FASTMNMF: JOINT DIAGONALIZATION BASED ACCELERATED ALGORITHMS FOR MULTICHANNEL NONNEGATIVE MATRIX FACTORIZATION

Nobutaka Ito, Tomohiro Nakatani

NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan
ito.nobutaka@lab.ntt.co.jp, nakatani.tomohiro@lab.ntt.co.jp

ABSTRACT

A multichannel extension of nonnegative matrix factorization (NMF) for audio/music data, called *multichannel NMF (MNMF)*, has been proposed by Sawada *et al.* ["Multichannel extensions of non-negative matrix factorization with complex-valued data," *IEEE Trans. ASLP*, vol. 21, no. 5, pp. 971–982, May 2013]. However, conventional MNMF algorithms have a major drawback of a heavy computational load due to numerous matrix operations, such as matrix inversions and matrix multiplications. Here we propose *FastMNMF*, accelerated algorithms for the MNMF based on joint diagonalization of matrices. It is well known that, for diagonal matrices, matrix operations reduce to mere scalar operations on diagonal entries. Because of this property, the joint diagonalization results in a significantly reduced computational load compared to conventional MNMF algorithms. This makes the proposed FastMNMF even applicable to a situation with a large database or restricted computational resources.

***Index Terms*—** Nonnegative matrix factorization, joint diagonalization, source separation, microphone arrays.

1. INTRODUCTION

Multichannel NMF (MNMF) [1, 2] is a multichannel extension of the nonnegative matrix factorization (NMF) [3–5] for audio/music data. Unlike the conventional, single-channel NMF, the MNMF exploits not only *spectral* information but also *spatial* one. This makes it possible to group together basis components corresponding to the same source to realize source separation, which is challenging in the single-channel NMF [6, 7]. On the other hand, as compared with conventional source separation [8–10], the *spectral* information allows the MNMF to naturally circumvent a permutation problem among frequency bins [10, 11], which is problematic in many techniques such as frequency-domain independent component analysis (ICA).

A main drawback of conventional multichannel NMF (MNMF) algorithms is a heavy computational load. Specifically, these algorithms require numerous matrix operations of complexity $O(M^3)$ (M: matrix order), such as matrix inversions and matrix multiplications. In particular, they require matrix inversion at *each time-frequency point* and each iteration. This results in a significant computational load, because the number of time-frequency points is typically tens of thousands or larger.

To overcome this drawback, here we propose *FastMNMF*, accelerated algorithms for the MNMF in [2] based on joint diagonalization. For diagonal matrices, matrix operations such as matrix inversions and matrix multiplications are nothing but scalar operations on diagonal entries of complexity $O(M)$. Owing to this property, the joint diagonalization leads to a significantly reduced computational load compared to conventional MNMF algorithms. Indeed, source separation experiments showed that the FastMNMF was up to 35 times faster than a conventional MNMF algorithm without much degrading the separation performance.

2. CONVENTIONAL MULTICHANNEL NMF

This section describes formulations and conventional algorithms of the MNMF [2]. Section 2.2 treats a basic MNMF formulation, and Section 2.3 an extended formulation for source separation. Before that, we first describe observed data in Section 2.1.

2.1. Observed Data

Suppose L source signals are mixed and observed by M microphones, where the number of sources, L, is assumed to be given. These observed signals are represented by $x_{1ij}, \ldots, x_{Mij} \in \mathbb{C}$ in the STFT domain with $i = 1, \ldots, I$ being the frequency-bin index and $j = 1, \ldots, J$ the frame index. The observed signals are used to compute an observed covariance matrix

$$\mathbf{X}_{ij} := \mathbf{x}_{ij}\mathbf{x}_{ij}^H, \tag{1}$$

where $\mathbf{x}_{ij} := \begin{bmatrix} x_{1ij} & \ldots & x_{Mij} \end{bmatrix}^T$. In this paper, ^{T} denotes transposition, and ^{H} Hermitian transposition.

2.2. Conventional Multichannel NMF: Basic Formulation

In the basic MNMF formulation, $\mathbf{X}_{ij}$ in (1) is approximated by the sum of K ($> L$) components

$$\hat{\mathbf{X}}_{ij} := \textstyle\sum_{k=1}^{K} t_{ik} v_{kj} \mathbf{H}_{ik}, \tag{2}$$

where t_{ik}, v_{kj}, and $\mathbf{H}_{ik}$ are unknown parameters to be estimated. The components $t_{ik}v_{kj}\mathbf{H}_{ik}$ in (2) are called *basis components* with $k = 1, \ldots, K$ being their index. Here, t_{ik} (≥ 0) models the power spectrum of the kth basis component, v_{kj} (≥ 0) its activation in the jth frame, and $\mathbf{H}_{ik}$ (Hermitian positive semidefinite) its spatial characteristics, such as the location of the corresponding source [9]. The (m_1, m_2) entry of $\mathbf{H}_{ik}$ is a scaled cross-spectrum between the m_1th and m_2th microphones of the kth basis component.

The MNMF problem is formulated as one of finding the parameters such that $\hat{\mathbf{X}}_{ij}$ in (2) "best" approximates $\mathbf{X}_{ij}$. One way of doing this is to minimize a cost function

$$\textstyle\sum_{i=1}^{I}\sum_{j=1}^{J} d(\mathbf{X}_{ij}, \hat{\mathbf{X}}_{ij}) = \sum_{i=1}^{I}\sum_{j=1}^{J} d(\mathbf{X}_{ij}, \sum_{k=1}^{K} t_{ik}v_{kj}\mathbf{H}_{ik}), \tag{3}$$

where $d(\mathbf{X}_{ij}, \hat{\mathbf{X}}_{ij})$ is an error of fit between $\mathbf{X}_{ij}$ and $\hat{\mathbf{X}}_{ij}$. For brevity, here we focus on a multichannel Itakura-Saito divergence[1] [12–14] $d(\mathbf{X}, \hat{\mathbf{X}}) = -\ln\det(\mathbf{X}\hat{\mathbf{X}}^{-1}) + \mathrm{tr}(\mathbf{X}\hat{\mathbf{X}}^{-1}) - M$, which has been empirically shown to be effective [2].

In [2], an MNMF algorithm was proposed, which minimizes (3) by alternating the following updates called *multiplicative update rules* after initializing the parameters.

$$\hat{\mathbf{X}}_{ij} \leftarrow \textstyle\sum_{k=1}^{K} t_{ik} v_{kj} \mathbf{H}_{ik} \tag{4}$$

$$t_{ik} \leftarrow t_{ik} \sqrt{\frac{\sum_{j=1}^{J} v_{kj}\, \mathrm{tr}(\hat{\mathbf{X}}_{ij}^{-1}\mathbf{X}_{ij}\hat{\mathbf{X}}_{ij}^{-1}\mathbf{H}_{ik})}{\sum_{j=1}^{J} v_{kj}\, \mathrm{tr}(\hat{\mathbf{X}}_{ij}^{-1}\mathbf{H}_{ik})}} \tag{5}$$

$$v_{kj} \leftarrow v_{kj} \sqrt{\frac{\sum_{i=1}^{I} t_{ik}\, \mathrm{tr}(\hat{\mathbf{X}}_{ij}^{-1}\mathbf{X}_{ij}\hat{\mathbf{X}}_{ij}^{-1}\mathbf{H}_{ik})}{\sum_{i=1}^{I} t_{ik}\, \mathrm{tr}(\hat{\mathbf{X}}_{ij}^{-1}\mathbf{H}_{ik})}} \tag{6}$$

$$\mathbf{A}_{ik} \leftarrow \textstyle\sum_{j=1}^{J} v_{kj} \hat{\mathbf{X}}_{ij}^{-1} \tag{7}$$

$$\mathbf{B}_{ik} \leftarrow \mathbf{H}_{ik} \big(\textstyle\sum_{j=1}^{J} v_{kj} \hat{\mathbf{X}}_{ij}^{-1} \mathbf{X}_{ij} \hat{\mathbf{X}}_{ij}^{-1}\big) \mathbf{H}_{ik} \tag{8}$$

$$\mathbf{H}_{ik} \leftarrow \mathbf{A}_{ik}^{-1} \# \mathbf{B}_{ik} \tag{9}$$

In (9), $\mathbf{A}\#\mathbf{B}$ denotes the geometric mean of Hermitian positive definite matrices $\mathbf{A}$ and $\mathbf{B}$ [15, 16]. See [2] for an algorithm for computing (9).

2.3. Conventional Multichannel NMF: Source Separation

This subsection treats the extended MNMF formulation for source separation.

To realize source separation, basis clustering, *i.e.,* clustering of the basis components $t_{ik}v_{kj}\mathbf{H}_{ik}$ into sources is crucial. The spatial information encoded in $\mathbf{H}_{ik}$ can be exploited for this basis clustering. To this end, $\mathbf{H}_{ik}$ is parametrized as

$$\mathbf{H}_{ik} = \textstyle\sum_{n=1}^{N} z_{kn} \mathbf{G}_{in}, \tag{10}$$

where z_{kn} and $\mathbf{G}_{in}$ are unknown parameters to be estimated, and N the assumed number of sources[2]. Here, z_{kn} (≥ 0) denotes the responsibility of the nth source for the kth basis component, and its estimation can be thought of as the basis clustering. The matrix $\mathbf{G}_{in}$ (Hermitian positive semidefinite) models the spatial characteristics of the nth source signal.

Plugging (10) into (2), we obtain the following extended MNMF model:

$$\hat{\mathbf{X}}_{ij} := \textstyle\sum_{n=1}^{N} \big(\sum_{k=1}^{K} z_{kn} t_{ik} v_{kj}\big) \mathbf{G}_{in}. \tag{11}$$

In this case, the MNMF problem is formulated as one of finding the parameters t_{ik}, v_{kj}, z_{kn}, and $\mathbf{G}_{in}$ such that the following cost function is minimized:

$$\begin{aligned} &\textstyle\sum_{i=1}^{I} \sum_{j=1}^{J} d(\mathbf{X}_{ij}, \hat{\mathbf{X}}_{ij}) \\ &= \textstyle\sum_{i=1}^{I} \sum_{j=1}^{J} d\big(\mathbf{X}_{ij}, \sum_{n=1}^{N} \big(\sum_{k=1}^{K} z_{kn} t_{ik} v_{kj}\big) \mathbf{G}_{in}\big). \end{aligned} \tag{12}$$

Once the parameters have been estimated, they can be used to estimate the source signals $\mathbf{y}_{ijn}$. For example, this can be done by multichannel Wiener filtering as follows:

$$\hat{\mathbf{y}}_{ijn} = \big(\textstyle\sum_{k=1}^{K} z_{kn} t_{ik} v_{kj}\big) \mathbf{G}_{in} \hat{\mathbf{X}}_{ij}^{-1} \mathbf{x}_{ij} \tag{13}$$

[1] Also known as a log-determinant divergence or Stein's loss.

[2] In this paper, we distinguish N and L: N denotes the assumed number of sources, whereas L the actual number. Since L is given, N can be set at $N = L$, but does not have to. In an experiment in Section 4, for example, N is initialized so that $N > L$, and then decreased gradually down to $N = L$.

Table 1. The number of matrix operations of complexity $O(M^3)$ per iteration with M being the matrix order. "$\mathbf{A}^{-1}$" stands for matrix inversions, "$\mathbf{AB}$" matrix multiplications, and "$\mathbf{A}^{-1}\#\mathbf{B}$" matrix geometric means in (9) and (19).

	conventional MNMF algorithms [2]		
	basic formulation (Section 2.2)	source separation (Section 2.3)	FastMNMF
$\mathbf{A}^{-1}$	IJ	IJ	$I(M+1)$
$\mathbf{AB}$	$2IK$	$2IN$	0
$\mathbf{A}^{-1}\#\mathbf{B}$	IK	IN	0
total	$I(3K+J)$	$I(3N+J)$	$I(M+1)$

with $\hat{\mathbf{X}}_{ij}$ computed as in (11).

In [2], an algorithm for this extended MNMF formulation was also proposed, which minimizes (12) by alternating the following updates after initializing the parameters.

$$\mathbf{H}_{ik} \leftarrow \textstyle\sum_{n=1}^{N} z_{kn} \mathbf{G}_{in} \tag{14}$$

$$\hat{\mathbf{X}}_{ij}, t_{ik}, \text{and } v_{kj} \text{ updated by (4), (5), and (6).} \tag{15}$$

$$z_{kn} \leftarrow z_{kn} \sqrt{\frac{\sum_{i=1}^{I} \sum_{j=1}^{J} t_{ik} v_{kj}\, \mathrm{tr}(\hat{\mathbf{X}}_{ij}^{-1}\mathbf{X}_{ij}\hat{\mathbf{X}}_{ij}^{-1}\mathbf{G}_{in})}{\sum_{i=1}^{I} \sum_{j=1}^{J} t_{ik} v_{kj}\, \mathrm{tr}(\hat{\mathbf{X}}_{ij}^{-1}\mathbf{G}_{in})}} \tag{16}$$

$$\mathbf{Q}_{in} \leftarrow \textstyle\sum_{k=1}^{K} z_{kn} t_{ik} \sum_{j=1}^{J} v_{kj} \hat{\mathbf{X}}_{ij}^{-1} \tag{17}$$

$$\mathbf{R}_{in} \leftarrow \mathbf{G}_{in} \big(\textstyle\sum_{k=1}^{K} z_{kn} t_{ik} \sum_{j=1}^{J} v_{kj} \hat{\mathbf{X}}_{ij}^{-1} \mathbf{X}_{ij} \hat{\mathbf{X}}_{ij}^{-1}\big) \mathbf{G}_{in} \tag{18}$$

$$\mathbf{G}_{in} \leftarrow \mathbf{Q}_{in}^{-1} \# \mathbf{R}_{in} \tag{19}$$

2.4. Drawback of Conventional Multichannel NMF

A major drawback of the conventional MNMF algorithms is a heavy computational load. Indeed, these algorithms require numerous matrix operations of complexity $O(M^3)$. Table 1 shows the number of such matrix operations per iteration[3]. We see that the number of matrix inversions depends on the number of time-frequency points, IJ, which is typically tens of thousands or larger.

For example, consider the algorithm in Section 2.3 with $M = 3$, $N = 3$, $I = 513$, and $J = 904$, which corresponds to an experimental condition in Section 4. In this case, the algorithm requires $IJ = 463752$ matrix inversions, $2IN = 3078$ matrix multiplications, and $IN = 1539$ matrix geometric means in each iteration.

See Section 4 for evaluation in terms of the computation time.

3. FASTMNMF: ACCELERATED ALGORITHMS FOR THE MULTICHANNEL NMF

3.1. Our Approach: Joint Diagonalization

To overcome the above drawback, here we propose *FastMNMF*, accelerated algorithms for the MNMF.

The proposed FastMNMF exploits the well-known fact that, for diagonal matrices, matrix operations such as matrix inversions and matrix multiplications are nothing but scalar operations on diagonal entries, which are only of complexity $O(M)$. For example,

$$\begin{pmatrix} \alpha_1 & & 0 \\ & \ddots & \\ 0 & & \alpha_M \end{pmatrix}^{-1} = \begin{pmatrix} \alpha_1^{-1} & & 0 \\ & \ddots & \\ 0 & & \alpha_M^{-1} \end{pmatrix}. \tag{20}$$

[3] Note that $\hat{\mathbf{X}}_{ij}^{-1}\mathbf{X}_{ij}\hat{\mathbf{X}}_{ij}^{-1} = \hat{\mathbf{X}}_{ij}^{-1}\mathbf{x}_{ij}(\hat{\mathbf{X}}_{ij}^{-1}\mathbf{x}_{ij})^H$ can be computed from $\hat{\mathbf{X}}_{ij}^{-1}$ and $\mathbf{x}_{ij}$ without matrix multiplications.

With this in mind, consider for example the algorithm in Section 2.2. If $\mathbf{H}_{i1}, \ldots, \mathbf{H}_{iK}$ were all diagonal, the matrix inversion $\hat{\mathbf{X}}_{ij}^{-1} = (\sum_{k=1}^{K} t_{ik} v_{kj} \mathbf{H}_{ik})^{-1}$ for instance would be mere inversion of the diagonal entries as in (20). However, the off-diagonal entries of $\mathbf{H}_{ik}$ are rarely zero in practice, because the basis components are normally highly correlated between microphones.

This motivates us to consider joint diagonalization of the spatial covariance matrices $\mathbf{H}_{i1}, \ldots, \mathbf{H}_{iK}$ in line with [17–20]. That is, we consider transforming $\mathbf{H}_{i1}, \ldots, \mathbf{H}_{iK}$ into some diagonal matrices $\boldsymbol{\mathcal{H}}_{i1}, \ldots, \boldsymbol{\mathcal{H}}_{iK}$ by a single nonsingular matrix $\mathbf{P}_i$ as

$$\begin{cases} \mathbf{P}_i^H \mathbf{H}_{i1} \mathbf{P}_i = \boldsymbol{\mathcal{H}}_{i1}, \\ \cdots\cdots \\ \mathbf{P}_i^H \mathbf{H}_{iK} \mathbf{P}_i = \boldsymbol{\mathcal{H}}_{iK}. \end{cases} \tag{21}$$

In this paper, *we use bold calligraphic fonts (e.g., $\boldsymbol{\mathcal{H}}_{i1}$) to represent diagonal matrices.* In signal-processing terms, this joint diagonalization corresponds to joint decorrelation of the K basis components. Here it is paramount how to obtain matrices $\boldsymbol{\mathcal{H}}_{ik}$ and $\mathbf{P}_i$ that satisfy (21).

In the case with only $K = 2$ basis components, this can be realized by solving a generalized eigenvalue problem as in [17, 18]. However, the restriction $K = 2$ largely limits the performance, since tens of basis components are typically needed for high performance [2].

To deal with the general case with an arbitrary K as in [19, 20], here we solve (21) for $\mathbf{H}_{ik}$ as

$$\begin{cases} \mathbf{H}_{i1} = (\mathbf{P}_i^{-1})^H \boldsymbol{\mathcal{H}}_{i1} \mathbf{P}_i^{-1}, \\ \cdots\cdots \\ \mathbf{H}_{iK} = (\mathbf{P}_i^{-1})^H \boldsymbol{\mathcal{H}}_{iK} \mathbf{P}_i^{-1}, \end{cases} \tag{22}$$

and estimate $\boldsymbol{\mathcal{H}}_{ik}$ and $\mathbf{P}_i$ from the observed signals by minimizing the multichannel Itakura-Saito divergence. This joint diagonalization approach leads to a significantly reduced computational load compared to the conventional MNMF algorithm in Section 2.2.

This joint diagonalization approach is also applicable to the algorithm in Section 2.3, where $\mathbf{G}_{i1}, \ldots, \mathbf{G}_{iN}$ instead of $\mathbf{H}_{i1}, \ldots, \mathbf{H}_{iK}$ are jointly diagonalized.

3.2. FastMNMF: Basic Formulation

The proposed FastMNMF for the basic MNMF formulation in Section 2.2 alternates the following update rules after initializing the parameters t_{ik}, v_{kj}, $\boldsymbol{\mathcal{H}}_{ik}$, and $\mathbf{P}_i$. The derivation is omitted because of space limitations.

$$\boldsymbol{\mathcal{X}}_{ij} \leftarrow D(\mathbf{P}_i^H \mathbf{X}_{ij} \mathbf{P}_i) \tag{23}$$

$$\hat{\boldsymbol{\mathcal{X}}}_{ij} \leftarrow \textstyle\sum_{k=1}^{K} t_{ik} v_{kj} \boldsymbol{\mathcal{H}}_{ik} \tag{24}$$

$$t_{ik} \leftarrow t_{ik} \sqrt{\frac{\sum_{j=1}^{J} v_{kj} \operatorname{tr}(\hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1} \boldsymbol{\mathcal{X}}_{ij} \hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1} \boldsymbol{\mathcal{H}}_{ik})}{\sum_{j=1}^{J} v_{kj} \operatorname{tr}(\hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1} \boldsymbol{\mathcal{H}}_{ik})}} \tag{25}$$

$$v_{kj} \leftarrow v_{kj} \sqrt{\frac{\sum_{i=1}^{I} t_{ik} \operatorname{tr}(\hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1} \boldsymbol{\mathcal{X}}_{ij} \hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1} \boldsymbol{\mathcal{H}}_{ik})}{\sum_{i=1}^{I} t_{ik} \operatorname{tr}(\hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1} \boldsymbol{\mathcal{H}}_{ik})}} \tag{26}$$

$$\boldsymbol{\mathcal{A}}_{ik} \leftarrow \textstyle\sum_{j=1}^{J} v_{kj} \hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1} \tag{27}$$

$$\boldsymbol{\mathcal{B}}_{ik} \leftarrow \boldsymbol{\mathcal{H}}_{ik} (\textstyle\sum_{j=1}^{J} v_{kj} \hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1} \boldsymbol{\mathcal{X}}_{ij} \hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1}) \boldsymbol{\mathcal{H}}_{ik} \tag{28}$$

$$\boldsymbol{\mathcal{H}}_{ik} \leftarrow \boldsymbol{\mathcal{A}}_{ik}^{-1} \# \boldsymbol{\mathcal{B}}_{ik} \tag{29}$$

$$[\mathbf{P}_i]_m \leftarrow ((1/J) \textstyle\sum_{j=1}^{J} \mathbf{X}_{ij} / [\hat{\boldsymbol{\mathcal{X}}}_{ij}]_{mm})^{-1} [(\mathbf{P}_i^{-1})^H]_m \tag{30}$$

Once the parameters have been obtained by the above algorithm, $\mathbf{H}_{ik}$ are obtained by (22).

We see that (24)–(29) have the same forms as (4)–(9), but *the matrices $\mathbf{X}_{ij}$, $\hat{\mathbf{X}}_{ij}$, $\mathbf{H}_{ik}$, $\mathbf{A}_{ik}$, and $\mathbf{B}_{ik}$ have been replaced by the diagonal matrices $\boldsymbol{\mathcal{X}}_{ij}$, $\hat{\boldsymbol{\mathcal{X}}}_{ij}$, $\boldsymbol{\mathcal{H}}_{ik}$, $\boldsymbol{\mathcal{A}}_{ik}$, and $\boldsymbol{\mathcal{B}}_{ik}$.* This has turned the matrix operations in (4)–(9) into ones on diagonal matrices[4], which are only of complexity $O(M)$. Equation (30) updates the matrix $\mathbf{P}_i$ for the joint diagonalization, where $[\mathbf{A}]_{mn}$ denotes the (m, n) entry of a matrix $\mathbf{A}$, and $[\mathbf{A}]_m$ its mth column. Although (30) requires a few matrix inversions, this is not a big issue because they are not required in each frame. Equation (23) transforms $\mathbf{X}_{ij}$ by $\mathbf{P}_i$, and replaces the resulting off-diagonal entries by zeros, where D denotes the operator that replaces the off-diagonal entries by zeros. Note that $D(\mathbf{P}_i^H \mathbf{X}_{ij} \mathbf{P}_i)$ can be computed without matrix multiplications because $D(\mathbf{P}_i^H \mathbf{X}_{ij} \mathbf{P}_i) = \operatorname{diag}(|([\mathbf{P}_i]_1)^H \mathbf{x}_{ij}|^2, \ldots, |([\mathbf{P}_i]_M)^H \mathbf{x}_{ij}|^2)$.

As shown in Table 1, the number of matrix operations of complexity $O(M^3)$ in the above FastMNMF is only $I(M + 1)$, which is independent of the number of frames, J. This is in sharp contrast to the conventional MNMF algorithm in Section 2.2, in which the number of such matrix operations depends on the number of time-frequency points, IJ. Note that matrix operations on diagonal matrices were not counted in Table 1, because their complexity is $O(M)$ instead of $O(M^3)$.

For example, consider the example in Section 2.4. In this case, each iteration of the above FastMNMF requires only $I(M + 1) = 2052$ matrix inversions of complexity $O(M^3)$ and no matrix multiplications or matrix geometric means of complexity $O(M^3)$.

3.3. FastMNMF: Source Separation

The proposed FastMNMF for the extended formulation in Section 2.3 alternates the following update rules after initializing the parameters t_{ik}, v_{kj}, z_{kn}, $\boldsymbol{\mathcal{G}}_{in}$, and $\mathbf{P}_i$.

$$\boldsymbol{\mathcal{X}}_{ij} \text{ updated by (23).} \tag{31}$$

$$\boldsymbol{\mathcal{H}}_{ik} \leftarrow \textstyle\sum_{n=1}^{N} z_{kn} \boldsymbol{\mathcal{G}}_{in} \tag{32}$$

$$\hat{\boldsymbol{\mathcal{X}}}_{ij}, t_{ik}, \text{ and } v_{kj} \text{ updated by (24), (25), and (26).} \tag{33}$$

$$z_{kn} \leftarrow z_{kn} \sqrt{\frac{\sum_{i=1}^{I} \sum_{j=1}^{J} t_{ik} v_{kj} \operatorname{tr}(\hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1} \boldsymbol{\mathcal{X}}_{ij} \hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1} \boldsymbol{\mathcal{G}}_{in})}{\sum_{i=1}^{I} \sum_{j=1}^{J} t_{ik} v_{kj} \operatorname{tr}(\hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1} \boldsymbol{\mathcal{G}}_{in})}} \tag{34}$$

$$\boldsymbol{\mathcal{Q}}_{in} \leftarrow \textstyle\sum_{k=1}^{K} z_{kn} t_{ik} \sum_{j=1}^{J} v_{kj} \hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1} \tag{35}$$

$$\boldsymbol{\mathcal{R}}_{in} \leftarrow \boldsymbol{\mathcal{G}}_{in} (\textstyle\sum_{k=1}^{K} z_{kn} t_{ik} \sum_{j=1}^{J} v_{kj} \hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1} \boldsymbol{\mathcal{X}}_{ij} \hat{\boldsymbol{\mathcal{X}}}_{ij}^{-1}) \boldsymbol{\mathcal{G}}_{in} \tag{36}$$

$$\boldsymbol{\mathcal{G}}_{in} \leftarrow \boldsymbol{\mathcal{Q}}_{in}^{-1} \# \boldsymbol{\mathcal{R}}_{in} \tag{37}$$

$$\mathbf{P}_i \text{ updated by (30).} \tag{38}$$

Once the parameters are obtained by the above algorithm, $\mathbf{G}_{in}$ are obtained by

$$\mathbf{G}_{in} = (\mathbf{P}_i^{-1})^H \boldsymbol{\mathcal{G}}_{in} \mathbf{P}_i^{-1}. \tag{39}$$

Estimates $\hat{\mathbf{y}}_{ijn}$ of the source signals $\mathbf{y}_{ijn}$ are also obtained by (13).

Again, (32)–(37) have the same forms as (14)–(19), but the matrices $\mathbf{X}_{ij}$, $\hat{\mathbf{X}}_{ij}$, $\mathbf{H}_{ik}$, $\mathbf{G}_{in}$, $\mathbf{Q}_{in}$, and $\mathbf{R}_{in}$ have been replaced by the

[4] The geometric mean $\boldsymbol{\mathcal{A}}_{ik}^{-1} \# \boldsymbol{\mathcal{B}}_{ik}$ in (29) can be computed from $\boldsymbol{\mathcal{A}}_{ik}$ and $\boldsymbol{\mathcal{B}}_{ik}$ in $O(M)$, because $\boldsymbol{\mathcal{A}}_{ik}^{-1} \# \boldsymbol{\mathcal{B}}_{ik} = \operatorname{diag}(\sqrt{[\boldsymbol{\mathcal{B}}_{ik}]_{11} / [\boldsymbol{\mathcal{A}}_{ik}]_{11}}, \ldots, \sqrt{[\boldsymbol{\mathcal{B}}_{ik}]_{MM} / [\boldsymbol{\mathcal{A}}_{ik}]_{MM}})$. Here, $\operatorname{diag}(\alpha_1, \ldots, \alpha_M)$ denotes the diagonal matrix with $\alpha_1, \ldots, \alpha_M$ on its diagonal.

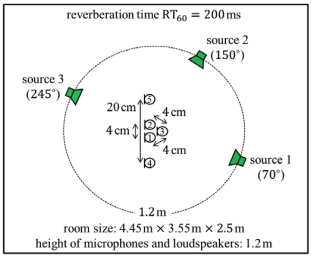

Fig. 1. Experimental setup.

diagonal matrices $\boldsymbol{\mathcal{X}}_{ij}$, $\hat{\boldsymbol{\mathcal{X}}}_{ij}$, $\boldsymbol{\mathcal{H}}_{ik}$, $\boldsymbol{\mathcal{G}}_{in}$, $\boldsymbol{\mathcal{Q}}_{in}$, and $\boldsymbol{\mathcal{R}}_{in}$. The above FastMNMF has the same computational complexity as the FastMNMF in Section 3.2 in terms of the number of matrix operations of complexity $O(M^3)$.

3.4. Relation to Prior Work

The above joint diagonalization approach has already been applied to a source separation method called full-rank spatial covariance analysis (FCA) [17–20], and, later but independently, to an unsupervised learning method called correlated tensor factorization (CTF) [21]. These methods are related to but different from the MNMF treated in this paper.

Ikeshita *et al.* [22] have also proposed a joint diagonalization based acceleration of the MNMF [2]. An assumption underlying this method is that the source signals are sparse in the sense that at most two of them are active at a time-frequency point. This allows for joint exact diagonalization of the spatial covariance matrices for each possible pair of source signals in line with [17, 18, 23]. However, source signals such as music parts often overlap in the time-frequency domain, which may violate the above assumption. The FastMNMF in this paper resolves this issue by dropping such a sparsity assumption, while still realizing acceleration thanks to the joint diagonalizability constraint (39) on the spatial covariance matrices.

Taniguchi *et al.* [24] have proposed preprocessing for joint diagonalization using an unmixing matrix obtained by independent vector analysis (IVA) [25] to accelerate the MNMF, where the number of sources needs be equal to that of microphones (*i.e.,* the even determined case). This is in a sharp contrast to the proposed MNMF, in which the number of sources can differ from that of microphones.

Apart from accelerating algorithms, the joint diagonalization has been employed in the signal processing literature in the contexts of independent component analysis [13, 26] and a subspace method [27].

4. EXPERIMENTS

We conducted music source separation experiments to compare the conventional MNMF algorithm in Section 2.3 and the proposed FastMNMF in Section 3.3. These algorithms were implemented in MATLAB (R2011a) and run on an Intel Core i7-7820X (3.6 GHz) processor. As in [2], we took four songs (”Bearlin”, ”Another Dreamer”, ”Fort Minor”, and ”Ultimate NZ Tour”) from a ”professionally produced music recordings” database of the signal separation and evaluation campaign (SiSEC) [28]. For each song, music parts from the database were mixed after being convolved with room impulse responses measured in a real room [10]. The setup for room impulse response measurement is depicted in Fig. 1. The sampling rate of the mixtures was 16 kHz after downsampling. Some other conditions are found in Table 2. The parameters t_{ik}, v_{kj}, and z_{kn} were randomly initialized by nonnegative values, and $\mathbf{G}_{in}$, $\boldsymbol{\mathcal{G}}_{in}$, and $\mathbf{P}_i$ by the identity matrix. Note that there is scale indeterminacy

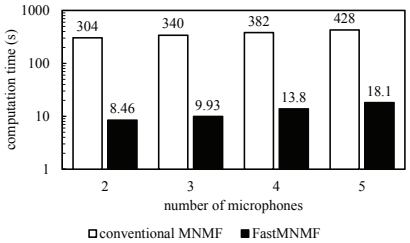

Fig. 2. Computation time for ten iterations for the song ”Bearlin” (14-s long).

Table 2. Experimental conditions.

frame length	1024 (64 ms)
frame shift	256 (16 ms)
window	square root of Hann
number of microphones, M	2, 3, 4, or 5
number of sources, L	3

in $\mathbf{P}_i$ in the FastMNMF. To prevent this from causing numerical instability, each column of $\mathbf{P}_i$ is scaled by $\mathbf{P}_i \leftarrow \mathbf{P}_i(D(\mathbf{P}_i))^{-1}$ in each iteration.

In the first experiment, we measured the computation time for ten iterations of these algorithms with $N = 3$ and $K = 10$. As seen from Fig. 2, the FastMNMF was up to 35 times faster than the conventional MNMF algorithm. This acceleration is attributed to efficient computation of matrix inverses, matrix products, and matrix geometric means enabled by the joint diagonalization.

In the second experiment, we evaluated source separation performance. The same procedure as in [2] was followed, which is briefly described in the following. We set $K = 30$ and $M = 2$. The assumed number of sources, N, was initialized by $N \leftarrow 9$, and then decreased gradually down to $N = 3$, which reportedly leads to robust parameter estimation [2]. In the first 20 iterations, only t_{ik} and v_{kj} were updated with the other parameters fixed. The total number of iterations was 471. The estimated parameters were used to estimate the source signals by multichannel Wiener filtering. For each song, ten trials were conducted.

Figure 3 shows the source separation performance SDR (signal-to-distortion ratio) [29] averaged over all sources and all trials. We can confirm that the above acceleration was realized without much degrading the source separation performance. The standard deviation shown by the error bar was up to 1.7 dB, which implies that the algorithms were sensitive to the initialization. Development of an algorithm robust to the initialization is an important open problem.

5. CONCLUSIONS

This paper proposed the FastMNMF, accelerated MNMF algorithms. The experiments implied that the joint diagonalization made the FastMNMF up to 35 times faster than a conventional MNMF algorithm without much degrading the separation performance.

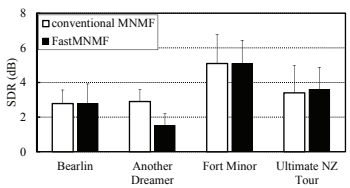

Fig. 3. Source separation performance SDR.

6. REFERENCES

[1] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Trans. ASLP*, vol. 18, no. 3, pp. 550–563, Mar. 2010.

[2] H. Sawada, H. Kameoka, S. Araki, and N. Ueda, "Multichannel extensions of non-negative matrix factorization with complex-valued data," *IEEE Trans. ASLP*, vol. 21, no. 5, pp. 971–982, May 2013.

[3] D. D. Lee and H. S. Seung, "Learning the parts of objects with nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, Oct. 1999.

[4] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. WASPAA*, Oct. 2003, pp. 177–180.

[5] 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, Mar. 2009.

[6] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Trans. ASLP*, vol. 15, no. 1, pp. 1–12, Jan. 2007.

[7] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Trans. ASLP*, vol. 15, no. 3, pp. 1066–1074, Mar. 2007.

[8] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 22, no. 1–3, pp. 21–34, Nov. 1998.

[9] N. Q. K. Duong, E. Vincent, and R. Gribonval, "Underdetermined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Trans. ASLP*, vol. 18, no. 7, pp. 1830–1840, Sep. 2010.

[10] H. Sawada, S. Araki, and S. Makino, "Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment," *IEEE Trans. ASLP*, vol. 19, no. 3, pp. 516–527, Mar. 2011.

[11] H. Sawada, R. Mukai, S. Araki, and S. Makino, "A robust and precise method for solving the permutation problem of frequency-domain blind source separation," *IEEE Trans. ASLP*, vol. 12, no. 5, pp. 530–538, Sep. 2004.

[12] W. James and C. Stein, "Estimation with quadratic loss," in *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, 1961.

[13] D.-T. Pham and J.-F. Cardoso, "Blind separation of instantaneous mixtures of non stationary sources," *IEEE Trans. SP*, vol. 49, no. 9, pp. 1837–1848, Sep. 2001.

[14] B. Kulis, M. Sustik, and I. Dhillon, "Learning low-rank kernel matrices," in *Proc. International Conference on Machine Learning (ICML)*, Jun. 2006, pp. 505–512.

[15] G. Pedersen and M. Takesaki, "The operator equation $THT = K$," in *Proceedings of the American Mathematical Society*, Nov. 1972, pp. 993–1022.

[16] J. Lawson and Y. Lim, "The geometric mean, matrices, metrics, and more," *The American Mathematical Monthly*, pp. 797–812, Nov. 2001.

[17] N. Ito, S. Araki, and T. Nakatani, "FastFCA: Fast algorithm for audio source separation using time-varying complex Gaussian distribution based on joint diagonalization of spatial covariance matrices," in *Proceedings of the Spring Meeting of the Acoustical Society of Japan*, Feb. 2018, pp. 427–430 (in Japanese).

[18] ——, "FastFCA: Joint diagonalization based acceleration of audio source separation using a full-rank spatial covariance model," in *Proc. EUSIPCO*, Sep. 2018 (a preprint arXiv: 1805.06572 published on May 17th, 2018).

[19] N. Ito and T. Nakatani, "FastFCA-AS: Joint diagonalization based acceleration of full-rank spatial covariance analysis for separating any number of sources," in *Proc. IWAENC*, Sep. 2018 (a preprint arXiv: 1805.09498 published on May 24th, 2018).

[20] ——, "Multiplicative updates and joint diagonalization based acceleration for under-determined BSS using a full-rank spatial covariance model," in *Proc. GlobalSIP*, Nov. 2018, pp. 231–235.

[21] K. Yoshii, K. Kitamura, Y. Bando, E. Nakamura, and T. Kawahara, "Independent low-rank tensor analysis for audio source separation," in *Proc. EUSIPCO*, Sep. 2018, pp. 1671–1675.

[22] R. Ikeshita, Y. Kawaguchi, and K. Nagamatsu, "Fast multichannel nonnegative matrix factorization with constraints on active source candidate," in *Proc. IWAENC*, Sep. 2018, pp. 520–524.

[23] N. Ito, C. Schymura, S. Araki, and T. Nakatani, "Noisy cGMM: Complex Gaussian mixture model with non-sparse noise model for joint source separation and denoising," in *Proc. EUSIPCO*, Sep. 2018, pp. 1662–1666.

[24] T. Taniguchi and T. Masuda, "Linear demixed domain multichannel nonnegative matrix factorization for speech enhancement," in *Proc. ICASSP*, Mar. 2017, pp. 476–480.

[25] N. Ono, "Stable and fast update rules for independent vector analysis based on auxiliary function technique," in *Proc. WASPAA*, Oct. 2011, pp. 189–192.

[26] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second order statistics," *IEEE Trans. SP*, pp. 434–444, Feb. 1997.

[27] S. Doclo and M. Moonen, "GSVD-based optimal filtering for single and multimicrophone speech enhancement," *IEEE Trans. SP*, vol. 50, no. 9, pp. 2230–2244, Sep. 2002.

[28] E. Vincent, S. Araki, F. Theis, G. Nolte, P. Bofill, H. Sawada, A. Ozerov, V. Gowreesunker, D. Lutter, and N. Duong, "The signal separation evaluation campaign (2007—2010): Achievements and remaining challenges," *Signal Processing*, vol. 92, no. 8, pp. 1928–1936, Aug. 2012.

[29] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. ASLP*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.