Error-controlling algorithm for simultaneous block-diagonalization and its application to independent component analysis

Takanori Maehara and Kazuo Murota

1 Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo 113-8656, Japan
E-mail maehara@misojiro.t.u-tokyo.ac.jp
Received April 14, 2010, Accepted August 28, 2010

Abstract

The finest simultaneous block-diagonalization for a given set of square matrices has been studied independently in the area of independent component analysis (ICA) and semidefinite programming. A new algorithm for this problem, which finds the finest decomposition with a capability of coping with numerical errors, has recently been proposed by the present authors. In this paper we indicate the use of the algorithm for ICA by describing its main features and comparing the method with the other existing methods.

Keywords simultaneous block-diagonalization, independent component analysis, eigenvalue problem

1. Introduction

In this paper, we consider the following problem:

Given a set of \( n \times n \) real matrices \( A_1, \ldots, A_N \), find an \( n \times n \) orthogonal matrix \( P \) such that \( P^\top A_1 P, \ldots, P^\top A_N P \) are in a common block-diagonal form.

Recently, this problem has been studied in the area of signal processing, independent component analysis (ICA). ICA is an effective method for signal processing proposed independently by Ans, Hérault, and Jutten and by Barness, Carlin, and Steinberger in the 1980’s.

Let \( X \) be a (given) \( d \)-dimensional signal. In the multidimensional ICA, proposed by Cardoso [1], we decompose \( X \) into mutually independent signals by finding an invertible (constant) matrix \( W \) and mutually independent (possibly multidimensional) signals \( Y_1, \ldots, Y_m \) such that \( W^{-1}X = (Y_1, \ldots, Y_m)^\top \). There are several methods, to be explained in Section 2, that reduce this task in ICA to simultaneous block-diagonalization of matrices. Standard methods for this reduction are JADE by Cardoso and Souloumiac [2] and SOBI by Belouchrani, Abed-Meraim, Cardoso, and Moulines [3]. For simultaneous block-diagonalization, the Jacobi-like algorithm of Cardoso and Souloumiac [2] is accepted as a standard algorithm in this area; see [4–6] and Section 3.

Methods for simultaneous block-diagonalization are also studied independently in the area of semidefinite programming (SDP), which is a major field of optimization. When the data matrices of SDP are simultaneously block-diagonalized, the associated SDP can be solved efficiently [7–9]. The methods for simultaneous block-diagonalization proposed in this area focus on algebraic structures, such as group symmetry, and find a decomposition using the theory of algebra. In particular, Murota, Kanno, Kojima, and Kojima [10] and Maehara and Murota [11] recently proposed a numerical algorithm, to be called MKKKM algorithm in this paper, for simultaneous block-diagonalization. The main idea of the algorithm is to consider the matrix \( * \)-algebra generated by the given matrices and to use the Artin-Wedderburn type structure theorem for matrix \( * \)-algebra. The algorithm is the first algorithm that finds the finest decomposition of given matrices from the numerical data of the matrices without knowing the underlying group symmetry. The algorithm is highly sensitive to the numerical errors contained in the given matrices and sometimes fails to find reasonable or plausible decompositions, although this is a correct “theoretical” behavior that is consistent with the Artin-Wedderburn theorem applied to matrices contaminated by numerical errors.

Very recently Maehara and Murota [12] have proposed an algorithm for simultaneous block-diagonalization, to be called the MM algorithm in this paper, which can find the finest decomposition and has a capability of controlling numerical errors. The algorithm can be seen as a dual of the MKKKM algorithm, in the sense explained in Section 4. This paper is intended to be a quick report on the use of the algorithm of [12] in the context of ICA.

2. Independent component analysis via simultaneous block-diagonalization

We here review two standard methods, JADE [2] and SOBI [3], that reduce the ICA problem to simultaneous block-diagonalization of matrices. Recall that the ICA problem is: given a \( d \)-dimensional signal \( X \), find an invertible matrix \( W \) and mutually independent signals
The JADE method first normalizes the given signal $X$ to that with zero-mean and unit-matrix variance via affine transformation. This process is called whitening, and is performed by the following algorithm. Here $(\cdot)$ means the expected value, which is replaced by the sample mean in practice.

Algorithm 1 (Whitening)

**Input:** Signal $X$

**Output:** Whitened signal $X$

1. $X \leftarrow X - \langle X \rangle$
2. $X \leftarrow (XX^\top)^{-1/2} X$

After whitening, we can assume that $W$ is an orthogonal matrix. The reason is the following. If some decomposition $Y = W^{-1} X$ with invertible $W$ is obtained, we modify $Y$ and $W$ by multiplying block-diagonal matrix so that $Y$ has unit-matrix variance. Then we have $W^{-1} W^\top = (W^{-1} X X^\top W^\top)^{-1} = I$.

After whitening, JADE considers the fourth-order cumulant matrices $C_{ij}$ $(i,j = 1, \ldots, d)$ of $X$, where each $C_{ij}$ is a $d \times d$ matrix whose $(k,l)$ entry is defined as

$$ (C_{ij})_{kl} = \langle X_i X_j X_k X_l \rangle - \langle X_i X_j \rangle \langle X_k X_l \rangle - \langle X_i X_k \rangle \langle X_j X_l \rangle + \langle X_i X_l \rangle \langle X_j X_k \rangle $$

for $k,l = 1, \ldots, d$. The matrices have the property that the $(k,l)$ entry of $C_{ij}$ is zero for $i,j = 1, \ldots, d$ if $X_k$ and $X_l$ are contained in different independent components. Accordingly, if these matrices are brought to a common block-diagonal form with $m$ diagonal blocks, it is understood that $X$ is decomposed into $m$ independent components. On the basis of this fact JADE reduces the ICA problem to simultaneous block-diagonalization of the $N = d^2$ fourth-order cumulant matrices $C_{ij}$.

The SOBI method also preprocesses the given signal by whitening. Then the method considers the time-delayed autocorrelation matrices

$$ R(t, \tau) = \langle X(t) X(t + \tau)^\top \rangle $$

with some $t$ and $\tau$. Here $X(t)$ denotes the delayed signal of $X$ with delay time $t$. The matrices also have the property that the $(k,l)$ entry of $R(t, \tau)$ is zero if $X_k$ and $X_l$ are mutually independent. This property allows the SOBI method to reduce the ICA problem to the simultaneous block-diagonalization of the matrices $R(t, \tau)$.

Here we emphasize that both methods need to compute the expectations but, in practice, we can only compute approximate values of the expectations. As a consequence the matrices generated by either method are not exactly decomposable in the algebraic sense. Hence a method of block-diagonalization with some error-controlling mechanism is needed.

3. Jacobi-like algorithm

Cardoso and Souloumiac’s algorithm [2] for simultaneous block-diagonalization is an extension of the Jacobi algorithm for eigenvalue decomposition. It applies successive Givens rotations to the given matrices until the sum of squares of off-diagonals, which denotes a diagonality criterion, becomes minimal. More concretely, let $R(i, j, \theta)$ be the Givens rotation [13] with $(i, j)$ plane with angle $\theta$ and let $\text{off}(A) = \sum_{i \neq j} a_{ij}$ for a matrix $A = (a_{ij})$. The algorithm for $A_1, \ldots, A_N$ is described as follows.

Algorithm 2 (Jacobi-like algorithm [2])

**Input:** $n \times n$ matrices $A_1, \ldots, A_N$

**Output:** block-diagonalized matrices $A_1, \ldots, A_N$

1. **repeat**
2. Find a Givens rotation $R = R(i, j, \theta)$ which minimizes $\sum_{k=1}^N \text{off}(R^\top A_k R)$
3. $A_k \leftarrow R^\top A_k R$ $(k = 1, \ldots, N)$
4. **until** convergence

Step 2 is performed as follows. For each $(i,j)$ compute the optimal Givens angle $\theta$ that results in the largest decrement of $\sum_k \text{off}(R^\top A_k R)$ and then choose the best $(i,j)$. The closed forms of the Givens angle and the decrement are given in [5].

It is easily seen that if the algorithm converges and $\sum_{k=1}^N \text{off}(A_k) = 0$, then $A_1, \ldots, A_N$ are decomposed to a simultaneous diagonal form. Conversely, it is proved by Bunse-Gerstner, Byers, and Mehrmann [4] that if the matrices $A_1, \ldots, A_N$ can be decomposed into a simultaneous diagonal form, this algorithm locally converges to $\sum_{k=1}^N \text{off}(A_k) = 0$. However, global convergence of the algorithm is still open.

This algorithm, originally proposed for simultaneous diagonalization [2], can also be used for simultaneous block-diagonalization [1]. Recently, Theis [6] afforded a reasonable theoretical basis for the method although the algorithm is not rigorously guaranteed to find the finest decomposition.

4. MM algorithm

The error-controlling algorithm of [12] for simultaneous block-diagonalization is described. A set $\mathcal{T}$ of $n \times n$ matrices is said to be a matrix $*$-algebra if $\mathcal{T}$ contains the identity matrix and is closed under sum, product, transpose, and scalar product. Let $\mathcal{T}'$ denote the commutant algebra of a matrix $*$-algebra $\mathcal{T}$, which is the set of all matrices that commute with all elements in $\mathcal{T}$, i.e.,

$$ \mathcal{T}' = \{ X \mid AX = XA = O \ (\forall A \in \mathcal{T}) \}.$$ 

The outline of the algorithm in [12] is the following. Consider the matrix $*$-algebra $\mathcal{T}$ generated by the given matrices $A_1, \ldots, A_N$. Sample a generic symmetric matrix $X$ from the commutant algebra $\mathcal{T}'$ and output an orthogonal matrix $P$ that diagonalizes $X$. An error-control parameter $\epsilon$ is introduced for robustness against numerical errors. The whole algorithm reads as follows.

Algorithm 3 (MM algorithm [12])

**Input:** $n \times n$ matrices $A_1, \ldots, A_N$

**Output:** Orthogonal matrix $P$ that diagonalizes input matrices

1. Set error-control parameter $\epsilon \geq 0$.
2. Sample a symmetric matrix $X$ with $\|A_k X - X A_k\| \leq \epsilon$ for $k = 1, \ldots, N$.
3. Find an orthogonal matrix $P$ that diagonalizes $X$, and output $P$.

The MM algorithm can be seen as a dual version of the...
MKKKM algorithm in the sense that the MKKKM algorithm computes an orthogonal matrix that diagonalizes a randomly sampled symmetric matrix in $T$ itself, and not in its commutant $T'$. Once the orthogonal matrix $P$ is found, the block-diagonal decomposition can be determined from the zero-nonzero pattern of the off-diagonal entries of $P^T A_k P$ for $k = 1, \ldots, N$.

The correctness of the MM algorithm can be stated as follows.

**Theorem 4 (Maehara and Murota [12])** If $\epsilon$ is set to zero, the orthogonal matrix $P$ found by Algorithm 3 gives the finest block-diagonal decomposition of $A_1, \ldots, A_N$. If $\epsilon$ is set to nonzero, the algorithm gives an error-controlled block-diagonal decomposition in the sense that the off-block-diagonal entries are of the order of $\epsilon$.

The proof of this theorem relies on some algebraic facts about matrix $*$-algebras and their commutant algebras, and also on the following linear-algebraic fact.

**Lemma 5 (Maehara and Murota [12])** Let $A$ be an $n \times n$ matrix and $X$ be an $n \times n$ symmetric matrix. If $\|AX - XA\| \leq \epsilon$, then, for any orthogonal matrix $P$ that diagonalizes $X$ as $P^T X P = \text{diag}(\lambda_1, \ldots, \lambda_n)$, we have

$$|(P^T A P)_{ij}| \cdot |\lambda_i - \lambda_j| \leq \epsilon.$$ 

Step 2 of Algorithm 3 is performed as follows. Since the expression $A_k X - X A_k$ is linear in the entries $x_{ij}$ of $X$, there exists an $n^2 \times n^2$ matrix $T_k$ such that $\text{vec}(A_k X - X A_k) = T_k \text{vec}(X)$, where $\text{vec}(X) = (x_{11}, x_{21}, \ldots, x_{nn})^\top$ is the vectorization operation. We can show that $T_k = I \otimes A_k - A_k^\top \otimes I$. Let

$$S = \sum_{k=1}^N (T_k^T T_k + T_k T_k^T).$$ 

The matrix $S$ has the following property.

**Proposition 6** Let $S$ be the matrix in (3). Let $u$ be a vector with $\|u\| = 1$ and $X$ be a matrix with $\text{vec}(X) = u$.

(a) If $u^T S u = 0$, then $A_k X - X A_k = O$ and $A_k^T X - X A_k^T = O$ ($k = 1, \ldots, N$).

(b) If $u^T S u \leq \epsilon^2$ then $\|A_k X - X A_k\| \leq \epsilon$ and $\|A_k^T X - X A_k^T\| \leq \epsilon$ ($k = 1, \ldots, N$).

A vector $u$ in Proposition 6-(b) can be found by the eigenvalue decomposition of $S$. The algorithm for Step 2 is given as follows.

**Algorithm 7 (Sample a symmetric matrix [12])**

**Input:** $n \times n$ matrices $A_1, \ldots, A_N$

**Output:** A symmetric matrix $X$ for Step 2 in Algorithm 3

1. Construct the matrix $S$ of (3) from $A_1, \ldots, A_N$.

2. Find normalized eigenvectors, say, $v_1, \ldots, v_r$, of $S$ that correspond to eigenvalues smaller than $\epsilon^2$.

3. Sample real numbers $c_1, \ldots, c_r$ randomly subject to $c_1^2 + \cdots + c_r^2 = 1$.

4. Put $u = c_1 v_1 + \cdots + c_r v_r$, let $X$ be the matrix such that $\text{vec}(X) = u$, and output $X = (X^T + X)/2$.

When the error-control parameter $\epsilon$ in Step 1 is not prescribed, a reasonable value of $\epsilon$ can be chosen from the eigenvalues of the matrix $S$. Suppose that the given matrices $A_1, \ldots, A_N$ can be regarded as being perturbed from the nominal values, say $\overline{A}_1, \ldots, \overline{A}_N$, and let $\overline{S}$ be the corresponding nominal values of the matrix $S$. By Proposition 6, we are to identify the null space of $\overline{S}$, which may be approximated by subspace spanned by the eigenvectors of $\overline{S}$ that correspond to small eigenvalues. Therefore, as a reasonable choice of $\epsilon$, we can adopt a threshold value that separates nearly zero eigenvalues of $S$ from nonzero eigenvalues.

5. **Numerical results**

Here we work with a problem of ICA, which is a standard setting in the area of ICA. Let $Y_1, Y_2$, and $Y_3$ be 2-dimensional signals of length $T = 10000$, representing three sequences of $(x, y)$-coordinates of $T = 10000$ points that correspond to the characters “R”, “C”, and “H” shown in Fig. 1-(a). Let $Y_4$ and $Y_5$ be 1-dimensional Gaussian noises of length $T = 10000$ drawn from $N(50, 50)$ independently. It is assumed that $Y_1, \ldots, Y_5$ are mutually independent. Let $W$ be a fixed random matrix and $X = W(Y_1, Y_2, Y_3, Y_4, Y_5)^\top$, which is 8-dimensional signal of length 10000. The ICA problem is to obtain $Y_1, Y_2$, and $Y_3$ from $X$; we have $d = 8, m = 5$ in the notation of Section 2.

According to the JADE method, we consider the fourth-order cumulant matrices $A_1, \ldots, A_N$, where $N = 64$ and each $A_k$ is of size $8 \times 8$. We used Cardoso's implementation for building these matrices. To decompose these matrices we apply the MM algorithm (Algorithm 3). We plot the eigenvalues of the matrix $S$ (see Fig. 2) to set $\epsilon = 1$ as the error-control parameter.

The signals decomposed by the MM algorithm are depicted in Fig. 1-(b). The signals obtained using the Jacobi-like algorithm are shown in Fig. 1-(c) and those using the MKKKM algorithm are in Fig. 1-(d). It is legitimate that the obtained signals are rotated or reflected. The MM algorithm and the Jacobi-like algorithm afford results of comparable quality, much sharper than the outputs of the MKKKM algorithm.

To see the difference between the MM algorithm and the Jacobi-like algorithm we work on a noisy ICA problem which is generated by adding noise to the above example. (more specifically, adding random matrix drawn from $N(0, 0.03)$ to the fourth cumulant matrices.) The results are shown in Fig. 3. It can be observed that the MM algorithm gives a slightly sharper result than the Jacobi-like algorithm. (We here omit the result of the MKKKM algorithm because it does not produce any reasonable result for this problem.) We remark that if larger noises, e.g., drawn from $N(0, 0.08)$, are added, the MM algorithm does not recover the original image.

In addition, the MM algorithm is equipped with the theoretical performance supports explained Section 4. The MM algorithm is thus a promising candidate for practical use in ICA.

**Acknowledgments**

This work is supported by the Global COE “The Research and Training Center for New Development in Mathematics”. 

- 133 –
(a) Sources $S_1$, $S_2$, and $S_3$.

(b) MM algorithm.

(c) Jacobi-like algorithm.

(d) MKKKM algorithm.

Fig. 1. The scatter plots of the signals for an ICA problem.

Fig. 2. Sorted eigenvalues of $S$ for the ICA problem. The broken line is a threshold of $\epsilon = 1$. Note that if we set $\epsilon = 0.2$ to split $\sigma_9$, the MM algorithm also recovers the original images.

Fig. 3. The scatter plots of the signals for a noisy ICA problem.

Fig. 4. Sorted eigenvalues of $S$ for a noisy ICA problem. The broken line is a threshold of $\epsilon = 1$.

References


