Signal Processing 128 (2016) 440–448

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

New Jacobi-like algorithms for non-orthogonal joint diagonalization of Hermitian matrices

Guang-Hui Cheng [a,*], Shan-Man Li [a], Eric Moreau [b]

[a] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, P. R. China
[b] Université de Toulon, CNRS, LSIS, UMR 7296, 83957 La Garde, France

ARTICLE INFO

Article history:
Received 5 December 2015
Received in revised form
27 April 2016
Accepted 13 May 2016
Available online 15 May 2016

Keywords:
Joint diagonalization
Blind source separation
Hermitian matrix
Jacobi-like algorithm

ABSTRACT

In this paper, two new algorithms are proposed for non-orthogonal joint matrix diagonalization under Hermitian congruence. The idea of these two algorithms is based on the so-called Jacobi algorithm for solving the eigenvalues problem of Hermitian matrix. The algorithms are then called 'general Jabobi-like diagonalization' algorithms (GERALD). They are based on the search of two complex parameters by the minimization of a quadratic criterion corresponding to a measure of diagonality. Lastly, numerical simulations are conducted to illustrate the effective performances of the GERALD algorithms.

1. Introduction

Recently, the joint diagonalization (JD) by congruence problem of a set of Hermitian matrices has arisen an increasing interest in the area of blind source separation (BSS) [1–11], independent component analysis (ICA) [12,13] and biomedical signal processing [14]. It is also noted that the canonical polyadic decomposition of third order tensors can be seen as a JD problem, see e.g. [15]. Hence, a number of algorithms have been proposed in the literatures [6–24], and could be basically divided into two categories: orthogonal and non-orthogonal ones. This depends on whether the searched diagonalizing matrix is unitary or not.

We notice that the unitary case has first been considered yielding the two well-known orthogonal algorithms JADE [19,20,3] and SOBI [2]. For the JADE algorithm, the mixing matrix is decomposed as a product of Givens (planar) rotations and each rotation parameter is derived analytically by minimizing a given objective function. Unfortunately, it was shown, see e.g. [21], that the pre-whitened operation required in practice for any orthogonal JD algorithm can adversely affect the overall performances. It is the main reason why nowadays the non-orthogonal case has received much attention.

In [6], the AC-DC (Alternating Columns-Diagonal Centers) algorithm is proposed. It is based on the minimization of the error model weighted least squares criterion and has a linear convergence rate. In [7], based on the property of strictly diagonally dominant matrices, the FFDIAG algorithm is proposed in the real case. It allows a quite simple iterative estimation of the diagonalizing matrix. A generalization of this algorithm to the two complex cases is proposed in [24]. In [9], the FAJD algorithm is proposed and is based on a column by column estimation of the diagonalizing matrix. by minimizing a weighted least squares criterion associated to a constraint to avoid the degenerate solution. In [5], the UWEDGE algorithm is proposed. It uses weighted matrices and has a connection to FFDIAG. In [16], a block diagonal algorithm is proposed that is based on an efficient nonlinear conjugate gradient (NCG) optimization procedure.

More recently, the ideas of Jacobi algorithms was developed in the non-orthogonal case. The main reason is the potential simplicity of such algorithms allowing a possible parallelization. Perhaps one of the first such related work was proposed in [13] where two algorithms in the real case was proposed based on the LU and QR decompositions of the diagonalizing matrix. In [22], using the same decompositions, the complex case is considered. In [17], the JDi based both on rotations and hyperbolic rotations is proposed in the real case. A generalization to the complex case is proposed in [18]. In [23], the ALUJA (a new decoupled Jacobi-like algorithm) algorithm is proposed. It is based on both an LU decomposition of the diagonalizing matrix and on an adapted local criterion.

In this paper, we propose a new approach for Jacobi-like algorithm where the elementary matrix under consideration for the Jacobi like procedure depends explicitly on two parameters. These

* Corresponding author.
E-mail address: ghcheng@uestc.edu.cn (G.-H. Cheng).

http://dx.doi.org/10.1016/j.sigpro.2016.05.013
0165-1684/

two parameters are derived optimally altogether. The generic algorithm is called 'GEneRal JAcobi-Like Diagonalization' (GERALD) The proposed algorithm is also shown to minimize the considered criterion at each iteration and the convergence to a local minima is ensured. In a second time, inspired by the works in [17] and [7,24], we show that by using the assumption that we are close to a diagonalizing solution, we can obtain an analytical minimizing solution for the two parameters. Interestingly enough, we also show that this last algorithm has a close link with the one in [24].

Our approach is different from the ones in [13,23] because there the elementary matrix only depends on one parameter. As shown by the following derivations, the proposed approach is then different. It is also clearly different from the ones in [17,18] since we consider a new parametrization of the elementary matrix with no rotation (classical nor hyperbolic).

The paper is organized as follows. In Section 2, we will review the JD problem, present the considered criterion and the overall description of the proposed algorithm. In Section 3, the two optimal parameters of each elementary matrix will be derived. In Section 4, numerical simulations are given for illustrating the performance of the two proposed algorithms. Finally, a conclusion will be drawn in Section 5.

Notation. Scalars are denoted by a lower case (a), vectors by a boldface lower case ($\mathbf{a}$) and matrices by a boldface upper case ($\mathbf{A}$). a_i is the i-th element of the vector $\mathbf{a}$ and a_{ij} is the (i, j)-th element of the matrix $\mathbf{A}$. $\mathbf{I}$ is the identity matrix. ZDiag$\{\cdot\}$ is a matrix operator that sets to zero the diagonal of the argument matrix. $\|\cdot\|_F$ is the Frobenius norm of the argument matrix. Let c be a complex number, we denote its conjugate by $\bar{c}$ and its modulus by $|c|$. The superscript $(\cdot)^H$ denotes the conjugate transpose.

2. Problem formulation and proposed algorithm

We assume that we have a set of Hermitian matrices $\mathcal{A} = \{\mathbf{A}_k \equiv (a_{k,ij}) \in \mathbb{C}^{N\times N},\ k = 1, \ldots, K\}$ that all share the following common decomposition:

$$\mathbf{A}_k = \mathbf{B}\mathbf{D}_k\mathbf{B}^H + \mathbf{E}_k, \tag{1}$$

where $\mathbf{B} \in \mathbb{C}^{N\times M} (N \geq M)$ denotes the so-called mixing matrix that is assumed full column rank, $\mathbf{D}_k \in \mathbb{C}^{M\times M}$ and $\mathbf{E}_k \in \mathbb{C}^{N\times N}$ are respectively diagonal matrices and additive noise matrices for all $k = 1, \ldots, K$.

The aim of JD by congruence is to seek a full row rank matrix $\mathbf{Q} \in \mathbb{C}^{M\times N}$ that makes the transformed matrices

$$\mathbf{T}_k = \mathbf{Q}\mathbf{A}_k\mathbf{Q}^H \tag{2}$$

all as diagonal as possible. The matrix $\mathbf{Q}$ is referred to as "demixing" matrix in the context of BBS.

In this paper, we consider the following objective function to be minimized

$$\mathcal{J}_1(\mathbf{Q}) = \sum_{k=1}^{K} \| \text{ZDiag}\{\mathbf{T}_k\} \|_F^2. \tag{3}$$

Ignoring the diagonal components, the goal of this objective function is to minimize the squares of all off diagonal components for all considered matrices. It can be noticed that a number of JD algorithms are based on this objective function [13,22,17,18,23].

In Jacobi like algorithms, the estimation of $\mathbf{Q}$ is carried by using a 'simple' multiplicative update. It is written as:

$$\mathbf{Q}_{\text{new}} = (\mathbf{I} + \mathbf{V})\mathbf{Q}_{\text{old}},$$

where $\mathbf{V} \in \mathbb{C}^{M\times M}$ and has a simple structure. In such case, $\mathbf{I} + \mathbf{V}$ is called the elementary matrix. The main advantage of the multiplicative update is that the property of full row rank of $\mathbf{Q}$ can more easily be maintained. This thus requires that $\mathbf{I} + \mathbf{V}$ be invertible.

By successive multiplicative updates, the minimization of the objective function $\mathcal{J}_1(\mathbf{Q})$ is solved by considering

$$\mathbf{A}_{k,\text{new}} \equiv \mathbf{Q}_{\text{new}}\mathbf{A}_k\mathbf{Q}_{\text{new}}^H = (\mathbf{I} + \mathbf{V})\mathbf{A}_{k,\text{old}}(\mathbf{I} + \mathbf{V})^H \tag{4}$$

for all possible elementary matrices and that until convergence.

In [13,23], the considered matrix $\mathbf{V}$ only depends on one non-zero component at a given position (i,j). Here we propose to consider a matrix $\mathbf{V}$ with two non-zero components at positions (i, j) and (j,i). Hence, the concrete form of this matrix is given as

$$\mathbf{V} = \mathbf{V}_{ij} \equiv \begin{pmatrix} 0 & \vdots & 0 & \vdots & 0 \\ \ldots & 0 & \ldots & v_{ij} & \ldots \\ 0 & \vdots & 0 & \vdots & 0 \\ \ldots & v_{ji} & \ldots & 0 & \ldots \\ 0 & \vdots & 0 & \vdots & 0 \end{pmatrix}.$$

It is easy to see that the matrix $I + \mathbf{V}_{ij}$ is non-singular if and only if $v_{ij}v_{ji} \neq 1$. In fact numerous computer simulations have proved that it is really not a problem. However if, very unfortunately, such a case occurs, then we drop the corresponding update and pursue with another indexes pair.

For the algorithm, the overall consideration of all indexes pairs is called a sweep. The algorithm is called 'GEneRal JAcobi-Like Diagonalization' (GERALD) and it is more precisely described in the table Algorithm 1.

Algorithm 1. The GERALD algorithm.

Input: $\mathbf{A}_k,\ k = 1, \ldots, K$ (matrices to be joint diagonalized)
1: **Initial** $\mathbf{Q} \in \mathbb{C}^{M\times N}$, full row rank matrix
2: Update $\mathbf{T}_k \longleftarrow \mathbf{Q}\mathbf{A}_k\mathbf{Q}^H$ for $k = 1, \ldots, K$.
3: **Repeat**
4: **for** $i = 1, \ldots, M$ **do**
5: **for** $j = i + 1, \ldots, M$ **do**
6: Compute $\mathbf{V}_{ij}$ to minimize $\mathcal{J}_1$ in (3).
7: $\mathbf{Q} \longleftarrow (\mathbf{I} + \mathbf{V}_{ij})\mathbf{Q}$.
8: $\mathbf{T}_k \longleftarrow (\mathbf{I} + \mathbf{V}_{ij})\mathbf{T}_k(\mathbf{I} + \mathbf{V}_{ij})^H$ for $k = 1, \ldots, K$.
9: **end for**
10: **end for**
11: **Until convergence**
Output: $\mathbf{Q}$ and $\mathbf{T}_k$ for $k = 1, \ldots, K$.

For the GERALD algorithm, a sweep corresponds to the repeat step 3 is called and the step 5 is called one transformation.

3. Derivations of the optimal updating matrix

Corresponding to the step 6 of the above algorithm description, our goal is now to find the two optimal parameters v_{ij} and v_{ji} minimizing $\mathcal{J}_1$ defined in (3). In this section, we will consider three strategies for establishing concrete algorithms.

3.1. Direct derivation

Let us denote