Nonnegative Joint Diagonalization by Congruence Based on LU Matrix Factorization
Lu Wang, Laurent Albera, Amar Kachenoura, Huazhong Shu, Lotfi Senhadji

To cite this version:

HAL Id: inserm-00841128
https://inserm.hal.science/inserm-00841128
Submitted on 7 Jul 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Nonnegative Joint Diagonalization by Congruence Based on LU Matrix Factorization

Lu Wang, Student Member, IEEE, Laurent Albera, Senior Member, IEEE, Amar Kachenoura, Huazhong Shu, Senior Member, IEEE, and Lotfi Senhadji, Senior Member, IEEE

Abstract—In this letter, a new algorithm for joint diagonalization of a set of matrices by congruence is proposed to compute the nonnegative joint diagonalizer. The nonnegativity constraint is imposed by means of a square change of variables. Then we formulate the high-dimensional optimization problem into several sequential polynomial subproblems using LU matrix factorization. Numerical experiments on simulated matrices emphasize the advantages of the proposed method, especially in the case of degeneracies such as for low SNR values and a small number of matrices. An illustration of blind separation of nuclear magnetic resonance spectroscopy confirms the validity and improvement of the proposed method.

Index Terms—Blind source separation, independent component analysis, LU factorization, nonnegative joint diagonalization by congruence, nuclear magnetic resonance spectroscopy.

I. INTRODUCTION

Consider a set $\mathcal{C} = \{C^{(k)}\}_{k=1}^{K}$ of $K$ real symmetric matrices sharing the common structure:

$$C^{(k)} = AD^{(k)}A^T$$  \hspace{1cm} (1)

where $A \in \mathbb{R}^{N \times P}$ denotes an unknown transformation matrix, and $D^{(k)} = \{D^{(k)}\}_{k=1}^{K}$ is a set of $(P \times P)$ unknown real diagonal matrices. $A$ is called a joint diagonalizer of $\mathcal{C}$. According to different applications, the $K$ matrices $C^{(k)}$ can be time-delayed covariance matrices, or higher order cumulant matrices. The objective of Joint Diagonalizing by Congruence (JDC) such matrices consists of estimating $A$ up to a scale factor and a permutation of columns. Such a problem appears in many signal processing contexts such as Blind Source Separation (BSS) and Independent Component Analysis (ICA). Some BSS applications involve a nonnegative mixing matrix $A$. For instance, in Nuclear Magnetic Resonance (NMR) spectroscopy, columns of $A$ represent the positive concentration of source metabolites [13]. Therefore we propose to impose a nonnegativity constraint on the matrix $A$ in the JDC problem. Generally, $A$ can be estimated either indirectly or directly [4]:

1) Indirect algorithms, such as JAD [3], FFDIAG [22], QDIAG [18], LUJD [1], FLEXJD [21], J-DI [14] and CVFFDIAG [19], estimate $A$ from the inverse of a transformation matrix $V$, which minimizes the non-diagonal parts of $VC^{(k)}V^T$ using the following criterion:

$$U(V) = \sum_{k=1}^{K} \| V C^{(k)}V^T - \text{diag} \left( VC^{(k)}V^T \right) \|_F^2$$  \hspace{1cm} (2)

where $\text{diag}(\cdot)$ eliminates the non-diagonal parts of its input and $\| \cdot \|_F$ denotes the Frobenius norm. These algorithms use additional constraint on $V$, such as orthogonality [3] or unit-determinant [1], in order to avoid the trivial solution $0$. There also exist some other criterions. For example, LUJD [1] minimizes a scale and permutation invariant criterion. An information theoretic criterion is proposed by Pham [12] and adopted by Joho [8] and Todros et al. [16], which requires the target set $C$ to be positive definite. However, the nonnegativity of $A$ can be hardly guaranteed in such approaches.

2) ACDC [20] and the subspace fitting algorithm [17] directly estimate $A$ and the set $D$ in order to best approximate the target set $C$ by minimizing the following criterion:

$$J(A, D) = \sum_{k=1}^{K} \| C^{(k)} - AD^{(k)}A^T \|_F^2$$  \hspace{1cm} (3)

Herein the nonnegativity of $A$ can be enforced explicitly.

In this letter, we present a nonnegative JDC algorithm based on criterion (3), which is committed to seek a nonnegative joint diagonalizer $A$. The nonnegativity constraint is imposed by means of a square change of variable. Then the high-dimensional optimization problem is formulated into several sequential polynomial subproblems using LU matrix factorization. Numerical experiments on simulated matrices emphasize the advantages of the proposed method, especially in the case of degeneracies such as for low Signal to Noise Ratio (SNR) values and a small number $K$ of matrices. An illustration of blind separation of NMR spectroscopy confirms the validity and improvement of the proposed method.

II. NONNEGATIVE JOINT DIAGONALIZATION

A way of including the nonnegativity constraint in (3) is through a square change of variable $A = B^{1/2}$ with $B \in \mathbb{R}^{N \times P}$ as originally proposed in [5] for Nonnegative Matrix Factorization (NMF), where $\prec$ stands for Hadamard product. Then (3) can be reformulated as the following unconstrained problem:

$$J(B, D) = \sum_{k=1}^{K} \| C^{(k)} - B^{1/2}D^{(k)}B^{1/2} \|_F^2$$  \hspace{1cm} (4)
Minimizing (4) is the main purpose of this paper. To estimate \( \mathbf{B} \), based on LU matrix factorization, the high dimensional optimization is reduced to search a sequence of sparse triangular matrices. Let’s recall the following definition:

**Definition 1:** A unit triangular matrix is a triangular matrix whose main diagonal elements are equal to 1.

Definition 1: An elementary triangular matrix \( T^{(i,j)}(a) \) is a unit triangular matrix whose non-diagonal elements are zeros except the \((i,j)\)-th entry, which is equal to \( a \).

**Lemma 1:** Any unit lower (or upper) triangular matrix belonging to \( \mathbb{R}^{N \times F} \) can be factorized as a product of \((2N - F - 1)/2\) elementary lower (or upper, respectively) triangular matrices.

The proof of lemma 1 is straightforward by reducing \( \mathbf{L} \) (or \( \mathbf{U} \)) into an elementary triangular matrix whose all non-diagonal elements are zeros, using Gaussian elimination. Lemma 1 yields that \( \mathbf{B} \) can be written as a product of elementary triangular matrices:

\[
\begin{align*}
\mathbf{B} &= \prod_{j=1}^{N} \prod_{i=j+1}^{N} \mathbf{L}^{(i,j)}(t_{i,j}) \prod_{i=1}^{P} \prod_{j=i+1}^{P} \mathbf{U}^{(i,j)}(u_{i,j}) \\
&= \prod_{i,j \in \mathcal{D}} T^{(i,j)}(t_{i,j}) 
\end{align*}
\]

(5)

where \( T^{(i,j)}(t_{i,j}) \) is defined for the sake of convenience: it stands for \( \mathbf{L}^{(i,j)}(t_{i,j}) \) when \( i > j \) and for \( \mathbf{U}^{(i,j)}(u_{i,j}) \) when \( i < j \) with \( M = F(N-1) \) being the total number of elementary lower and upper triangular matrices. The sizes of all the \( \mathbf{L}^{(i,j)}(t_{i,j}) \) and \( \mathbf{U}^{(i,j)}(u_{i,j}) \) matrices are \((N \times N)\) and \((P \times P)\), respectively, except that of \( \mathbf{L}^{(N,i)}(t_{N,i}) \), which is \((N \times P)\). Ideally the minimization of (4) consists of finding \( M \) matrices \( T^{(i,j)}(t_{i,j}) \) and the diagonal matrix set \( \mathbf{D} \) such that \( J(\prod_{i,j \in \mathcal{D}} T^{(i,j)}(t_{i,j}), \mathbf{D}) \) is a global minimum. As Yeredor’s ACDC (Alternating Columns and Diagonal Centers) algorithm [20], the algorithm proposed in this paper alternates between two minimization schemes:

- **The AC (“Alternating Columns”) phase** minimizes (4) with respect to \( \{U_{i,j}\} \) of \( \mathbf{B} \), while keeping \( \{L_{i,j}\} \) fixed.
- **The DC (“Diagonal Centers”) phase** minimizes (4) w.r.t the diagonal matrix set \( \mathbf{D} \) while keeping \( \mathbf{B} \) fixed.

Hence, we name the proposed algorithm ACDC\(_{\mathcal{L},\mathcal{U}}\).

### A. AC Phase

In this phase, we minimize (4) w.r.t \( T^{(i,j)}(t_{i,j}) \). Instead of simultaneously computing these \( M \) matrices, each \( T^{(i,j)}(t_{i,j}) \) with a selected \((i,j)\)-index is identified sequentially. Suppose that \( \mathbf{A} \) and \( \mathbf{B} \) are the current estimate of \( \mathbf{A} \) and \( \mathbf{B} \), respectively. The update of \( \mathbf{B} \) by one \( T^{(i,j)}(t_{i,j}) \) is \( \mathbf{BT}^{(i,j)}(t_{i,j}) \). Then we can show the following lemma:

**Lemma 2:** The Hadamard square of the update is given by:

\[
\left( \mathbf{BT}^{(i,j)}(t_{i,j}) \right)^{\text{HS}} = \left( \mathbf{B}^{\text{HS}} \right) \mathbf{T}^{(i,j)}(t_{i,j}) + 2t_{i,j} (\mathbf{b} \square \mathbf{b}) e_j^T \quad (6)
\]

where \( \mathbf{b} \) and \( \mathbf{b} \) denote the \( i \)-th and \( j \)-th columns of \( \mathbf{B} \) respectively, and \( e_j \) is the \( j \)-th column of the identity matrix.

The proof of lemma 2 is omitted due to lack of space. The optimization of (4) w.r.t \( T^{(i,j)}(t_{i,j}) \) only consists of minimizing the following cost function:

\[
J_{\mathcal{AC}}(t_{i,j}) = \sum_{k=1}^{K} \left\| \mathbf{C}^{(k)} - \left( \left( \mathbf{BT}^{(i,j)}(t_{i,j}) \right)^{\text{HS}} \right)^{D^{(k)}} \right\|_F^2 \quad (7)
\]

Inserting (6) into (7), the only unknown variable \( t_{i,j} \) can be computed by minimizing the following cost function:

\[
J_{\mathcal{AC}}(t_{i,j}) = \sum_{k=1}^{K} \left\| \mathbf{F}^{(k)}_d t_{i,j} + \mathbf{F}^{(k)}_3 \mathbf{t}_{i,j} + \mathbf{F}^{(k)}_2 t_{i,j}^2 \\
+ \mathbf{F}^{(k)}_1 t_{i,j} + \mathbf{F}^{(k)}_0 \right\|_F^2 \quad (8)
\]

with:

\[
\begin{align*}
\mathbf{F}^{(k)}_d &= -d^{(k)}_j \left[ \mathbf{b} \square \mathbf{b}^T \right] \\
\mathbf{F}^{(k)}_3 &= -2d^{(k)}_j \left[ \mathbf{b} \square \mathbf{b}^T + \mathbf{b} \square \mathbf{b}^T \right] \\
\mathbf{F}^{(k)}_2 &= -d^{(k)}_j \left[ \mathbf{b} \square \mathbf{b}^T + \mathbf{b} \square \mathbf{b}^T \right] + 4d^{(k)}_j \mathbf{b} \square \mathbf{b}^T \\
\mathbf{F}^{(k)}_1 &= -2d^{(k)}_j \left[ \mathbf{b} \square \mathbf{b}^T + \mathbf{b} \square \mathbf{b}^T \right] + 4d^{(k)}_j \mathbf{b} \square \mathbf{b}^T \\
\mathbf{F}^{(k)}_0 &= \mathbf{C}^{(k)} - \left( \mathbf{B}^{\text{HS}} \right) \mathbf{D}^{(k)} \mathbf{B}^{\text{HS}}^T
\end{align*}
\]

(9)

where \( d^{(k)}_j \) denotes the \((j, j)\)-th entry of \( \mathbf{D}^{(k)} \). Equation (8) shows that \( J_{\mathcal{AC}}(t_{i,j}) \) is a 8-th degree polynomial in \( t_{i,j} \). The global minimum \( t_{i,j} \) can be obtained by computing the roots of its derivative and selecting the root yielding the smallest value of \( J_{\mathcal{AC}}(t_{i,j}) \). Once the optimal \( t_{i,j} \) is computed, the matrix \( \mathbf{B} \) is updated by computing \( \mathbf{BT}^{(i,j)}(t_{i,j}) \).

However, it is observed that if the matrix set \( \mathbf{C} \) well fulfills the model (1), classical non-constrained JDC methods can give nonnegative \( \mathbf{A} \). In this situation, imposing the nonnegativity constraint explicitly is unnecessary and increases the computational burden. Therefore, the nonnegativity constraint can be relaxed by decomposing \( \mathbf{A} = \prod_{i,j \in \mathcal{L}} T^{(i,j)}(t_{i,j}) \) directly instead of using \( \mathbf{B} \). Each parameter \( t_{i,j} \) can be found by minimizing the following 4-th degree polynomial:

\[
J_{\mathcal{AC}}(t_{i,j}) = \sum_{k=1}^{K} \left\| \mathbf{F}^{(k)}_d t_{i,j}^2 + \mathbf{F}^{(k)}_2 t_{i,j} + \mathbf{F}^{(k)}_1 t_{i,j} \right\|_F^2 \quad (10)
\]

In practice, it is suggested to compute \( t_{i,j} \) by minimizing (10) first. If all the elements in the \( j \)-th column of \( \mathbf{AT}^{(i,j)}(t_{i,j}) \) have the same sign, the update \( \mathbf{A} = \left| \mathbf{AT}^{(i,j)}(t_{i,j}) \right| \) is adopted, where \( | \cdot | \) denotes the absolute value. Otherwise, \( t_{i,j} \) is computed by minimizing (8). Then the AC procedure is repeated to estimate
with the next \((i, j)\) index. The processing of all the \(M\) factors is called a full AC iteration.

B. DC Phase

In this phase, the minimization of (4) w.r.t the diagonal matrices \((D^{(k)})_{k=1}^K\) can be separated into \(K\) distinct linear least square subproblems:

\[
J_{DC}^k(D^{(k)}) = \left\| C^{(k)} - AD^{(k)}A^T \right\|^2_F \tag{11}
\]

The optimal solution of \(D^{(k)}\) given by [20] is:

\[
D^{(k)} = \text{diag}\left( (H^T H)^{-1} H^T e^{(k)} \right) \tag{12}
\]

where \(H = (A^T A)^{1/2}, A = [B]^{1/2}\) and \(e^{(k)} = \text{vec}(C^{(k)})\), \(\text{vec}(\cdot)\) converts a matrix into a column vector by stacking its columns vertically.

In this context, one sweep is referred to the combination of one full AC iteration and one DC phase. The overall numerical complexity per sweep is \(O(N^3PK)\), which is higher than \(O(N^2PK)\) of ACDC [20]. In practice, several sweeps are necessary to ensure convergence. One can stop the algorithm when the value of \(J(B, D)\), or the decrease of \(J(B, D)\) between two successive sweeps falls below a fixed positive threshold. Such a stopping criterion is guaranteed to be met since \(J(B, D)\) is non-increasing in each sweep. We observed empirically that the proposed algorithm converges linearly.

III. SIMULATION RESULTS

In this section, the proposed ACDC\(_{LU}\) method is compared with several state-of-the-art JDC methods and BSS algorithms. The performance is measured in terms of the error between the true joint diagonalizer \(A\) and its estimate \(\hat{A}\), as well as the source \(s\) and its estimate \(\hat{s}\) when a BSS context is considered. So the scale and permutation invariant distance defined in [7] is chosen as the preferred measure:

\[
D(A, \hat{A}) = \min_{\Pi} \Psi(A, \Pi \hat{A}) \tag{13}
\]

where the solution (13) requires to sweep all the \((P \times P)\) permutation matrices \(\Pi\), and:

\[
\Psi(M, \hat{M}) = \frac{1}{P} \sum_{p=1}^P \left| m_p - \frac{\hat{m}_p^T m_p}{\hat{m}_p} \right| \tag{14}
\]

with \(m_p\) and \(\hat{m}_p\) denoting the \(p\)-th columns of \(M\) and \(\hat{M}\), respectively. Moreover, we repeat all the experiments with 500 independent Monte Carlo trials. All the algorithms stop either when the relative error of the corresponding criterion between two successive sweeps is less than \(10^{-4}\) or when the number of sweeps exceeds 2000. In the following experiments, all the algorithms are initialized by identity matrices. All the simulations are implemented in Matlab v7.14 and run on Intel Duad-Core 2.8 GHz CPU with 32 Gb memory.

A. Simulated JDC Model

In this test, the behavior of the proposed ACDC\(_{LU}\) method is evaluated and compared with several classical nonorthogonal JDC methods, including one direct-type method, namely ACDC [20], and three indirect-type methods, namely FFDIAG [22], LU1J1D [1] and UWedGE [15]. The synthetic matrix set \(C\) is generated randomly according to (1) with \(N = 3, P = 3\) and \(T_{ij}^{(k)}(t, j)\) with the next \((i, j)\) index. The processing of all the \(M\) factors is called a full AC iteration.

B. BSS Application on NMR Spectroscopy Data

In addition to the good performance of the proposed method in simulated JDC model, we further illustrate its capability for BSS. ACDC\(_{LU}\) is compared with three well-known BSS algorithms, namely the ICA methods CoM2 [6] and SOBI [2], and a
a nonnegative joint diagonalizer. Simulation results illustrate its improvement in dealing with degenerate JDC models and its potential usefulness in BSS applications.

### REFERENCES


