Blind Identification of Underdetermined Mixtures via Reduced Tensor Decomposition

Andreas Devogel, Lieven De Lathauwer
Department of Electrical Engineering (ESAT)
STADIUS, KU Leuven, Leuven, Belgium

***Abstract*—We present improved versions of two algorithms for blind source separation in the case where there are fewer sensors than sources. These methods rely on computing the decomposition of an auxiliary tensor. As generating the auxiliary tensor is computationally expensive, we propose an alternative methodology that only computes a reduced version of the tensor, which moreover holds the most reliable part of the information. Additionally we introduce state-of-the methods for computing the tensor decomposition, which are new in independent component analysis. Eventually, we arrive at a reliable scheme that only requires the computation of a limited number of partial matrix eigenvalue decompositions.**

***Index Terms*—Higher-order tensor, canonical polyadic decomposition (CPD), higher-order statistics, independent component analysis (ICA), underdetermined mixtures.**

I. Introduction

Independent component analysis (ICA) [1], or blind source separation (BSS), aims to identify a set of sources based only on measurements of the simultaneously received signal combinations. This problem can be represented by the following linear mixture model

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n}. \tag{1}$$

Here, the stochastic vector $\mathbf{s} \in \mathbb{C}^R$ represents R unknown source signals. The mixing matrix $\mathbf{A} \in \mathbb{C}^{J\times R}$, which is a priori unknown, characterizes how the source signals are combined to produce the J multichannel observations represented by the stochastic vector $\mathbf{x} \in \mathbb{C}^J$. Lastly, $\mathbf{n} \in \mathbb{C}^J$ denotes additive noise on the observations.

The goal of ICA is now to estimate the source signals $\mathbf{s}$ and/or mixing matrix $\mathbf{A}$ from the observations $\mathbf{x}$ in (1), assuming that the sources are statistically independent from one another. In the overdetermined case, meaning that there are at least as many sensors as signals, or $J \geq R$, the sources can be approximately separated by left multiplying the observations with the pseudo-inverse of the mixing matrix estimate. In the underdetermined or overcomplete case, meaning that there are fewer sensors than signals, or $J < R$, this no longer holds. In this case, for any observational sample $\mathbf{x}_t$, there exists an affine variety of dimension $R-J$ with possibly corresponding source samples $\mathbf{s}_t$, satisfying $\mathbf{x}_t = \mathbf{A}\mathbf{s}_t$. This being said, the mixture matrix and source distributions are still unique under mild assumptions [2]. Under such conditions, estimation of the mixing matrix itself is an overdetermined problem, even if the underlying ICA problem is underdetermined. From this mixing matrix estimate, the sources may then be separated using probabilistic tools [3], or by exploiting source properties such as sparsity of signals or finite alphabets. Therefore, estimation of the mixing matrix first, and sources second, remains an important strategy also for underdetermined ICA.

In this paper, we will focus on the direct estimation of the mixing matrix from observed samples, by means of the extension of two algebraic methods for underdetermined ICA. These methods are based on the decomposition of a higher-order tensor into a minimal sum of rank-1 terms, via the canonical polyadic decomposition (CPD) [4]. We specifically consider the *Second-Order Blind Identification of Underdetermined Mixtures* (SOBIUM) algorithm [5], and the *Fourth-Order-Only Blind Identification* (FOOBI) algorithm [6]. These methods compute certain statistics of the observations and construct a tensor based on them, which admits a CPD that reveals the underlying mixing matrix. Both methods crucially rely on the computation of the nullspace of a large matrix to obtain an auxiliary tensor that is subsequently decomposed. Since the size of this matrix scales as $O(R^2J^4)$, this computation quickly becomes computationally very expensive. We propose an alternative methodology where only a subspace of this nullspace is used, which contains the most relevant information for the problem. This would not only lower the cost of computing the nullspace, it also results in a reduced version of the auxiliary tensor, making the decomposition of the latter cheaper as well. Furthermore, we introduce current state-of-the-art methods for CPD computation, in order to more accurately and efficiently compute the decomposition for the reduced auxiliary tensor.

The paper is laid out as follows. Section II introduces the statistical quantities used in the SOBIUM and FOOBI algorithms, detailing how their structure can reveal the mixing matrix, and explains how this matrix can effectively be computed via the CPD of an auxiliary tensor. Section III discusses how a reduced version of this tensor may lead to more efficient computation without significant loss of accuracy. Experimental results based on simulation data are discussed in section IV. Section V finally contains our conclusions and future work.

This work was funded by: (1) the Flemish Government: AI Research Program. Lieven De Lathauwer and Andreas Devogel are affiliated to Leuven.AI - KU Leuven institute for AI, B-3000, Leuven, Belgium. (2) KU Leuven Internal Funds: iBOF/23/064, C14/22/096.

Notation: We denote scalars by lower case italic (a), vectors by bold lower case ($\mathbf{a}$), matrices by bold upper case ($\mathbf{A}$), and tensors by calligraphic upper case ($\mathcal{A}$). We use capital italic to denote upper bounds of index values ($i = 1, 2, \ldots, I$). Entries of of vectors, matrices and tensors are written with indices in subscript, and are often denoted as scalar representations ($(\mathbf{a})_i = a_i$, $(\mathbf{A})_{ij} = a_{ij}$, $(\mathcal{A})_{ijk} = a_{ijk}$, $\ldots$). The jth column of a matrix $\mathbf{A}$ is then denoted as $\mathbf{a}_j$.

The Khatri–Rao product $\odot$ of two matrices $\mathbf{A}$ and $\mathbf{B}$ with L columns is defined by $(\mathbf{A} \odot \mathbf{B})_{(i-1)L+j,\,l} = a_{il}\, b_{jl}$. The outer product $\otimes$ of an arbitrary number of vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$ is defined by $(\mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} \otimes \cdots)_{ijk\ldots} = a_i\, b_j\, c_k \cdots$.

Finally, we denote with $\cdot^{\mathrm{T}}$, $\cdot^{\mathrm{H}}$, $\cdot^*$ the transpose, conjugate transpose and complex conjugate respectively, and $\|\cdot\|$ represents the Frobenius norm.

II. Problem Derivation

In this section we briefly summarize the SOBIUM and FOOBI approach. We refer to [5], [6], [7] for further details.

A. SOBIUM

The SOBIUM algorithm estimates the mixing matrix based on second-order statistics [8]. To this end, it makes the assumption that the sources are individually correlated in time, in addition to being mutually uncorrelated. The spatial covariance matrices of the noiseless observations $\mathbf{x}_t$ then satisfy $\mathbf{C}_k = E\{\mathbf{x}_t\, \mathbf{x}_{t+\tau_k}^{\mathrm{H}}\} = \mathbf{A} \cdot \mathbf{D}_k \cdot \mathbf{A}^{\mathrm{H}}$, for different delays τ_k, where the $\mathbf{D}_k = E\{\mathbf{s}_t\, \mathbf{s}_{t+\tau_k}^{\mathrm{H}}\} \in \mathbb{C}^{R\times R}$ are diagonal, $k = 1, \ldots, K$. Define now the matrix $\mathbf{D} \in \mathbb{C}^{K\times R}$ by $(\mathbf{D})_{kr} = (\mathbf{D}_k)_{rr}$. If we construct the matrix $\mathbf{C} \in \mathbb{C}^{J^2\times K}$ such that $(\mathbf{C})_{(i-1)J+j,\,k} = (\mathbf{C}_k)_{ij}$, it then holds that

$$\mathbf{C} = (\mathbf{A} \odot \mathbf{A}^*) \cdot \mathbf{D}^{\mathrm{T}}. \tag{2}$$

Let the reduced SVD of $\mathbf{C}$ now be given by

$$\mathbf{C} = \mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^{\mathrm{H}} \tag{3}$$

with $\mathbf{U} \in \mathbb{C}^{J^2\times R}$, $\mathbf{\Sigma} \in \mathbb{R}^{R\times R}$, $\mathbf{V} \in \mathbb{C}^{K\times R}$, and define the matrix $\mathbf{H} = \mathbf{U} \cdot \mathbf{\Sigma} \in \mathbb{C}^{J^2\times R}$. Under the assumption that $\mathbf{A} \odot \mathbf{A}^*$ and $\mathbf{D}$ are of full column rank, implying that $R \leq J^2$ and $R \leq K$, it follows from (2) and (3) that

$$\mathbf{A} \odot \mathbf{A}^* = \mathbf{H} \cdot \mathbf{W} \tag{4}$$

for some nonsingular matrix $\mathbf{W} \in \mathbb{C}^{R\times R}$. In this way, the problem of estimating the mixing matrix $\mathbf{A}$ from $\mathbf{C}$ can be translated to finding the matrix $\mathbf{W}$ that satisfies (4).

B. FOOBI

The FOOBI algorithm estimates the mixing matrix based on fourth-order statistics [8]. Consider the fourth-order cumulant tensor $\mathcal{C}^{\mathbf{x}} \in \mathbb{C}^{J\times J\times J\times J}$ of the noiseless observations, whose elements are given by $c^{\mathbf{x}}_{ijkl} = \sum_{r=1}^{R} \kappa_r a_{ir} a^*_{jr} a^*_{kr} a_{lr}$, where κ_r is the kurtosis of the rth source. Define now the diagonal matrix $\tilde{\mathbf{C}}^{\mathbf{s}} \in \mathbb{R}^{R\times R}$ by $\tilde{c}^{\mathbf{s}}{}_{rr} = \kappa_r$. If we construct the matrix $\mathbf{C}^{\mathbf{x}} \in \mathbb{C}^{J^2\times J^2}$ such that $(\mathcal{C}^{\mathbf{x}})_{ijkl} = (\mathbf{C}^{\mathbf{x}})_{(i-1)J+j,\,(k-1)J+l}$, it then holds that

$$\mathbf{C}^{\mathbf{x}} = (\mathbf{A} \odot \mathbf{A}^*) \cdot \tilde{\mathbf{C}}^{\mathbf{s}} \cdot (\mathbf{A} \odot \mathbf{A}^*)^{\mathrm{H}}. \tag{5}$$

Since $\mathbf{C}^{\mathbf{x}}$ is Hermitian by (5), it admits a symmetric eigenvalue decomposition, given by

$$\mathbf{C}^{\mathbf{x}} = \mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{U}^{\mathrm{H}} \tag{6}$$

with $\mathbf{U} \in \mathbb{C}^{J^2\times R}$ and $\mathbf{\Sigma} \in \mathbb{R}^{R\times R}$. Define now the matrix $\mathbf{H} = \mathbf{U} \cdot \mathbf{\Sigma}^{\frac{1}{2}} \in \mathbb{C}^{J^2\times R}$. Under the assumption that $\mathbf{A} \odot \mathbf{A}^*$ is of full column rank, implying $R \leq J^2$, it follows from (5) and (6) that

$$(\mathbf{A} \odot \mathbf{A}^*) \cdot (\tilde{\mathbf{C}}^{\mathbf{s}})^{\frac{1}{2}} = \mathbf{H} \cdot \mathbf{W} \tag{7}$$

for some nonsingular matrix $\mathbf{W} \in \mathbb{C}^{R\times R}$. This again defers finding the mixing matrix $\mathbf{A}$ to finding this matrix $\mathbf{W}$.

C. Auxiliary CPD problem

From (4) and (7), we have obtained that right multiplication of $\mathbf{H}$ with a corresponding matrix $\mathbf{W}$ equals $\mathbf{A} \odot \mathbf{A}^*$, up to scaling. Let the matricization of the rth column vector be defined as $\mathrm{mat}((\mathbf{A} \odot \mathbf{A}^*)_r)_{ij} = (\mathbf{A} \odot \mathbf{A}^*)_{(i-1)J+j,\,r}$, then we see that $\mathrm{mat}((\mathbf{A} \odot \mathbf{A}^*)_r) = \mathbf{a}_r\, \mathbf{a}_r^{\mathrm{H}}$ adheres to a rank-1 structure, $r = 1, \ldots, R$. This suggests that we find W by imposing every column of $\mathbf{H} \cdot \mathbf{W}$ to have rank-1 structure. To identify this structure, we introduce the "rank-1 detecting map"

$$\Phi : (\mathbf{X}, \mathbf{Y}) \in \mathbb{C}^{I_1\times I_2} \times \mathbb{C}^{I_1\times I_2} \to \Phi(\mathbf{X}, \mathbf{Y}) \in \mathbb{C}^{I_1\times I_2\times I_1\times I_2}$$

$$\text{with } \Phi(\mathbf{X}, \mathbf{Y})_{ijkl} = x_{ij}\, y_{kl} + y_{ij}\, x_{kl} - x_{il}\, y_{kj} + y_{il}\, x_{kj}$$

which has the property that $\Phi(\mathbf{X}, \mathbf{X}) = \mathcal{O}$ if and only if $\mathbf{X}$ is of rank at most one. Define now the matricizations $\mathbf{H}_s = \mathrm{mat}(\mathbf{h}_s) \in \mathbb{C}^{J\times J}$ as before, and define the tensors $\mathcal{P}_{st} = \Phi(\mathbf{H}_s, \mathbf{H}_t) \in \mathbf{C}^{J\times J\times J\times J}$, $s, t = 1, 2, \ldots, R$. It can be shown [6], [7] that, assuming the tensors $\Phi(\mathbf{a}_s\mathbf{a}_s^{\mathrm{H}}, \mathbf{a}_s\mathbf{a}_s^{\mathrm{H}})$ are all linearly independent, there exist R linearly independent symmetric matrices $\mathbf{M}_r \in \mathbb{C}^{R\times R}$ such that

$$\sum_{s,t=1}^{R} (\mathbf{M}_r)_{st}\, \mathcal{P}_{st} = \mathcal{O}. \tag{8}$$

Moreover, the matrices $\mathbf{M}_r$ satisfy $\mathbf{M}_r = \mathbf{W} \cdot \mathbf{\Lambda}_r \cdot \mathbf{W}^{\mathrm{T}}$, where the matrices $\mathbf{\Lambda}_r \in \mathbb{C}^{R\times R}$ are diagonal, $r = 1, \ldots, R$, and $\mathbf{W}$ is the matrix we wish to compute. In other words, by solving the linear system (8) for the matrices $\mathbf{M}_r$, we obtain a simultaneous matrix diagonalization problem from which we can find the desired matrix $\mathbf{W}$. This problem can equivalently be interpreted as the CPD of an auxiliary third-order tensor. To this end, define the matrix $\mathbf{\Lambda} \in \mathbb{C}^{R\times R}$ by $(\mathbf{\Lambda})_{ri} = (\mathbf{\Lambda}_r)_{ii}$. If we now construct the tensor $\mathcal{M} \in \mathbb{C}^{R\times R\times R}$ by stacking the $\mathbf{M}_r$ matrices frontal-slice-wise, it adheres to the CPD

$$\mathcal{M} = \sum_{r=1}^{R} \mathbf{w}_r \otimes \mathbf{w}_r \otimes \boldsymbol{\lambda}_r \tag{9}$$

which we denote as $\mathcal{M} = [\![\mathbf{W}, \mathbf{W}, \mathbf{\Lambda}]\!]$. By computing the frontal slices $\mathbf{M}_r$ via (8) and solving the associated CPD (9), we can thus retrieve the matrix $\mathbf{W}$. We can then compute $\mathbf{H} \cdot \mathbf{W}$, which equals $\mathbf{A} \odot \mathbf{A}^*$ up to scaling by (4) and (7). This allows us to estimate the columns $\mathbf{a}_r$ of the mixing matrix from $\mathrm{mat}((\mathbf{H} \cdot \mathbf{W})_r)$ through best rank-1 approximation.

III. Reduced Tensor Computation

Due to the symmetry of the matrices $\mathbf{M}_r$ and the fact that $\mathcal{P}_{st} = \mathcal{P}_{ts}$, we can rewrite (8) more compactly as

$$\sum_{s=1}^{R} (\mathbf{M}_r)_{ss}\, \mathcal{P}_{ss} \;+\; 2 \sum_{s,t=1;\; s<t}^{R} (\mathbf{M}_r)_{st}\, \mathcal{P}_{st} = \mathcal{O}. \qquad (10)$$

If we then place the entries of $\mathbf{M}_r$ in some order into a row vector $\mathbf{m}_r \in \mathbb{C}^{R(R+1)/2}$, and stack vectorizations of the tensors $\mathcal{P}_{st}$ columnwise in the same order into a matrix $\mathbf{P} \in \mathbb{C}^{J^4 \times R(R+1)/2}$, we can write (10) as the linear system $\mathbf{P} \cdot \mathbf{m}_r = \mathbf{0}$, from which the R matrices $\mathbf{M}_r$ may be computed. In real-world applications where noise is present on the observations $\mathbf{x}$, the matrices $\mathbf{H}$ obtained from the estimated statistics might not exactly satisfy (4) and (7), and this linear system might not exactly admit an R-dimensional solution space. In this case, we can approximate the linearly independent solutions by the R right singular vectors corresponding to the R smallest singular values of the matrix $\mathbf{P}$. For convenience, we will enumerate these singular vectors in reverse order, meaning that $\mathbf{m}_1$ and the associated symmetric matrix $\mathbf{M}_1$ correspond to the smallest singular value σ_1, and so on.

We can at this point use the singular values σ_r as a measure of the quality of their associated solution matrix $\mathbf{M}_r$. If σ_r is very close to 0, it is very close to being in the nullspace of the noisy matrix $\mathbf{P}$, and thus likely to be close to a noiseless solution. Such matrices $\mathbf{M}_r$ will contain relevant information for the estimation of $\mathbf{W}$. On the other hand, if σ_r is not close to 0, we expect its associated $\mathbf{M}_r$ matrix to not provide much added information. We therefore propose to only consider a subset of $R_M < R$ matrices $\mathbf{M_r}$, corresponding to the R_M smallest singular values σ_r that are sufficiently close to 0. We then obtain the reduced tensor $\mathcal{M}^{(R_M)}$, consisting of only the first R_M frontal slices of $\mathcal{M}$. In this approach, (9) reduces to

$$\mathcal{M}^{(R_M)} = \sum_{r=1}^{R} \mathbf{w}_r \otimes \mathbf{w}_r \otimes \boldsymbol{\lambda}_r^{(R_M)} = [\![\mathbf{W}, \mathbf{W}, \boldsymbol{\Lambda}^{(R_M)}]\!]$$

where $\boldsymbol{\Lambda}^{(R_M)}$ is a reduced version of $\boldsymbol{\Lambda}$, containing only the first R_M rows. In the noiseless case, this CPD will still exactly retrieve the matrix $\mathbf{W}$ if no two columns $\boldsymbol{\lambda}_r^{(M_R)}$ are proportional, which is generically satisfied if $R_M \geq 2$ [9]. In the noisy case, exclusion of frontal slices with larger corresponding singular values should not lead to a significant reduction in accuracy for the estimation of $\mathbf{W}$. As a matter of fact, this may speed up the identification process significantly. Because the computation of a right singular vector of a tall matrix with m rows and n columns has a complexity of $O(mn^2)$ [10], the cost of computing any matrix $\mathbf{M}_r$ from $\mathbf{P}$ scales as $O(J^4R^4)$. Computing only $R_M < R$ of these matrices then accordingly reduces the cost by a factor R/R_M. Since the matric $\mathbf{P} \in \mathbb{C}^{J^2 \times R(R+1)/2}$ is much larger in size than the auxiliary tensor $\mathcal{M} \in \mathbb{C}^{R \times R \times R}$, and the reduced auxiliary tensor $\mathcal{M}^{(R_M)} \in \mathbb{C}^{R \times R \times R}$ is even smaller, we obtain a highly significant reduction of the overall complexity.

Table I shows the outline of the reduced tensor decomposition scheme for both SOBIUM and FOOBI as described above. We will refer to this as algorithm I. The actual source separation happens in the computation of the CPD of $\mathcal{M}$, in step 6 of the algorithm. Methods for CPD computation are generally optimization-based. A popular choice is the *Alternating Least Squares* (ALS) algorithm, since it is quite intuitive and allows for easy implementation. However, this method only achieves linear convergence at best. We propose the use of a *Nonlinear Least Squares* (NLS) algorithm [11] for CPD computation. This method is guaranteed to converge and achieves up-to-quadratic asymptotic convergence. An important part in realizing good performance for optimization is the use of an adequate initialization strategy. One options is to just initialize randomly, but then all the work is left to the optimization method. The most common choice for effective initialization is the *Generalized EigenValue Decomposition* (GEVD) algorithm [12], which uses a simple scheme based on the eigenvalue decomposition of a pair of frontal slices of the tensor. Here, we propose the use of a recent extension of this method, the *Generalized EigenSpace Decomposition* (GESD) algorithm [13]. This method is more robust than GEVD, as it does not rely on a single pair of frontal slices, but combines information obtained from different pairs of slices. Moreover, given such a pair of slices, it only extracts the information that can reliably be estimated from it. Consequently, GESD is somewhat more expensive than GEVD, though it produces a higher-quality initialization that in turn reduces the cost of the overall CPD computation. The use of such advanced methods is new in the setting of ICA.

TABLE I: Reduced Tensor Underdetermined ICA

	SOBIUM
1.	Estimate covariance matrices $\mathbf{C}_k$ and stack in $\mathbf{C}$
2.	Compute SVD $\mathbf{C} = \mathbf{U} \cdot \boldsymbol{\Sigma} \cdot \mathbf{V}^{\mathrm{H}}$; $\quad \mathbf{H} = \mathbf{U} \cdot \boldsymbol{\Sigma}$
	FOOBI
1.	Estimate covariance tensor $\mathcal{C}^{\mathbf{x}}$ and matricize to $\mathbf{C}^{\mathbf{x}}$
2.	Compute EVD $\mathbf{C}^{\mathbf{x}} = \mathbf{U} \cdot \boldsymbol{\Sigma} \cdot \mathbf{U}^{\mathrm{H}}$; $\quad \mathbf{H} = \mathbf{U} \cdot \boldsymbol{\Sigma}^{\frac{1}{2}}$
	Auxiliary CPD
3.	Compute $\mathcal{P}_{st} = \Phi(\mathbf{H}_s, \mathbf{H}_s)$, $1 \leq s < t \leq R$, and stack in $\mathbf{P}$
4.	Extract R_M right singular vectors $\mathbf{m}_r$ of $\mathbf{P}$ with small σ_r
5.	Form matrices $\mathbf{M}_r$ and stack in $\mathcal{M}^{(R_M)}$
6.	Compute CPD $\mathcal{M}^{(R_M)} = [\![\mathbf{W}, \mathbf{W}, \boldsymbol{\Lambda}^{(R_M)}]\!]$; $\quad \mathbf{T} = \mathbf{H} \cdot \mathbf{W}$
7.	Estimate $\mathbf{a}_r$ by best rank-1 approximation of mat$(\mathbf{t}_r)$, $r = 1, \ldots, R$

IV. Experimental results

We test the performance of algorithm I using synthetic data for the simulation of source signals and observations. We consider $R = 6$ sources received by a uniform circular array (UCA) of radius R_a with $J = 5$ identical sensors. We assume free space propagation such that the entries of the mixing matrix before normalization are given by

$$\tilde{a}_{jr} = \exp\left(2\pi\iota(x_j \cos(\theta_r)\cos(\phi_r) + y_j \cos(\theta_r)\sin(\phi_r))\right)$$

where ι is the imaginary unit, $x_j = (R_a/\lambda)\cos(2\pi(j-1)/J)$, $y_j = (R_a/\lambda)\sin(2\pi(j-1)/J)$, and $R_a/\lambda = 0.55$. The directions-of-arrival (DoA) of the sources are given by

$$\theta_1 = \frac{3\pi}{10}, \theta_2 = \frac{3\pi}{10}, \theta_3 = \frac{2\pi}{5}, \theta_4 = 0\ , \theta_5 = \frac{\pi}{10}, \theta_6 = \frac{3\pi}{5}$$
$$\phi_1 = \frac{7\pi}{10}, \phi_2 = \frac{9\pi}{10}, \phi_3 = \frac{3\pi}{5}, \phi_4 = \frac{4\pi}{5}, \phi_5 = \frac{3\pi}{5}, \phi_6 = \frac{3\pi}{5}.$$

The mixing matrix is then obtained by normalizing each column of $\tilde{\mathbf{A}}$ with respect to its Frobenius norm. The sources are unit-variance QAM4 in baseband, implying that they take their values equally likely in the set $\{\pm 1/\sqrt{2} \pm i/\sqrt{2}\}$. In the case of SOBIUM, we consider multipath signals, ensuring autocorrelation of the sources by applying random filters of length 51 with complex standard normally distributed coefficients. Additive zero-mean complex Gaussian noise is then added to the observations. Unless stated otherwise, we use 10000 samples with a signal-to-noise ratio (SNR) of 20dB.

Algorithm I was implemented in Matlab using the Tensorlab package [14], based on the original code for SOBIUM and FOOBI from Tensorlab+ [15], [16]. In the case of SOBIUM, we use $K = 31$ covariance matrices with lags $\{0, 1, \ldots, 30\}$. For the CPD computation in step 6 of the algorithm, we compare **1)** *Simultaneous Generalized Schur Decomposition* (SGSD) [17] initialized with a *Generalized Schur Decomposition* (GSD) of the first two frontal slices $\mathbf{M}_1$ and $\mathbf{M}_2$, against **2)** the NLS algorithm initialized with GESD, also obtained from Tensorlab+. The first option was used in the original implementation of SOBIUM and FOOBI, while the second option consists of state-of-the-art methods. For both SGSD and NLS, we use a function value tolerance of 10^{-9}, all other parameters use default values.

Performance of the algorithm is measured in terms of execution time and accuracy, for which we use two metrics. The first measures the error on $\mathcal{M}^{(R_M)}$ and is defined as $\epsilon_{\mathcal{M}} = \left\|\mathcal{M}^{(R_M)} - [\![\hat{\mathbf{W}}_1, \hat{\mathbf{W}}_2, \hat{\mathbf{\Lambda}}_{R_M}]\!]\right\| / \|\mathcal{M}^{(R_M)}\|$, where $\hat{\mathbf{W}}_1, \hat{\mathbf{W}}_2, \hat{\mathbf{\Lambda}}_{R_M}$ are the computed factors of the reduced auxiliary tensor. The second measure the error on $\mathbf{A}$ and is defined as $\epsilon_{\mathbf{A}} = \|\mathbf{A} - \hat{\mathbf{A}}\|/\|\mathbf{A}\|$, where $\hat{\mathbf{A}}$ is the optimally scaled and permuted mixing matrix estimate. We conduct Monte-Carlo experiments where the mean is taken over 100 runs.

Figure 1 shows the performance of the FOOBI version of algorithm I as a function of the maximal singular value σ_r of the frontal slices $\mathbf{M}_r$ that are kept to construct $\mathcal{M}^{(R_M)}$, where we always keep at least three slices. The top-left plot shows the distribution of the $R = 6$ minimal singular values of the normalized matrix $\mathbf{P}/\|\mathbf{P}\|$. We can see here that these singular values exhibit heavy outliers towards larger values or r. The remaining plots show the execution time t for computing the auxiliary CPD as well as the errors $\epsilon_{\mathcal{M}}$ and $\epsilon_{\mathbf{A}}$, both after initialization (dashed lines) and after further optimization (full line). We see that the execution time of GESD is stable and overall quite fast. The comparison to SGSD in this setting is not entirely fair as this method benefits from highly efficient built-in Matlab functionality. In terms of the error on the auxiliary CPD, we can see that it tends to increase with the singular value threshold, as more frontal slices $\mathbf{M}_r$ are used. This is because only a *row* is added to the third factor matrix of the CPD to capture a whole additional *matrix* slice of the noisy tensor $\mathcal{M}^{(R_M)}$, making the decomposition less exact. This is further amplified by the fact that the added slices are of decreasing quality. We also see that GSD initialization on its own is not sufficient to produce an adequate estimation of $\mathbf{W}$ from the CPD approximation based on only two slices. This is mirrored in its ability to estimate the mixing matrix. Looking at the error on the mixing matrix, we can see that virtually no accuracy is gained by including frontal slices $\mathbf{M}_r$ with corresponding singular values larger than about 0.025. This value nicely excludes most of the heavy singular value outliers, and substantiates the idea that corresponding matrices $\mathbf{M}_r$, which are not sufficiently close to the nullspace of $\mathbf{P}$, do not aid in the estimation process. Furthermore, we note that GESD is already able to compute a sufficiently accurate CPD approximation, such that little accuracy is gained through further optimization. The relatively high cost of this optimization procedure can thus be avoided. Notice further that the top-right figure shows only part of the computational time, since significant speedup of algorithm I as a whole is realized when only computing fewer slices $\mathbf{M}_r$ from $\mathbf{P}$, as explained in section III.

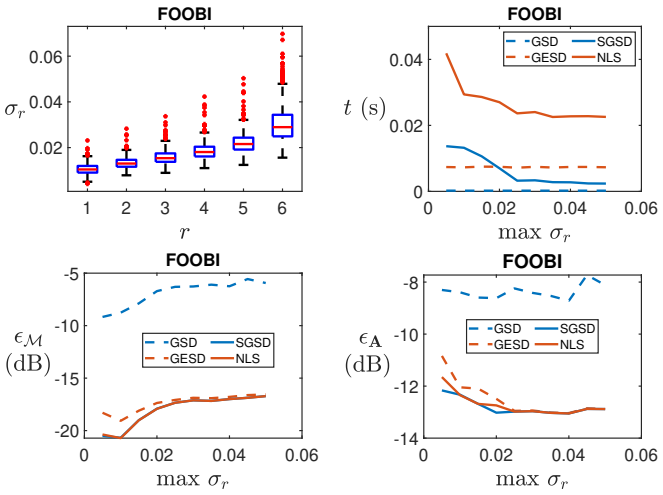

Fig. 1: Performance of FOOBI as a function of the maximal singular value σ_r for which a slice $\mathbf{M}_r$ of $\mathcal{M}$ is computed.

Figure 2 now shows the accuracy of the SOBIUM version of algorithm I as a function of the number R_M of frontal slices used to construct $\mathcal{M}^{(R_M)}$. The trends here are similar as for FOOBI. It is now more clear that NLS is able to consistently compute the best approximation of the auxiliary CPD, if only slightly. For the error on the mixing matrix, we can see that essentially no accuracy is gained when using more than four frontal slices $\mathbf{M}_r$. Even compared to using three slices, no significant accuracy is gained, as an overall error difference of 0.5dB only amounts to a factor 1.122. We see again that GESD performs about as well as the optimization methods.

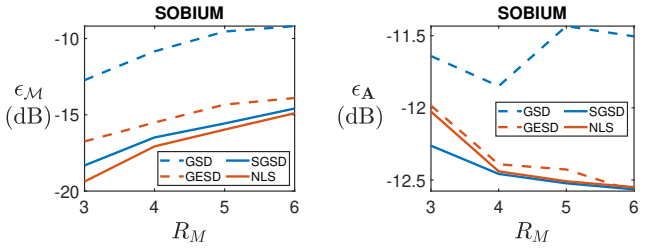

Fig. 2: Performance of SOBIUM as a function of the number R_M of frontal slices $\mathbf{M}_r$ of $\mathcal{M}$.

Figure 3 finally shows the accuracy of the SOBIUM version of algorithm I for SGSD and NLS as a function of the SNR on the observations, with either all $R = 6$ frontal slices $\mathbf{M}_r$ or only $R_M = 3$ slices used to construct $\mathcal{M}^{(R_M)}$. As expected, both errors decrease as the SNR increases, up to an SNR value of about 10dB, after which they stagnate as the statistical error of estimating the covariances from the observations dominates the accuracy. We can see that the error on the auxiliary CPD is about 5dB lower when using only three slices than when using all 6. We also see that, independently of the SNR, using only three slices leads to mixing matrix estimates that are practically as good as when using the full tensor.

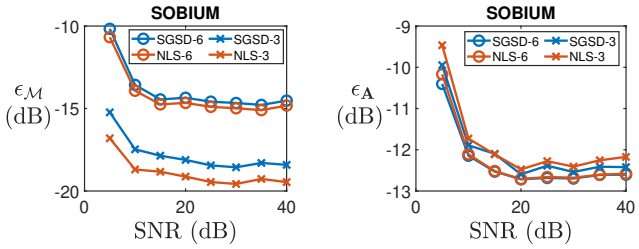

Fig. 3: Performance of SOBIUM as a function of the SNR for $R_M = 6$ and $R_M = 3$.

V. Conclusion and Future Work

We have introduced improved versions of the SOBIUM and FOOBI algorithms for blind identification of underdetermined mixtures. These methods rely on the computation of the CPD of an auxiliary tensor, of which the frontal slices are computed through a large linear system. We have shown that by computing only a small subset of these slices to construct a reduced version of the tensor, the computational cost of this construction as well as the reduced auxiliary CPD problem can be reduced significantly, without loss of accuracy. Furthermore, we have introduced state-of-the art methods for CPD computation, in the form of NLS optimization with GESD initialization, which are new to the setting of ICA. We have shown that GESD by itself already computes reliable the mixing matrix estimates, using only partial matrix EVDs.

Future work in this context incudes devising a method for the effective computation of only a subset of matrices $\mathbf{M}_r$ from the large matrix $\mathbf{P}$, in order to fully take advantage of the gain in computational complexity. The structure of $\mathbf{P}$ induced by the mapping Φ can additionally be exploited for efficiency. Furthermore, the obtained results can prove more generally relevant, as for instance a variant of the *Analytical Constant Modulus Algorithm* (ACMA) [18] could be developed using similar techniques.

References

[1] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

[2] J. Eriksson and V. Koivunen, "Identifiability, separability, and uniqueness of linear ICA models," *IEEE signal processing letters*, vol. 11, no. 7, pp. 601–604, 2004.

[3] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural computation*, vol. 12, no. 2, pp. 337–365, 2000.

[4] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE signal processing magazine*, vol. 32, no. 2, pp. 145–163, 2015.

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

[6] L. De Lathauwer, J. Castaing, and J.-F. Cardoso, "Fourth-order cumulant-based blind identification of underdetermined mixtures," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2965–2973, 2007.

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

[8] E. Kofidis and P. A. Regalia, "Tensor approximation and signal processing applications," *Contemporary Mathematics*, vol. 280, pp. 103–134, 2001.

[9] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on signal processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[10] L. N. Trefethen and D. Bau, *Numerical linear algebra*. SIAM, 2022.

[11] 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 Journal on Optimization*, vol. 23, no. 2, pp. 695–720, 2013.

[12] S. E. Leurgans, R. T. Ross, and R. B. Abel, "A decomposition for three-way arrays," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 4, pp. 1064–1083, 1993.

[13] E. Evert, M. Vandecappelle, and L. De Lathauwer, "A recursive eigenspace computation for the canonical polyadic decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 43, no. 1, pp. 274–300, 2022.

[14] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer. (2016, Mar.) Tensorlab 3.0. Available online. [Online]. Available: https://www.tensorlab.net

[15] S. Hendrikx, M. Boussé, N. Vervliet, M. Vandecappelle, R. Kenis, and L. De Lathauwer, "Tensorlab$^+$," available online, Version of Dec 2022 downloaded from https://www.tensorlabplus.net.

[16] S. Hendrikx, R. Widdershoven, N. Vervliet, and L. De Lathauwer, "Tensorlab+: A case study on reproducibility in tensor research," *IEEE Computing in Science & Engineering*, vol. 25, no. 5, pp. 6–13, 2023.

[17] L. De Lathauwer, B. De Moor, and J. Vandewalle, "Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 26, no. 2, pp. 295–327, 2004.

[18] A.-J. Van Der Veen and A. Paulraj, "An analytical constant modulus algorithm," *IEEE Transactions on Signal Processing*, vol. 44, no. 5, pp. 1136–1155, 1996.