IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 20, 2017, accepted April 6, 2017, date of publication April 25, 2017, date of current version June 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2695497

Underdetermined Joint Blind Source Separation of Multiple Datasets

LIANG ZOU[1], XUN CHEN[2], (Member, IEEE), XIANGYANG JI[3], (Member, IEEE), AND Z. JANE WANG[1], (Fellow, IEEE)

[1]Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada
[2]Department of Biomedical Engineering, Hefei University of Technology, Hefei 230009, China
[3]Department of Automation, Tsinghua University, Beijing 100084, China

Corresponding author: Xun Chen (xunchen@ece.ubc.ca)

This work was supported by NPRP from the Qatar National Research Fund (a member of Qatar Foundation) under Grant 7-684-1-127. The statements made herein are solely the responsibility of the authors.

ABSTRACT In this paper, we tackle the problem of jointly separating instantaneous linear underdetermined mixtures of latent sources from multiple data sets, where the number of sources exceeds that of observations in each data set. Currently available blind source separation (BSS) methods, including joint BSS (JBSS) and underdetermined BSS (UBSS), cannot address this underdetermined problem effectively. We exploit the second-order statistics of observations, and present a novel BSS method, referred to as underdetermined joint BSS for multiple data sets (UJBSS-m), as a generalization of our previous work on two data sets. In this paper, the cross correlation between each pair of data sets is modeled by a third-order tensor in which a set of spatial covariance matrices corresponding to different time delays are stacked. Considering the latent common structure of these constructed tensors, the mixing matrices are jointly estimated via joint canonical polyadic decomposition of these specialized tensors. Furthermore, we recover the sources from each data set separately based on the estimated mixing matrices. Simulation results demonstrate that the proposed UJBSS-m method yields superior performances when compared with commonly used single-set UBSS and JBSS methods.

INDEX TERMS Underdetermined joint blind source separation, joint canonical polyadic decomposition, cross correlation.

I. INTRODUCTION

The increasing availability of multiset and multimodal signals has posed new challenges for conventional blind source separation (BSS) methods which are originally designed to analyze one data set at a time. There are many applications involving multiple datasets which have dependence relationships between them and need to be jointly analyzed [1]–[3], such as electroencephalography (EEG), electrocardiography (ECG), and magnetic resonance imaging (MRI) data. Hence, joint blind source separation (JBSS) algorithms have attracted great interest in the fields of signal processing owing to their ability to simultaneously recover the underlying and physiologically meaningful components from multiple datasets. The crucial difference between BSS and JBSS is reflected by the fact that BSS only examines each dataset separately, whereas JBSS generalizes BSS to consider the dependence across multiple datasets [4]. Compared with conventional single-set BSS methods, JBSS generally could yield better performances. In addition, JBSS can keep the extracted components aligned across different datasets, an important feature that is not provided by single-set BSS methods.

The most original JBSS method was likely canonical correlation analysis (CCA), which has been popular to analyze relationships between two sets of variables [5]. It seeks a linear transformation of the observations such that the obtained corresponding source components across two datasets are maximally correlated. A generalization of CCA from two datasets to multiple datasets, the multiset canonical correlation analysis (MCCA), was shown to be flexible and powerful for discovering associations across multiple datasets [6]. Another recent extension of CCA is the joint diagonalization of many cross-cumulate matrices [7], which is especially effective when there is no explicit source distribution known in advance [7]. In addition to CCA-type algorithms, numerous models have been introduced to generalize the idea of single-set BSS to JBSS. Independent component analysis (ICA), for example, has been extended to handle

2169-3536 © 2017 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

multiple datasets [8]. The group ICA and the joint ICA attempt to concatenate multiple datasets into one dataset in the vertical and horizontal dimension respectively, and then the standard ICA can be performed on the concatenated single dataset [8]. Based on the modular Bayesian framework, Groves *et al.* proposed a novel Linked ICA [9], encapsulating the ideas from both the group ICA and joint ICA. Independent vector analysis (IVA) generalizes ICA to multiple datasets by exploring statistical dependences across datasets [8].

It is worth noting that the above mentioned JBSS algorithms were originally proposed for the *determined case*, since they generally assume that the number of sources is equal to or less than that of the observations. This assumption may not be true in some practical applications, due to concerns such as the cost or time issues [10]. However, to our best knowledge, in the current literature there is only very limited work on JBSS methods specifically designed for the *underdetermined case* (i.e., the number of sources is greater than that of observations), even though there have been more single-set underdetermined BSS (UBSS) methods [11]–[17] which can be used to unmix the mixtures from each dataset separately. These UBSS methods can be divided mainly into two categories [18]. Most UBSS methods rely upon the sparsity of source signals in a specific domain, e.g., the time-frequency domain [13], [16], [17]. This category of methods usually require exhaustive computation, especially when the number of sources is large. Many algebraic methods were also proposed for unmixing the mixtures in the underdetermined case, most of which are based on decomposition of different data structures, e.g., covariance matrices [12].

In our previous paper [19], we proposed an underdetermined joint blind source separation method for two datasets (UJBSS-2) based on the decomposition of a specialized tensor. However, it can only jointly estimate the mixing matrices from two datasets and cannot be extended directly to unmix the mixtures from multiple (larger than two) datasets [20]. To fill this gap in the literature, in this paper, we plan to extend the idea of JBSS to the underdetermined case and generalize the idea of underdetermined joint blind source separation (UJBSS) for two datasets to that for multiple datasets.

More specifically, inspired by the MCCA model and the simultaneous diagonalization of covariance matrices [12], [21], [22], we exploit second-order statistics of the observations in each pair of datasets and propose a novel BSS method, termed as the underdetermined joint blind source separation for multiple datasets (UJBSS-m). Unlike the traditional (over)determined JBSS methods, the proposed UJBSS-m consists of two steps: 1) jointly estimate the mixing matrices from multiple datasets, and 2) recover the underlying sources individually based on each mixing matrix estimated in step 1). The most challenging task is to estimate the unknown mixing matrices precisely, which is the main concern of this paper. In this work, this problem is tackled via joint canonical polyadic decomposition of specialized tensors. The dependence information between each pair of datasets is modeled by a third-order tensor where a set of spatial covariance matrices related to different time delays are stacked. Considering the possible combinations of two datasets, the pairs of the corresponding tensors share a common factor and then the mixing matrices (i.e., factor matrices of those tensors) can be jointly estimated by optimization-based methods. The estimated mixing matrices are further used to recover the sources from each dataset. In this work, we explore a novel subspace representation based method [13] to recover the sources.

Our main contributions are summarized as follows:

1) This paper extends the idea of (over)determined JBSS to that of the underdetermined case.

2) Exploiting the cross correlation between each pair of datasets, we propose a novel and effective method to jointly estimate the mixing matrices for multiple datasets. More precise estimates of the mixing matrices can be achieved via the proposed UJBSS-m method compared to several classical single-set UBSS methods and JBSS methods.

3) The proposed UJBSS-m method can be used to solve single set UBSS problems and could achieve better performance in some cases, as demonstrated in the promising application of noise enhanced signal processing.

4) The proposed UJBSS-m method does not rely upon the sparsity of signals and therefore it can be applied to a wide class of signals, e.g., audio/speech and biomedical signals.

II. NOTATIONS AND PRELIMINARIES

In this paper, we generally use the notation of [23], which was adapted by [24] and [25]. A tensor can be interpreted as multi-index numerical array, whereby the order of a tensor is the dimensionality of the array. Scalars, denoted as lowercase letters, e.g., x, are said to be tensors of zero order. Vectors (first-order tensors) are denoted by boldface lowercase letters, e.g., $\boldsymbol{x}$. Matrices (second-order tensors) are denoted by boldface capital letters $\boldsymbol{X}$. Third-order or higher-order tensors are denoted by boldface Euler script letters, e.g., $\mathcal{X}$. The transpose, inverse, Moore-Penrose pseudo inverse, norm are denoted by $(\cdot)^T$, $(\cdot)^{-1}$, $(\cdot)^\dagger$, $\|\cdot\|$.

The operation of matricization reorders the elements of a higher-order tensor into a matrix. For example, mode-n matricization of a Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ yields a matrix $\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times I_2 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N)}$ whose columns are all mode-n fibers arranged in a specifically predefined order. In this paper, tensor element $x_{i_1,i_2,\ldots,i_N}$ corresponds to matrix elements $x_{(n)(i_n,j)}$, where

$$j = 1 + \sum_{\substack{l=1 \\ l \neq n}}^{N} (i_l - 1)\left(\prod_{\substack{k=1 \\ k \neq n}}^{l-1} I_k\right).$$

The inner product of two same-sized tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the sum of the products of their elements, i.e.,

$$< \mathcal{X}, \mathcal{Y} >= \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1,i_2,\ldots,i_N} y_{i_1,i_2,\ldots,i_N}.$$

The Frobenius norm of a tensor $\mathcal{X}$ is the square root of its inner product with itself, i.e.,

$$\|\mathcal{X}\| = \sqrt{<\mathcal{X}, \mathcal{X}>}.$$

The outer product of vectors $\{\boldsymbol{a}^{(n)}\} \in \mathbb{R}^{I_n}, n = 1, 2, \ldots, N$ yields a rank-one tensor $\mathcal{X} = \boldsymbol{a}^{(1)} \circ \boldsymbol{a}^{(2)} \circ \ldots \boldsymbol{a}^{(N)}$ with entries $x_{i_1,i_2,\ldots,i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \ldots a^{(N)}_{i_N}$, where the $\circ$ represents the outer product operation. The superscript in parentheses represents one element in a sequence, e.g., $\boldsymbol{a}^{(n)}$ represents the nth vector in a sequence of vectors.

In order to demonstrate multi-way models, the usual matrix product, such as Kronecker product and Khatri-Rao product, is not sufficient. A frequently used operation is the mode-n product, denoted by $\times_n$. The mode-n product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\boldsymbol{A} \in \mathbb{R}^{J_n \times I_n}$ amounts to the product of all mode-n fibers with $\boldsymbol{A}$ and yields a tensor with the size of $(I_1 \times I_2 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \cdots \times I_N)$, whose entries are given by

$$(\mathcal{X} \times_n \boldsymbol{A})_{i_1,i_2,\ldots,i_{n-1},j_n,i_{n+1}\ldots,i_N} = \sum_{i_n=1}^{I_n} x_{i_1,i_2,\ldots,i_{n-1},i_n,i_{n+1}\ldots i_N} a_{j_n,i_n}.$$

The mode-n product of a tensor and a vector is a special case of the mode-n product of a tensor and a matrix with the size of $(1 \times I_n)$. Note that the order of the result is $(N-1)$, one less than the order of the original tensor. It is often useful to calculate the product of a tensor with a sequence of vectors. Let $\mathcal{X}$ denote a tensor with the size of $I_1 \times I_2 \times \cdots \times I_N$, and let $\{\boldsymbol{a}^{(n)}\}$ $(n = 1, 2, \ldots, N)$, be a sequence of vectors, each with the length of I_n. Then the product of $\mathcal{X}$ with a sequence of vectors in all modes yields a scalar, i.e.,

$$y = \mathcal{X} \times_1 \boldsymbol{a}^{(1)} \times_2 \boldsymbol{a}^{(2)} \times_3 \cdots \times_N \boldsymbol{a}^{(N)} = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N}^{I_N} x_{i_1,i_2,\ldots,i_N} a^{(1)}_{i_1} a^{(2)}_{i_2} \ldots a^{(N)}_{i_N}.$$

We refer the readers to [24] and [25] for further details and discussions about various tensor operations.

III. PROBLEM FORMULATION

The problem of interest here is the underdetermined JBSS for multiple datasets, e.g., K datasets. The M observations of each dataset contain the linear mixtures of the corresponding N sources. We can model the mixing process as follows,

$$\boldsymbol{X}^{(k)} = \boldsymbol{A}^{(k)} \boldsymbol{S}^{(k)} + \boldsymbol{E}^{(k)}, \quad k = 1, 2, \ldots, K. \tag{1}$$

$\boldsymbol{X}^{(k)} = [\boldsymbol{x}^{(k)}_1, \boldsymbol{x}^{(k)}_2, \ldots, \boldsymbol{x}^{(k)}_M]^T$ denotes the M-dimensional observations with real values and $\boldsymbol{x}^{(k)}_m$ is the m^{th} channel of the observations in dataset k. $\boldsymbol{S}^{(k)} = [\boldsymbol{s}^{(k)}_1, \boldsymbol{s}^{(k)}_2, \ldots, \boldsymbol{s}^{(k)}_N]^T$ means the underlying N-dimensional sources with real values and $\boldsymbol{s}^{(k)}_n$ is the n^{th} source for dataset k. $\boldsymbol{A}^{(k)} = [\boldsymbol{a}^{(k)}_1, \boldsymbol{a}^{(k)}_2, \ldots, \boldsymbol{a}^{(k)}_N] \in \mathbb{R}^{M \times N}$ with $M < N$ (i.e., the underdetermined case) denotes the unknown mixing matrix, whose n^{th} column $\boldsymbol{a}^{(k)}_n$ corresponds to the source $\boldsymbol{s}^{(k)}_n$ for dataset k. $\boldsymbol{E}^{(k)}$ means the possible additive noise which is generally assumed to be zero mean, temporally white and uncorrelated with the source signals.

Similar to several existing JBSS methods, e.g. MCCA [6] and JDAIG-SOS [7], we have the following assumptions regarding the sources:

(1) The sources are uncorrelated within each dataset:

$$E\{\boldsymbol{s}^{(k)}_i(t)(\boldsymbol{s}^{(k)}_j(t+\tau))^T\} = 0 \quad \forall\tau, \quad 1 \le i \ne j \le N, \ k = 1, 2, \ldots, K, \tag{I}$$

where $\boldsymbol{s}^{(k)}_i(t)$ is the i-th source in dataset k and $\boldsymbol{s}^{(k)}_j(t+\tau)$ represents the j-th source with the time delay τ in dataset k.

(2) Only the corresponding sources from two different datasets have non-zero correlations:

$$\boldsymbol{D}^{(\tau)} = E\{\boldsymbol{S}^{(k_1)}(t)(\boldsymbol{S}^{(k_2)}(t+\tau))^T\} = Diag(\rho_1(\tau), \rho_2(\tau), \ldots, \rho_N(\tau)), \tag{II}$$

where $Diag(\cdot)$ represents the diagonal matrix, the $\rho_n(\tau) = E\{\boldsymbol{s}^{(k_1)}_n(t)(\boldsymbol{s}^{(k_2)}_n(t+\tau))^T\}$ denotes the covariance between $\boldsymbol{s}^{(k_1)}_n(t)$ and $\boldsymbol{s}^{(k_2)}_n(t+\tau)$. This assumption means that the corresponding sources in multiple datasets are second-order correlated with each other. In addition, the sources within $[\boldsymbol{s}^{(1)}_i, \boldsymbol{s}^{(2)}_i, \ldots, \boldsymbol{s}^{(K)}_i]$ are uncorrelated with the sources within $[\boldsymbol{s}^{(1)}_j, \boldsymbol{s}^{(2)}_j, \ldots, \boldsymbol{s}^{(K)}_j]$ for $1 \le i \ne j \le N$.

The task of estimating the mixing matrices $\{\boldsymbol{A}^{(k)}\}$ and retrieving the underlying sources are not equivalent in the underdetermined case. Therefore, most UBSS methods consist of two stages: estimate the mixing matrices first and then retrieve the underlying sources. The major problem under consideration is to estimate $\{\boldsymbol{A}^{(k)}\}$ jointly up to permutation and scaling. In this paper, this problem is addressed via a specially designed joint tensor decomposition. In addition, retrieving the underlying sources when the mixing matrices are estimated or known is a classic inverse problem [11]. In order to further demonstrate the performance of the proposed mixing matrices estimation method, we also implement an approach for source recovering based on the estimated $\boldsymbol{A}^{(k)}$.

IV. CANONICAL POLYADIC DECOMPOSITION OF TENSOR

A polyadic decomposition aims to decompose a higher-order tensor as a linear combination of rank-one tensors [25], [26]. For the case of a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, it can be written in the form

$$\mathcal{X} = \sum_{n=1}^{N} \boldsymbol{a}_n \circ \boldsymbol{b}_n \circ \boldsymbol{c}_n, \tag{2}$$

where N is a positive integer and $\boldsymbol{a}_n \in R^I, \boldsymbol{b}_n \in R^J, \boldsymbol{c}_n \in R^K$. Equivalently, it can be written element wisely as

$$x_{i,j,k} = \sum_{n=1}^{N} a_{i,n} b_{j,n} c_{k,n}, \tag{3}$$

where $i = 1, 2, \ldots, I, j = 1, 2, \ldots, J$ and $k = 1, 2, \ldots, K$. The rank of a tensor is the smallest number of rank-one

tensors that yield the tensor in the way as (2). If $rank(\mathcal{X}) = N$, (2) is the Canonical Polyadic Decomposition (CPD) of $\mathcal{X}$, which is also known as the canonical decomposition (CANDECOMP) or parallel factor analysis (PARAFAC) [23]. The canonical polyadic approximation means that

$$\begin{aligned}\mathcal{X} &\approx [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!] \\ &\equiv \sum_{n=1}^{N} \boldsymbol{a}_n \circ \boldsymbol{b}_n \circ \boldsymbol{c}_n, \end{aligned} \tag{4}$$

where $N = rank(\mathcal{X})$.

The factor matrices refer to the combination of the vectors corresponding to each rank-one tensor and can be written as

$$\begin{aligned}\boldsymbol{A} &= [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N] \in R^{I\times N} \\ \boldsymbol{B} &= [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_N] \in R^{J\times N} \\ \boldsymbol{C} &= [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_N] \in R^{K\times N}. \end{aligned} \tag{5}$$

To a large extent, the power of CPD mainly stems from its uniqueness property. The uniqueness of CPD means that the decomposition is the only possible combination of rank-one tensors which sum to the objective tensor with the exception of the indeterminacies of column permutation and scaling. The permutation indeterminacy refers to the fact that we can permute the rank-one terms arbitrarily. The scaling indeterminacy means that we can scale the individual column of the factor matrices as long as their product remains the same, i.e.,

$$\mathcal{X} = \sum_{n=1}^{N} (\alpha_n^1 \boldsymbol{a}_n) \circ (\alpha_n^2 \boldsymbol{b}_n) \circ (\alpha_n^3 \boldsymbol{c}_n) \quad \text{if } \alpha_n^1 \alpha_n^2 \alpha_n^3 = 1. \tag{6}$$

The uniqueness condition is based on the rank of tensors. The most famous result on uniqueness of CPD was reported by J. Kruskal [27]. Kruskal's theorem states that the CPD of a third-order tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ is deterministically unique if N (where $N = rank(\mathcal{X})$) satisfies

$$N \leq \frac{k_{\boldsymbol{A}} + k_{\boldsymbol{B}} + k_{\boldsymbol{C}} - 2}{2}, \tag{7}$$

where $k_{\cdot}$ denotes the k-rank of a given matrix $(\cdot)$, meaning that $k_{\cdot}$ is the largest integer that any $k_{\cdot}$ columns of the matrix $(\cdot)$ are linearly independent. Checking deterministic conditions can be cumbersome. De Lathauwer have studied different methods to determine the rank of a tensor and concluded that the decomposition of a third-order tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ is generically unique (i.e., with probability one) [28] provided that N satisfies

$$N \leq K \quad and \quad N(N-1) \leq IJ(I-1)(J-1)/2. \tag{8}$$

Domanov and Lathauwer further complemented the existing bounds for generic uniqueness of the CPD [29] and concluded that the CPD of a third-order tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ of rank N is generically unique if

$$\begin{aligned}&2 \leq I \leq J \leq K \leq N \\ &N \leq \frac{I + J + 2K - 2 - \sqrt{(I-J)^2 + 4K}}{2}, \end{aligned} \tag{9}$$

or

$$\begin{aligned}&3 \leq I \leq J \leq N \leq K \\ &N \leq (I-1)(J-1). \end{aligned} \tag{10}$$

There are two main approaches to compute the CPD of a tensor, namely the linear algebra [30] and optimization based methods [23], [31]. Both types of methods have their own strengths and weaknesses. For a thorough study of the uniqueness conditions and computation, we refer to [25], [28], [32], and the references therein.

V. ALGORITHM FOR ESTIMATING THE MIXING MATRICES IN UJBSS

How to estimate the mixing matrix is still a challenging problem, even for underdetermined case of single dataset. In this paper, we propose a novel and effective algorithm to jointly estimate the mixing matrices from multiple dataset, which can be regarded as an extension of the method based on statistical property of signals, e.g., simultaneous diagonalization of the second order autocovariance and CPD of a specialized tensor [12], [19], [21]. For ease of presentation, we take the case of 3 datasets as an example, e.g., $\boldsymbol{X}^{(1)}$, $\boldsymbol{X}^{(2)}$ and $\boldsymbol{X}^{(3)}$, and it can be easily generalized to the case of more than 3 datasets. The problem is reformulated as joint canonical polyadic decomposition of a sequence of third-order tensors, which share common factor matrices. It should be mentioned that the proposed method is limited to real-valued problems and cannot be directly generalized to complex-valued cases.

A. TENSOR CONSTRUCTION

The cross covariance of the observations with time delay τ, such as the observations in dataset k_1, $\boldsymbol{X}^{(k_1)}(t)$, and the observations in dataset k_2 with time delay τ, $\boldsymbol{X}^{(k_2)}(t+\tau)$, can be formulated as

$$\begin{aligned}&E\{\boldsymbol{X}^{(k_1)}(t)\boldsymbol{X}^{(k_2)}(t+\tau)^T\} \\ &\quad = (\boldsymbol{A}^{(k_1)})E\{\boldsymbol{S}^{(k_1)}(t)\boldsymbol{S}^{(k_2)}(t+\tau)^T\}(\boldsymbol{A}^{(k_2)})^T, \end{aligned} \tag{11}$$

where k_1 and k_2 represent the index of each dataset and range from 1 to 3. Considering the correlations within and between each pair of datasets, the covariance matrices between $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$ with time delay τ satisfy

$$\begin{aligned}\boldsymbol{P}^{(1)} &= E\{\boldsymbol{X}^{(1)}(t)\boldsymbol{X}^{(2)}(t+\tau_1)^T\} = (\boldsymbol{A}^{(1)})\boldsymbol{U}^{(\tau_1)}(\boldsymbol{A}^{(2)})^T, \\ \boldsymbol{P}^{(2)} &= E\{\boldsymbol{X}^{(1)}(t)\boldsymbol{X}^{(2)}(t+\tau_2)^T\} = (\boldsymbol{A}^{(1)})\boldsymbol{U}^{(\tau_2)}(\boldsymbol{A}^{(2)})^T, \\ &\vdots \\ \boldsymbol{P}^{(L)} &= E\{\boldsymbol{X}^{(1)}(t)\boldsymbol{X}^{(2)}(t+\tau_L)^T\} = (\boldsymbol{A}^{(1)})\boldsymbol{U}^{(\tau_L)}(\boldsymbol{A}^{(2)})^T, \end{aligned} \tag{12}$$

in which τ_l means the time delay and the matrix $\boldsymbol{U}^{(\tau_l)} = E\{\boldsymbol{S}^{(1)}(t)\boldsymbol{S}^{(2)}(t+\tau_l)^T\}$ is diagonal, for $l = 1, 2, \ldots, L$.

We stack the sequence of covariance matrices $\boldsymbol{P}^{(1)}, \boldsymbol{P}^{(2)}, \ldots, \boldsymbol{P}^{(L)}$, denoted as $\{\boldsymbol{P}^{(l)}\}$, in a tensor $\mathcal{P} \in \mathbb{R}^{M\times M\times L}$ as follows: $(\mathcal{P})_{i,j,l} = (\boldsymbol{P}^{(l)})_{i,j}$, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, M$, $l = 1, 2, \ldots, L$. We define the matrix $\boldsymbol{U}$

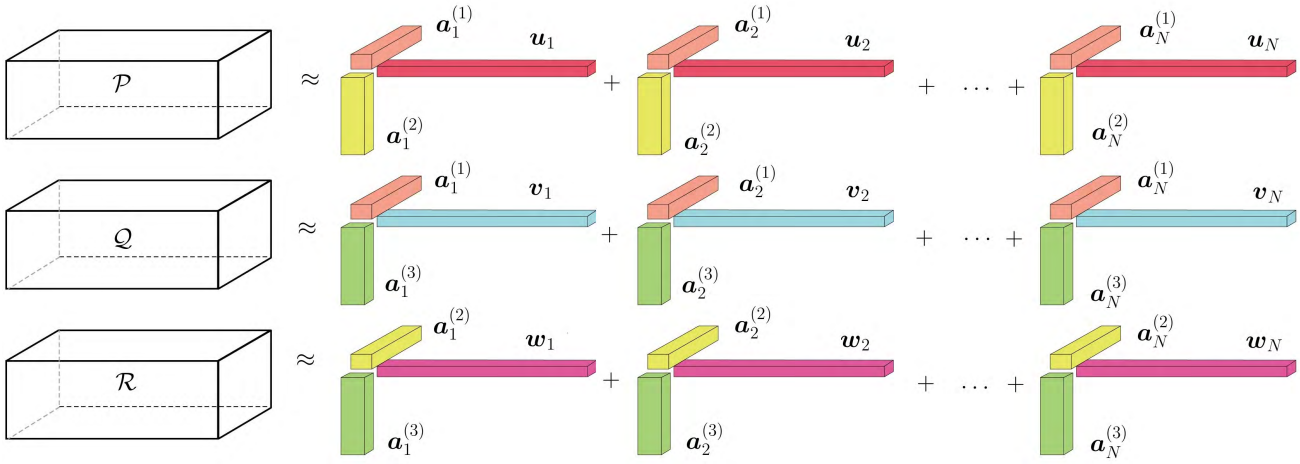

FIGURE 1. Illustration of how to generate tensors by incorporating the dependence information between each pair of datasets.

of size $L \times N$ with the element $\boldsymbol{U}_{l,n} = (\boldsymbol{U}^{(\tau_l)})_{n,n}$, for $l = 1, 2, \ldots, L$, $n = 1, 2, \ldots, N$. Then we can represent $\mathcal{P}$ as (see Fig. 1):

$$\mathcal{P} = \sum_{n=1}^{N} \boldsymbol{a}_n^{(1)} \circ \boldsymbol{a}_n^{(2)} \circ \boldsymbol{u}_n, \tag{13}$$

in which $\boldsymbol{a}_n^{(1)}$ and $\boldsymbol{a}_n^{(2)}$ are the n^{th} column of the mixing matrices $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$ respectively, and $\boldsymbol{u}_n$ is the n^{th} column of the matrix $\boldsymbol{U}$.

Similarly, the covariance matrix between the other two pairs of observations with time delay τ_l, denoted as $\boldsymbol{Q}^{(l)}$ and $\boldsymbol{R}^{(l)}$ satisfy

$$\begin{aligned} \boldsymbol{Q}^{(l)} &= E\{\boldsymbol{X}^{(1)}(t)\boldsymbol{X}^{(3)}(t+\tau_l)^T\} = (\boldsymbol{A}^{(1)})\boldsymbol{V}^{(\tau_l)}(\boldsymbol{A}^{(3)})^T, \\ \boldsymbol{R}^{(l)} &= E\{\boldsymbol{X}^{(2)}(t)\boldsymbol{X}^{(3)}(t+\tau_l)^T\} = (\boldsymbol{A}^{(2)})\boldsymbol{W}^{(\tau_l)}(\boldsymbol{A}^{(3)})^T, \end{aligned} \tag{14}$$

where $\boldsymbol{V}^{(\tau_l)} = E\{\boldsymbol{S}^{(1)}(t)\boldsymbol{S}^{(3)}(t+\tau_l)^T\}$ and $\boldsymbol{W}^{(\tau_l)} = E\{\boldsymbol{S}^{(2)}(t)\boldsymbol{S}^{(3)}(t+\tau_l)^T\}$ for $l = 1, 2, \ldots, L$. Stack these two sequence of covariance matrices $\{\boldsymbol{Q}\}$ and $\{\boldsymbol{R}\}$ in tensors $\mathcal{Q} \in \mathbb{R}^{M\times M\times L}$ and $\mathcal{R} \in \mathbb{R}^{M\times M\times L}$ as follows: $(\mathcal{Q})_{i,j,l} = (\boldsymbol{Q}^{(l)})_{i,j}$, $(\mathcal{R})_{i,j,l} = (\boldsymbol{R}^{(l)})_{i,j}$, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, M$, $l = 1, 2, \ldots, L$. To simplify the notation, we further define the matrix $\boldsymbol{V} \in \mathbb{R}^{L\times N}$ and $\boldsymbol{W} \in \mathbb{R}^{L\times N}$ with the element $\boldsymbol{V}_{l,n} = (\boldsymbol{V}^{(\tau_l)})_{n,n}$ and $\boldsymbol{W}_{l,n} = (\boldsymbol{W}^{(\tau_l)})_{n,n}$, for $l = 1, 2, \ldots, L$, $n = 1, 2, \ldots, N$. Then these two tensors can be represented as (see Fig. 1):

$$\begin{aligned} \mathcal{Q} &= \sum_{n=1}^{N} \boldsymbol{a}_n^{(1)} \circ \boldsymbol{a}_n^{(3)} \circ \boldsymbol{v}_n, \\ \mathcal{R} &= \sum_{n=1}^{N} \boldsymbol{a}_n^{(2)} \circ \boldsymbol{a}_n^{(3)} \circ \boldsymbol{w}_n, \end{aligned} \tag{15}$$

in which $\boldsymbol{a}_n^{(k)}$ is the n^{th} column of the mixing matrices $\boldsymbol{A}^{(k)}$ for $k = 1, 2, 3$, $\boldsymbol{v}_n$ and $\boldsymbol{w}_n$ are the n^{th} column of the matrix $\boldsymbol{V}$ and $\boldsymbol{W}$ respectively.

It should be mentioned that the choices of $\tau_1, \tau_2, \ldots, \tau_L$ may affect the estimation precision of the mixing matrices. If τ_l is too large, the correlation between two related sources with the delay will be close to 0 and then the covariance matrix might be ill conditioned. It is desired to select $\tau_1, \tau_2, \ldots, \tau_L$ such that $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{W}$ are well conditioned. In addition, if the time delay τ is too large, the covariance matrix of the sources in two datasets (e.g., $\boldsymbol{U}^{(\tau)}$) will be close to a null matrix and thus the assumption (II) may not hold. Here, we heuristically choose the time delay as $\tau_l \in [0, 200]$ data samples.

Fig. 1 illustrates how to generate these tensors by incorporating the dependence information between each pair of datasets. It is worth noting that each pair of tensors share a common factor matrix, e.g., $\mathcal{P}$ and $\mathcal{Q}$ are coupled in the mode of $\boldsymbol{A}^{(1)}$.

B. JOINT TENSOR POLYADIC DECOMPOSITION

Considering the common latent structure, now the problem of estimating the mixing matrices $\boldsymbol{A}^{(k)}$ can be reformulated as a problem of joint CPD of a collection of tensors, e.g., $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{R}$ for the case of three datasets. There are two main approaches to jointly decompose a sequence of tensors, i.e., linear algebra [33] and optimization based methods [34]–[36]. Sørensen *et al.* [33] took into account the coupling between multiple tensors and developed a linear algebra based algorithm. This method can provide an explicit solution for exact tensor decomposition. However, in practice data are noisy and consequently the estimation may be not accurate. In addition, it is notable that the linear algebra based method requires the full column rank of the common factor matrices whereas the common factors in our problem are rank deficient [33]. In this paper, we generalize the idea of coupled matrix and tensor factorization (CMTF) and jointly decompose a sequence of tensors via gradient-based optimization method [34]–[36].

The uniqueness condition of the joint CPD is important in practice. Simply said, the solution of the joint CPD will be

generic unique if all the individual CPDs are unique. In this paper, we can get the unique solution of each mixing matrix generically, providing the number of sources satisfies the condition (9) or (10). It is worth mentioning that this uniqueness condition of the joint CPD might be further relaxed, but the topic itself deserves a stand-alone theoretical paper and it is out of scope of the current paper.

The aim is to find the factor matrices $\{\boldsymbol{A}^{(k)}\} \in \mathbb{R}^{M\times N}$ and the covariance of sources between different datasets $\boldsymbol{U}, \boldsymbol{V}$ and $\boldsymbol{W} \in \mathbb{R}^{L\times N}$ which can minimize the following objective function, a variant of Frobenius norm of the difference between the given tensors and their canonical polyadic approximation, written as

$$\begin{aligned} &f(\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \\ &= \frac{1}{2}\underbrace{\|\mathcal{P} - [\![\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{U}]\!]\|^2}_{f^{(1)}(\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\boldsymbol{U})} + \frac{1}{2}\underbrace{\|\mathcal{Q} - [\![\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(3)}, \boldsymbol{V}]\!]\|^2}_{f^{(2)}(\boldsymbol{A}^{(1)},\boldsymbol{A}^{(3)},\boldsymbol{V})} \\ &\quad + \frac{1}{2}\underbrace{\|\mathcal{R} - [\![\boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}, \boldsymbol{W}]\!]\|^2}_{f^{(3)}(\boldsymbol{A}^{(2)},\boldsymbol{A}^{(3)},\boldsymbol{W})}. \end{aligned} \tag{16}$$

where $[\![\cdot]\!]$ denotes the canonical polyadic approximation of a given tensor. This equation simultaneously takes the coupling information between different tensors into account. We propose to solve this problem via a gradient-based optimization method. Proposition 1 elaborates the partial derivative of the objective function f with respect to each column of the desired matrices, i.e. $\{\boldsymbol{a}_n^{(k)}\}, \boldsymbol{u}_n, \boldsymbol{v}_n$ and $\boldsymbol{w}_n$ for $n = 1, 2, \ldots, N$.

The equations in Proposition 1 is proved in the Appendix. Then the gradient of f can be assembled via stacking the partial derivatives with respect to each column of the factor matrices, as

$$\nabla f = \left[\frac{\partial f}{\partial \boldsymbol{a}_1^{(1)}}; \frac{\partial f}{\partial \boldsymbol{a}_2^{(1)}}; \ldots; \frac{\partial f}{\partial \boldsymbol{a}_N^{(1)}}; \ldots; \frac{\partial f}{\partial \boldsymbol{w}_1}; \ldots; \frac{\partial f}{\partial \boldsymbol{w}_N}\right]^T. \tag{17}$$

Once we get this gradient, we can calculate the factor matrices, including the mixing matrices and the covariance matrices, based on any first-order optimization method. In this paper, we employ the nonlinear conjugate gradient algorithm (NCG) implemented in [37] to solve the unconstrained optimization problem and estimate the mixing matrices of multiple datasets simultaneously. Compared with second-order optimization methods, such as Newton-based methods, NCG always requires less computation and memory [36].

VI. SOURCE EXTRACTION BASED ON THE ESTIMATED MIXING MATRICES

Unlike the (over)determined case, the estimation of the mixing matrix is not equivalent to recovering the underlying sources in UBSS. A complete UBSS approach always consists of both mixing matrix estimation and source extraction, even though our main focus in this paper is the estimation of mixing matrices. Extracting the sources when the mixing matrix is estimated is a classic inverse problem. Many techniques have already been proposed in the literature, including array processing techniques [38] and methods exploiting the sparsity of sources in a domain, e.g., the time-frequency (TF) domain [13]. In order to demonstrate the performance of the proposed mixing matrices estimation method, we adopt a recently-developed subspace representation method [15] to recover the latent sources based on the estimated mixing matrices. For simplicity, the proposed method for extracting sources is derived without considering the background noise. However, it was shown to be robust to the background noise [15].

For any underdetermined non-homogeneous linear equation, the complete solution can be represented as the sum of its particular solution and a general solution of the corresponding homogeneous equation. As to the case in this paper, $\boldsymbol{A}^{(k)}\boldsymbol{S}^{(k)} = \boldsymbol{X}^{(k)}$, the general solution of source $\boldsymbol{S}^{(k)}$ can be written as

$$\boldsymbol{S}^{(k)} = \boldsymbol{S}_p^{(k)} + \boldsymbol{S}_h^{(k)}, \tag{18}$$

where the $\boldsymbol{S}_p^{(k)}$ denotes its particular solution and $\boldsymbol{S}_h^{(k)}$ denotes a general solution of the corresponding homogeneous equation $\boldsymbol{A}^{(k)}\boldsymbol{S}^{(k)} = \boldsymbol{0}$. One particular solution of the above mentioned non-homogeneous equation is

$$\boldsymbol{S}_p^{(k)} = (\boldsymbol{A}^{(k)})^{\dagger}\boldsymbol{X}^{(k)}, \tag{19}$$

where $(\boldsymbol{A}^{(k)})^{\dagger}$ denotes the pseudo-inverse of the mixing matrix $\boldsymbol{A}^{(k)}$. In addition, the general solution of the homogeneous equation $\boldsymbol{A}^{(k)}\boldsymbol{S}^{(k)} = \boldsymbol{0}$ can be expressed as

$$\boldsymbol{S}_h^{(k)} = \boldsymbol{V}\boldsymbol{Z}^{(k)}, \tag{20}$$

where $\boldsymbol{V}$ is an $N * (N - M)$ matrix whose columns are bases of the nullspace of $\boldsymbol{A}^{(k)}$ and $\boldsymbol{Z}^{(k)}$ is an arbitrary matrix with the size of $(N - M) * T$ (T represents the total number of samples in each channel) [39]. The basis matrix $\boldsymbol{V}$ can be obtained from the mixing matrix $\boldsymbol{A}^{(k)}$ and then the problem which aims to estimate the N dimensional observations boils down to the problem of estimating $N - M$ dimensional latent variable $\boldsymbol{Z}^{(k)}$.

In order to be applicable to a wide class of signals, such as audio and biological signals EEG, EMG, the Generalized Gaussian Distribution (GGD) [40] is utilized to model the source distributions. Mathematically, it is expressed in the following equation

$$p_y(y; \sigma, \beta) = \frac{v(\beta)}{\sigma} exp\{-c(\beta)|\frac{y-\mu}{\sigma}|^{2/(1+\beta)}\}, \tag{21}$$

where

$$\begin{aligned} c(\beta) &= \left(\frac{\Gamma(3/2(1+\beta))}{\Gamma(1/2(1+\beta))}\right)^{1/(1+\beta)} \\ v(\beta) &= \frac{\Gamma(3/2(1+\beta))^{1/2}}{(1+\beta)\Gamma(1/2(1+\beta))^{3/2}}, \end{aligned} \tag{22}$$

in which $\Gamma(\cdot)$ is the Gamma function. σ is the standard derivation and μ is the mean of a continuous random variable y. In this paper, the mean of source is assumed to be 0.

Proposition 1: The partial derivative of the objective function f with respect to each column of the desired matrices , i.e., $\{\boldsymbol{a}_n^{(k)}\}$, $\boldsymbol{u}_n$, $\boldsymbol{v}_r$ and $\boldsymbol{w}_n$, are given by

$$\frac{\partial f}{\partial \boldsymbol{a}_n^{(1)}} = -\mathcal{P} \times_2 \boldsymbol{a}_n^{(2)} \times_3 \boldsymbol{u}_n - \mathcal{Q} \times_2 \boldsymbol{a}_n^{(3)} \times_3 \boldsymbol{v}_n + \sum_{c=1}^{N}[(\boldsymbol{a}_n^{(2)})^T \boldsymbol{a}_c^{(2)} (\boldsymbol{u}_n)^T \boldsymbol{u}_c + (\boldsymbol{a}_n^{(3)})^T \boldsymbol{a}_c^{(3)} (\boldsymbol{v}_n)^T \boldsymbol{v}_c]\boldsymbol{a}_c^{(1)}$$

$$\frac{\partial f}{\partial \boldsymbol{a}_n^{(2)}} = -\mathcal{P} \times_1 \boldsymbol{a}_n^{(1)} \times_3 \boldsymbol{u}_n - \mathcal{R} \times_2 \boldsymbol{a}_n^{(3)} \times_3 \boldsymbol{w}_n + \sum_{c=1}^{N}[(\boldsymbol{a}_n^{(1)})^T \boldsymbol{a}_c^{(1)} (\boldsymbol{u}_n)^T \boldsymbol{u}_c + (\boldsymbol{a}_n^{(3)})^T \boldsymbol{a}_c^{(3)} (\boldsymbol{w}_n)^T \boldsymbol{w}_c]\boldsymbol{a}_c^{(2)}$$

$$\frac{\partial f}{\partial \boldsymbol{a}_n^{(3)}} = -\mathcal{Q} \times_1 \boldsymbol{a}_n^{(1)} \times_3 \boldsymbol{v}_n - \mathcal{R} \times_1 \boldsymbol{a}_n^{(2)} \times_3 \boldsymbol{w}_n + \sum_{c=1}^{N}[(\boldsymbol{a}_n^{(1)})^T \boldsymbol{a}_c^{(1)} (\boldsymbol{v}_n)^T \boldsymbol{v}_c + (\boldsymbol{a}_n^{(2)})^T \boldsymbol{a}_c^{(2)} (\boldsymbol{w}_n)^T \boldsymbol{w}_c]\boldsymbol{a}_c^{(3)}$$

$$\frac{\partial f}{\partial \boldsymbol{u}_n} = -\mathcal{P} \times_1 \boldsymbol{a}_n^{(1)} \times_2 \boldsymbol{a}_n^{(2)} + \sum_{c=1}^{N}[(\boldsymbol{a}_n^{(1)})^T \boldsymbol{a}_c^{(1)} (\boldsymbol{a}_n^{(2)})^T \boldsymbol{a}_c^{(2)}]\boldsymbol{u}_c$$

$$\frac{\partial f}{\partial \boldsymbol{v}_n} = -\mathcal{Q} \times_1 \boldsymbol{a}_n^{(1)} \times_2 \boldsymbol{a}_n^{(3)} + \sum_{c=1}^{N}[(\boldsymbol{a}_n^{(1)})^T \boldsymbol{a}_c^{(1)} (\boldsymbol{a}_n^{(3)})^T \boldsymbol{a}_c^{(3)}]\boldsymbol{v}_c$$

$$\frac{\partial f}{\partial \boldsymbol{w}_n} = -\mathcal{R} \times_1 \boldsymbol{a}_n^{(2)} \times_2 \boldsymbol{a}_n^{(3)} + \sum_{c=1}^{N}[(\boldsymbol{a}_n^{(2)})^T \boldsymbol{a}_c^{(2)} (\boldsymbol{a}_n^{(2)})^T \boldsymbol{a}_c^{(3)}]\boldsymbol{w}_c.$$

We define the parameter set $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \boldsymbol{\sigma}\}$ for simplicity, where each component of $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_N]$ and $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_N]$ correspond to each channel of the sources. The parameters of the GGD $\boldsymbol{\theta}$ can be estimated to maximize the likelihood of the observed mixtures $\boldsymbol{X}^{(k)}$ based on Expectation-maximization (EM) algorithm. Then $\boldsymbol{Z}$ can be obtained by sampling from $p(\boldsymbol{Z}^{(k)}|\boldsymbol{X}^{(k)}, \theta)$ as

$$\hat{\boldsymbol{Z}}^{(k)} = \frac{1}{G}\sum_{g=1}^{G} \boldsymbol{Z}_g^{(k)}, \tag{23}$$

where $\{\boldsymbol{Z}_1^{(k)}, \ldots, \boldsymbol{Z}_G^{(k)}\}$ are the G samples drawn from $p(\boldsymbol{Z}^{(k)}|\boldsymbol{X}^{(k)}, \boldsymbol{\theta})$ using the Markov Chain Monte Carlo (MCMC) method. Then we recover the underlying sources based on

$$\hat{\boldsymbol{S}}^{(k)} = (\boldsymbol{A}^{(k)})^{\dagger}\boldsymbol{X}^{(k)} + \boldsymbol{V}\boldsymbol{Z}^{(k)}. \tag{24}$$

The major steps of the proposed UJBSS-m algorithm are summarized in Algorithm 1. The number of time delays is 20 in default. The step size of time delays, i.e. $\tau_{l+1} - \tau_l$, is suggested to be 2 samples (corresponding to 0.25ms) and 5 samples (corresponding to 5ms) for audio signals and physiological signals respectively.

VII. NUMERICAL STUDY FOR THE MULTIPLE DATASET CASE

To demonstrate the joint separation performance for multiple datasets, simulations are performed on both audio and biological signals when applying the proposed UJBSS-m and several commonly used BSS methods. Two performance indices are used to evaluate the separation performances. One is the estimation error of the mixing matrices, defined as:

$$Error = 10log_{10}\{mean(\frac{||\boldsymbol{A} - \hat{\boldsymbol{A}}||}{||\boldsymbol{A}||})\}, \tag{25}$$

where $\hat{\boldsymbol{A}}$ denotes the optimally ordered estimate of $\boldsymbol{A}$. The other measures the Pearson correlation coefficient (PCC) between the estimated sources and the original ones, which is defined as

$$PCC(\boldsymbol{s}_n^{(k)}, \hat{\boldsymbol{s}}_n^{(k)}) = \frac{cov(\boldsymbol{s}_n^{(k)}, \hat{\boldsymbol{s}}_n^{(k)})}{\sigma_{\boldsymbol{s}_n^{(k)}}\sigma_{\hat{\boldsymbol{s}}_n^{(k)}}}, \tag{26}$$

where $\hat{\boldsymbol{s}}_n^{(k)}$ means the estimate of the source $\boldsymbol{s}_n^{(k)}$ in the k_{th} dataset, $cov(\cdot, \cdot)$ means the covariance between two variables and σ means the standard deviation. In order to ensure the dependence between the sources of each pair of datasets, the sources are synthesized as follows,

$$\begin{aligned} \boldsymbol{S}^{(1)} &= [\boldsymbol{s}_1^{(1)}, \boldsymbol{s}_2^{(1)}, \ldots, \boldsymbol{s}_N^{(1)}]^T; \\ \boldsymbol{S}^{(2)} &= \boldsymbol{S}^{(1)}. * (unifrnd(0, 1, \boldsymbol{S}^{(1)}) \\ \boldsymbol{S}^{(3)} &= \boldsymbol{S}^{(1)}. * (unifrnd(0, 1, \boldsymbol{S}^{(1)}), \end{aligned} \tag{27}$$

where $unifrnd(0, 1, \boldsymbol{S}^{(1)})$ generates a matrix with the same size of $\boldsymbol{S}^{(1)}$ and each element of the matrix is randomly drawn from the continuous uniform distribution on the interval (0,1). The average correlation between the source $\boldsymbol{s}_n^{(1)}$ and the corresponding source $\boldsymbol{s}_n^{(k)}(k = 2, 3)$ is about 0.85; the average correlation between the source $\boldsymbol{s}_n^{(2)}$ and the corresponding source $\boldsymbol{s}_n^{(3)}$ is about 0.7, both of which can be regarded as highly correlated.

A. SIMULATION 1: AUDIO SIGNALS

The sources used in this simulation include 8 audio signals, such as two pieces of sound from the cable news network (CNN) news and a piece of sound of an anonymous singer, all of which are publicly available.[1] The sampling rate is 8000Hz. The mixing matrices are generated randomly with

[1] http://research.ics.aalto.fi/ica/cocktail/sounds.html

Algorithm 1 The UJBSS-m Algorithm Based on Joint Tensor Decomposition

Input: M-dimensional observations $\{\boldsymbol{X}^{(k)}\}$ and the number of sources N in each datasets, for $k = 1, 2, \ldots, K$.
Output: the estimated mixing matrices $\{\boldsymbol{A}^{(k)}\}$ and the recovered N-dimensional sources $\{\boldsymbol{S}^{(k)}\}$, for $k = 1, 2, \ldots, K$.

STEP 1: For each pair of datasets, e.g., $\boldsymbol{X}^{(k_1)}$ and $\boldsymbol{X}^{(k_2)}(k_1 \neq k_2)$, we calculate the cross covariance matrices as (11) and stack them to construct a third-order tensor as in Section V-A. Considering the combination of datasets, we get $\binom{K}{2}$ tensors where each pair of tensors share a common factor matrix, as shown in Fig. 1;
STEP 2: Calculate the joint polyadic decomposition of the tensors constructed in step 1 via optimization based method and estimate the mixing matrices $\{\boldsymbol{A}^{(k)}\}$ as in Section V-B;
STEP 3: Estimate the parameters of the Generalized Gaussian distribution based on the EM algorithm.

Initialize: initialize the parameter $\boldsymbol{\theta}$ to some random values.

E-step: calculate the expected value of the log likelihood function with respect to the conditional distribution of $\boldsymbol{Z}^{(k)}$ given the observation $\boldsymbol{X}^{(k)}$ under the current estimate of $\boldsymbol{\theta}$. It can be expressed as $E_{p(\boldsymbol{Z}^{(k)}|\boldsymbol{X}^{(k)},\boldsymbol{\theta}^*)}(log(p(\boldsymbol{Z}^{(k)}|\boldsymbol{X}^{(k)}, \boldsymbol{\theta})))$, where $\boldsymbol{\theta}^*$ means the parameter value got in the initialization or the previous M-step.

M-step: update the parameter set $\boldsymbol{\theta}$ to maximize the above expected value. The updated value is

$$\begin{aligned}\boldsymbol{\theta} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, E_{p(\boldsymbol{Z}^{(k)}|\boldsymbol{X}^{(k)},\boldsymbol{\theta}^*)}(log(p(\boldsymbol{Z}^{(k)}|\boldsymbol{X}^{(k)}, \boldsymbol{\theta})))\\ &\approx \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \frac{1}{G}\sum_{g=1}^{G} log(p(\boldsymbol{Z}_g^{(k)}|\boldsymbol{X}^{(k)}, \boldsymbol{\theta})),\end{aligned}$$

where $\{\boldsymbol{Z}_1^{(k)}, \ldots, \boldsymbol{Z}_G^{(k)}\}$ are G samples drawn from $p(\boldsymbol{Z}^{(k)}|\boldsymbol{X}^{(k)}, \boldsymbol{\theta})$ based on the MCMC method.

Iterate: iterate the E-step and M-step until convergence.
STEP 4: Recover the sources $\boldsymbol{S}^{(k)}$ based on the minimum mean-square error criterion as in Equation (24).

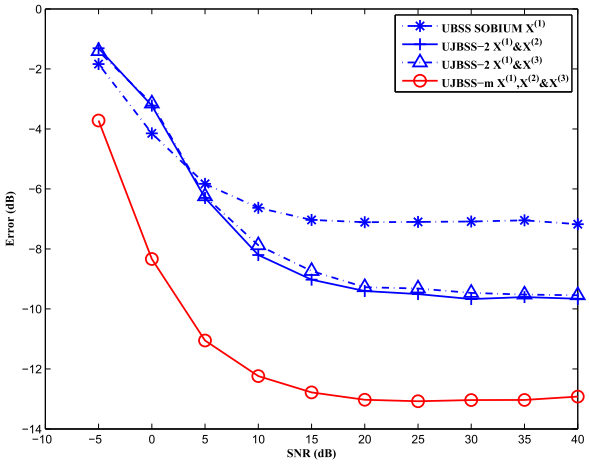

FIGURE 2. Simulation 1: Performance comparisons on audio signals when using the proposed UJBSS-m method and other UBSS methods, including the single-set UBSS method SOBIUM [12] and the UJBSS method for two dataset, i.e., UJBSS-2 [19]. Here the number of sources $N = 5$ and the number of observations $M = 4$. The number of time delays $L = 20$ and the step size of time delays (i.e., $\tau_l - \tau_{l-1}$) is 2 data samples, corresponding to 0.25ms. Similar results are observed for $A^{(2)}$ and $A^{(3)}$.

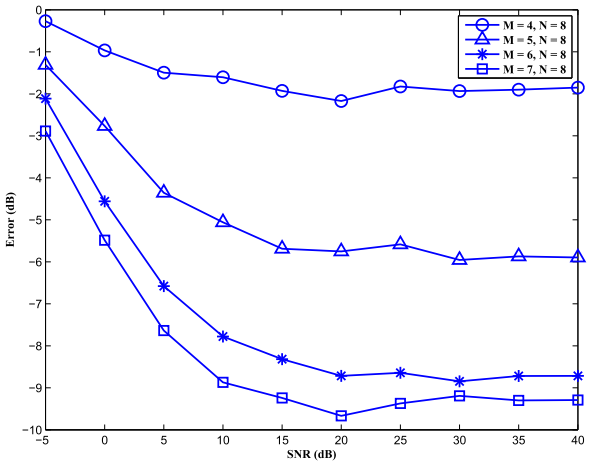

FIGURE 3. Simulation 1: Estimation error of $A^{(1)}$ when employing the proposed UJBSS method. Here the number of sources $N = 8$ and the number of observations M varies from 4 to 7. The number of time delays $L = 20$ and the step size of time delays (i.e., $\tau_l - \tau_{l-1}$) is 2 data samples, corresponding to 0.25ms.

elements following the uniform distribution $U[-1, 1]$. For simplicity, each column of the mixing matrices is normalized into a unit vector. Three datasets are generated following (27). In our first setting, 5 sources are mixed into 4 observations in each dataset and the corresponding sources in the different datasets are highly correlated. With different signal-to-noise ratios (SNRs), we compare the proposed UJBSS-m method with a commonly-used single-set UBSS method, SOBIUM [12], when it is applied to each dataset separately. We also test the performance of our recent work on UJBSS for two datasets, UJBSS-2 [19], when two datasets are available, e.g., $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$. We repeat the simulation 1000 times and the performance is shown in Fig. 2. Results are given according to the SNR level in the range of -5dB - 40dB. Benefiting from dependence information between different datasets, the proposed UJBSS can provide more accurate estimation of the mixing matrices, while SOBIUM neglects the possible inter-dataset information. Compared with UJBSS-2, the proposed UJBSS-m takes into account more dependence information, among three datasets rather than between two datasets, and yields better performance. We also note that the *Error* measure from the single set UBSS and UJBSS methods decreases with the increase of the SNR. The proposed UJBSS-m consistently provides the best results over the whole SNR range, suggesting the performance stability of the proposed algorithm.

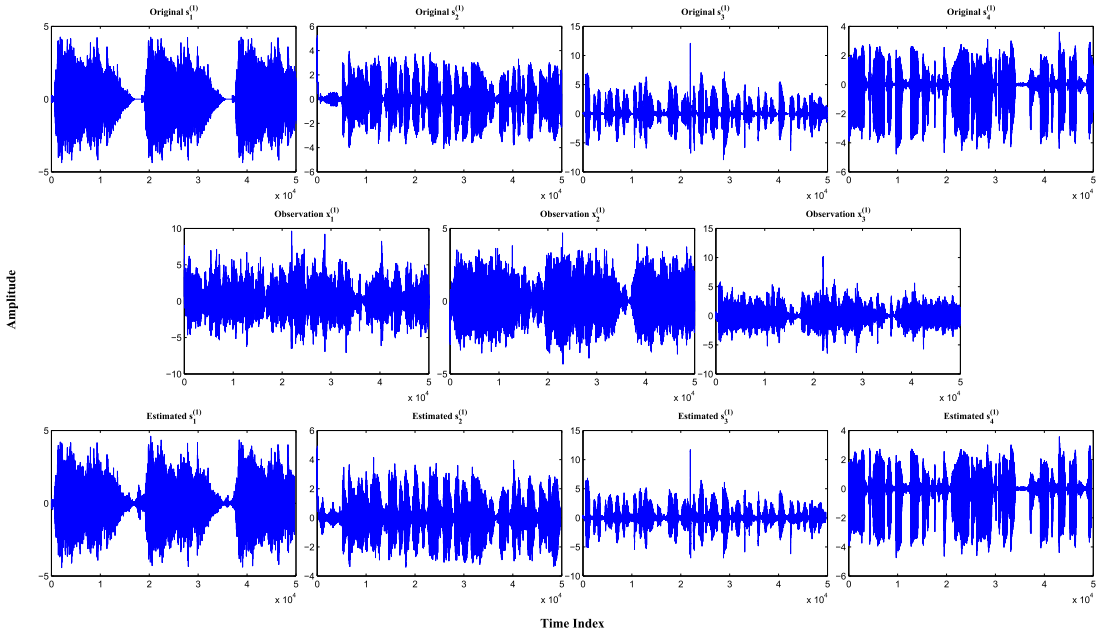

FIGURE 4. Simulation 1: An illustrative example from the proposed UJBSS-m method. First row: The original 4 sources; Second row: 3 channels of the mixed observations; Third row: the recovered 4 sources from the first dataset.

We also examine the performance of the proposed method with the decrement of the under-determinacy level, i.e., the number of the observations increases from 4 to 7 while the number of sources is set to be 8. As noted in Fig. 3, the estimation performance is getting better when more observations are available. Besides the degraded estimation precision, a higher under-determinacy level also requires higher computational complexity. The performance is getting better when the SNR is increased from -5dB to 20dB. The change of estimation error is not obvious when the SNR is greater than 20dB even there are some fluctuations. It is shown that the estimation performance relies upon several factors such as the noise level SNR, under-determinacy level (i.e., the number of sources for a given number of sensors) and the correlation between each pair of datasets.

We recover the latent sources from each dataset based on the estimated mixing matrices. In an illustrative example, we linearly mix 4 audio sources into 3 observations in each dataset. Fig. 4 shows the separation results of the first dataset in the time domain. The top four subfigures of Fig. 4 represent the original sources, the middle three subfigures are the mixed observations and the bottom four subfigures are the recovered sources via the proposed UJBSS-m method. In addition, we compare the proposed method with other three single-set UBSS methods, including SOBIUM, UBSS based on subspace representation (UBSS-SR for short) [15] and UBSS based on sparse coding (UBSS-SC for short) [14], as well as the JBSS method MCCA [6] and UJBSS for two datasets UJBSS-2 [19], in term of the PCC between the original sources and the recovered ones. Both UBSS-SR and UBSS-SC are based on single source detection, which assumes that the TF points are occupied by a single source or the corresponding single source possesses dominant energy. However, the performance of estimating the mixing matrix deteriorates when this assumption is not satisfied. Furthermore, given that the time-frequency analysis method [13], [19] is memory-intensive and time-consuming, we estimate the mixing matrices via UJBSS-2 and SOBIUM respectively, and then extract the sources using the same method as in UBSS-SR [15]. The performance results of these six methods are reported in Table 1.

Despite adopting the same technology in extracting sources, the performance of the proposed method is significantly better than that of the single-set SOBIUM and UBSS-SR method. This observation confirms the importance of estimating mixing matrices accurately. In addition, the proposed method also outperforms a recently proposed UBSS method UBSS-SC. The main reason is that such UBSS methods always require the sparsity of the sources to some extent, while the assumption may not be satisfied in reality. MCCA, which has been successfully used in many fields [41], assumes that the number of sources is equal to the number of observations in each dataset and it could not be used to separate sources in the underdetermined case directly. We add one observation in each dataset so that MCCA can be applied. Therefore it is not really a fair setting and comparison to the proposed method. However, we note that the performance of MCCA is not as good as that of the proposed method, even with an additional observation signal. The following reasons could contribute to the worse performance of MCCA: it is mainly due to the fact that the correlation coefficients between sources in two datasets are quite close [6], [42]; the performance of MCCA may suffer from error accumulation of the deflation-based separation methods [7].

TABLE 1. PCC performance results in Simulation 1.

	Methods	s_1	s_2	s_3	s_4
Dataset 1	UJBSS-m	0.993	1.000	0.881	0.992
	UJBSS-2 [19]	0.989	1.000	0.859	0.990
	SOBIUM [12]	0.980	1.000	0.512	0.967
	UBSS-SC [14]	0.951	0.965	-0.066	0.934
	UBSS-SR [15]	0.901	0.999	0.226	0.967
	MCCA* [6]	0.571	0.909	0.675	0.732
Dataset 2	UJBSS-m	0.910	0.998	0.740	0.944
	UJBSS-2 [19]	0.897	0.998	0.714	0.892
	SOBIUM [12]	0.877	0.998	0.686	0.962
	UBSS-SC [14]	0.795	0.940	-0.401	0.607
	UBSS-SR [15]	0.885	0.981	0.730	0.954
	MCCA* [6]	-0.574	0.911	-0.678	-0.733
Dataset 3	UJBSS-m	0.899	1.000	0.783	0.870
	UJBSS-2 [19]	0.720	1.000	0.515	0.869
	SOBIUM [12]	0.756	-0.951	0.078	0.880
	UBSS-SC [14]	0.381	-0.434	-0.614	-0.630
	UBSS-SR [15]	-0.557	0.987	0.620	-0.561
	MCCA* [6]	0.577	-0.911	-0.679	0.732

(1) *We add one additional observation in each dataset when we evaluate the MCCA.

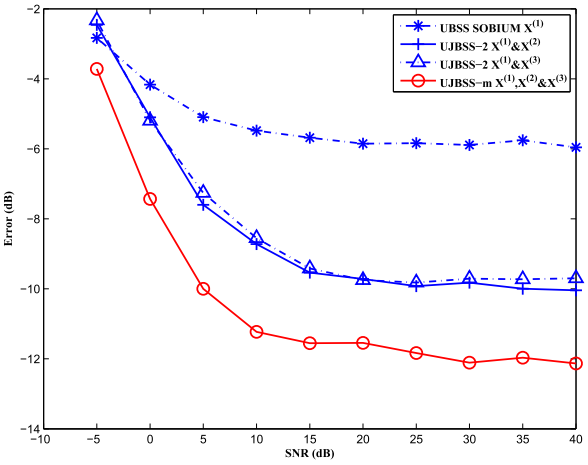

FIGURE 5. Simulation 2: Performance comparisons on physiological signals between the proposed UJBSS-m method and two other methods (i.e., the single-set UBSS method SOBIUM [12] and UJBSS method for two dataset, UJBSS-2 [19]). Here the number of sources $N = 4$ and the number of observations $M = 3$. The number of time delays $L = 20$ and the step size of time delays (i.e., $\tau_l - \tau_{l-1}$) is 2 data samples. Similar results are observed for $A^{(2)}$ and $A^{(3)}$.

B. SIMULATION 2: PHYSIOLOGICAL SIGNALS

In this experiment, we employ four physiological signals as sources, including ECG, EEG, electrooculography (EOG) and electromyography (EMG) from a publicly available database [43]. The sampling rate is 1000 Hz. The sources corresponding to the other two datasets are generated following (27). We get similar results as that in the simulation 1. As can be seen from Fig. 5, the estimation performance is getting better with the increase of the SNR of observations. In the whole SNR range, the proposed UJBSS-m method estimates the mixing matrices with higher accuracy than the single-set UBSS method SOBIUM and UJBSS method for two datasets UJBSS-2.

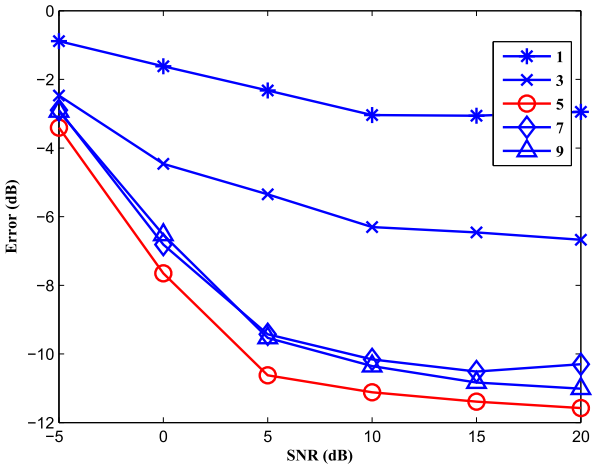

FIGURE 6. Simulation 2: Performance of the proposed UJBSS-m method when the step size of time delays (i.e., $\tau_l - \tau_{l-1}$) varies from 1 to 9. Here the number of sources $N = 4$ and the number of observations $M = 3$. The number of time delays $L = 20$. Similar results are observed for $A^{(2)}$ and $A^{(3)}$.

We also investigate the effect of the time delays, as shown in Fig. 6. At high SNR level, e.g. SNR = 20dB, the average Error of the proposed UJBSS-m is −11.57dB when the step size of the time delays is 5 data samples,corresponding to 5ms. However, the average Error corresponding to 3 data samples is −6.67dB, significantly larger than that of 5 data samples. The main reason is that the change of the covariance matrices is not obvious for the small step size and the covariance matrices related to these delays could not provide enough information to estimate the common factors, i.e. the mixing matrices. If the time delay is too large, such as more than 500 data samples (corresponding to 500ms), the covariance between two datasets will be close to $\mathbf{0}$. Here, we select 5 data samples as the step size of the time delays. In practice, we should select the time delays empirically based on the characters of the sources, e.g., we suggest the time delays smaller than 100ms for physiological signals. In addition, we evaluate the role of the number of time delays and find that it has less impact on the performance. In this paper, we set the number of time delays to 20.

We further show the performances in term of the PCC results between the original sources and the estimated ones. As shown in Table 2, the proposed method yields promising results when it is used to separate the latent and underdetermined mixtures. Compared to the classical JBSS method MCCA, the proposed UJBSS approach needs fewer number of observations in each dataset, while it yields a better performance.

VIII. A CASE STUDY: SOLVE A SINGLE SET UBSS PROBLEM BASED ON UJBSS-m

In this section, we show that the proposed UJBSS-m method can be employed to solve a single set UBSS problem, the noise enhanced signal processing problem, with a superior performance. As in Simulation 1, we employ 5 real audio

TABLE 2. PCC performance results in Simulation 2.

	Methods	ECG	EOG	EEG	EMG
Dataset 1	UJBSS-m	0.985	0.994	0.910	0.831
	UJBSS-2 [19]	0.975	0.986	0.872	0.781
	SOBIUM [12]	-0.815	-0.985	0.800	-0.382
	UBSS-SC [14]	0.090	0.949	-0.707	-0.679
	UBSS-SR [15]	0.504	-0.906	0.115	0.503
	MCCA* [6]	0.613	-0.754	0.695	0.674
Dataset 2	UJBSS-m	0.956	0.821	1.000	0.832
	UJBSS-2 [19]	0.883	0.705	0.942	0.308
	SOBIUM [12]	0.851	-0.467	0.999	0.726
	UBSS-SC [14]	-0.420	0.544	0.776	0.726
	UBSS-SR [15]	-0.894	0.288	-0.967	-0.203
	MCCA* [6]	0.634	-0.754	0.678	-0.676
Dataset 3	UJBSS-m	0.779	0.738	0.998	0.997
	UJBSS-2 [19]	0.777	0.738	0.998	0.997
	SOBIUM [12]	0.443	0.697	-0.993	-0.983
	UBSS-SC [14]	0.394	0.638	0.722	0.848
	UBSS-SR [15]	0.582	0.609	0.426	0.871
	MCCA* [6]	-0.624	-0.756	0.689	0.676

(1) *We add one additional observation in each dataset when we evaluate the MCCA.

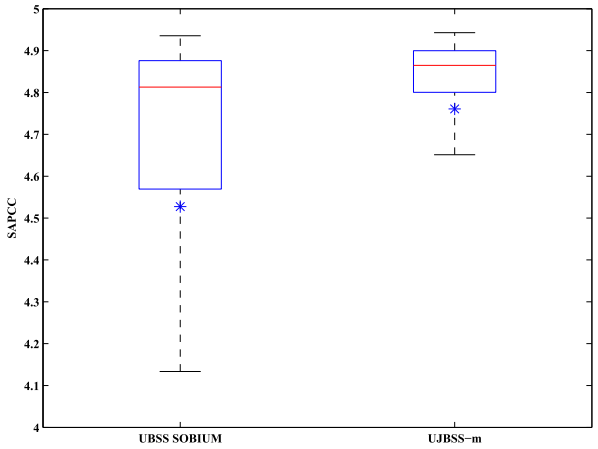

FIGURE 7. The sum of the absolute correlation coefficients between the recovered sources and the original ones. The blue asterisks represent the averages and the red lines stand for the medians. The edges of the box are the lower and upper quartiles.

signals as the sources. These 5 audio signals are mixed into 4 observations with a mixing matrix $\boldsymbol{A}^{(1)}$ whose elements follow the uniform distribution $U[-1, 1]$. We generate three datasets as

$$\begin{aligned} \boldsymbol{X}^{(1)} &= \boldsymbol{A}^{(1)}\boldsymbol{S}^{(1)} \\ \boldsymbol{X}^{(2)} &= awgn(\boldsymbol{X}^{(1)}, 20\ \text{dB}) \\ \boldsymbol{X}^{(3)} &= awgn(\boldsymbol{X}^{(1)}, 20\ \text{dB}), \end{aligned} \tag{28}$$

where $awgn(\boldsymbol{X}^{(1)}, 20\ \text{dB})$ represents adding white Gaussian noise to the signals $\boldsymbol{X}^{(1)}$ (i.e., the real observations) with SNR of 20dB. Noise, traditionally regarded as the unwanted signal, can play a very important constructive role in estimation problems, which is known as noise enhanced signal processing. $\boldsymbol{X}^{(2)}$ and $\boldsymbol{X}^{(3)}$ are random noise added signals based on $\boldsymbol{X}^{(1)}$.

The problem of interest here is to estimate the mixing matrix $\boldsymbol{A}^{(1)}$. Traditionally, we can estimate $\boldsymbol{A}^{(1)}$ from the dataset $\boldsymbol{X}^{(1)}$ based on the single set UBSS method SOBIUM. Here we also can apply the proposed algorithm as mentioned in Section V. Then we recover the sources via the method mentioned in Section VI based on the estimated mixing matrix $\boldsymbol{A}^{(1)}$. We repeat the experiment for 1000 times and calculate the sum of the absolute PCC (SAPCC) between the recovered sources and the original ones, which is calculated as

$$SAPCC = \sum_{n=1}^{5} abs(PCC(\boldsymbol{s}_n^{(1)}, \hat{\boldsymbol{s}}_n^{(1)})), \tag{29}$$

where $abs(\cdot)$ represents the absolute value function. Fig. 7 shows the distribution of the performance for 1000 repeats of the experiments. The average SAPCC for UBSS is 4.53 while that for the proposed UJBSS-m is 4.76, even with the same source extraction technology. The one-way analysis of variance (ANOVA) is performed on the results provided by these 1000 repeats. The obtained p value is 1.5677e-11, which means that the results of the proposed UJBSS-m method and single set UBSS method are significantly different. The proposed UJBSS-m algorithm demonstrates more robust and better performance. This example illustrates that the estimation accuracy could be improved by adding suitable noises to the input signals.

IX. CONCLUSIONS AND DISCUSSION

In this paper, we extend the idea of (over)determined JBSS to the underdetermined case and further generalize the UJBSS for two datasets (i.e., UJBSS-2) to the case of multiple datasets. The basic idea is similar to that in UJBSS-2, which estimate the mixing matrices jointly first and then restore the source signals. In this paper, we exploit the cross correlation of the observations between each pair of datasets and present a novel underdetermined joint blind source separation method, namely UJBSS-m, to jointly estimate the mixing matrices from multiple datasets when the number of observations is smaller than that of the sources. The mixing matrices are accurately estimated through joint canonical polyadic decomposition of a sequence of specialized tensors in which a set of covariance matrices are stacked. Further the sources are recovered based on the estimated mixing matrices. Numerical results on multiple datasets demonstrate the superior performances of the proposed method when compared to the commonly used JBSS and single-set UBSS methods. As an example application for noise enhanced signal processing, we also show that the proposed UJBSS-m method can be utilized to solve the single-set UBSS problem when suitable noise is added to the observations. In addition, the proposed UJBSS-m method does not rely upon sparsity of signals and therefore it can be applied to a wide class of signals.

APPENDIX

The Appendix is the proof of **Proposition** 1.

Proof: The three components of the objective function in (16), i.e., $f^{(1)}(\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{U})$, $f^{(2)}(\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(3)}, \boldsymbol{V})$ and $f^{(3)}(\boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}, \boldsymbol{W})$, share similar structure, which is the difference between one tensor and the corresponding estimated results. Therefore, we take $f^{(1)}(\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{U})$ and its partial derivative with respect to $\boldsymbol{a}_n^{(1)}$ for further analysis. It can be rewritten as

$$\begin{aligned} & f^{(1)}(\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{U}) \\ &= \|\mathcal{P} - [\![\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{U}]\!]\|^2 \\ &= \underbrace{\|\mathcal{P}\|^2}_{f_1^{(1)}} - 2\underbrace{< \mathcal{P}, [\![\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{U}]\!] >}_{f_2^{(1)}} + \underbrace{\|[\![\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{U}]\!]\|^2}_{f_3^{(1)}}. \end{aligned} \tag{A.1}$$

The first summand $f_1^{(1)}$ does not involve any variable and therefore

$$\frac{\partial f_1^{(1)}}{\partial \boldsymbol{a}_n^{(1)}} = \boldsymbol{0}, \tag{A.2}$$

where $\boldsymbol{0}$ is the zero vector with the same length as $\boldsymbol{a}_n^{(1)}$. The second summand $f_2^{(1)}$ is the inner product of the tensor $\mathcal{P}$ with its its polyadic decomposition, and it can be computed as

$$\begin{aligned} f_2^{(1)} &= < \mathcal{P}, [\![\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{U}]\!] > \\ &= < \mathcal{P}, \sum_{n=1}^{N} \boldsymbol{a}_n^{(1)} \circ \boldsymbol{a}_n^{(2)} \circ \boldsymbol{u}_n > \\ &= \sum_{n=1}^{N} \sum_{i_1=1}^{M} \sum_{i_2=1}^{M} \sum_{i_3=1}^{L} p_{i_1,i_2,i_3} a_{i_1,n}^{(1)} a_{i_2,n}^{(2)} u_{i_3,n} \\ &= \sum_{n=1}^{N} (\mathcal{P} \times_1 \boldsymbol{a}_n^{(1)} \times_2 \boldsymbol{a}_n^{(2)} \times_3 \boldsymbol{u}_n) \\ &= \sum_{n=1}^{N} (\mathcal{P} \times_2 \boldsymbol{a}_n^{(2)} \times_3 \boldsymbol{u}_n)^T \boldsymbol{a}_n^{(1)}. \end{aligned} \tag{A.3}$$

The partial derivative of $f_2^{(1)}$ with respect to each column of $\boldsymbol{A}^{(1)}$ is

$$\frac{\partial f_2^{(1)}}{\partial \boldsymbol{a}_n^{(1)}} = \mathcal{P} \times_2 \boldsymbol{a}_n^{(2)} \times_3 \boldsymbol{u}_n. \tag{A.4}$$

The third summand is the square of the Frobenius norm of $\mathcal{P}$'s polyadic decomposition, and it can be computed as

$$\begin{aligned} f_3^{(1)} &= \|[\![\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{U}]\!]\|^2 \\ &= < \sum_{n=1}^{N} \boldsymbol{a}_n^{(1)} \circ \boldsymbol{a}_n^{(2)} \circ \boldsymbol{u}_n, \sum_{n=1}^{N} \boldsymbol{a}_n^{(1)} \circ \boldsymbol{a}_n^{(2)} \circ \boldsymbol{u}_n > \\ &= \sum_{b=1}^{N} \sum_{c=1}^{N} \underbrace{((\boldsymbol{a}_b^{(1)})^T (\boldsymbol{a}_c^{(1)})(\boldsymbol{a}_b^{(2)})^T (\boldsymbol{a}_c^{(2)})(\boldsymbol{u}_b)^T (\boldsymbol{u}_c))}_{F(b,c)} \\ &= F(n,n) + \sum_{\substack{b=1 \\ b\neq n}}^{N} \sum_{\substack{c=1 \\ c\neq n}}^{N} F(b,c) + 2 \sum_{\substack{c=1 \\ c\neq n}}^{N} F(n,c), \end{aligned} \tag{A.5}$$

where b and c denote the indices of the factor matrices. The first summand of $f_3^{(1)}$ is

$$F(n,n) = (\boldsymbol{a}_n^{(1)})^T (\boldsymbol{a}_n^{(1)})(\boldsymbol{a}_n^{(2)})^T (\boldsymbol{a}_n^{(2)})(\boldsymbol{u}_n)^T (\boldsymbol{u}_n), \tag{A.6}$$

and its partial derivative with respect to the n^{th} column of the factor matrix $\boldsymbol{A}^{(1)}$ is

$$\frac{\partial F(n,n)}{\partial \boldsymbol{a}_n^{(1)}} = 2((\boldsymbol{a}_n^{(2)})^T \boldsymbol{a}_n^{(2)} \boldsymbol{u}_n^T \boldsymbol{u}_n) \boldsymbol{a}_n^{(1)}. \tag{A.7}$$

The second summand of $f_3^{(1)}$ does not involve the variable $\boldsymbol{a}_n^{(1)}$ and therefore the corresponding partial derivative with respect to $\boldsymbol{a}_n^{(1)}$ is the zero vector with the same length as $\boldsymbol{a}_n^{(1)}$. The third summand of $f_3^{(1)}$ is

$$2 \sum_{\substack{c=1 \\ c\neq n}}^{N} F(n,c) = 2 \sum_{\substack{c=1 \\ c\neq n}}^{N} (\boldsymbol{a}_n^{(1)})^T (\boldsymbol{a}_c^{(1)})(\boldsymbol{a}_n^{(2)})^T (\boldsymbol{a}_c^{(2)})(\boldsymbol{u}_n)^T (\boldsymbol{u}_c), \tag{A.8}$$

and its partial derivative with respect to the $\boldsymbol{a}_n^{(1)}$ can be computed as $2 \sum_{\substack{c=1 \\ c\neq n}}^{N} [(\boldsymbol{a}_n^{(2)})^T \boldsymbol{a}_c^{(2)} (\boldsymbol{u}_n)^T \boldsymbol{u}_c] \boldsymbol{a}_c^{(1)}$. Therefore,

$$\begin{aligned} \frac{\partial f_3^{(1)}}{\partial \boldsymbol{a}_n^{(1)}} &= 2((\boldsymbol{a}_n^{(2)})^T \boldsymbol{a}_n^{(2)} \boldsymbol{u}_n^T \boldsymbol{u}_n) \boldsymbol{a}_n^{(1)} \\ &\quad + 2 \sum_{\substack{c=1 \\ c\neq n}}^{N} [(\boldsymbol{a}_n^{(2)})^T \boldsymbol{a}_c^{(2)} (\boldsymbol{u}_n)^T \boldsymbol{u}_c] \boldsymbol{a}_c^{(1)} \\ &= 2 \sum_{c=1}^{N} [(\boldsymbol{a}_n^{(2)})^T \boldsymbol{a}_c^{(2)} (\boldsymbol{u}_n)^T \boldsymbol{u}_c] \boldsymbol{a}_c^{(1)}. \end{aligned} \tag{A.9}$$

Combining all the above results, i.e., equation (A.2), (A.4) and (A.9), the partial derivative of $f^{(1)}(\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{U})$ with respect to the $\boldsymbol{a}_n^{(1)}$ can be computed as

$$\begin{aligned} & \frac{\partial f^{(1)}(\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{U})}{\partial \boldsymbol{a}_n^{(1)}} \\ &= \frac{\partial f_1^{(1)}}{\partial \boldsymbol{a}_n^{(1)}} - 2\frac{\partial f_2^{(1)}}{\partial \boldsymbol{a}_n^{(1)}} + \frac{\partial f_3^{(1)}}{\partial \boldsymbol{a}_n^{(1)}} \\ &= -2\mathcal{P} \times_2 \boldsymbol{a}_n^{(2)} \times_3 \boldsymbol{u}_n + 2 \sum_{c=1}^{N} [(\boldsymbol{a}_n^{(2)})^T \boldsymbol{a}_c^{(2)} (\boldsymbol{u}_n)^T \boldsymbol{u}_c] \boldsymbol{a}_c^{(1)}. \end{aligned} \tag{A.10}$$

Similarly, we can calculate the partial derivative of $f^{(2)}(\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(3)}, \boldsymbol{V})$ with respect to the $\boldsymbol{a}_n^{(1)}$ as

$$\begin{aligned} & \frac{\partial f^{(2)}(\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(3)}, \boldsymbol{V})}{\partial \boldsymbol{a}_n^{(1)}} \\ &= -2\mathcal{Q} \times_2 \boldsymbol{a}_n^{(3)} \times_3 \boldsymbol{v}_n + 2 \sum_{c=1}^{N} [(\boldsymbol{a}_n^{(3)})^T \boldsymbol{a}_c^{(3)} (\boldsymbol{v}_n)^T \boldsymbol{v}_c] \boldsymbol{a}_c^{(1)}. \end{aligned} \tag{A.11}$$

$f^{(3)}(\boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}, \boldsymbol{W})$ does not involve the variable $\boldsymbol{a}_n^{(1)}$ and therefore

$$\frac{\partial f^{(3)}(\boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}, \boldsymbol{W})}{\partial \boldsymbol{a}_n^{(1)}} = \boldsymbol{0}. \tag{A.12}$$

Consequently, the partial derivative of the objective function with respect to $\boldsymbol{a}_n^{(1)}$ is

$$\begin{aligned}
&\frac{\partial f(\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})}{\partial \boldsymbol{a}_n^{(1)}} \\
&= \frac{1}{2}\frac{\partial f^{(1)}}{\partial \boldsymbol{a}_n^{(1)}} + \frac{1}{2}\frac{\partial f^{(2)}}{\partial \boldsymbol{a}_n^{(1)}} + \frac{1}{2}\frac{\partial f^{(3)}}{\partial \boldsymbol{a}_n^{(1)}} \\
&= -\mathcal{P} \times_2 \boldsymbol{a}_n^{(2)} \times_3 \boldsymbol{u}_n - \mathcal{Q} \times_2 \boldsymbol{a}_n^{(2)} \times_3 \boldsymbol{v}_n \\
&\quad + \sum_{c=1}^{N} [(\boldsymbol{a}_n^{(2)})^T \boldsymbol{a}_c^{(2)} (\boldsymbol{u}_n)^T \boldsymbol{u}_c + (\boldsymbol{a}_n^{(3)})^T \boldsymbol{a}_c^{(3)} (\boldsymbol{v}_n)^T \boldsymbol{v}_c] \boldsymbol{a}_c^{(1)}.
\end{aligned} \tag{A.13}$$

This completes the proof of the first equation in Proposition 1. The proof of other equations is similar to that of (A.13) and thus omitted here.

REFERENCES

[1] T. Adali, M. Anderson, and G.-S. Fu, "Diversity in independent component and vector analyses: Identifiability, algorithms, and applications in medical imaging," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 18–33, Mar. 2014.

[2] G. Zhou, Q. Zhao, Y. Zhang, T. Adali, S. Xie, and A. Cichocki, "Linked component analysis from matrices to high-order tensors: Applications to biomedical data," *Proc. IEEE*, vol. 104, no. 2, pp. 310–331, Feb. 2016.

[3] S. Ge *et al.*, "A brain-computer interface based on a few-channel EEG-fNIRS bimodal system," *IEEE Access*, vol. 5, pp. 208–218, Mar. 2017.

[4] X. Chen, Z. J. Wang, and M. McKeown, "Joint blind source separation for neurophysiological data analysis: Multiset and multimodal methods," *IEEE Signal Process. Mag.*, vol. 33, no. 3, pp. 86–107, May 2016.

[5] J. R. Kettenring, "Canonical analysis of several sets of variables," *Biometrika*, vol. 58, no. 3, pp. 433–451, 1971.

[6] Y.-O. Li, T. Adalı, W. Wang, and V. D. Calhoun, "Joint blind source separation by multiset canonical correlation analysis," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3918–3929, Oct. 2009.

[7] X.-L. Li, T. Adalı, and M. Anderson, "Joint blind source separation by generalized joint diagonalization of cumulant matrices," *Signal Process.*, vol. 91, no. 10, pp. 2314–2322, 2011.

[8] T. Adali, Y. Levin-Schwartz, and V. D. Calhoun, "Multimodal data fusion using source separation: Two effective models based on ICA and IVA and their properties," *Proc. IEEE*, vol. 103, no. 9, pp. 1478–1493, Sep. 2015.

[9] A. R. Groves, C. F. Beckmann, S. M. Smith, and M. W. Woolrich, "Linked independent component analysis for multimodal data fusion," *Neuroimage*, vol. 54, no. 3, pp. 2198–2217, 2011.

[10] M. Kleinsteuber and H. Shen, "Blind source separation with compressively sensed linear mixtures," *IEEE Signal Process. Lett.*, vol. 19, no. 2, pp. 107–110, Feb. 2012.

[11] Z. Koldovský, P. Tichavský, A. H. Phan, and A. Cichocki, "A two-stage MMSE beamformer for underdetermined signal separation," *IEEE Signal Process. Lett.*, vol. 20, no. 12, pp. 1227–1230, Dec. 2013.

[12] L. De Lathauwer and J. Castaing, "Blind identification of underdetermined mixtures by simultaneous matrix diagonalization," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1096–1105, Apr. 2008.

[13] S. Xie, L. Yang, J.-M. Yang, G. Zhou, and Y. Xiang, "Time-frequency approach to underdetermined blind source separation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 2, pp. 306–316, Feb. 2012.

[14] L. Zhen, D. Peng, Z. Yi, Y. Xiang, and P. Chen, "Underdetermined blind source separation using sparse coding," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2016.2610960.

[15] S. Kim and C. D. Yoo, "Underdetermined blind source separation based on subspace representation," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2604–2614, Jul. 2009.

[16] V. G. Reju, S. N. Koh, and Y. Soon, "An algorithm for mixing matrix estimation in instantaneous blind source separation," *Signal Process.*, vol. 89, no. 9, pp. 1762–1773, 2009.

[17] G. Wunder, H. Boche, T. Strohmer, and P. Jung, "Sparse signal processing concepts for efficient 5G system design," *IEEE Access*, vol. 3, pp. 195–208, Sep. 2015.

[18] P. Tichavsky and Z. Koldovsky, "Weight adjusted tensor method for blind separation of underdetermined mixtures of nonstationary sources," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1037–1047, Apr. 2011.

[19] L. Zou, X. Chen, and Z. J. Wang, "Underdetermined joint blind source separation for two datasets based on tensor decomposition," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 673–677, May 2016.

[20] L. Zou, Z. J. Wang, X. Chen, and X. Ji, "Underdetermined joint blind source separation based on tensor decomposition," in *Proc. IEEE Electr. Comput. Eng. (CCECE)*, Apr. 2016, pp. 1–4.

[21] X.-F. Gong, X.-L. Wang, and Q.-H. Lin, "Generalized non-orthogonal joint diagonalization with lu decomposition and successive rotations," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 8–12, May 2015.

[22] M. Congedo, R. Phlypo, and J. Chatel-Goldman, "Orthogonal and non-orthogonal joint blind source separation in the least-squares sense," in *Proc. IEEE 20th Eur. Signal Process. Conf. (EUSIPCO)*, Apr. 2012, pp. 1885–1889.

[23] E. Acar, D. M. Dunlavy, and T. G. Kolda, "A scalable optimization approach for fitting canonical tensor decompositions," *J. Chemometrics*, vol. 25, no. 2, pp. 67–86, 2011.

[24] H. A. Kiers, "Towards a standardized notation and terminology in multiway analysis," *J. Chemometrics*, vol. 14, no. 3, pp. 105–122, 2000.

[25] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[26] G. Zhou and A. Cichocki, "Canonical polyadic decomposition based on a single mode blind source separation," *IEEE Signal Process. Lett.*, vol. 19, no. 8, pp. 523–526, Aug. 2012.

[27] J. B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra Appl.*, vol. 18, no. 2, pp. 95–138, 1977.

[28] L. De Lathauwer, "A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 28, no. 3, pp. 642–666, 2006.

[29] I. Domanov and L. D. Lathauwer, "Generic uniqueness conditions for the canonical polyadic decomposition and indscal," *SIAM J. Matrix Anal. Appl.*, vol. 36, no. 4, pp. 1567–1589, 2015.

[30] I. Domanov and L. D. Lathauwer, "Canonical polyadic decomposition of third-order tensors: Reduction to generalized eigenvalue decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 2, pp. 636–660, 2014.

[31] L. Sorber, M. Van Barel, and L. De Lathauwer, "Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-(L_r, L_r, 1) terms, and a new generalization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 695–720, 2013.

[32] I. Domanov and L. De Lathauwer, "Canonical polyadic decomposition of third-order tensors: Relaxed uniqueness conditions and algebraic algorithm," *Linear Algebra Appl.*, vol. 513, pp. 342–375, Apr. 2017.

[33] M. Sørensen, I. Domanov, and L. De Lathauwer, "Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-(L_r, n, L_r, n, 1) terms—Part II: Algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 36, no. 3, pp. 1015–1045, 2015.

[34] E. Acar, T. G. Kolda, and D. M. Dunlavy. *All-at-Once Optimization for Coupled Matrix and Tensor Factorizations*. [Online]. Available: https://arxiv.org/abs/1105.3422

[35] L. Sorber, M. V. Barel, and L. D. Lathauwer, "Structured data fusion," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 586–600, Jun. 2015.

[36] N. Vervliet, O. Debals, and L. De Lathauwer, "Tensorlab 3.0-numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization," in *Proc. Conf. Rec. IEEE 50th Asilomar Conf. Signals, Syst. Comput.*, Sep. 2016.

[37] D. M. Dunlavy, T. G. Kolda, and E. Acar, "Poblano v1. 0: A matlab toolbox for gradient-based optimization," Sandia Nat. Lab., Albuquerque, NM and Livermore, CA, USA, Tech. Rep. SAND2010-1422, 2010, pp. 1–48.

[38] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Optimum Array Processing*. Hoboken, NJ, USA: Wiley, 2004.

[39] G. Strang, *Introduction to Linear Algebra*. Cambridge, MA, USA: Wellesley-Cambridge Press, 2003.

[40] S. Kim and C. D. Yoo, "Underdetermined blind source separation based on generalized gaussian distribution," in *Proc. 16th IEEE Signal Process. Soc. Workshop Mach. Learn. Signal Process.*, Apr. 2006, pp. 103–108.

[41] N. M. Correa, T. Adali, Y.-O. Li, and V. D. Calhoun, "Canonical correlation analysis for data fusion and group inferences," *IEEE Signal Process. Mag.*, vol. 27, no. 4, pp. 39–50, Jun. 2010.
[42] X. Chen, Z. J. Wang, and M. J. McKeown, "A three-step multimodal analysis framework for modeling corticomuscular activity with application to Parkinson's disease," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 4, pp. 1232–1241, Apr. 2014.
[43] A. L. Goldberger *et al.*, "Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

LIANG ZOU received the B.Sc. degree in microelectronics from Anhui University, China, in 2010, and the M.Sc. degree in biomedical engineering from the University of Science and Technology of China in 2013. He is currently pursuing the Ph.D. degree with The University of British Columbia, Canada. His current research interest includes statistical signal processing and machine learning.

XUN CHEN received the B.S. degree in electrical engineering from the University of Science and Technology of China in 2009 and the Ph.D. degree in biomedical engineering from The University of British Columbia, Canada, in 2014. He was a Research Scientist with the Department of Electrical and Computer Engineering, The University of British Columbia. He is currently with the Department of Biomedical Engineering, Hefei University of Technology, China, as a Full Professor. His research interests include the broad areas of statistical signal processing and machine learning in biomedical applications.

XIANGYANG JI received the B.S. degree in materials science and the M.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1999 and 2001, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He joined Tsinghua University, Beijing, in 2008, where he is currently a Full Professor with the Department of Automation, School of Information Science and Technology. His research interests include signal processing, image/video processing, image/video compression and communication, 3-D representation, and reconstruction.

Z. JANE WANG (F'17) received the B.S. degree from Tsinghua University, China, in 1996, and the M.S. and Ph.D. degrees from the University of Connecticut in 2000 and 2002, respectively, all in electrical engineering. She was a Research Associate with the Electrical and Computer Engineering Department, University of Maryland, College Park. Since 2004, she has been with the Department Electrical and Computer Engineering, The University of British Columbia, Canada, where she is currently a Full Professor. Her research interests include statistical signal processing theory and applications with a current focus on brain data analytics. She has been an Associate Editor of the *IEEE Signal Processing Magazine*, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II.