Original Paper

Repeated Update of Demixing Vectors in Independent Low-rank Matrix Analysis for Better Separation

Taishi Nakashima* and Nobutaka Ono

Tokyo Metropolitan University, Tokyo, Japan

ABSTRACT

In this paper, we propose a better update algorithm for independent low-rank matrix analysis (ILRMA). ILRMA has two types of parameters, demixing vectors and non-negative matrix factorization parameters, which are estimated by minimizing the same objective function. Although many extensions of ILRMA have been proposed, the importance of the order of parameter updates in ILRMA is not investigated sufficiently. Because of the observation that iterative projection two (IP2) shows a higher performance than IP1, we propose a repeated update of demixing vectors with the source model fixed in one iteration; this approximates a simultaneous update of all demixing vectors together. We conducted music source separation experiments with more than 100 songs. The results showed that the proposed algorithm with the repeated update of demixing vectors outperforms the conventional ILRMA regarding separation performance and convergence speed.

*Corresponding author: Taishi Nakashima, taishi@ieee.org.

1   Introduction

Blind source separation (BSS) aims to separate a mixture of multiple signals into individual signals [2]. It has been applied in a vast number of applications, such as multichannel audio processing [11], music information retrieval [1], wireless communication systems [10], soundscape information retrieval [9], and brain imaging [29].

Independent vector analysis (IVA) considers frequency-wise correlations and avoids permutation ambiguity by assuming a multivariate distribution of sources [3, 5]. The demixing matrices of IVA are estimated by updating with the gradient descent method, which is sensitive to a step-size parameter and may be unstable. Auxiliary-function-based IVA (AuxIVA) has been proposed to improve convergence and performance [21]. AuxIVA estimates demixing matrices faster and more stably than the conventional IVA with no tuning parameters using the majorization-minimization algorithm [8]. Some extensions of AuxIVA for online/real-time processing [27] and its application to hearing aids [26] have been proposed. To further improve the source model, AuxIVA was extended to independent low-rank matrix analysis (ILRMA) [7] by exploiting a low-rank source model of nonnegative matrix factorization. ILRMA models the spectrograms of separated sources more accurately and thus achieves a higher separation performance than AuxIVA. Many extensions to other source models of ILRMA were proposed [6, 14, 15]. Update rules of demixing matrices and source models have been extended from IVA to ILRMA.

Update rules of demixing matrices of AuxIVA have been extensively studied in recent years. Iterative projection (IP) was initially proposed for AuxIVA and updates each row vector of a demixing matrix (*demixing vector*) per iteration [21]. The pairwise update rule called iterative projection 2 (IP2) was also proposed and updates two demixing vectors simultaneously in each iteration [20]. We refer to the original IP as IP1 to distinguish it from IP2 in this paper. As an alternative approach, iterative source steering (ISS), which updates the entire demixing matrix using elementary row operations, has been proposed [24]. ISS can eliminate matrix inversions and thus updates more rapidly while keeping the same separation performance as IP1. All these update methods are derived by minimizing the likelihood function of the input mixture signal, leading to a system of quadratic equations called hybrid exact-approximate joint diagonalization (HEAD) [30, 31] (also known as "sequentially drilled" joint congruence (SeDJoCo) [28]). When the number of sources is three or more, HEAD is still an open problem. IP2 can yield a global optimal solution of HEAD only for two sources by solving a generalized eigenvalue problem. IP1, IP2, and ISS are available for ILRMA because ILRMA includes AuxIVA as a special case [7]. The original ILRMA employs IP1 to update demixing vectors. ILRMA with IP2 or ISS has also been proposed [17, 18].

Interestingly, it has been reported that IVA with IP2 [19] and ILRMA with IP2 [18] showed a higher performance than those with IP1, even using the same source model. These results imply that how to update the parameters also contributes the better separation performance. Similarly, it has been shown that a slower updating of source model parameters tends to improve the separation performance [13].

Motivated by these studies, we here focus on the parameter updates in ILRMA. The contribution of this study is summarized as follows:

1. We propose a new algorithm of ILRMA named ILRMA-Rep. It repeatedly updates the demixing vectors by fixing the source model at each iteration. It works as approximately updating all the demixing vectors at once.

2. We derive faster update rules of ILRMA-Rep based on the matrix inversion lemma (MIL). Especially the update rule based on IP2 with MIL, which we call IP2-MIL, is new and applicable to other BSS methods, such as online AuxIVA.

3. We conducted simulated separation experiments with 144 music signals to validate the efficacy of our proposed method. The results show that the proposed methods improve the separation performance.

The rest of this paper is organized as follows. We briefly summarize the background of the conventional ILRMA and its update rules in Section 2. Section 3 describes ILRMA with repeated updates of demixing vectors (ILRMA-Rep), the proposed ILRMA updates with repeated updates of demixing vectors. We evaluate the performance of ILRMA-Rep and compare it with that of the conventional ILRMA with IP1, IP2, and ISS in Section 4. Finally, Section 5 concludes this paper.

2 Overview of Problem Formulation and Conventional ILRMA

2.1 Notations

In the rest of this paper, we use lower- and uppercase bold symbols for vectors and matrices, respectively. Lowercase normal symbols denote vector entries or scalars. For example, x_{kft} denotes the kth entry of a vector $\boldsymbol{x}_{ft}$. The transpose and the conjugate transpose of a vector and a matrix are respectively denoted as $(\cdot)^{\mathsf{T}}$ and $(\cdot)^{\mathsf{H}}$. We denote $\mathbb{C}$ and $\mathbb{R}_+$ as the sets of all complex numbers and non-negative real numbers, respectively. Unless specified otherwise, indices f, t, and m always take the ranges from 1 to F, T, and M, respectively. Also, we omit the bounds of sums or products over these indices when they span the ranges. For example, $\sum_k$ is the sum over all k in the range of $k = 1, \ldots, K$, and $\sum_{ft}$ is the double-sum over $f = 1, \ldots, F$ and $t = 1, \ldots, T$. Similarly,

$\{\boldsymbol{W}_f\}_f$ denotes the set of $\boldsymbol{W}_f$ for all f; $\{x_{kft}\}_{kft}$ denotes the set of x_{kft} for all k, f, and t, for example.

2.2 Signal Model

We formulate the short-time Fourier transform (STFT) domain BSS as

$$\boldsymbol{x}_{ft} = \boldsymbol{A}_f \boldsymbol{s}_{ft}, \tag{1}$$

where $\boldsymbol{x}_{ft} \in \mathbb{C}^M$ denotes the observed mixture signals recorded by M microphones and $\boldsymbol{s}_{ft} \in \mathbb{C}^K$ denotes the source signals of K sources, respectively, at frequency f and time t. Here, $\boldsymbol{A}_f = \begin{bmatrix} \boldsymbol{a}_{1f} & \cdots & \boldsymbol{a}_{Kf} \end{bmatrix} \in \mathbb{C}^{M \times K}$ is the mixing matrix whose kth column vector $\boldsymbol{a}_{kf}$ corresponds to the steering vector from source k to each microphne. The goal of BSS is to estimate the frequency-wise demixing matrices,

$$\boldsymbol{W}_f = \begin{bmatrix} \boldsymbol{w}_{1f} & \cdots & \boldsymbol{w}_{Mf} \end{bmatrix}^{\mathsf{H}} \in \mathbb{C}^{K \times M}, \tag{2}$$

such that the estimated source is

$$\boldsymbol{y}_{ft} = \boldsymbol{W}_f \boldsymbol{x}_{ft}, \tag{3}$$

given only mixture signals $\boldsymbol{x}_{ft}$, where $\boldsymbol{y}_{ft} \in \mathbb{C}^K$ denotes the estimated signals. The kth row vector of the demixing matrix $\boldsymbol{w}_{kf}$ is called the *demixing vector* for source k. We henceforth consider the determined situation, i.e., the number of sources equals that of microphones $M = K$.

2.3 Independent Low-rank Matrix Analysis

We briefly summarize ILRMA as a maximum likelihood estimation problem. The likelihood function is derived with the following assumptions.

1. All the sources are statistically independent; their joint distribution is the product of their individual distributions.

2. The estimated signals follow this zero-mean multivariate complex Gaussian distribution,

$$p_y(y_{kft}) = \frac{1}{\pi r_{kft}} \exp\left(-\frac{|y_{kft}|^2}{r_{kft}}\right), \tag{4}$$

where $r_{kft} \in \mathbb{R}_+$ is called the variance of the distribution corresponding to the kth source in a time–frequency point f and t.

3. The variance r_{kft} is decomposed into L parts as

$$r_{kft} = \sum_{l=1}^{L} b_{kfl} c_{klt}, \tag{5}$$

where b_{kfl} and $c_{klt} \in \mathbb{R}_+$ are called the *basis* and *activity* coefficients, respectively. Henceforth, we call these coefficients the *source model parameters*.

The kth estimated signal is denoted as $y_{kft} = \boldsymbol{w}_{kf}^{\mathsf{H}} \boldsymbol{x}_{ft}$ from (3). By using these assumptions, we can determine the likelihood of the observation as

$$\mathcal{L}(\mathcal{W}) = \prod_{ft} p_x \left(\boldsymbol{x}_{ft} \right) \tag{6}$$

$$= \prod_{ft} p_y \left(\boldsymbol{y}_{ft} \right) |\det \boldsymbol{W}_f|^2 \tag{7}$$

$$= \prod_{ft} \left(\prod_{k} p_y \left(y_{kft} \right) \right) |\det \boldsymbol{W}_f|^2, \tag{8}$$

where p_x is the probability density function of the observed signals and $\mathcal{W}$ is the set defined as $\mathcal{W} = \{\boldsymbol{W}_f\}_f$. The determinant term is the Jacobian that comes from the change of variable. The goal of ILRMA is to estimate $\boldsymbol{W}_f$ by minimizing the following function:

$$\mathcal{L}^+(\mathcal{W}, \mathcal{B}, \mathcal{C}) = \sum_{ft} \left[\sum_{k} \left(\frac{|\boldsymbol{w}_{kf}^{\mathsf{H}} \boldsymbol{x}_{ft}|^2}{r_{kft}} + \log r_{kft} \right) - \log|\det \boldsymbol{W}_f|^2 \right] + \text{const.}, \tag{9}$$

where $\mathcal{B}$ and $\mathcal{C}$ are the sets defined as $\{b_{kfl}\}_{kfl}$ and $\{c_{klt}\}_{klt}$, respectively. The aim of ILRMA is to estimate the demixing vectors $\boldsymbol{w}_{kf}$ and the source model parameters b_{kfl}, c_{klt} that minimize (9) with only the observed mixture $\boldsymbol{x}_{ft}$.

2.4 Update of Source Model Parameters

We can derive the following multiplicative update rules of the source model parameters by applying the auxiliary-function method to (9) [7]:

$$b_{kfl} \leftarrow b_{kfl} \cdot \frac{\sum_t |y_{kft}|^2 c_{klt} (\sum_i b_{kfi} c_{kit})^{-2}}{\sum_t c_{klt} (\sum_i b_{kfi} c_{kit})^{-1}}, \tag{10}$$

$$c_{klt} \leftarrow c_{klt} \cdot \frac{\sum_f |y_{kft}|^2 b_{kfl} (\sum_i b_{kfi} c_{kit})^{-2}}{\sum_f b_{kfl} (\sum_i b_{kfi} c_{kit})^{-1}}. \tag{11}$$

These update rules guarantee the convergence of (9).

2.5   Update of Demixing Vectors

We summarized the well-used update rules of demixing vectors for AuxIVA and ILRMA, which is based on our proposed method in the next section.

2.5.1   Sequential Update of Demixing Vector: Iterative Projection One (IP1)

We calculate the derivative of (9) with respect to $\boldsymbol{w}_{kf}$ $(k = 1, \ldots, K)$ to derive the update rules of demixing vectors. Then, we obtain the following system of quadratic equations:

$$\boldsymbol{w}_{mf}^{\mathsf{H}} \boldsymbol{U}_{kf} \boldsymbol{w}_{kf} = \begin{cases} 0 & (m \neq k) \\ 1 & (m = k) \end{cases} \quad (m, k = 1, \ldots, K), \tag{12}$$

where $\boldsymbol{U}_{kf} \in \mathbb{C}^{K \times K}$ is called the *weighted covariance matrix* defined as

$$\boldsymbol{U}_{kf} = \frac{1}{T} \sum_t \frac{\boldsymbol{x}_{ft} \boldsymbol{x}_{ft}^{\mathsf{H}}}{r_{kft}}.$$

This system of quadratic equations is called *hybrid exact-approximate joint diagonalization (HEAD)* [28, 30, 31]. When $K = 2$, the closed-form solution of (12) can be derived by solving a generalized eigenvalue problem [19, 25]. However, for $K \geq 3$, the closed-form solution of the HEAD problem has not yet been found. Instead, we minimize the objective function (9) with respect to only one demixing vector $\boldsymbol{w}_{kf}$ while keeping the others fixed. The resulting update rule is given by

$$\boldsymbol{w}_{kf} \leftarrow \boldsymbol{U}_{kf}^{-1} \boldsymbol{W}_f^{-1} \boldsymbol{e}_k, \tag{13}$$

$$\boldsymbol{w}_{kf} \leftarrow \frac{\boldsymbol{w}_{kf}}{\sqrt{\boldsymbol{w}_{kf}^{\mathsf{H}} \boldsymbol{U}_{kf} \boldsymbol{w}_{kf}}}. \tag{14}$$

It is referred to as IP1 [21]. Algorithm 1 summarizes the conventional ILRMA with IP1.

2.5.2   Pairwise Update Rules of Demixing Vectors: Iterative Projection Two (IP2)

The closed-form solution of HEAD is available for two sources by solving the following generalized eigenvalue problem:

$$\boldsymbol{U}_{2f} \boldsymbol{w}_{kf} = \lambda_{kf} \boldsymbol{U}_{1f} \boldsymbol{w}_{kf} \quad (k = 1, 2), \tag{15}$$

where λ_{1f} and λ_{2f} $(\lambda_{1f} \geq \lambda_{2f})$ are the corresponding eigenvalues of $\boldsymbol{w}_{1f}$ and $\boldsymbol{w}_{2f}$. Therefore, the two demixing vectors $\boldsymbol{w}_{1f}, \boldsymbol{w}_{2f}$ can be updated simultaneously [20, 32].

Algorithm 1: Conventional ILRMA with IP1.

Input   : Observed signals $\boldsymbol{x}_{ft}$ $(\forall f, t)$
Output : Estimated signals $\boldsymbol{y}_{ft}$ $(\forall f, t)$
Set the number of iterations N_{itr}
Initialize demixing matrices $\boldsymbol{W}_f$ $(\forall f)$
Initialize source model parameters $b_{kfl}, c_{klt},$ $(\forall k, f, t, l)$
$\boldsymbol{y}_{ft} \leftarrow \boldsymbol{W}_f \boldsymbol{x}_{ft},$ $(\forall f, t)$
for $n_{itr} = 1, \ldots, N_{itr}$ **do**

    for $k = 1, \ldots, K$ **do**

        $b_{kfl} \leftarrow b_{kfl} \dfrac{\sum_t |y_{kft}|^2 c_{klt} \left(\sum_i b_{kfi} c_{kit}\right)^{-2}}{\sum_t c_{klt} \left(\sum_i b_{kfi} c_{kit}\right)^{-1}}$ $(\forall l, f)$

        $c_{klt} \leftarrow c_{klt} \dfrac{\sum_f |y_{kft}|^2 b_{kfl} \left(\sum_i b_{kfi} c_{kit}\right)^{-2}}{\sum_f b_{kfl} \left(\sum_i b_{kfi} c_{kit}\right)^{-1}}$ $(\forall l, t)$

        $r_{kft} \leftarrow \sum_l b_{kfl} c_{ktl}$ $(\forall f, t)$
        $\boldsymbol{U}_{kf} \leftarrow \frac{1}{T} \sum_t \frac{1}{r_{kft}} \boldsymbol{x}_{ft} \boldsymbol{x}_{ft}^{\mathsf{H}}$ $(\forall f)$
        $\boldsymbol{w}_{kf} \leftarrow \boldsymbol{U}_{kf}^{-1} \boldsymbol{W}_f^{-1} \boldsymbol{e}_k$
        $\boldsymbol{w}_{kf} \leftarrow \dfrac{\boldsymbol{w}_{kf}}{\sqrt{\boldsymbol{w}_{kf}^{\mathsf{H}} \boldsymbol{U}_{kf} \boldsymbol{w}_{kf}}}$

    end

    $\boldsymbol{y}_{ft} \leftarrow \boldsymbol{W}_f \boldsymbol{x}_{ft}$ $(\forall f, t)$
end

For $K \geq 3$, the joint update method for two demixing vectors in the case of three or more sources has been proposed [19]. This method was originally proposed for AuxIVA and achieved a higher separation performance with fewer iteration steps. We obtain the following system of $2K$ quadratic equations by calculating $\partial \mathcal{L}^+ / \partial \boldsymbol{w}_{mf} = \boldsymbol{0}$ and $\partial \mathcal{L}^+ / \partial \boldsymbol{w}_{nf} = \boldsymbol{0}$ for all $k \neq m, n$ and $m \neq n$:

$$\boldsymbol{w}_{mf}^{\mathsf{H}} \boldsymbol{U}_{mf} \boldsymbol{w}_{mf} = 1, \quad \boldsymbol{w}_{mf}^{\mathsf{H}} \boldsymbol{U}_{nf} \boldsymbol{w}_{nf} = 0, \tag{16}$$

$$\boldsymbol{w}_{nf}^{\mathsf{H}} \boldsymbol{U}_{mf} \boldsymbol{w}_{mf} = 0, \quad \boldsymbol{w}_{nf}^{\mathsf{H}} \boldsymbol{U}_{nf} \boldsymbol{w}_{nf} = 1, \tag{17}$$

$$\boldsymbol{w}_{kf}^{\mathsf{H}} \boldsymbol{U}_{mf} \boldsymbol{w}_{mf} = 0, \quad \boldsymbol{w}_{kf}^{\mathsf{H}} \boldsymbol{U}_{nf} \boldsymbol{w}_{nf} = 0. \tag{18}$$

The pairwise update rule of $\boldsymbol{w}_\ell$ $(\ell = m, n)$ is derived by solving (16)–(18) as follows [19];

$$\boldsymbol{P}_{\ell f} \leftarrow \boldsymbol{U}_{\ell f}^{-1} \boldsymbol{W}_f^{-1} \begin{bmatrix} \boldsymbol{e}_m & \boldsymbol{e}_n \end{bmatrix}, \tag{19}$$

$$\boldsymbol{Z}_{\ell f} \leftarrow \boldsymbol{P}_{\ell f}^{\mathsf{H}} \boldsymbol{U}_{\ell f} \boldsymbol{P}_{\ell f}, \tag{20}$$

$$\boldsymbol{w}_{\ell f} \leftarrow \dfrac{\boldsymbol{P}_{\ell f} \boldsymbol{z}_{\ell f}}{\sqrt{\boldsymbol{z}_{\ell f}^{\mathsf{H}} \boldsymbol{Z}_{\ell f} \boldsymbol{z}_{\ell f}}}, \tag{21}$$

where $\boldsymbol{z}_{\ell f}$ is the eigenvector of $\boldsymbol{Z}_{\ell f}$ (see [20] for details). Note that the choice of the two indices m and n is arbitrary as long as $m \neq n$.

2.5.3   Iterative Source Steering (ISS)

Instead of estimating the demixing vector $\boldsymbol{w}_{kf}$ in IP1, ISS [24] updates the entire demixing matrix by estimating a new vector $\boldsymbol{v}_{kf} = \begin{bmatrix} v_{1kf} & \dots & v_{Kkf} \end{bmatrix}^{\mathsf{T}}$,

$$\boldsymbol{W}_f \leftarrow \boldsymbol{W}_f - \boldsymbol{v}_{kf}\boldsymbol{w}_{kf}^{\mathsf{H}}, \tag{22}$$

where the update rule that minimizes (9) with respect to v_{mkf} is given by

$$v_{mkf} = \begin{cases} 1 - (\boldsymbol{w}_{kf}^{\mathsf{H}}\boldsymbol{U}_{kf}\boldsymbol{w}_{kf})^{-\frac{1}{2}} & (m = k), \\ \dfrac{\boldsymbol{w}_{mf}^{\mathsf{H}}\boldsymbol{U}_{mf}\boldsymbol{w}_{kf}}{\boldsymbol{w}_{kf}^{\mathsf{H}}\boldsymbol{U}_{mf}\boldsymbol{w}_{kf}} & (m \neq k). \end{cases} \tag{23}$$

Furthermore, from the demixing model $y_{kft} = \boldsymbol{w}_{kf}^{\mathsf{H}}\boldsymbol{x}_{ft}$, the following are the inverse-free update rules of $\boldsymbol{v}_{kf}$ and the output estimated signal $\boldsymbol{y}_{ft}$:

$$v_{mkf} = \frac{\sum_t \frac{y_{mft}y_{kft}^*}{r_{mft}}}{\sum_t \frac{|y_{kft}|^2}{r_{mft}}}, \tag{24}$$

$$\boldsymbol{y}_{ft} \leftarrow \boldsymbol{y}_{ft} - \boldsymbol{v}_{kf}y_{kft}. \tag{25}$$

3   Repeated Update of Demixing Vectors for ILRMA

3.1   Our Motivation and Approach

IVA with IP2 [19] and ILRMA with IP2 [18] showed a higher performance than those with IP1. These results imply that the performance will be further improved if more demixing vectors, ideally all of them, are updated simultaneously such as

$$\boldsymbol{W}_f \leftarrow \arg\min_{\boldsymbol{W}_f} \mathcal{L}^+(\mathcal{W}, \mathcal{B}, \mathcal{C}). \tag{26}$$

However, it leads to the HEAD problem and no closed-form solution has yet been found for $K \geq 3$ as mentioned in the previous section.

Instead of solving the HEAD problem in a closed-form manner, we propose an "approximately" simultaneous update of demixing vectors by simply repeating the demixing vector updates with IP1, IP2, or ISS several times while keeping the source model parameters fixed. Since each update reduces the

objective function of (9), it is expected that the demixing matrix converges the solution of the HEAD problem by repeating the demixing matrix update.

We can see this as follows. From (12), we rewrite the HEAD problem as

$$\boldsymbol{W}_f \begin{bmatrix} \boldsymbol{U}_{1f}\boldsymbol{w}_{1f} & \dots & \boldsymbol{U}_{Kf}\boldsymbol{w}_{Kf} \end{bmatrix} = \boldsymbol{E}_K, \tag{27}$$

where $\boldsymbol{E}_K$ is the $K \times K$ identity matrix. Therefore, we can check how close the demixing matrix is to the solution of HEAD by visualizing the left side of (27). Figure 1 shows one example of convergence to the solution of HEAD by iteration steps. Each colormap shows the element-wise absolute value of the left-hand side of (27). The initial demixing matrices were set to the identity matrices. The weighted covariance matrices were set to the random Hermitian matrices. As shown in Figure 1, the repeated updates of demixing vectors considerably improve the estimated solution of HEAD. This example implies that a set of repeated updates of demixing vectors can work as a simultaneous update of demixing vectors.

| Method	Iteration				
	1	2	5	10	20
IP1					
IP2					
ISS					
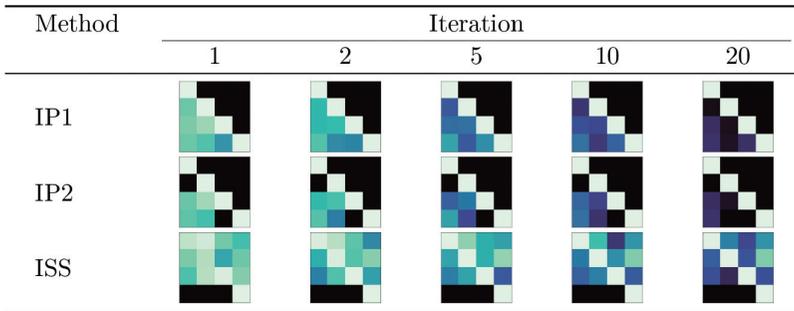
Figure 1: Convergence of demixing vectors to the solution of HEAD by repeated updates. The ideal solution is the identity matrix; only the diagonal elements are bright, and the others are dark. This figure shows an example of convergence for four sources. Each column shows element-wise absolute values of the demixing matrices after applying each update method.

The outline of the proposed procedures is summarized in Algorithm 2. We call this algorithm *ILRMA-Rep*. Figure 2 illustrates how the demixing vectors and the source model parameters are updated.

3.2 Efficient Algorithm for Repeated Update

Updates of demixing matrices with IP1 or IP2 include a matrix inversion. It could cause a large computational complexity for repeating the update of demixing vectors. However, the matrix inversion lemma (MIL) is available to reduce complexities because the covariance matrices are fixed within each

---

Algorithm 2: Proposed ILRMA-Rep.

Input   : Observed signals $\boldsymbol{x}_{ft}$ $(\forall f, t)$
Output : Estimated signals $\boldsymbol{y}_{ft}$ $(\forall f, t)$
Set the update method of demixing vectors
Set the number of iterations N_{itr}
Set the number of "repeats" N_{rep}
Initialize demixing matrices $\boldsymbol{W}_f$ $(\forall f)$
Initialize source model parameters b_{kfl}, c_{klt}, $(\forall k, f, t, l)$
$\boldsymbol{y}_{ft} \leftarrow \boldsymbol{W}_f \boldsymbol{x}_{ft}$, $(\forall f, t)$
for $n_{itr} = 1, \ldots, N_{itr}$ **do**

  for $k = 1, \ldots, K$ **do**

  $\quad b_{kfl} \leftarrow b_{kfl} \dfrac{\sum_t |y_{kft}|^2 c_{klt} \left(\sum_i b_{kfi} c_{kit}\right)^{-2}}{\sum_t c_{klt} \left(\sum_i b_{kfi} c_{kit}\right)^{-1}}$ $(\forall l, f)$

  $\quad c_{klt} \leftarrow c_{klt} \dfrac{\sum_f |y_{kft}|^2 b_{kfl} \left(\sum_i b_{kfi} c_{kit}\right)^{-2}}{\sum_f b_{kfl} \left(\sum_i b_{kfi} c_{kit}\right)^{-1}}$ $(\forall l, t)$

  $\quad r_{kft} \leftarrow \sum_l b_{kfl} c_{ktl}$ $(\forall f, t)$
  $\quad \boldsymbol{U}_{kf} \leftarrow \frac{1}{T} \sum_t \frac{1}{r_{kft}} \boldsymbol{x}_{ft} \boldsymbol{x}_{ft}^{\mathsf{H}}$ $(\forall f)$

  end

  for $n_{rep} = 1, \ldots, N_{rep}$ **do**

  switch *update method of demixing vectors* **do**

    case *IP1* **do**

      for $k = 1, \ldots, K$ **do**
      $\quad$ Update $\boldsymbol{w}_{kf}$ by (13) and (14)
      end

    case *IP2* **do**

      for $(m, n) = (1, 2), \ldots, (K-1, K)$ **do**
        Calculate $\boldsymbol{P}_{mf}$ and $\boldsymbol{P}_{nf}$ by (19)
        Calculate $\boldsymbol{Z}_{mf}$ and $\boldsymbol{Z}_{nf}$ by (20)
        Update $\boldsymbol{w}_{mf}$ and $\boldsymbol{w}_{nf}$ by (21)
      end

    case *ISS* **do**

      for $k = 1, \ldots, K$ **do**
        for $m = 1, \ldots, K$ **do**
        $\quad$ Calculate v_{kmf} by (23)
        end
        Update $\boldsymbol{W}_f$ with $\boldsymbol{v}_{kf}$ by (22)
      end

    end

  end

  end
  $\boldsymbol{y}_{ft} \leftarrow \boldsymbol{W}_f \boldsymbol{x}_{ft}$ $(\forall f, t)$
end

$\boldsymbol{w}_k$: demixing vector for k-th source
$\boldsymbol{U}_k$: covariance matrix for k-th source

(a) Conventional ILRMA

(b) Proposed ILRMA-Rep
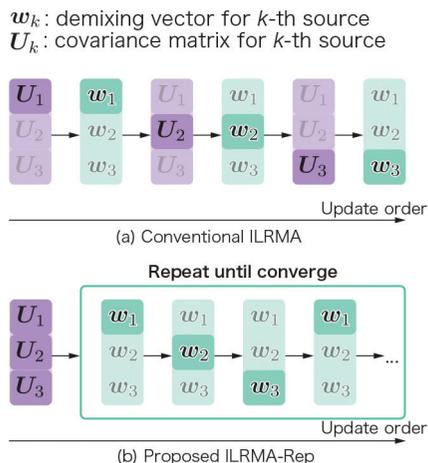
Figure 2: The diagram between conventional ILRMA (a) and proposed ILRMA-Rep (b) where $K = 3$, for example. Blight boxes in each subfigure indicate the update of corresponding parameters.

"repeat." In this section, we derive an efficient algorithm for the repeated update of demixing vectors.

MIL states that:

$$(\boldsymbol{X} + \boldsymbol{\Psi}\boldsymbol{\Omega})^{-1} = \boldsymbol{X}^{-1} - \boldsymbol{X}^{-1}\boldsymbol{\Psi}(\boldsymbol{E}_J + \boldsymbol{\Omega}\boldsymbol{X}^{-1}\boldsymbol{\Psi})^{-1}\boldsymbol{\Omega}\boldsymbol{X}^{-1}, \qquad (28)$$

where $\boldsymbol{X}$ is an invertible square matrix of shape $K \times K$, $\boldsymbol{\Psi}$ and $\boldsymbol{\Omega}$ are matrices of shape $K \times J$ and $J \times K$ ($J \leq K$) respectively, and $\boldsymbol{E}_J$ is the $J \times J$ identity matrix.

Let us consider avoiding the matrix inversion of IP1 included in (13). Let $\boldsymbol{A}_f = \boldsymbol{W}_f^{-1}$ and $\boldsymbol{a}_{kf} = \boldsymbol{A}_f\boldsymbol{e}_k$, if we update $\boldsymbol{A}_f$ together with $\boldsymbol{W}_f$, (13) can be easily calculated as $\hat{\boldsymbol{w}}_{kf} = \boldsymbol{U}_{kf}^{-1}\boldsymbol{a}_{kf}$ where $\hat{\boldsymbol{w}}_{kf}$ denote the left side of (13). Then, the point is how to update $\boldsymbol{A}_f$ without the matrix inversion. We rewrite (13) as

$$\boldsymbol{W}_f \leftarrow \boldsymbol{W}_f + \boldsymbol{e}_k(\hat{\boldsymbol{w}}_{kf} - \hat{\boldsymbol{w}}_{kf})^{\mathsf{H}}, \qquad (29)$$

Then, by using the matrix inversion lemma with $\boldsymbol{\Psi} \leftarrow \boldsymbol{e}_k$ and $\boldsymbol{\Omega} \leftarrow (\hat{\boldsymbol{w}}_{kf} - \boldsymbol{w}_{kf})^{\mathsf{H}}$ in (28), we obtain the update rule of $\boldsymbol{A}_f$ without matrix inversion. Finally, the update rules without the matrix inversion are summarized as follows.

$$\hat{\boldsymbol{w}}_{kf} \leftarrow \boldsymbol{U}_{kf}^{-1}\boldsymbol{a}_{kf}, \qquad (30)$$

$$\hat{\boldsymbol{w}}_{kf} \leftarrow \frac{\hat{\boldsymbol{w}}_{kf}}{\sqrt{\hat{\boldsymbol{w}}_{kf}^{\mathsf{H}}\boldsymbol{U}_{kf}\hat{\boldsymbol{w}}_{kf}}}, \qquad (31)$$

$$\boldsymbol{d}_{kf} := \hat{\boldsymbol{w}}_{kf} - \boldsymbol{w}_{kf} \tag{32}$$

$$\boldsymbol{A}_f \leftarrow \boldsymbol{A}_f - \frac{\boldsymbol{a}_{kf}\left(\boldsymbol{d}_{kf}^{\mathsf{H}}\boldsymbol{A}_f\right)}{1 + \boldsymbol{d}_{kf}^{\mathsf{H}}\boldsymbol{a}_{kf}}, \tag{33}$$

$$\boldsymbol{W}_f \leftarrow \boldsymbol{W}_f + \boldsymbol{e}_k \boldsymbol{d}_{kf}^{\mathsf{H}}. \tag{34}$$

These update rules were first derived in the context of online BSS [27]. Henceforth, we call this version *IP1-MIL*.

Likewise, by setting $\boldsymbol{\Psi} \leftarrow [\boldsymbol{e}_m\ \boldsymbol{e}_n]$ and $\boldsymbol{\Omega} \leftarrow [(\hat{\boldsymbol{w}}_{mf} - \boldsymbol{w}_{mf})\,(\hat{\boldsymbol{w}}_{nf} - \boldsymbol{w}_{nf})]^{\mathsf{H}}$ in (28), the efficient update rule of IP2 is given by

$$\boldsymbol{P}_{\ell f} \leftarrow \boldsymbol{U}_{\ell f}^{-1}[\boldsymbol{a}_{\ell f}\ \boldsymbol{a}_{nf}] \qquad\qquad (\ell = m,\, n), \tag{35}$$

$$\boldsymbol{Z}_{\ell f} \leftarrow \boldsymbol{P}_{\ell f}^{\mathsf{H}}\boldsymbol{U}_{\ell f}\boldsymbol{P}_{\ell f} \qquad\qquad (\ell = m,\, n), \tag{36}$$

$$\hat{\boldsymbol{w}}_{\ell f} \leftarrow \frac{\boldsymbol{P}_{\ell f}\boldsymbol{z}_{\ell f}}{\sqrt{\boldsymbol{z}_{\ell f}^{\mathsf{H}}\boldsymbol{Z}_{\ell f}\boldsymbol{z}_{\ell f}}}, \qquad\qquad (\ell = m,\, n), \tag{37}$$

$$\boldsymbol{d}_{\ell f} := \hat{\boldsymbol{w}}_{\ell f} - \boldsymbol{w}_{\ell f} \qquad\qquad (\ell = m, n), \tag{38}$$

$$\boldsymbol{G} := \boldsymbol{E}_2 + \begin{bmatrix} \boldsymbol{d}_{mf}^{\mathsf{H}} \\ \boldsymbol{d}_{nf}^{\mathsf{H}} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_{mf} & \boldsymbol{a}_{nf} \end{bmatrix}, \tag{39}$$

$$\boldsymbol{A}_f \leftarrow \boldsymbol{A}_f - \begin{bmatrix} \boldsymbol{a}_{mf} & \boldsymbol{a}_{nf} \end{bmatrix} \boldsymbol{G}^{-1} \left(\begin{bmatrix} \boldsymbol{d}_{mf}^{\mathsf{H}} \\ \boldsymbol{d}_{nf}^{\mathsf{H}} \end{bmatrix} \boldsymbol{A}_f \right), \tag{40}$$

$$\boldsymbol{W}_f \leftarrow \boldsymbol{W}_f + \boldsymbol{e}_m \boldsymbol{d}_{mf}^{\mathsf{H}} + \boldsymbol{e}_n \boldsymbol{d}_{nf}^{\mathsf{H}}, \tag{41}$$

where $\boldsymbol{E}_2$ is the 2×2 identity matrix. Note that $\boldsymbol{G}$ is a matrix of shape 2×2 and thus the cost of its inversion is much less than those of $\boldsymbol{W}_f$ and $\boldsymbol{U}_{kf}$. To the best of our knowledge, this is the first derivation of IP2 with the MIL. Henceforth, we call this version *IP2-MIL*.

3.3   *Complexity Analysis*

This subsection outlines the time complexity of the demixing matrix update method for each iteration and frequency bin. For IP1 and IP2, the calculation of $(\boldsymbol{W}_f\boldsymbol{U}_{kf})^{-1}$ in (13) and (19), respectively, takes $O(K^3)$ time at each source k. For ISS by (23), the calculation of $\boldsymbol{v}_{kf}$ takes $O(K^3)$ time. Therefore, the total time complexity of updating the demixing matrix $\boldsymbol{W}_f$ for all sources $k = 1, \dots, K$ is $O(K^4)$ with these methods. In ILRMA-Rep, the covariance matrices $\boldsymbol{U}_{kf}$ are fixed, and the repeat of demixing matrix updates takes $O(N_{\text{rep}}K^4)$ time, where N_{rep} is the number of repeats. The use of MIL allows for efficient updating of $\boldsymbol{W}_f^{-1}$. More specifically, $O(K^4)$ can be computed for the first repeat only, and $O(K^3)$ for subsequent $(N_{\text{rep}} - 1)$ repeats. Therefore, the total time complexity of IP1-MIL and IP2-MIL for all the sources takes

the time complexity of $O(N_{\mathrm{rep}}K^3 + K^4)$. Note that MIL cannot be applied to ISS because $\boldsymbol{W}_f^{-1}$ is not calculated in ISS. In addition, we have an alternative for ISS using (24) instead of (23). As for ISS by (24), the calculation of $\boldsymbol{v}_{kf}$ takes $O(TK)$ time, and the total time complexity of updating the demixing matrix $\boldsymbol{W}_f$ for all sources $k = 1, \ldots, K$ is $O(TK^2)$. From this, ILRMA-Rep using ISS by (24) takes $O(N_{\mathrm{rep}}TK^2)$ in total. Repeating the update by (24) should take more time than ISS by (23) because $T \gg K^2$ in most cases for normal BSS. Therefore, we hereafter use (23) for ISS update in this paper. Table 1 summarizes the complexities of the update rules.

Table 1: Complexities per each iteration and frequency bin.

Method	Complexity	
	$\boldsymbol{U}_{kf}\ (\forall k)$	$\boldsymbol{w}_{kf}$ or $\boldsymbol{v}_{kf}\ (\forall k)$
IP1, IP2	$O(TK^3)$	$O(N_{\mathrm{rep}}K^4)$
IP1-MIL, IP2-MIL	$O(TK^3)$	$O(N_{\mathrm{rep}}K^3 + K^4)$
ISS by (24)	$O(TK^3)$	$O(N_{\mathrm{rep}}K^4)$
ISS by (25)	——	$O(N_{\mathrm{rep}}TK^2)$

4 Experimental Validation

4.1 Solving the HEAD Problem

4.1.1 Runtime Comparison

Since the runtime performance may differ from theoretical complexity, as described in Section 3.3, we first compare the runtime performance characteristics of the different methods. This simulation was run using Python language on a workstation powered by a 96-core AMD EPYC 7643 processor at 2.3 GHz. We generate sets of K random Hermitian matrices $\boldsymbol{U}_{kf}\ (k = 1, \ldots, K)$ by the following procedure:

1. Generate random complex values

$$y_{kft}, z_{kft} \sim \mathcal{N}(0, 1) \quad (\forall k, f, t),$$
$$x_{kft} := y_{kft} + j z_{kft} \quad (\forall k, f, t).$$

2. Generate random observed signals as $\boldsymbol{x}_{ft} := \begin{bmatrix} x_{1ft} & \cdots & x_{Kft} \end{bmatrix}^{\mathsf{T}} \quad (\forall f, t).$

3. Calculate random covariance matrices as

$$\boldsymbol{U}_{kf} \leftarrow \frac{1}{T} \sum_t \frac{\boldsymbol{x}_{ft} \boldsymbol{x}_{ft}^{\mathsf{H}}}{\sqrt{\sum_f |x_{kft}|^2}} \quad (\forall k, f).$$

The initial demixing matrices were set to the identity matrix. We set the number of time frame T to 100 and the number of frequency bins F to 257. For ISS, we update the demixing matrices by (23) in this experiment because T is large. We ran 20 iterations of IP1, IP1-MIL, IP2, IP2-MIL, and ISS by changing $K = 2, 4, 6, 8, 10, 12, 14, 16$. The covariance matrices are fixed in each K. Table 2 shows the average runtimes for IP1, IP1-MIL, IP2, IP2-MIL, and ISS over 100 trials.

Table 2: Runtimes (ms) with and without the matrix inversion lemma (MIL).

Methods	Number of channels							
	2	4	6	8	10	12	14	16
IP1	5.7	16.9	40.7	79.5	151.1	222.3	342.6	503.6
IP1-MIL	4.8	14.5	32.6	62.6	112.6	180.8	312.6	465.0
IP2	12.4	33.8	68.3	122.9	202.4	302.0	461.7	720.3
IP2-MIL	13.6	32.1	55.2	90.0	135.4	188.5	283.1	501.2
ISS	4.4	16.4	45.2	84.8	174.1	265.2	511.6	793.1

We can see that the fastest algorithm depends on the number of channels. For example, ISS is the best in $K = 2$, but IP1-MIL is the best in $K = 6$. The table also shows that IP1-MIL and IP2-MIL were almost consistently faster than IP1 and IP2, except for IP2 and IP2-MIL in $K = 2$. This exception might be because computing temporal variables in IP2-MIL take more time than the matrix inversion when $K = 2$. In the following evaluation of separation performance, IP1-MIL and IP2-MIL will respectively be referred to as IP1 and IP2 because the resulting outputs are identical to those of the original IP1 and IP2.

4.2   Blind Separation of Music Mixtures

4.2.1   Evaluation Criteria

We evaluated the separation and convergence performance characteristics of our proposed ILRMA-Rep with convolutive music mixtures. The separation performance criterion is the scale-invariant source-to-distortion ratio (SI-SDR) [S.], defined as

$$\text{SI-SDR}(\boldsymbol{s}, \hat{\boldsymbol{s}}) = 10 \log_{10} \frac{\|\alpha \boldsymbol{s}\|^2}{\|\alpha \boldsymbol{s} - \hat{\boldsymbol{s}}\|^2}, \tag{42}$$

$$\alpha = \frac{\hat{\boldsymbol{s}}^{\mathsf{T}} \boldsymbol{s}}{\|\boldsymbol{s}\|^2}, \tag{43}$$

where $\boldsymbol{s} \in \mathbb{R}^L$ is the reference signal and $\hat{\boldsymbol{s}} \in \mathbb{R}^L$ is the estimated signal in the discrete time domain of length L. A higher SI-SDR indicates higher separation

performance and quality. We also measured SI-SDR improvement (SI-SDRi) defined as the difference in SI-SDR between the estimated and mixture signals:

$$\text{SI-SDRi}(\boldsymbol{s}, \hat{\boldsymbol{s}}) = \text{SI-SDR}(\boldsymbol{s}, \hat{\boldsymbol{s}}) - \text{SI-SDR}(\boldsymbol{s}, \boldsymbol{x}), \qquad (44)$$

where $\boldsymbol{x} \in \mathbb{R}^L$ is the mixture signal in the discrete time domain. In the following experiment, we use the individual reverberant signals at each microphone as reference signals.

4.2.2   Dataset and Simulation Setup

We used the MUSDB18-HQ dataset [22] as a source signal, including 150 stereo-recorded songs. Each song in the dataset consists of four parts: `bass`, `drums`, `other`, and `vocals`. To perform the image-source method and reduce the computational load, we extracted the left channel of each signal and then downsampled each signal by 16 kHz from the original 44.1 kHz. We created simulated convolutive mixtures using the Python package `pyroomacoustics` [23]. As shown in Figure 3, the number of sound sources and microphones was four, and the microphone array was uniformly linear with a spacing of 2 cm. The reverberation time was approximately 200 ms.
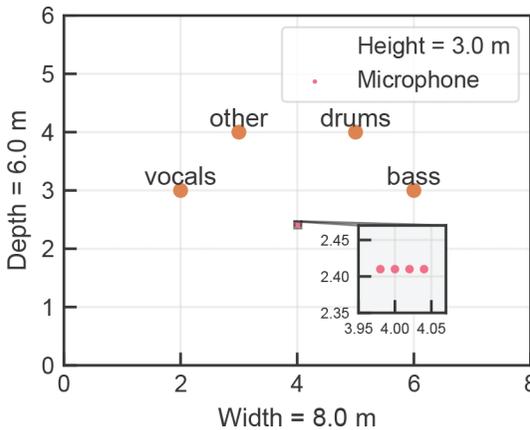
Figure 3: Room geometry and locations of sources and microphone array.

4.2.3   Experimental Conditions

We use the STFT with a 4096-points Hamming analysis window and half-overlap. We set the initial demixing matrices $\boldsymbol{W}_f$ to the identity matrices for all f. Also, we set the initial values of the source models b_{kfl} to 1 and c_{klt} to uniformly distributed values over $]0, 1]$ for all l, k, f, and t. We performed

ILRMA-Rep with 100 iteration steps and several "repeates" N_{rep} in each iteration. After separation, the scale of the estimated signal was restored by back-projection onto the first microphone [16]. Then, we calculated the SI-SDRi for each music mixture and obtained its average along all the songs and channels.

4.2.4   *Implementation Notes*

We implemented all the algorithms in `Python 3.9.6`. For numerical stability, we performed the following regularization after the respective updates in each iteration:

$$b_{kfl} \leftarrow \max(b_{kfl}, \varepsilon), \quad c_{klt} \leftarrow \max(c_{klt}, \varepsilon), \tag{45}$$

where ε is a user-defined parameter. We set ε to 10^{-6} in this experiment.

4.2.5   *Results and Discussion*

Figure 4 shows the separation performance at the end of the 100 iterations for the number of bases $L = 1, 2, 5, 10,$ and 20. Note that the conventional ILRMA corresponds to $N_{\text{rep}} = 1$ in the brightest color bar in Figure 4, and the case of $L = 1$ in the leftmost plot in Figure 4 is nearly equivalent to that of AuxIVA.
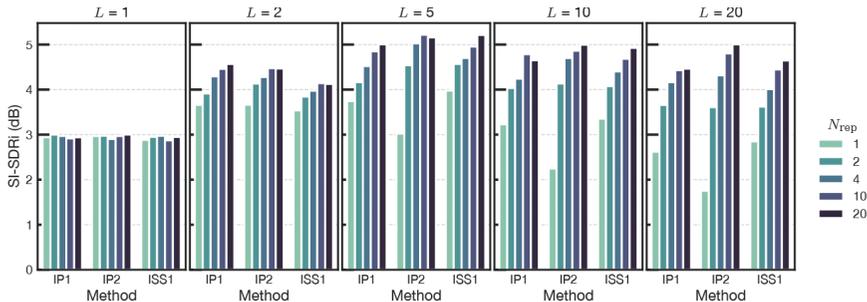
Figure 4: Averaged SI-SDR improvements over 144 samples and four channels at the end of 100 iterations, with various numbers of repeats and bases.

For $L = 1$, the SI-SDRi of all methods did not change markedly with increasing N_{rep}. For $L = 2$ or larger, SI-SDRi was steadily increased by N_{rep} in each L. These results support that performance can be improved by simply repeating the updates of demixing matrices.

One possible explanation for these results would be as follows. ILRMA is a difficult optimization problem in terms of two different types of parameters: demixing matrices and source model parameters (NMF parameters). It is a non-convex optimization, and then it should have local minima. For example, if the

source model is optimized too fast, it may fit mixed (or insufficiently-separated) signals. Then, the demixing matrices are no longer updated, and a poor result might be obtained. Previous studies suggest that quickly updating the demixing matrices [18, 19] or slowly updating the source model parameters [13] leads to better separation performance. Our proposed method repeatedly updates the demixing matrices by fixing the source model parameters. It works by updating the demixing matrices faster than the source model parameters and thus improves the separation performance.

Figure 5 shows the relationship between SI-SDRi at the end of the iteration with $N_{\mathrm{rep}} = 1$ and $N_{\mathrm{rep}} \geq 2$. Each point corresponds to the value before becoming the average in Figure 4. The dashed line in the figure is the line
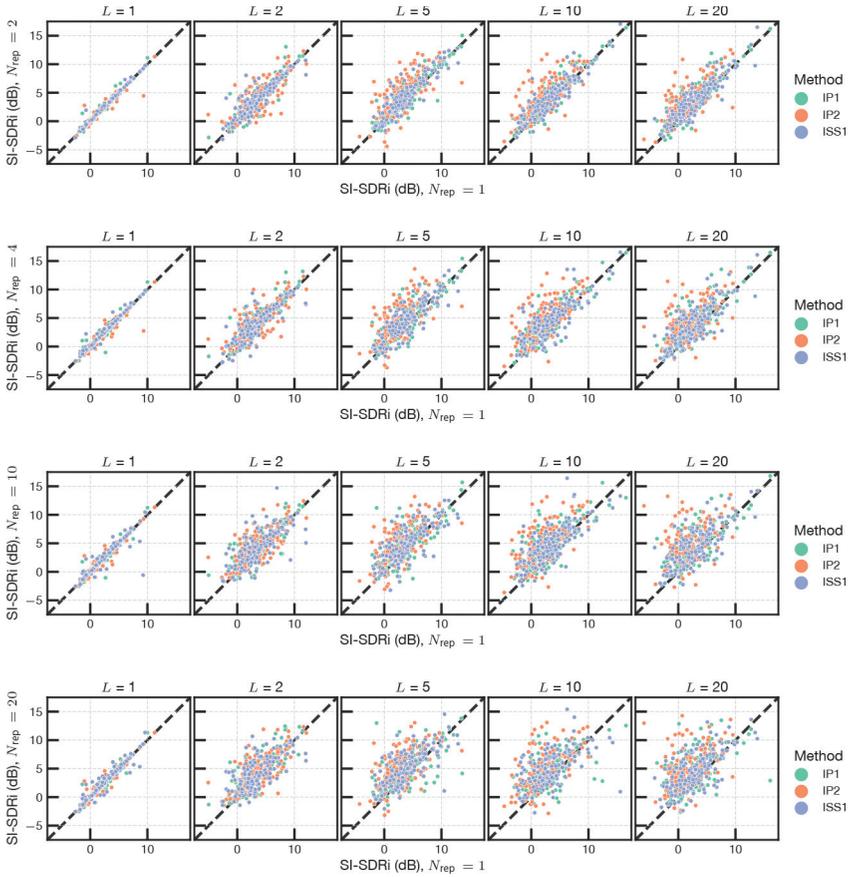
Figure 5: Scatter plots of SI-SDR improvements with and without repeats for each number of bases L.

(a) $L = 1$

(b) $L = 2$

(c) $L = 5$

(d) $L = 10$

(e) $L = 20$
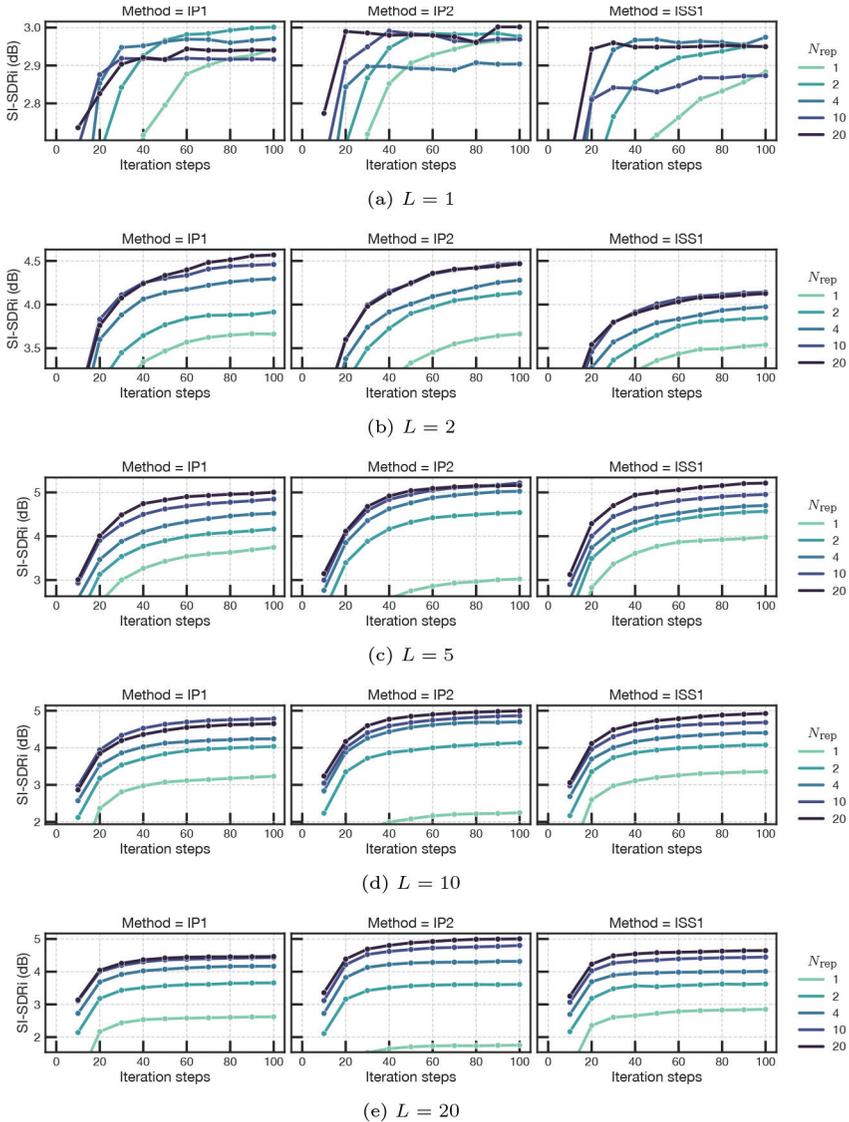
Figure 6: Averaged SI-SDR improvements over 144 samples and four channels along with iteration steps, with various numbers of repeats and bases. Note that vertical axes are aligned for each row, but not for different rows.

where the vertical and horizontal axes are equal; the points above the dashed line indicate that repeats of the demixing vector update improve SI-SDRi. For $L = 1$, the points are clustered near the dashed line for any number of

repetitions. By contrast, when $L = 2$ or larger, more points are clustered above the dashed line, and this tendency is stronger with a larger number of iterations. This result supports the observation that the separation performance improves with repeated updates of the demixing matrix.

Figure 6 shows the separation performance with the number of iteration steps for the number of bases $L = 1, 2, 5, 10,$ and 20. For $L = 1$, the convergence became faster as N_{rep} increased, but the final performance was almost the same, approximately $3\,\text{dB}$. By contrast, for $L = 2$ or larger, both the convergence speed and the final performance became much higher than those for $L = 1$.

5 Conclusion

In this paper, we proposed a repeated update scheme of demixing vectors for independent low-rank matrix analysis. We also derived an efficient update rule for the repeated version of IP2 by applying the matrix inversion lemma. Music source separation experiments with more than 100 songs were performed to evaluate the separation performance. The experimental results implied that the proposed update scheme improved the separation performance. In particular, we experimentally found that when the number of bases is large, the separation performance is markedly improved by repeating parameter updates. In future work, we will further investigate the performance of repeated updates of demixing vectors on other source models using IP1, e.g., in independent deeply learned matrix analysis [12] or the multichannel variational autoencoder method [4].

Biographies

Taishi Nakashima received his B.E. in Engineering from Osaka University, Osaka, Japan, in 2019 and his M.S. in Informatics from Tokyo Metropolitan University, Tokyo, Japan, in 2021. He is pursuing a Ph.D. at Tokyo Metropolitan University and is also a recipient of the JSPS Research Fellowship (DC1) from April 2021. He is an esteemed Student Member of the Acoustical Society of Japan (ASJ) and the IEEE Signal Processing Society (SPS). He received the 24th Best Student Presentation Award of ASJ and the 16th IEEE SPS Japan Student Conference Paper Award in 2022. His research interests primarily focus on blind source separation and acoustic signal processing.

Nobutaka Ono received his B.E., M.S., and Ph.D. degrees from the University of Tokyo, Japan, in 1996, 1998, and 2001, respectively. He became a research associate in 2001 and a lecturer in 2005 at the University of Tokyo.

He moved to the National Institute of Informatics in 2011 as an associate professor and then to Tokyo Metropolitan University in 2017 as a full professor. His research interests include acoustic signal processing, especially microphone array processing, source localization and separation, machine learning, and optimization algorithms. He is a member of IEEE, EURASIP, APSIPA, IPSJ, IEICE, and ASJ. He was a member of IEEE Audio and Acoustic Signal Processing (AASP) Technical Committee from 2014 to 2019. He served as Associate Editor of IEEE Transactions on Audio, Speech, and Language Processing from 2012 to 2015. He received the best paper award at APSIPA ASC in 2018 and 2021 and Sadaoki Furui Prize Paper Award from APSIPA in 2021.

References

[1] E. Cano, D. FitzGerald, A. Liutkus, M. D. Plumbley, and F.-R. Stöter, "Musical Source Separation: An Introduction," *IEEE Signal Processing Magazine*, 36(1), 2019, 31–40, DOI: 10.1109/MSP.2018.2874719.

[2] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, 1st edition, USA: Academic Press, Inc., 2010.

[3] A. Hiroe, "Solution of Permutation Problem in Frequency Domain ICA, Using Multivariate Probability Density Functions," in *Proc. ICA*, 2006, 601–8.

[4] H. Kameoka, L. Li, S. Inoue, and S. Makino, "Supervised Determined Source Separation with Multichannel Variational Autoencoder," *Neural Computation*, 31(9), 2019, 1891–914, DOI: 10.1162/neco_a_01217.

[5] T. Kim, H. T. Attias, S.-Y. Lee, and T.-W. Lee, "Blind Source Separation Exploiting Higher-Order Frequency Dependencies," *IEEE/ACM Trans. Audio, Speech, Language Process*, 15(1), 2006, 70–9.

[6] D. Kitamura, S. Mogami, Y. Mitsui, N. Takamune, H. Saruwatari, N. Ono, Y. Takahashi, and K. Kondo, "Generalized Independent Low-rank Matrix Analysis Using Heavy-Tailed Distributions for Blind Source Separation," *EURASIP Journal on Advances in Signal Processing*, 2018(1), 2018, 28.

[7] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Determined Blind Source Separation Unifying Independent Vector Analysis and Nonnegative Matrix Factorization," *IEEE/ACM Trans. Audio, Speech, Language Process*, 24(9), 2016, 1622–37, DOI: 10.1109/TASLP.2016.2577880.

[8] K. Lange, *MM Optimization Algorithms*, Society for Industrial & Applied Mathematics, U.S., 2016, DOI: 10.1137/1.9781611974409.ch1.

[9] T.-H. Lin and Y. Tsao, "Source Separation in Ecoacoustics: A Roadmap Towards Versatile Soundscape Information Retrieval," *Remote Sensing in Ecology and Conservation*, 6(3), 2020, 236–47.

[10]   Z. Luo, C. Li, and L. Zhu, "A Comprehensive Survey on Blind Source Separation for Wireless Adaptive Processing: Principles, Perspectives, Challenges and New Research Directions," *IEEE Access*, 6, 2018, 66685–708, DOI: 10.1109/ACCESS.2018.2879380.

[11]   S. Makino, ed., *Audio Source Separation*, Springer International Publishing, August 2018, DOI: 10.007/978-3-319-73031-8.

[12]   N. Makishima, S. Mogami, N. Takamune, D. Kitamura, H. Sumino, S. Takamichi, H. Saruwatari, and N. Ono, "Independent Deeply Learned Matrix Analysis for Determined Audio Source Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(10), 2019, 1601–15.

[13]   Y. Mitsui, D. Kitamura, N. Takamune, H. Saruwatari, Y. Takahashi, and K. Kondo, "Independent Low-rank Matrix Analysis based on Parametric Majorization-Equalization Algorithm," in *Proceedings of IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, December 2017, 1–5.

[14]   S. Mogami, Y. Mitsui, N. Takamune, D. Kitamura, H. Saruwatari, Y. Takahashi, K. Kondo, H. Nakajima, and H. Kameoka, "Independent Low-Rank Matrix Analysis Based on Generalized Kullback–Leibler Divergence," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E102-A(2), 2019, 458–63.

[15]   S. Mogami, N. Takamune, D. Kitamura, H. Saruwatari, Y. Takahashi, K. Kondo, and N. Ono, "Independent Low-Rank Matrix Analysis Based on Time-Variant Sub-Gaussian Source Model for Determined Blind Source Separation," *IEEE/ACM Trans. Audio, Speech, Language Process*, 28, 2020, 503–18.

[16]   N. Murata, S. Ikeda, and A. Ziehe, "An Approach to Blind Source Separation Based on Temporal Structure of Speech Signals," *Neurocomputing*, 41(1-4), 2001, 1–24.

[17]   T. Nakashima, R. Scheibler, M. Togami, and N. Ono, "Joint Dereverberation and Separation with Iterative Source Steering," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, June 2021.

[18]   T. Nakashima, R. Scheibler, Y. Wakabayashi, and N. Ono, "Faster Independent Low-rank Matrix Analysis with Pairwise Updates of Demixing Vectors," in *Proceedings of European Signal Processing Conference (EUSIPCO)*, January 2021, 301–5.

[19]   N. Ono, "Fast Algorithm for Independent Component/Vector/Lowrank Matrix Analysis with Three or More Sources," in *Proceedings Spring Meeting of Acoustical Society of Japan*, in Japanese, March 2018, 437–8.

[20]   N. Ono, "Fast Stereo Independent Vector Analysis and Its Implementation on Mobile Phone," in *Proceedings of IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, September 2012.

[21] N. Ono, "Stable and Fast Update Rules for Independent Vector Analysis based on Auxiliary Function Technique," in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2011, 189–92, DOI: 10.1109/ASPAA.2011.6082320.

[22] Z. Rafii, A. Liutkus, F.-R. Sttöter, S. I. Mimilakis, and R. Bittner, "MUSDB18-HQ - an uncompressed version of MUSDB18," August 2019, https://doi.org/10.5281/zenodo.3338373.

[23] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A Python Package for Audio Room Simulation and Array Processing Algorithms," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, 351–5, DOI: 10.1109/ICASSP.2018.8461310.

[24] R. Scheibler and N. Ono, "Fast and Stable Blind Source Separation with Rank-1 Updates," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, 236–40.

[25] R. Scheibler and N. Ono, "MM Algorithms for Joint Independent Subspace Analysis with Application to Blind Single and Multi-Source Extraction," 2020, arXiv: 2004.03926 [eess.SP].

[26] M. Sunohara, C. Haruta, and N. Ono, "Low-latency real-time blind source separation for hearing aids based on time-domain implementation of online independent vector analysis with truncation of non-causal components," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, 216–20, DOI: 10.1109/ICASSP.2017.7952149.

[27] T. Taniguchi, N. Ono, A. Kawamura, and S. Sagayama, "An Auxiliary-Function Approach to Online Independent Vector Analysis for Real-time Blind Source Separation," in *Proceedings of Hands-Free Speech Communication and Microphone Arrays (HSCMA)*, May 2014, 107–11.

[28] A. Weiss, A. Yeredor, S. A. Cheema, and M. Haardt, "The Extended "Sequentially Drilled" Joint Congruence Transformation and Its Application in Gaussian Independent Vector Analysis," *IEEE Transactions on Signal Processing*, 65(23), 2017, 6332–44.

[29] J. Yang, S. Gohel, and B. Vachha, "Current Methods and New Directions in Resting State fMRI," *Clinical Imaging*, 65, 2020, 47–53, DOI: https://doi.org/10.1016/j.clinimag.2020.04.004.

[30] A. Yeredor, "Blind Separation of Gaussian Sources With General Covariance Structures: Bounds and Optimal Estimation," *IEEE Transactions on Signal Processing*, 58(10), 2010, 5057–68, DOI: 10.1109/TSP.2010.2053362.

[31] A. Yeredor, "On Hybrid Exact-Approximate Joint Diagonalization," in *Proceedings of IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, December 2009, 312–5.

[32] T. Yoshioka, T. Nakatani, and M. Miyoshi, "An Integrated Method for Blind Separation and Dereverberation of Convolutive Audio Mixtures," in *Proceedings of European Signal Processing Conference (EUSIPCO)*, August 2008.