ALGORITHMS FOR NONORTHOGONAL APPROXIMATE JOINT BLOCK-DIAGONALIZATION

P. Tichavsky and Z. Koldovsky*

Institute of Information Theory and Automation
Pod vodárenskou věží 4, 18208 Prague 8, Czech Republic

ABSTRACT

Approximate joint block diagonalization (AJBD) of a set of matrices has applications in blind source separation, e.g., when the signal mixtures contain mutually independent subspaces of dimension higher than one. In this paper we present three novel AJBD algorithms which differ in the final target criterion to be minimized in the optimization process. Two of the algorithms extend the principle of Jacobi elementary rotations to the more general concept of matrix elementary rotations. The proposed algorithms are compared to existing state-of-the art AJBD algorithms.

Index Terms— Source separation, independent subspaces

1. INTRODUCTION

Consider a set of square symmetric matrices $\mathbf{M}_n$, $n = 1,\ldots,N$, that are all block diagonal, with K blocks along its main diagonal, $\mathbf{M}_n = \text{Bdiag}(\mathbf{M}_{n1},\ldots,\mathbf{M}_{nK})$, where $\mathbf{M}_{nk}$ is the $k-$th block of $\mathbf{M}_n$ and the Bdiag$(\cdot)$ operator constructs a block-diagonal matrix from its argument matrices. The size of the blocks might not be identical. We assume that the size of the block $\mathbf{M}_{nk}$ is $I_k \times I_k$. The dimension of the matrices $\mathbf{M}_n$ is $d = I_1 + \ldots I_K$.

Next, assume that (possibly perturbed) congruence transformations of these matrices are given as

$$\mathbf{R}_n = \mathbf{A}\mathbf{M}_n\mathbf{A}^T + \mathbf{N}_n, \qquad n = 1,\ldots,N \tag{1}$$

where the superscript T denotes a matrix transposition, $\mathbf{A}$ is an unknown square "mixing matrix", and $\mathbf{N}_n$ is a perturbation (or "noise") matrix. We shall refer to the case where all $\mathbf{N}_n = \mathbf{0}$, $n = 1,\ldots,N$ as the "unperturbed" (or "noiseless") case. The choice of symbol $\mathbf{R}$ reflects the fact that the matrices in the set often play a role of (sample-) covariance matrices of a partitioned data, or time-lagged (sample-) covariance matrices.

The goal in Approximate Joint Block Diagonalization (AJBD) is to find a "demixing" matrix $\mathbf{W}$, such that the congruence matrices

$$\mathbf{R}_n^{\mathbf{W}} = \mathbf{W}\mathbf{R}_n\mathbf{W}^T, \qquad n = 1,\ldots,N \tag{2}$$

are all approximately block diagonal, having the blocks on the main diagonal of the same size as the original matrices $\mathbf{M}_n$. Ideally, one may wish to estimate $\mathbf{W} = \mathbf{A}^{-1}$ and get $\mathbf{R}_n^{\mathbf{W}} \approx \text{Bdiag}(\mathbf{R}_{n1}^{\mathbf{W}},\ldots,\mathbf{R}_{nK}^{\mathbf{W}})$, where $\mathbf{R}_{nk}^{\mathbf{W}} \approx \mathbf{M}_{nk}$. Sometimes the task is reversed and consists in fitting the given data matrices $\mathbf{R}_n$ by the model (1) with parameters $\mathbf{A}$, $\{\mathbf{M}_n\}$.

In general, however, it is impossible to recover the original blocks $\mathbf{M}_{nk}$ (even in the "noiseless" case), because of inherent ambiguities of the problem (e.g., [6]), but it is possible to recover "independent subspaces", as explained below.

Let $\mathbf{W}_0 = \mathbf{A}^{-1}$, also called demixing matrix, be partitioned in K blocks $\mathbf{W}_k$ of size $I_k \times d$, $\mathbf{W}_0 = [\mathbf{W}_1^T,\ldots,\mathbf{W}_K^T]^T$. Each block $\mathbf{W}_k$ represents a linear space of all linear combinations of its rows. These linear spaces are in general uniquely identifiable [6, 2]. Let $\widehat{\mathbf{W}}$ be an estimated demixing matrix. We say that $\widehat{\mathbf{W}}$ is "essentially equivalent" to $\mathbf{W}_0$ (and therefore represents an ideal joint block diagonalization), if there exists a suitable $d \times d$ permutation matrix $\mathbf{\Pi}$ such that for each $k = 1,\ldots,K$ the subspaces spanned by $\mathbf{W}_k$ and by the respective k-th block of $\mathbf{\Pi}\widehat{\mathbf{W}}$ coincide, and their mutual angle[1] is zero.

Some existing AJBD algorithms are restricted to the case where $\mathbf{A}$ (and therefore also $\widehat{\mathbf{W}}$) are orthogonal [3], some other algorithms consider a general matrix $\mathbf{A}$ [4, 6]. In this paper, we examine the general case.

It was shown in [8] that reasonable solutions to AJBD can be obtained using a two step approach, by first applying an ordinary approximate joint diagonalization (AJD) algorithm, and then clustering the separated components (rows of the demixing matrix). More specifically, it was shown that unlike several popular AJD approaches, one recently proposed AJD method (U-WEDGE, Uniformly Weighted Exhaustive Diagonalization with Gauss itErations [9]) features a unique ability to attain ideal separation in the unperturbed ("noiseless") case, for general (not necessarily orthogonal) matrices

*Supported by Grant Agency of the Czech Republic through the project 102/09/1278

[1] The mutual angle between two subspaces can be obtained in Matlab using the `subspace` function.

$\mathbf{A}$. The above mentioned methods are computationally appealing, but it appeared that in the presence of the noise, dedicated block algorithms may perform better. This paper is therefore devoted to finding AJBD solutions that minimize some *a priori* chosen criteria.

The paper is organized as follows. Section 2 presents a survey of three main cost functions used for AJBD in the literature. In Section 3, a general AJBD algorithm based on generalized Jacobi rotations is proposed. Specific details of the general procedure for the two criteria are derived in Section 4. Section 5 presents an algorithm for minimizing the third criterion. In Section 6, a novel initialization method is proposed. A simulation study in Section 7 verifies effectiveness of the proposed AJBD methods.

2. SURVEY OF MAIN AJBD CRITERIA

In the literature, we can see three main criteria used for joint block diagonalization.

$$C_{\mathrm{LS}}(\mathbf{W}) = \frac{1}{2}\sum_{n=1}^{N} \|\mathrm{Boff}(\mathbf{R}_n^{\mathbf{W}})\|_F^2 \tag{3}$$

where the operator "Boff" nullifies the elements of a matrix that lie in the diagonal blocks. This is used in [3] to build a unitary block diagonalization algorithm. In the case of non-orthogonal diagonalization, the criterion has to be completed by adding a constraint that prevents the algorithm from converging to the trivial solution $\mathbf{W} = \mathbf{0}$. The constraint is that all rows of $\mathbf{W}$ have unit Euclidean norm. An example is [4].

A different criterion was proposed by Lahat *et al* [5].

$$C_{\mathrm{LL}}(\mathbf{W}) = \frac{1}{2}\sum_{n=1}^{N} \log \frac{\det \mathrm{Bdiag}(\mathbf{R}_n^{\mathbf{W}})}{\det(\mathbf{R}_n^{\mathbf{W}})} \tag{4}$$

where Bdiag is the opposite of the operator "Boff", nullifies the elements of a matrix that lie out of the diagonal blocks. The criterion in motivated by maximum likelihood estimation. It is a generalization of the criterion proposed for approximate joint diagonalization in [7]. Minimization of this criterion leads to the maximum likelihood estimator, if $\mathbf{R}_n$ are sample covariance matrices from Gaussian distributed iid random vectors that have covariance matrices admitting the exactly block diagonal structure.

The third possible criterion was suggested by Nion [6]. The latest method is based on fitting the block diagonal model to the given set of matrices, seeking the mixing matrix $\mathbf{A} = \mathbf{W}^{-1}$,

$$C_{\mathrm{FIT}}(\mathbf{A}) = \sum_{n=1}^{N} \|\mathbf{R}_n - \mathbf{A}\mathbf{M}_n^{\mathbf{A}}\mathbf{A}^T\|_F^2 \tag{5}$$

where

$$\mathbf{M}_n^{\mathbf{A}} = \arg \min_{\mathbf{M}_n = \mathrm{Bdiag}(\mathbf{M}_n)} \|\mathbf{R}_n - \mathbf{A}\mathbf{M}_n\mathbf{A}^T\|_F^2 \,. \tag{6}$$

Success of the joint block diagonalization can be measured by these three criteria or, if the ideal mixing/demixing matrix is known, by the angle between the true and estimated subspaces mentioned in the previous subsection.

3. THE ELEMENTARY MATRIX ROTATION ALGORITHM

Each of the criterion mentioned in the former section (and namely the first two criteria) can be optimized by applying set of elementary rotations that follow the idea of well known Jacobi rotations. The elementary rotations follow the block structure of the assumed block matrices $\mathbf{M}_n$. The rotations have the form

$$\mathbf{E}_{ij}(\mathbf{B},\mathbf{C}) = \begin{bmatrix} \mathbf{I} & \dots & & \mathbf{0} \\ & \ddots & \mathbf{B} & \\ & \mathbf{C} & \ddots & \\ \mathbf{0} & & & \mathbf{I} \end{bmatrix} \tag{7}$$

where the diagonal is as in the identity matrix and the only nonzero off-diagonal blocks $\mathbf{B}$ and $\mathbf{C}$ of the sizes $I_i \times I_j$ and $I_j \times I_i$ lie at positions (i, j) and (j, i), respectively, for $1 \le i < j \le K$.

The blocks $\mathbf{B}$ and $\mathbf{C}$ are selected so that the updated demixing matrix $\mathbf{W}' = \mathbf{E}_{ij}(\mathbf{B},\mathbf{C})\mathbf{W}$ approximates the lowest attainable criterion function $C(\mathbf{W}')$, where $C(\mathbf{W}')$ is one of the criteria (3), (4), (5). The idea of the minimization is that the elementary rotations $\mathbf{E}_{ij}(\mathbf{B},\mathbf{C})$ should be small, i.e. the matrices $\mathbf{B}$ and $\mathbf{C}$ should be small. The true criterion function $C(\mathbf{W}')$ is replaced by the second order approximation in terms of $\mathbf{B}$ and $\mathbf{C}$, and is minimized in a closed form.

We will show in the next section that the approximation of $C(\mathbf{W}')$ might have the form $C(\mathbf{W}') = \varphi(\mathbf{B},\mathbf{C})$,

$$\begin{aligned}\varphi(\mathbf{B},\mathbf{C}) = \varphi_0 + \sum_{n=1}^{N} [&\mathrm{tr}(\mathbf{J}_{n1}\mathbf{B}) + \mathrm{tr}(\mathbf{J}_{n2}\mathbf{C}) \\ &+ \mathrm{tr}(\mathbf{J}_{n3}\mathbf{B}\mathbf{J}_{n4}\mathbf{B}) + \mathrm{tr}(\mathbf{J}_{n5}\mathbf{B}\mathbf{J}_{n6}\mathbf{B}^T) \\ &+ \mathrm{tr}(\mathbf{J}_{n7}\mathbf{C}\mathbf{J}_{n8}\mathbf{C}) + \mathrm{tr}(\mathbf{J}_{n9}\mathbf{C}\mathbf{J}_{n10}\mathbf{C}^T) \\ &+\mathrm{tr}(\mathbf{J}_{n11}\mathbf{B}\mathbf{J}_{n12}\mathbf{C}) + \mathrm{tr}(\mathbf{J}_{n13}\mathbf{B}\mathbf{J}_{n14}\mathbf{C}^T)]\end{aligned} \tag{8}$$

where the matrices $\mathbf{J}_{nm}$, $m = 1, \dots, 14$, depend on $\mathbf{R}_n^{\mathbf{W}}$, $\varphi_0 = \varphi(\mathbf{0},\mathbf{0})$, and "tr" denotes the trace of a matrix (sum of diagonal elements). Explicit expressions for the matrices $\mathbf{J}_{nm}$, $m = 1, \dots, 14$, for the criteria, (3) and (4) will be derived in the next section.

The criterion in (8) can be rewritten as

$$\varphi(\mathbf{B},\mathbf{C}) = \varphi_0 + \mathbf{g}^T\mathbf{v}(\mathbf{B},\mathbf{C}) + \frac{1}{2}\mathbf{v}(\mathbf{B},\mathbf{C})^T\mathbf{H}\mathbf{v}(\mathbf{B},\mathbf{C}) \tag{9}$$

where

$$\mathbf{v}(\mathbf{B},\mathbf{C}) = \begin{bmatrix} \mathrm{vec}(\mathbf{B}) \\ \mathrm{vec}(\mathbf{C}) \end{bmatrix} \tag{10}$$

$\mathbf{g}$ is a suitable vector of the same length, and $\mathbf{H}$ is a symmetric matrix. It holds

$$\mathbf{g} = \sum_{n=1}^{N} \begin{bmatrix} \mathrm{vec}(\mathbf{J}_{n1}^T) \\ \mathrm{vec}(\mathbf{J}_{n2}^T) \end{bmatrix} \tag{11}$$

$$\mathbf{H} = \sum_{n=1}^{N} \begin{bmatrix} \mathbf{H}_{BBn} & \mathbf{H}_{BCn} \\ \mathbf{H}_{BCn}^T & \mathbf{H}_{CCn} \end{bmatrix} \tag{12}$$

where

$$\begin{aligned} \mathbf{H}_{BBn} &= \mathbf{J}_{n6} \otimes \mathbf{J}_{n5}^T + \mathbf{J}_{n6}^T \otimes \mathbf{J}_{n5} \\ &\quad +(\mathbf{J}_{n3} \otimes \mathbf{J}_{n4}^T + \mathbf{J}_{n4} \otimes \mathbf{J}_{n3}^T)\mathbf{P} && (13) \\ \mathbf{H}_{BCn} &= \mathbf{J}_{n14} \otimes \mathbf{J}_{n13}^T + (\mathbf{J}_{n12} \otimes \mathbf{J}_{n11}^T)\mathbf{P} && (14) \\ \mathbf{H}_{CCn} &= \mathbf{J}_{n10} \otimes \mathbf{J}_{n9}^T + \mathbf{J}_{n10}^T \otimes \mathbf{J}_{n9} \\ &\quad +(\mathbf{J}_{n7} \otimes \mathbf{J}_{n8}^T + \mathbf{J}_{n8} \otimes \mathbf{J}_{n7}^T)\mathbf{P}, && (15) \end{aligned}$$

$\otimes$ is the Kronecker product, and where $\mathbf{P}$ stands for a permutation matrix such that $\mathbf{P}\mathrm{vec}(\mathbf{M}) = \mathrm{vec}(\mathbf{M}^T)$ for any matrix $\mathbf{M}$ of a suitable dimension ($I_i \times I_j$).

Once $\mathbf{g}$ and $\mathbf{H}$ are computed, the optimum $\mathbf{v}(\mathbf{B}, \mathbf{C})$ that minimizes $\varphi(\mathbf{B}, \mathbf{C})$ can be found as

$$\mathbf{v}(\mathbf{B}, \mathbf{C}) = -\mathbf{H}^{-1}\mathbf{g}\,. \tag{16}$$

After each elementary transformation, i.e. update $\mathbf{W}' = \mathbf{E}_{ij}(\mathbf{B}, \mathbf{C})\mathbf{W}$, it is recommended to check if the original optimization criterion has really decreased its value. If this is not the case, it means that the quadratic approximation was not accurate enough. One can proceed by replacing the step (16) by another one as it is done in the so called damped Gauss-Newton algorithm, also known as Levenberg-Marquardt,

$$\mathbf{v}(\mathbf{B}, \mathbf{C}) = -(\mathbf{H} + \mu\mathbf{I})^{-1}\mathbf{g} \tag{17}$$

where μ is a suitable positive parameter. Discussion of this modification, however, exceeds the scope of this paper.

The main algorithm consists after a suitable initialization in cyclic minimization of the elementary rotations with respect to all pairs (i, j), $1 \le i < j \le K$, until norms of the matrices $\mathbf{B}$ and $\mathbf{C}$ are smaller than a threshold for all the pairs.

4. EXPANSION OF THE COST FUNCTIONS

In this section we analyze the criteria (3), (4) in order to derive matrices $\mathbf{J}_{nm}$ that are needed for computation of the elementary rotations considered in the previous section. For simplicity of presentation we assume here that there are only three blocks in the matrices $\mathbf{M}_n$, and the elementary rotation is sought for $i = 1$ and $j = 2$. We can assume without any loss in generality that the third block includes all remaining blocks. We skip the running index n.

4.1. Case LS

Let $\mathbf{R}^{\mathbf{W}}$ and $\tilde{\mathbf{R}} = \mathbf{E}\mathbf{R}^{\mathbf{W}}\mathbf{E}^T$ be partitioned to blocks $\mathbf{R}_{11}, \ldots, \mathbf{R}_{33}$ and $\tilde{\mathbf{R}}_{11}, \ldots, \tilde{\mathbf{R}}_{33}$, respectively. Since

$$\tilde{\mathbf{R}} = \begin{bmatrix} \mathbf{I} & \mathbf{B} & \mathbf{0} \\ \mathbf{C} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \mathbf{R}_{13} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathbf{R}_{23} \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{C}^T & \mathbf{0} \\ \mathbf{B}^T & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$$

we have $\tilde{\mathbf{R}}_{12} = \mathbf{R}_{12} + \mathbf{R}_{11}\mathbf{C}^T + \mathbf{B}\mathbf{R}_{22}$, $\tilde{\mathbf{R}}_{13} = \mathbf{R}_{13} + \mathbf{B}\mathbf{R}_{23}$ and $\tilde{\mathbf{R}}_{23} = \mathbf{R}_{23} + \mathbf{C}\mathbf{R}_{13}$. Then

$$\varphi_{LS}(\mathbf{B}, \mathbf{C}) = \|\tilde{\mathbf{R}}_{12}\|_F^2 + \|\tilde{\mathbf{R}}_{13}\|_F^2 + \|\tilde{\mathbf{R}}_{23}\|_F^2\,. \tag{18}$$

A straightforward computation gives $\mathbf{J}_1 = 2\mathbf{R}_{22}\mathbf{R}_{12}^T$, $\mathbf{J}_2 = 2\mathbf{R}_{11}\mathbf{R}_{12}$, $\mathbf{J}_5 = \mathbf{I}$, $\mathbf{J}_6 = \mathbf{R}_{22}^2$, $\mathbf{J}_9 = \mathbf{I}$, $\mathbf{J}_{10} = \mathbf{R}_{11}^2$, $\mathbf{J}_{11} = \mathbf{R}_{11}$, $\mathbf{J}_{12} = 2\mathbf{R}_{22}$, $\mathbf{J}_3 = \mathbf{J}_4 = \mathbf{J}_7 = \mathbf{J}_8 = \mathbf{J}_{13} = \mathbf{J}_{14} = \mathbf{0}$.

4.2. Case LL

Consider the following Taylor series expansion of the matrix function $\log\det\mathbf{R}$ in a neighborhood of a positive definite matrix $\mathbf{R}_0$. Let $\Delta\mathbf{R} = \mathbf{R} - \mathbf{R}_0$. It holds

$$\log\frac{\det\mathbf{R}}{\det\mathbf{R}_0} \approx \mathrm{tr}(\mathbf{R}_0^{-1}\Delta\mathbf{R}) - \frac{1}{2}\mathrm{tr}(\mathbf{R}_0^{-1}\Delta\mathbf{R}\mathbf{R}_0^{-1}\Delta\mathbf{R})\,.$$

We apply this approximation to the last line of the expression

$$\begin{aligned} \varphi_{\mathrm{LL}}(\mathbf{B}, \mathbf{C}) &= \frac{1}{2}\log\frac{\det\mathrm{Bdiag}(\tilde{\mathbf{R}})}{\det\tilde{\mathbf{R}}} \\ &= \varphi_0 + \frac{1}{2}\log\frac{\det\mathrm{Bdiag}(\tilde{\mathbf{R}})}{\det\tilde{\mathbf{R}}} - \frac{1}{2}\log\frac{\det\mathrm{Bdiag}(\mathbf{R}^W)}{\det\mathbf{R}^W} \\ &= \varphi_0 + \frac{1}{2}\log\frac{\det\mathrm{Bdiag}(\tilde{\mathbf{R}})}{\det\mathrm{Bdiag}(\mathbf{R}^W)} - \frac{1}{2}\log\frac{\det\tilde{\mathbf{R}}}{\det\mathbf{R}^W}\,. \end{aligned}$$

After some computation we get (8) with $\mathbf{J}_1 = \mathbf{J}_3 = \mathbf{R}_{21}\mathbf{R}_{11}^{-1}$, $\mathbf{J}_2 = \mathbf{J}_7 = \mathbf{R}_{12}\mathbf{R}_{22}^{-1}$, $\mathbf{J}_4 = -\frac{1}{2}\mathbf{J}_1$, $\mathbf{J}_5 = \frac{1}{2}\mathbf{R}_{11}^{-1}$, $\mathbf{J}_6 = \mathbf{R}_{22} - \mathbf{R}_{21}\mathbf{R}_{11}^{-1}\mathbf{R}_{12}$, $\mathbf{J}_8 = -\frac{1}{2}\mathbf{J}_7$, $\mathbf{J}_9 = \frac{1}{2}\mathbf{R}_{22}^{-1}$, $\mathbf{J}_{10} = \mathbf{R}_{11} - \mathbf{R}_{12}\mathbf{R}_{22}^{-1}\mathbf{R}_{21}$, $\mathbf{J}_{11} = \mathbf{J}_{12} = \mathbf{I}$, $\mathbf{J}_{13} = \mathbf{J}_{14} = \mathbf{0}$. The resultant algorithm is similar to the one proposed in [5].

5. MINIMIZING THE FIT COST FUNCTION

The generalized Jacobi rotations presented in Section 3, appear not to be so effective in the case of the FIT criterion (5). The reason is that while in the case of the former two criteria, the Hessian matrix $\mathbf{H}$ can be shown to be always positive (semi-)definite, it is no longer true in the last case. Therefore, we propose another simple iterative procedure for this cost function. Starting from an initial estimate of the mixing matrix $\mathbf{A}$, its update $\mathbf{A}'$ is obtained as

$$\mathbf{A}' = \arg\min_{\tilde{\mathbf{A}}} \sum_{n=1}^{N} \|\mathbf{R}_n - \tilde{\mathbf{A}}\mathbf{M}_n^{\mathbf{A}}\mathbf{A}^T\|_F^2\,. \tag{19}$$

Since

$$\|\mathbf{R}_n - \tilde{\mathbf{A}}\mathbf{M}_n^{\mathbf{A}}\mathbf{A}^T\|_F^2 = \|\mathrm{vec}(\mathbf{R}_n) - (\mathbf{A}\mathbf{M}_n^{\mathbf{A}} \otimes \mathbf{I})\mathrm{vec}(\tilde{\mathbf{A}})\|^2$$

the optimum $\mathrm{vec}(\tilde{\mathbf{A}})$ can be found by solving the overdetermined system of linear equations

$$\begin{bmatrix} \mathbf{A}\mathbf{M}_1^{\mathbf{A}} \otimes \mathbf{I} \\ \vdots \\ \mathbf{A}\mathbf{M}_N^{\mathbf{A}} \otimes \mathbf{I} \end{bmatrix} \mathrm{vec}(\tilde{\mathbf{A}}) \approx \begin{bmatrix} \mathrm{vec}(\mathbf{R}_1) \\ \vdots \\ \mathrm{vec}(\mathbf{R}_N) \end{bmatrix}$$

in the least square sense. The result is

$$\begin{aligned} \mathrm{vec}(\mathbf{A}') &= \left[\sum_{n=1}^{N} (\mathbf{A}\mathbf{M}_n^{\mathbf{A}} \otimes \mathbf{I})^T (\mathbf{A}\mathbf{M}_n^{\mathbf{A}} \otimes \mathbf{I})\right]^{-1} \\ & \left[\sum_{n=1}^{N} (\mathbf{A}\mathbf{M}_n^{\mathbf{A}} \otimes \mathbf{I})^T \mathrm{vec}(\mathbf{R}_n)\right] . \end{aligned} \tag{20}$$

After some simplifications we get

$$\mathbf{A}' = \left[\sum_{n=1}^{N} \mathbf{R}_n \mathbf{A}\mathbf{M}_n^{\mathbf{A}}\right] \left[\sum_{n=1}^{N} \mathbf{M}_n^{\mathbf{A}} \mathbf{A}^T \mathbf{A}\mathbf{M}_n^{\mathbf{A}}\right]^{-1} . \tag{21}$$

It remains to explain computation of $\mathbf{M}_n^{\mathbf{A}}$ in (6). Let the mixing matrix $\mathbf{A}$ be vertically partitioned as $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_K]$, where the size of blocks $\mathbf{A}_k$ is $d \times I_k$, and let $\mathbf{M}_{nk}$ be the kth diagonal block of $\mathbf{M}_n^{\mathbf{A}}$. Then

$$\begin{aligned} \|\mathbf{R}_n - \mathbf{A}\mathbf{M}_n^{\mathbf{A}}\mathbf{A}^T\|_F^2 &= \|\mathbf{R}_n - \sum_{k=1}^{K} \mathbf{A}_k \mathbf{M}_{nk} \mathbf{A}_k^T\|_F^2 \\ = \|\mathrm{vec}(\mathbf{R}_n) - \sum_{k=1}^{K} (\mathbf{A}_k \otimes \mathbf{A}_k)\mathrm{vec}(\mathbf{M}_{nk})\|^2 \end{aligned} \tag{22}$$

The desired elements of $\mathbf{M}_n^{\mathbf{A}}$ can be arranged in a vector

$$\mathbf{m}_n = [\mathrm{vec}(\mathbf{M}_{n1})^T, \ldots, \mathrm{vec}(\mathbf{M}_{nK})^T]^T .$$

The optimum $\mathbf{m}_n$ is given as

$$\mathbf{m}_n = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T \mathrm{vec}(\mathbf{R}_n)$$

where

$$\mathbf{G} = [\mathbf{A}_1 \otimes \mathbf{A}_1, \ldots, \mathbf{A}_K \otimes \mathbf{A}_K] .$$

The convergence of the iterative process (19) appears to be very good, as it is shown in the simulation section.

6. THE ALGORITHMS INITIALIZATION

The algorithms proposed in the previous session are iterative and convergence to the global minimum of the respective criteria (3), (4), and (5) is not guaranteed. In order to minimize probability of getting the algorithm stuck in a local but not global minimum, it is convenient to find a suitable initialization. We propose to use the method advocated in [8], apply the AJD algorithm U-WEDGE, followed by a suitable clustering of rows of the mixing matrix, which would reveal the block structure of the demixed matrices.

First, we propose a modified method of clustering the rows, compared to the one proposed in [8]. It is a greedy algorithm again. Given the AJD demixing matrix $\mathbf{W}$, compute an auxiliary matrix $\mathbf{D}$ as

$$\mathbf{D} = \sum_{n=1}^{N} |\mathbf{W}\mathbf{R}_n\mathbf{W}^T| \tag{23}$$

where the absolute value is taken elementwise. If the demixing is perfect, $\mathbf{D}$ should have, after arranging columns and rows, the same block structure as the original matrices $\mathbf{M}_n$. Let the block sizes be ordered increasingly, $I_1 \leq \ldots \leq I_K$. Elements of all columns of $\mathbf{D}$ are sorted decreasingly to form a matrix $\mathbf{D}'$. Then, sort the (I_1+1)-th row of $\mathbf{D}'$ and denote it $\mathbf{r}_1$ for easy reference. The first block of the demixing matrix will be composed of the indices that correspond to the smallest I_1 elements in $\mathbf{r}_1$. It means that $\mathbf{W}_1$ is built of the rows of $\mathbf{W}$ with these indices. The rows and columns of $\mathbf{D}$ at the positions $i_1, \ldots, i_{I_1}$ are set to zero, and the procedure iterated until $K-1$ subspaces (blocks) $\mathbf{W}_k$, $k = 1, \ldots, K-1$, have been determined. The K-th block $\mathbf{W}_K$ is then composed of the rows that have not been selected before.

7. SIMULATIONS

We have compared performance of five AJBD algorithms: (1) U-WEDGE completed by clustering of rows of a demixing matrix, this algorithm is used to initialize all subsequent ones, (2) algorithm JBD_NCG [6], (3) the ad hoc algorithm of Ghennioui et al [4], and two algorithms proposed in this paper: (4) LSJBD and (5) FITJBD, minimizing the criteria (3) and (5), respectively. The LLAJD algorithm is not included, because it would need a different setup with all target matrices positive (semi-)definite.

We consider ten target matrices, each having four diagonal blocks of the size 5×5. The blocks were taken at random, different at each simulation trial, normally distributed with a zero mean, and were symmetrized by adding their transposition. The mixing matrix $\mathbf{A}$ was taken at random, also new in each simulation trial. Finally, an additive noise was added as in (1); the noise matrices were symmetrized as well, and scaled to achieve desired signal-to-noise ratio (SNR), which is defined as 10 $\log_{10}$ of Frobenius norm of the mixtures divided by Frobenius norm of the noise.

The noisy mixtures were processed by each of the five algorithms in 100 independent trials. Each algorithm has its own cost function, which is optimized, therefore we have chosen to measure the performance by the angular difference of the subspaces spanned by corresponding groups of rows of the true and estimated demixing matrix. The resultant mean and median average angular differences, as a function of SNR are plotted in Figure 1. We note namely excellent performance of the FITJBD algorithm, which appears to work at lower SNR's than its competitors. Convergence of the algorithm is nearly linear, as in gradient methods or as in JBD_NCG. In this particular example, the latter algorithm does not work well, but it is not bad in general, namely in lower dimensions. Both algorithms need thousands of iterations to converge.

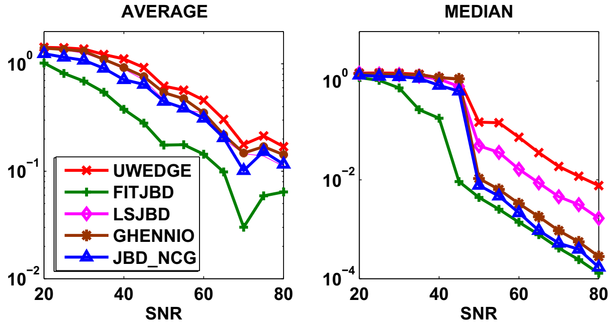

Fig. 1. Performance of the five AJBD techniques.

An advantage of the methods based on generalized Jacobi rotations (here LSJBD) is that they have nearly quadratic convergence. In general they require only 10-100 iterations to converge. The algorithm of Ghennioui is something between. Sometimes it converges quickly, as it had a quadratic convergence, namely at high SNR scenarios, in other cases it is slow. Numbers of iterations and the CPU time of the algorithms are shown in Figures 2 and 3, respectively.

8. CONCLUSIONS

We have presented novel algorithms for approximate joint block–diagonalization. They differ in the cost function that is optimized. Matlab codes of the proposed algorithms are posted on the Internet at http://si.utia.cas.cz/downloadPT.htm .

Acknowledgements

The authors wish to give thanks to Dr. Dimitri Nion for sending them a matlab code of his algorithm JBD_NCG.

9. REFERENCES

[1] K. Abed-Meraim and A. Belouchrani, "Algorithms for Joint Block Diagonalization", *Proc. of EUSIPCO* 2004, Vienna, Austria, pp. 209 - 212, 2004.

[2] L. de Lathauwer, "Decomposition of Higher-Order Tensor in Block Terms-Part II: Definitions and Uniqueness", *SIAM J. Matrix Anal. and Appl.* vol. 30, pp. 1033-1066, 2008.

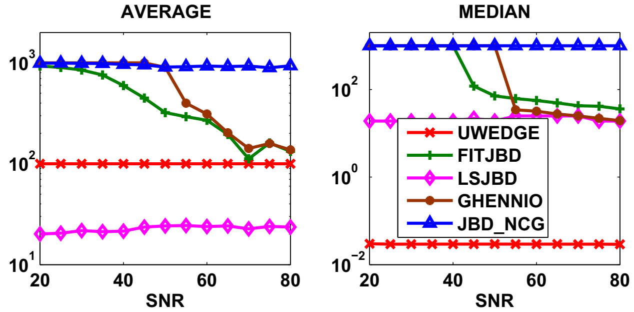

Fig. 2. Average and median number of iterations versus SNR.

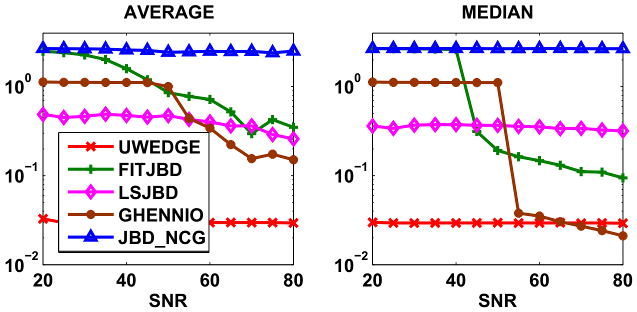

Fig. 3. Average and median CPU time versus SNR.

[3] C. Févotte and F. J. Theis, "Pivot Selection Strategies in Jacobi Joint Block-Diagonalization", *Proc. ICA'2007*, pp. 177-184.

[4] H. Ghennioui, F. El Mostafa, N. Thirion-Moreau, A. Adib, and E. Moreau, "A Nonunitary Joint Block Diagonalization algorithm for Blind Separation of Convolutive Mixtures of Sources", *IEEE Signal Processing Letters*, vol. 14, pp. 860–863, 2007.

[5] D. Lahat, J.-F. Cardoso, and H. Messer, "Joint Block Diagonalization Algorithms for Optimal Separation of Multidimensional Components", F. Theis et al. (Eds.): LVA/ICA 2012, LNCS 7191, pp. 155-162, 2012.

[6] D. Nion, "A Tensor Framework for Nonunitary Joint Block Diagonalization", *IEEE Trans. Signal Processing*, vol. 59, no. 10, pp. 4585-4594, Oct. 2011.

[7] D.-T. Pham , "Joint Approximate Diagonalization of Positive Definite Hermitian Matrices", *SIAM J. Matrix Anal. and Appl.* vol. 22, pp. 1136-1152, 2001.

[8] P. Tichavský, A. Yeredor, and Z. Koldovský, "On Computation of Approximate Joint Block-Diagonalization using Ordinary AJD", F. Theis et al. (Eds.): *LVA/ICA 2012*, LNCS 7191, pp. 163-171, 2012.

[9] P. Tichavský and A. Yeredor , "Fast Approximate Joint Diagonalization Incorporating Weight Matrices", *IEEE Tr. Signal Processing*, vol. 57, pp. 878-891, March 2009.