{W}_{kt}^{(c)} = \boldsymbol{Y}_{kt}^{(c)}\boldsymbol{Y}_t^{-1}$ and the total posterior covariance as follows:

$$\text{for } c \in \{s,d\}, \boldsymbol{\Sigma}_{kt}^{(c)} = \hat{\boldsymbol{x}}_{kt}^{(c)}\left(\hat{\boldsymbol{x}}_{kt}^{(c)}\right)^{\star} + \boldsymbol{Y}_{kt}^{(c)} - \boldsymbol{W}_{kt}^{(c)}\boldsymbol{Y}_{kt}^{(c)}.$$

Then, for the re-estimation of h_{kt}, we also need the total posterior covariance $\boldsymbol{\Sigma}_{kt}$ of $\boldsymbol{x}_{kt}$, which is different from $\boldsymbol{\Sigma}_{kt}^{(s)} + \boldsymbol{\Sigma}_{kt}^{(d)}$ because $\boldsymbol{x}_{kt}^{(s)}$ and $\boldsymbol{x}_{kt}^{(d)}$ are not independent conditionally on $\boldsymbol{x}_t$. It is straightforward to show that

$$\boldsymbol{\Sigma}_{kt} = \boldsymbol{\Sigma}_{kt}^{(s)} + \boldsymbol{\Sigma}_{kt}^{(d)} + \boldsymbol{\Sigma}_{kt}^{(ds)} + \boldsymbol{\Sigma}_{kt}^{(ds)\star}, \tag{11}$$

where

$$\boldsymbol{\Sigma}_{kt}^{(ds)} = \hat{\boldsymbol{x}}_{kt}^{(d)}\left(\hat{\boldsymbol{x}}_{kt}^{(s)}\right)^{\star} - \boldsymbol{W}_{kt}^{(d)}\boldsymbol{Y}_{kt}^{(s)}.$$

3.2.2. Modified M-step

The updates for $\boldsymbol{P}_k$ are given by:

$$\boldsymbol{P}_k \leftarrow \frac{1}{T}\sum_{t=1}^{T}\frac{\text{diag}\boldsymbol{\Sigma}_{kt}^{(s)}}{h_{kt}}. \tag{12}$$

The updates for $\boldsymbol{L}_k$ and $\boldsymbol{S}_k$ are obtained straightforwardly by the truncated eigenvalue decomposition of the weighted average of $\boldsymbol{\Sigma}_{kt}^{(d)}$ as follows:

$$\boldsymbol{L}_k, \boldsymbol{S}_k \leftarrow \text{eig}\left(\frac{1}{T}\sum_{t=1}^{T}\frac{\boldsymbol{\Sigma}_{kt}^{(d)}}{h_{kt}}, N\right), \tag{13}$$

where $\text{eig}\left(M,N\right)$ gives the N-truncated eigenvalue decomposition of matrix M, i.e., the first N eigenvectors along with their eigenvalues. Efficient randomized algorithms [15] may be used to compute this decomposition with a time complexity scaling as $\mathcal{O}\left(F^2N\right)$ or even less.

Concerning the update for h_{kt}, it is identical to (7), except that we use the more efficient expression (11) for $\boldsymbol{\Sigma}_{kt}$.

Algorithm 2 Inversion of the mixture covariance matrix $\boldsymbol{Y}_t$ under the structured template model (10).

- $\boldsymbol{M}_{0,t}^{-1} \leftarrow \left[\frac{1}{\sum_k h_{kt}\boldsymbol{P}_k}\right]$
- For $k = 1 \ldots K$: Calculate (16) and (17).
- Return $\boldsymbol{Y}_t^{-1} = \boldsymbol{M}_{K,t}^{-1}$

3.2.3. Efficient matrix inversion

For inference in the EM algorithm (and also in the auxiliary function approach of [10]), we need to compute the inverses of the $F \times F$ mix covariance matrices $\boldsymbol{Y}_t$. This step is one of the computational bottlenecks of the method. In this paper, we use the proposed structured model (10) in conjunction with the Woodbury matrix identity [16], which states

$$\begin{aligned}&\left(\boldsymbol{A}+\boldsymbol{U}\boldsymbol{C}\boldsymbol{V}\right)^{-1}\\ &= \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}\left(\boldsymbol{C}^{-1}+\boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}\right)^{-1}\boldsymbol{V}\boldsymbol{A}^{-1},\end{aligned} \tag{14}$$

for any matrices $\boldsymbol{A}$, $\boldsymbol{U}$, $\boldsymbol{C}$, and $\boldsymbol{V}$ of appropriate size. The strategy we propose is to proceed iteratively.

Assuming we now have the inverse of $\boldsymbol{M}_{k-1,t}$, (which is a diagonal matrix for $k = 1$), we compute

$$\boldsymbol{M}_{k,t}^{-1} = \left(\boldsymbol{M}_{k-1,t} + h_{kt}\boldsymbol{L}_k\left[\boldsymbol{S}_k\right]\boldsymbol{L}_k^{\star}\right)^{-1}. \tag{15}$$

Using Woodbury matrix identity (14) on (15), we have

$$\boldsymbol{M}_{k,t}^{-1} = \boldsymbol{M}_{k-1,t}^{-1} - \boldsymbol{M}_{k-1,t}^{-1}\boldsymbol{L}_k\boldsymbol{\Omega}_{k,t}^{-1}\boldsymbol{L}_k^{\star}\boldsymbol{M}_{k-1,t}^{-1}, \tag{16}$$

$$\boldsymbol{\Omega}_{k,t} = \left(\left[\frac{1}{h_{kt}\boldsymbol{S}_k}\right] + \boldsymbol{L}_k^{\star}\boldsymbol{M}_{k-1,t}^{-1}\boldsymbol{L}_k\right), \tag{17}$$

where $\frac{1}{a}$ denotes entry-wise inversion for a vector a. The procedure is iterated up to $k = K$, as shown in the Algorithm 2.

Note that the same identity (14) may be applied to compute $\boldsymbol{V}_k^{-1}$ in a computationally efficient manner for the update of h_{kt} in (7) as follows:

$$\boldsymbol{V}_k^{-1} = \left[\frac{1}{\boldsymbol{P}_k}\right] \tag{18}$$

$$\left(\boldsymbol{I} - \boldsymbol{L}_k\left(\left[\frac{1}{\boldsymbol{S}_k}\right] + \boldsymbol{L}_k^{\star}\left[\frac{1}{\boldsymbol{P}_k}\right]\boldsymbol{L}_k\right)^{-1}\boldsymbol{L}_k^{\star}\left[\frac{1}{\boldsymbol{W}_k}\right]\right). \tag{19}$$

3.3. Computational complexity

It can be shown that the computational complexity of each of the K iterations of the procedure presented in the algorithm box 2 in this structured case is $\mathcal{O}\left(N\left(F^2+FN+N^2\right)\right)$. Then, considering our modifications to the generalized EM algorithm, we also need the add the following computationally most demanding operations:

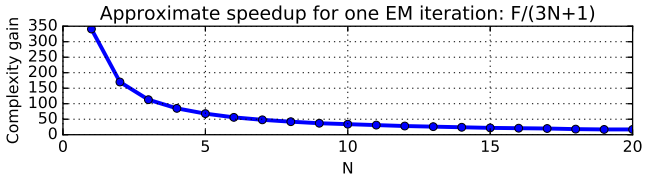

Fig. 3. Theoretical speedup $\boldsymbol{C}_b/\boldsymbol{C}_s$ over baseline PSDTF yield by the proposed inversion procedure permitted by the structured covariance model (10), for $K = 5$ sources and $F = 1024$ frequency bins. Observing this speedup in practice requires careful implementation of the inversion algorithm 2.

- **E-step:** $\mathcal{O}\left(F^2\right)$ for each $\boldsymbol{W}_{kt}^{(s)}$ and $\boldsymbol{\Sigma}_{kt}^{(s)}$. $O\left(NF^2\right)$ for each $\boldsymbol{W}_{kt}^{(d)}$ and $\boldsymbol{\Sigma}_{kt}^{(d)}$, as well as for $\boldsymbol{\Sigma}_{kt}$.
- **M-step:** $\mathcal{O}\left(TF^2 + F^2N\right)$ for the update of $\boldsymbol{L}_k$ and $\boldsymbol{S}_k$.

Keeping only the computations that need to be done for each frame, this brings the complexity of each iteration roughly down to

$$\boldsymbol{C}_s = \mathcal{O}\left(3KTF^2\left(N+1\right)\right). \tag{20}$$

In Fig. 3, we display the expected improvement $\boldsymbol{C}_b/\boldsymbol{C}_s$ in complexity as a function of N for fixed F and K. We check for actual improvements with our implementation in the next section. The effect of the proposed method naturally increases linearly with the number of frequency bands considered.

4. EVALUATION

In this section, we show that the simplified structured covariance model (10) leads to performance similar to the unconstrained PSDTF model [10], while allowing for a significant computational speedup.

4.1. Experimental conditions

We consider the same evaluation material as presented in [10], i.e., 4 synthetic mixtures, each of which being composed of $K = 3$ instrumental notes first presented separately, and then combined as various chords. Interestingly, the notes have a significant amount of overlap in the frequency domain. This dataset does not correspond to a real challenge in terms of modern source separation technology (see for instance the latest SiSEC report [17] for that matter). However, they are sufficient for our purpose because our objective here is indeed only to assess whether the proposed structured model leads to a degradation in performance compared to PSDTF, as a price for its computational effectiveness. Comparing both methods on the same dataset seems sufficient for this.

In terms of metrics, we compute the Source to Distortion Ratio (SDR [18]) between the true source signals and the estimates, both for PSDTF and for the fast PSDTF we propose, abbreviated as fPSDTF here. We display the difference between these two scores for all mixtures as a function of the order N of the approximation in Fig. 4. Then, for one of the mixture, we display the evolution of this difference over the iterations of the EM algorithm in Fig. 5.

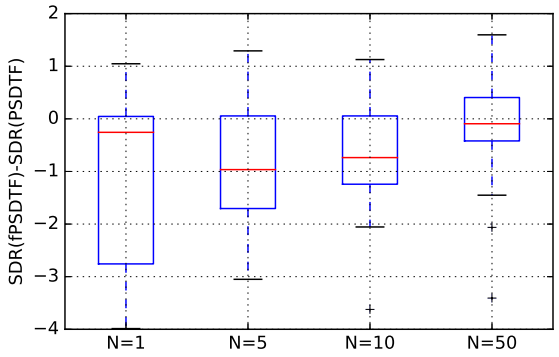

Fig. 4. Performance increase yield by the proposed approximation fPSDTF over baseline PSDTF, for different values of the low-rank component order N.

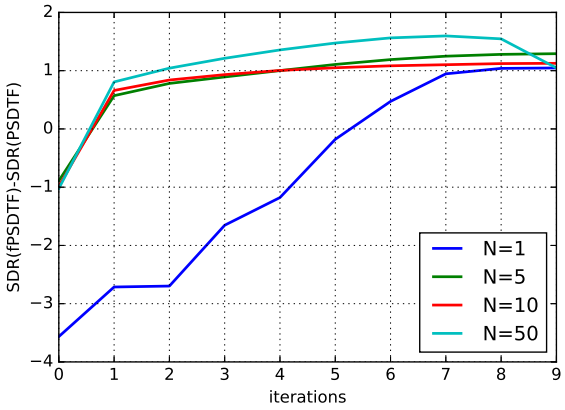

Fig. 5. Performance increase of fPSDTF over PSDTF vs the number of iterations for one particular mixture and different orders N of the deterministic component.

4.2. Experimental results

As shown in these Figures, the fPSDTF has roughly the same performance than PSDTF, even for small N. As expected, the difference decreases with increasing N. This fact strongly supports the proposed model as a good approximation to PSDTF that allows for effective implementations. Note that in our implementation, the computational speedup we observe was not as important as displayed on Fig. 3, but was rather of one order of magnitude, which is already noticeable. This is due to the fact that the PSDTF implementation benefited from multicore architectures, while fPSDTF did not.

Finally, for one particular excerpt and source k, we display the basis matrices $|\boldsymbol{L}_k|$ learned with PSDTF for different values of N, along with the corresponding loading factors $\boldsymbol{S}_k$. We can see that only a few basis vectors turn out to be active in this example, suggesting that assuming a diagonal plus low-rank structure for spectral covariances fits music analysis.

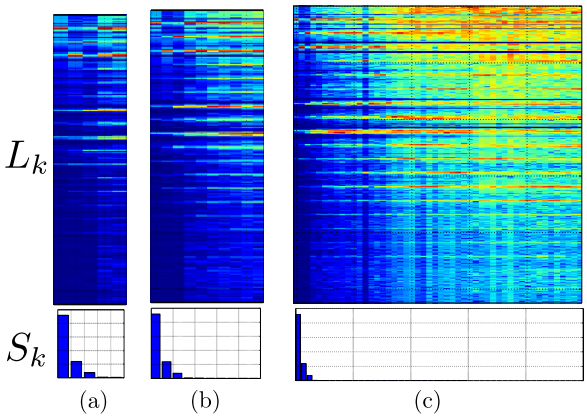

Fig. 6. Basis matrices $\boldsymbol{L}_k$ and the corresponding loading factors $\boldsymbol{S}_k$ for: a) $N = 5$, b) $N = 10$ and c) $N = 50$. We can notice that even for large N, only a small number of basis are typically active.

5. CONCLUSION

In this paper, we introduced a low-rank plus diagonal structure for approximating the positive semidefinite covariance matrices that are routinely used in source separation studies. We showed how straightforward applications of classical linear algebra methods could allow the inversion of sums of such matrices, leading to considerable speedups as compared to naive inversion. Considering the particular positive semidefinite tensor factorization (PSDTF) model, where those covariances are rather large, we showed that this approximation does not lead to noticeable decrease in performance as compared to the unconstrained model, while operating significantly faster. Interestingly, the proposed methodology and structured approximation may be used whenever large covariance matrices are considered, which may also happen in the case of massively multichannel signals. Future work includes using the potential of this structured approximation for scaling up the method to large-scale separation scenarios.

6. REFERENCES

[1] L. Benaroya, F. Bimbot, and R. Gribonval, "Audio source separation with a single sensor," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, 2006.

[2] A. Liutkus, R. Badeau, and G. Richard, "Gaussian processes for underdetermined source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 59, 2011.

[3] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, 2010.

[4] A. Liutkus and R. Badeau, "Generalized Wiener filtering with fractional power spectrograms," in *40th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, Apr. 2015, IEEE.

[5] U. Şimşekli, A. Liutkus, and A.T. Cemgil, "Alpha-stable matrix factorization," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2289–2293, 2015.

[6] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*, Wiley Publishing, Sept. 2009.

[7] P. Smaragdis, C. Févotte, G. Mysore, N. Mohammadiha, and M.D. Hoffman, "Static and dynamic source separation using nonnegative factorizations: A unified view," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 66–75, 2014.

[8] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.

[9] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the beta-divergence," *Neural Computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[10] K. Yoshii, R. Tomioka, D. Mochihashi, and M. Goto, "Infinite positive semidefinite tensor factorization for source separation of mixture signals," in *International Conference on Machine Learning (ICML)*. International Machine Learning Society (IMLS), 2013, vol. 28, pp. 576–584.

[11] R. Badeau, "Gaussian modeling of mixtures of non-stationary signals in the time-frequency domain (HR-NMF)," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2011.

[12] C. Févotte and M. Kowalski, "Low-rank time-frequency synthesis," in *Neural Information Processing Systems (NIPS)*, 2014, pp. 3563–3571.

[13] H. Kameoka, "Multi-resolution signal decomposition with time-domain spectrogram factorization," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015, pp. 86–90.

[14] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press, 2005.

[15] N. Halko, P. Martinsson, and J. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[16] M. Woodbury, "Inverting modified matrices," *Memorandum report*, vol. 42, pp. 106, 1950.

[17] A. Liutkus, F. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, Cham, 2017, pp. 323–332.

[18] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462 –1469, July 2006.