



HAL
open science

CorrIndex: a permutation invariant performance index

Elaheh Sobhani, Pierre Comon, Christian Jutten, Massoud Babaie-Zadeh

► **To cite this version:**

Elaheh Sobhani, Pierre Comon, Christian Jutten, Massoud Babaie-Zadeh. CorrIndex: a permutation invariant performance index. *Signal Processing*, 2022, 195, pp.108457. 10.1016/j.sigpro.2022.108457 . hal-03230210v2

HAL Id: hal-03230210

<https://hal.science/hal-03230210v2>

Submitted on 23 Jan 2022

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.

CorrIndex: a permutation invariant performance index

Elaheh Sobhani^{a,b,*}, Pierre Comon^a, Christian Jutten^a,
Massoud Babaie-Zadeh^b

^a*GIPSA-Lab, UMR CNRS 5216, Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab,
38000 Grenoble, France*

^b*Department of Electrical Engineering, Sharif University of Technology, Tehran
11365-11155, Iran*

Abstract

Permutation and scaling ambiguities are relevant issues in tensor decomposition and source separation algorithms. Although these ambiguities are inevitable when working on real data sets, it is preferred to eliminate these uncertainties for evaluating algorithms on synthetic data sets. As shown in the paper, the existing performance indices for this purpose are either greedy and unreliable or computationally costly. In this paper, we propose a new performance index, called CorrIndex, whose reliability can be proved theoretically. Moreover, compared to previous performance indices, it has a low computational cost. Theoretical results and computer experiments demonstrate these advantages of CorrIndex compared to other indices.

Keywords: Assignment, Performance index, Permutation and scaling ambiguity, Blind Source Separation, ICA, Tensor decomposition, Maximum Weighted Matching, Hungarian algorithm, Optimal transport

*Corresponding author

Email addresses:

elaheh.sobhani@gipsa-lab.grenoble-inp.fr, sobhani.es@gmail.com; EURASIP member (Elaheh Sobhani), pierre.comon@gipsa-lab.grenoble-inp.fr; EURASIP member (Pierre Comon), christian.jutten@gipsa-lab.grenoble-inp.fr; EURASIP member (Christian Jutten), mbzadeh@sharif.edu; EURASIP member (Massoud Babaie-Zadeh)

1. Introduction

Permutation and scaling ambiguities are relevant issues in some applications such as tensor decomposition [1] and Blind Source Separation (BSS) [2]. By scaling, we mean multiplication by diagonal matrix with non-zero entries, which may be complex in the most general case. Firstly, these two ambiguities are inherent in tensor representations, by definition of tensors [1]. Secondly, in BSS, statistical independence is not affected by scaling or permutation of the sources [2]. A mixing (or demixing) matrix can then only be estimated up to these ambiguities under the independence assumption. Although it is impossible to eliminate these ambiguities when working with real data sets, where the original parameters are not available, it is feasible to get rid of these uncertainties in evaluating algorithm performance on synthetic data sets. Furthermore, reasonable comparisons on synthetic data sets are very helpful to choose adequately an appropriate algorithm to be applied on real data sets. Therefore, in order to report reasonably the performance indices of existing algorithms on synthetic data sets where the desired parameters are accessible, it is important to employ proper methods to measure the performances.

Assume that the original and estimated components have been normalized, then the only remaining ambiguities are the permutation and scaling with complex numbers of unit modulus. The existing approaches to measure the performances of the algorithms of BSS and tensor decomposition can be classified in three main categories: “greedy approaches”, “graph-based methods” and “invariant indices”. Greedy approaches [3, 4, 5, 6] try to assign the most correlated components estimated by an algorithm, and then compute the error of estimation or decomposition. Although most of these methods return back an estimated permutation as well as a performance index, they are not reliable in noisy conditions. In other words, the reported index by these kinds of methods depends directly on the manner of computing and analyzing the correlation matrix.

Graph-based methods [7, 8, 9] are originated from the well-known *optimal*

assignment problem [10], which is itself a particular case of the *optimal transport* problem [11]. Although these kinds of methods have the guarantee to find the optimal permutation, they are computationally expensive (as we shall see, the minimum cost is $\mathcal{O}(8N^3)$ with a correlation matrix of size $N \times N$), especially
 35 when the correlation matrix is large.

However, the viewpoint of a third category, namely invariant indices [12, 13, 14], differs from the latter approaches. These *invariant* indices measure the performance regardless of permutation and scaling, and yield an index that can directly be used to compare algorithms. The reported indices of [12, 13, 14] are
 40 invariant to permutation and scaling, and the index of [12] provides the guarantee of a zero distance between estimated and original matrices up to column permutation and scaling, when the obtained index is zero. Nevertheless, the index of [12] is not bounded from above. More importantly, the upper bounds of indices of [13, 14] have not been investigated, and it seems that these bounds are
 45 not easy to interpret. In addition, these methods are in the literature of source separation, and the indices introduced therein utilize the inverse (or pseudo-inverse) of the mixing matrix, which may involve an additional computational burden.

In this paper, we studied critically other performance indices to point out
 50 their drawbacks such as being optimistic or pessimistic. Moreover, we introduce a new performance index, called *CorrIndex*, which can be considered to belong to the category of “invariant indices”. *CorrIndex* is based on some correlation matrix (in fact scalar products), and hence, it does not lead to a high computational cost since matrix inversion is avoided. In addition, not only *CorrIndex*
 55 is invariant, but also it provides an interpretable upper bound and guarantees a zero *distance* up to column permutation and scaling if *CorrIndex* = 0 (a more formal definition of this distance will be subsequently given). Hence, compared to greedy methods, *CorrIndex* is more reliable. Moreover, compared to graph-based methods and other invariant indices, it requires the lowest computational
 60 cost. More importantly, *CorrIndex* is not limited to BSS after pre-whitening: it applies to tensor decompositions as well, contrary to other similar indices

proposed in [12, 13, 14].

This paper is organized as follows. In Section 2, the problem is formulated and a brief review of previous methods and indices is provided. The proposed
65 index, CorrIndex, and its comparison with other existing indices is presented in
Section 3. Section 4 proposes experimental results with discussion. Eventually,
the remarks of Section 5 concludes the paper.

2. A critical survey

Let $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N] \in \mathbb{C}^{M \times N}$ and $\widehat{\mathbf{A}} = [\widehat{\mathbf{a}}_1, \widehat{\mathbf{a}}_2, \dots, \widehat{\mathbf{a}}_N] \in \mathbb{C}^{M \times N}$ be
70 the original and estimated matrices respectively, where $\mathbb{C}^{M \times N}$ stands for the
set of M by N complex-valued matrices. Let us denote the set of permutations
of N elements by $\text{Perm}(N)$, and denote by \mathbf{P}_σ the matrix associated with the
permutation $\sigma \in \text{Perm}(N)$. If the columns of \mathbf{A} and $\widehat{\mathbf{A}}$ are normalized by their
 L_2 norms, scaling ambiguity reduces to post-multiplication by a diagonal matrix
75 $\mathbf{\Lambda}$ with entries of unit modulus.

Assume $\widehat{\mathbf{A}} = \mathbf{A}\mathbf{P}_\sigma\mathbf{\Lambda} + \mathbf{W}$, where the columns of \mathbf{A} and $\widehat{\mathbf{A}}$ are normalized,
 $\mathbf{\Lambda}$ is a diagonal matrix with unit modulus entries and \mathbf{W} is an additive noise.
More formally, the goal is to measure the gap defined below:

$$\epsilon_0(\mathbf{A}, \widehat{\mathbf{A}}) = \min_{\sigma, \mathbf{\Lambda}} \|\mathbf{A}\mathbf{P}_\sigma\mathbf{\Lambda} - \widehat{\mathbf{A}}\|_F^2 \quad (1)$$

This gap can be computed with or without estimating permutation σ explicitly.
Seeking the optimal permutation σ can be written as the following optimization
problem:

$$\underset{\sigma}{\operatorname{argmin}} \frac{1}{2} \sum_{n=1}^N \|\mathbf{a}_n - \widehat{\mathbf{a}}_{\sigma(n)}\|_2^2 = \underset{\sigma}{\operatorname{argmax}} \sum_{n=1}^N |\mathbf{a}_n^H \widehat{\mathbf{a}}_{\sigma(n)}|. \quad (2)$$

Let $C_{ij} = |\mathbf{a}_i^H \widehat{\mathbf{a}}_j|$, and denote by \mathbf{C} the matrix whose entries are C_{ij} . Then, if
the columns of \mathbf{A} and $\widehat{\mathbf{A}}$ are normalized by their L_2 norms, we have $0 \leq C_{ij} \leq 1$.
In the sequel, three main approaches of measuring the distance between \mathbf{A} and
 $\widehat{\mathbf{A}}$ appeared in the literature are reviewed.

80 *2.1. Methods based on correlation matrix*

2.1.1. Greedy approach of [3]

In this approach, $\hat{\mathbf{a}}_j$ is assigned to \mathbf{a}_i if C_{ij} has the maximum value in the j^{th} column of \mathbf{C} . This straightforward approach has two drawbacks. On one hand, if two or more maximum values occurred in the same row, a reasonable assignment could not be concluded. This happens for instance in far-field antenna array processing when sources are angularly close, in the presence of noise [15].
 85 On the other hand, the delivered index is not reliable, since, even if the index is zero, one cannot guarantee $\hat{\mathbf{A}} = \mathbf{A}\mathbf{P}_\sigma\mathbf{\Lambda}$. The following toy numerical example illustrates this problem.

Assume that in an experiment matrix \mathbf{C} is:

$$\mathbf{C} = \begin{bmatrix} 0.8 & 0.3 & 0.1 \\ 0.85 & 0.9 & 0.5 \\ 0.5 & 0.2 & 0.7 \end{bmatrix}. \quad (3)$$

90 The concluded assignment by this method is $(\hat{\mathbf{a}}_1, \mathbf{a}_2)$, $(\hat{\mathbf{a}}_2, \mathbf{a}_2)$, $(\hat{\mathbf{a}}_3, \mathbf{a}_3)$ which is obviously not acceptable because column \mathbf{a}_2 is selected twice. Computing the square error via $\frac{1}{2} \sum_{n=1}^3 \|\mathbf{a}_n - \hat{\mathbf{a}}_{\sigma(n)}\|_2^2$ by considering the assumption of normalized \mathbf{a}_n and $\hat{\mathbf{a}}_{\sigma(n)}$ with respect to L_2 norm and by substituting the values of $|\mathbf{a}_i^H \hat{\mathbf{a}}_j|$ from C_{ij} , one obtains $3 - 0.85 - 0.9 - 0.7 = 0.55$, which is less than the
 95 exact error, $3 - 0.8 - 0.9 - 0.7 = 0.60$ (the exact error is given in Section 2.3 with the optimal permutation). This example shows that this algorithm outputs a matrix \mathbf{P} that may not be a permutation.

This index is always *optimistic* since it searches in a set of assignments larger than $\text{Perm}(N)$. In fact, if a set \mathcal{A} contains a set \mathcal{B} , *i.e.* $\mathcal{B} \subseteq \mathcal{A}$, then

$$\min_{x \in \mathcal{A}} f(x) \leq \min_{x \in \mathcal{B}} f(x) \quad (4)$$

for any function $f(x)$. Therefore, the reported error is always smaller than or equal to the exact error based on the optimal assignment.

100 2.1.2. Greedy approach of [4, 5]

In order to avoid a non-acceptable assignment, after detecting the maximum value of each column of \mathbf{C} , its row and column can be removed for the rest of the algorithm. In other words, if in j^{th} column of matrix \mathbf{C} , C_{ij} is the maximum value, then the i^{th} row and j^{th} column of \mathbf{C} will be ignored in the search of the
 105 next maximum value.

This is a *greedy* approach, since the index depends on the order of choosing the maximum values. For example, if this greedy algorithm is applied on matrix \mathbf{C} expressed in (3), the resulted assignment will be $(\hat{\mathbf{a}}_1, \mathbf{a}_2)$, $(\hat{\mathbf{a}}_2, \mathbf{a}_1)$, $(\hat{\mathbf{a}}_3, \mathbf{a}_3)$ provided that the columns are swept from left to right. However, if the columns
 110 are swept in the opposite way, the assignment will be $(\hat{\mathbf{a}}_1, \mathbf{a}_1)$, $(\hat{\mathbf{a}}_2, \mathbf{a}_2)$, $(\hat{\mathbf{a}}_3, \mathbf{a}_3)$. Compared to the optimistic index, the error output by this greedy approach by sweeping from left to right, $3 - 0.85 - 0.3 - 0.7 = 1.15$, is larger than the exact error, $3 - 0.8 - 0.9 - 0.7 = 0.60$, while by sweeping from right to left, the reported error equals to the exact error 0.6.

115 By imposing a column ordering, this greedy approach searches a set of assignments smaller than $\text{Perm}(N)$: following (4), one can conclude that the error measurement is always pessimistic. Therefore, the reported error is always larger than or equal to the exact error based on the optimal assignment.

2.1.3. Score measure [6]

This index, which is also known as *congruence* [16], is customized for tensors and is applied to evaluate the performance of a tensor decomposition in terms of estimating all the loading matrices (defined below) together. Let us explain the permutation ambiguity by means of a tensor decomposition example called Canonical Polyadic (CP) [1]. The CP decomposition of a third order tensor of rank 2 admits the following form:

$$\mathcal{T}^{I \times J \times K} = \sum_{r=1}^2 \mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)} \otimes \mathbf{a}_r^{(3)}, \quad (5)$$

120 where \otimes denotes the outer (tensor) product, and $\mathbf{a}_r^{(1)}$, $\mathbf{a}_r^{(2)}$ and $\mathbf{a}_r^{(3)}$ are some vectors of size I, J and K , respectively. Equation (5) can be represented in

a compact form as $\mathcal{T} = \llbracket \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)} \rrbracket$, where $\mathbf{A}^{(i)} = [\mathbf{a}_1^{(i)}, \mathbf{a}_2^{(i)}]$ is called the mode- i loading matrix of \mathcal{T} . Observe that the permuted version of loading matrices, *i.e.* $\mathbf{A}_p^{(i)} = [\mathbf{a}_2^{(i)}, \mathbf{a}_1^{(i)}]$, $i = 1, 2, 3$, results in the same tensor as \mathcal{T} in (5).

The score measure of the tensor $\mathcal{T} = \llbracket \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)} \rrbracket$ is calculated based on the correlation matrix $\mathbf{C} = \mathbf{C}^{(1)} \boxtimes \mathbf{C}^{(2)} \boxtimes \mathbf{C}^{(3)}$, where \boxtimes is the *Hadamard* product (element-wise product) and $C_{ij}^{(k)} \triangleq |\mathbf{a}_i^{(k)\text{H}} \hat{\mathbf{a}}_j^{(k)}|$, $k = 1, 2, 3$. This index is also greedy, since the assignment is concluded based on the maximum values of \mathbf{C} , which have been chosen in a way explained in Section 2.1.2, and the corresponding *score* is an average of these selected values.

2.2. Methods based on graph matching

The optimal assignment (or optimal transport) problem is an old, well-known and fundamental combinatorial optimization problem [8, 9, 11]. The first polynomial time algorithm for optimal assignment problems is the “Hungarian method” [7] also known as “Kuhn-Munkres” [9, 17], and the complexity of the algorithm is $\mathcal{O}(N^4)$ [9]. This algorithm has been employed in [17] for an optimal pairing of the sources in BSS.

The optimal assignment problem can also be considered as a special case of Maximum Weighted Matching (MWM), which is a well-known problem in graph theory, for which several polynomial time algorithms exist [8]. The best exact [10, 18, 19] and approximate [20] MWM algorithms cost $\mathcal{O}(8N^3)$ and $\mathcal{O}(N^2)$, respectively.

2.3. Methods based on optimal permutation

Searching for the optimal permutation σ , *i.e.* for the optimal permutation matrix \mathbf{P}^* , described at the beginning of Section 2, can be viewed as finding some entries of \mathbf{C} such that no pair among them lies in the same row or column, while the sum of these entries is maximum. One can formulate this as the following optimization problem [11]:

$$\mathbf{P}^* = \underset{\mathbf{P} \in \mathbb{R}_+^{N \times N}}{\operatorname{argmin}} \sum_{i,j} D_{ij} P_{ij} \quad \text{s.t.} \quad \mathbf{P} \mathbb{1}_N = \mathbf{P}^\top \mathbb{1}_N = \mathbb{1}_N, \quad (6)$$

where $\mathbf{D} = -\mathbf{C}$, $\mathbb{1}_N$ is a vector of ones of dimension N and the superscript \star denotes the optimal solution. In other words, we look for a *bistochastic* matrix, *i.e.* a square matrix of non-negative real numbers, whose rows and columns have unit L_1 norm [21]. By vectorizing (concatenating columns) \mathbf{P} and \mathbf{D} into vectors \mathbf{d} and \mathbf{p} , (6) can be rewritten in the standard form of linear program [11, Sec. 3.1]:

$$\mathbf{p}^\star = \underset{\mathbf{p} \in \mathbb{R}_+^{N^2}}{\operatorname{argmin}} \mathbf{d}^\top \mathbf{p} \quad \text{s.t.} \quad \mathbf{Q}\mathbf{p} = \mathbb{1}_{2N}, \quad (7)$$

145 where $\mathbf{Q} = [\mathbb{1}_N^\top \boxtimes \mathbf{I}_N, \mathbf{I}_N \boxtimes \mathbb{1}_N^\top]^\top \in \mathbb{R}^{2N \times N^2}$, \mathbf{I}_N and \boxtimes denote the identity matrix of size N and the Kronecker product, respectively. Yet, from Birkhoff's Theorem, the set of bistochastic matrices is a polyhedron¹ whose vertices are permutations [22, Theorem 8.7.1]. On the other hand, a fundamental theorem of linear programming [23, Theorem 2.7] tells that the minimum of a linear
 150 objective in a non-empty polytope (*i.e.* a finite polyhedron) is reached at a vertex of the polytope. This permits to relax the search for a permutation into (6) or (7): in fact, looking for the best bistochastic matrix will eventually yield a permutation and this is the whole power of the method.

For example, by employing MWM or the linear program described above,
 155 the optimal permutation in experiment (3) is the identity matrix.

Recently, by improving some required computational steps, the running time of implementing the linear programming algorithms has been reduced to $\mathcal{O}(q^{2+\frac{1}{6}})$ [24] and $\mathcal{O}(q^{2+\frac{1}{18}})$ [25], where q is the size of unknown vector in the linear programming problem. Therefore, as $q = N^2$ in (7), the lowest complex-
 160 ity to find the optimal permutation of the problem described at the beginning of Section 2 by the means of linear programming is approximately $\mathcal{O}(N^4)$. In addition, in [11, 26], authors proposed sub-optimal solutions, called Auction algorithms, which costs $\mathcal{O}(N^2)$ [26].

¹convex by definition.

2.4. Indices invariant to permutation

165 Methods described in Section 2.1 try first to estimate the permutation, and then measure some distances based on the estimated permutation. As it is seen in Section 2.1.1, these kinds of methods may actually not return a permutation, and there is no guarantee that $\epsilon_0(\mathbf{A}, \widehat{\mathbf{A}}) = 0$ even if the indices they output are zero. Conversely, the algorithms of Section 2.2 behave better (in terms of
170 returning optimal permutation) but may become very costly for large values of N .

However, in the literature of source separation [2], some indices have been proposed to measure the gap (based on their own definition of gap) between original and estimated mixing matrices without needing to find the correspond-
175 ing permutation [12, 13, 14]. Moreover, these indices are zero if and only if $\epsilon_0(\mathbf{A}, \widehat{\mathbf{A}}) = 0$, which offers a valuable guarantee.

The indices proposed in [12, 13, 14] are based on $\mathbf{S} = \mathbf{A}^{-1}\widehat{\mathbf{A}}$ (or $\mathbf{S} = \mathbf{A}^\dagger\widehat{\mathbf{A}}$ for non-square \mathbf{A}). The details of these indices are as follows.

2.4.1. Comon index [12]

Comon's index is a combination of L_1 and L_2 norms, and is calculated as:

$$\epsilon_1(\mathbf{S}) = \sum_{i=1}^N \left| \sum_{j=1}^N |S_{ij}| - 1 \right|^2 + \sum_{j=1}^N \left| \sum_{i=1}^N |S_{ij}| - 1 \right|^2, \\ \sum_{i=1}^N \left| \sum_{j=1}^N |S_{ij}|^2 - 1 \right| + \sum_{j=1}^N \left| \sum_{i=1}^N |S_{ij}|^2 - 1 \right|.$$

180 In [12], it has been proved that ϵ_1 is invariant to permutation, *i.e.* $\epsilon_1(\mathbf{A}, \widehat{\mathbf{A}}) = \epsilon_1(\mathbf{A}, \widehat{\mathbf{A}}\mathbf{P}\mathbf{A})$. Moreover, it has been shown that $\epsilon_1(\mathbf{A}, \widehat{\mathbf{A}}) = 0$ if and only if $\widehat{\mathbf{A}} = \mathbf{A}\mathbf{P}_\sigma\mathbf{A}$, where σ is the optimal permutation. However, ϵ_1 can increase enormously, depending on the values of matrix \mathbf{S} , hence, this index is not bounded from above.

185 *2.4.2. Moreau-Macchi index [13]*

The index proposed in [13] measures a gap between matrix \mathbf{S} and a permutation matrix. It is defined as:

$$\epsilon_2(\mathbf{S}) = \sum_{i=1}^N \left(\sum_{j=1}^N \frac{|S_{ij}|^2}{(\max_k |S_{ik}|)^2} - 1 \right) + \sum_{j=1}^N \left(\sum_{i=1}^N \frac{|S_{ij}|^2}{(\max_k |S_{kj}|)^2} - 1 \right).$$

Dividing by the maximum value (e.g. $(\max_k |S_{ik}|)^2$) provides an upper bound for ϵ_2 unlike ϵ_1 .

2.4.3. Amari index [14]

This performance index takes the form:

$$\epsilon_3(\mathbf{S}) = \sum_{i=1}^N \left(\sum_{j=1}^N \frac{|S_{ij}|}{\max_k |S_{ik}|} - 1 \right) + \sum_{j=1}^N \left(\sum_{i=1}^N \frac{|S_{ij}|}{\max_k |S_{kj}|} - 1 \right).$$

The only difference between Amari and Moreau-Macchi index is the power 2
 190 which exists in ϵ_2 . Therefore, calculating ϵ_3 is less costly compared to ϵ_2 . In addition, as for ϵ_2 , the division by the maximum value (e.g. $\max_k |S_{ik}|$) provides an upper bound for ϵ_3 .

An accurate investigation of indices reviewed in this section reveals that ϵ_1 is not bounded from above. Furthermore, the upper bounds on ϵ_2 and ϵ_3 have not
 195 been studied in [13, 14], so that their upper bound cannot be easily interpreted. In order to obtain interpretable upper bounds and to reduce computational cost, one may think of replacing $\mathbf{S} = \mathbf{A}^\dagger \widehat{\mathbf{A}}$ by \mathbf{C} , but in this case the property that $\epsilon_i = 0$ is equivalent to $\epsilon_0 = 0$, for $i \in \{1, 2, 3\}$ does not hold anymore.

3. Our proposed index: CorrIndex

200 In this section, we introduce, “*CorrIndex*”, which is based on a correlation matrix. Remind that we define $\mathbf{C} = |\mathbf{A}^\dagger \widehat{\mathbf{A}}|$, where $\mathbf{A} \in \mathbb{C}^{M \times N}$ and $\widehat{\mathbf{A}} \in \mathbb{C}^{M \times N}$ and modulus is understood entrywise. In addition, we assume that the columns of \mathbf{A} and $\widehat{\mathbf{A}}$ are normalized by their L_2 norms.

Basically, if $\epsilon_0(\mathbf{A}, \widehat{\mathbf{A}}) = 0$, N entries of \mathbf{C} are one, since $|\mathbf{a}_n| = |\widehat{\mathbf{a}}_n|$ and the columns of \mathbf{A} and $\widehat{\mathbf{A}}$ are normalized to unit L_2 norms. Remember that it is

desired that a performance index is zero if and only if $\epsilon_0(\mathbf{A}, \widehat{\mathbf{A}}) = 0$. In order to satisfy these basic requirements in the matrix case, *i.e.* $M > 1$, CorrIndex is defined as follows:

$$\text{CorrIndex}(\mathbf{C}) = \frac{1}{2N} \left[\sum_{i=1}^N \left| \max_k C_{ik} - 1 \right| + \sum_{j=1}^N \left| \max_k C_{kj} - 1 \right| \right]. \quad (8)$$

The coefficient $\frac{1}{2N}$ keeps CorrIndex values in the range $[0, 1]$. Moreover, modulus operator, *i.e.* $|\cdot|$, helps to guarantee a zero distance between \mathbf{A} and $\widehat{\mathbf{A}}$ if CorrIndex = 0 (cf. Proposition 2). In addition, according to (8), it is obvious that if the distance between \mathbf{A} and $\widehat{\mathbf{A}}$ is zero, then CorrIndex = 0. Therefore, the two requirements mentioned above are simply satisfied by (8). Further, CorrIndex is invariant to permutation and scaling (cf. Proposition 1).

Remark: It can be also observed that CorrIndex is bounded: $0 \leq \text{CorrIndex} \leq 1$. According to (8), unlike ϵ_2 and ϵ_3 , the upper bound of CorrIndex is easier to interpret when $M > 1$, since it is achieved when entries of \mathbf{C} are minimal (*i.e.* the largest possible angular distance between \mathbf{A} and $\widehat{\mathbf{A}}$). In particular, when $M \geq 2N$, $\mathbf{C} = \mathbf{0}$ when all the columns of \mathbf{A} and $\widehat{\mathbf{A}}$ are orthogonal to each other, which yields CorrIndex = 1. Next, as proved below, the zero lower bound is meaningful, since it corresponds to $\epsilon_0 = 0$.

The one-row case: On the other hand in the row vector case, *i.e.* $M = 1$, as CorrIndex is based on (2), we should return back to the basic minimization of finding optimal assignment (σ), which is a restatement of (2) as follows:

$$\underset{\sigma}{\operatorname{argmin}} \frac{1}{2} \sum_{n=1}^N (a_n - \hat{a}_{\sigma(n)})^2. \quad (9)$$

In order to respect (9) and to consider the scaling ambiguity, a new matrix \mathbf{C} with entries $C_{ij} = (|a_i| - |\hat{a}_j|)^2$ is used to define CorrIndex for row vectors:

$$\text{CorrIndex}(\mathbf{C}) = \frac{1}{2N} \left[\sum_{i=1}^N \min_k C_{ik} + \sum_{j=1}^N \min_k C_{kj} \right]. \quad (10)$$

Comparing (8) and (10) reveals that “max” and “1” have been replaced with “min” and “0”, respectively, which helps benefit from the same properties as (8) in the vector case.

220 In the following, it is shown that CorrIndex is invariant to scaling and permutation, *i.e.* $\text{CorrIndex}(\mathbf{A}, \widehat{\mathbf{A}}) = \text{CorrIndex}(\mathbf{A}, \widehat{\mathbf{A}}\mathbf{P}\mathbf{\Lambda})$. Moreover, it is shown that $\text{CorrIndex}(\mathbf{C}) = 0$ if and only if $\widehat{\mathbf{A}} = \mathbf{A}\mathbf{P}_\sigma\mathbf{\Lambda}$, *i.e.* $\epsilon_0(\mathbf{A}, \widehat{\mathbf{A}}) = 0$.

Proposition 1. *CorrIndex is invariant to permutation and scaling:*

$$\text{CorrIndex}(\mathbf{A}, \widehat{\mathbf{A}}) = \text{CorrIndex}(\mathbf{A}\mathbf{P}\mathbf{\Lambda}, \widehat{\mathbf{A}}) = \text{CorrIndex}(\mathbf{A}, \widehat{\mathbf{A}}\mathbf{P}\mathbf{\Lambda}). \quad (11)$$

Proof. Assume that $\mathbf{C}_1 = |\mathbf{A}^H \widehat{\mathbf{A}}|$ and $\mathbf{C}_2 = |(\mathbf{A}\mathbf{P}\mathbf{\Lambda})^H \widehat{\mathbf{A}}|$. As modulus operator
 225 is insensitive to matrix $\mathbf{\Lambda}$, then $\mathbf{C}_2 = \mathbf{P}^H \mathbf{C}_1$, and since CorrIndex is invariant to row permutation according to (8), the proof is complete. The same proof applies to $\mathbf{C}_3 = |\mathbf{A}^H \widehat{\mathbf{A}}\mathbf{P}\mathbf{\Lambda}|$, because of the invariance of CorrIndex to column permutation. \square

Proposition 2. *Suppose that $\mathbf{A} \in \mathbb{C}^{M \times N}$ and $\widehat{\mathbf{A}} \in \mathbb{C}^{M \times N}$. $\text{CorrIndex}(\mathbf{A}, \widehat{\mathbf{A}}) = 0$ if and only if $\widehat{\mathbf{A}}$ can be written as a permuted version of \mathbf{A} :*

$$\text{CorrIndex}(\mathbf{A}, \widehat{\mathbf{A}}) = 0 \iff \widehat{\mathbf{A}} = \mathbf{A}\mathbf{P}_\sigma\mathbf{\Lambda}. \quad (12)$$

Proof. Firstly, if $\widehat{\mathbf{A}} = \mathbf{A}\mathbf{P}_\sigma\mathbf{\Lambda}$, then $\max_k C_{ik} = 1, \forall i$ and $\max_k C_{kj} = 1, \forall j$.
 230 Thus $\text{CorrIndex}(\mathbf{A}, \widehat{\mathbf{A}}) = 0$. Secondly, we prove the converse. If $\text{CorrIndex}(\mathbf{A}, \widehat{\mathbf{A}}) = 0$, then it implies that $\max_k C_{ik} = 1, \forall i$ and $\max_k C_{kj} = 1, \forall j$. From these two equalities, it can be inferred that there is at least one 1 in each column and row of \mathbf{C} . Let us assume $C_{ij} = |\mathbf{a}_i^H \widehat{\mathbf{a}}_j| = 1$. According to the Cauchy–Schwarz inequality and the assumption of normalized columns of \mathbf{A} and $\widehat{\mathbf{A}}$, we have
 235 $|\mathbf{a}_i^H \widehat{\mathbf{a}}_j| \leq \|\mathbf{a}_i\| \|\widehat{\mathbf{a}}_j\|$, where the equality of two sides occurs if and only if $\mathbf{a}_i = \widehat{\mathbf{a}}_j$. Since such a conclusion holds for all other associated pairs of columns of \mathbf{A} and $\widehat{\mathbf{A}}$, therefore, $\widehat{\mathbf{A}} = \mathbf{A}\mathbf{P}_\sigma\mathbf{\Lambda}$. \square

As mentioned before, it is hard to assess the relative error made on loading matrices in tensor decompositions, because of scaling and permutation ambiguities [1]. So as to overcome these ambiguities, we can use CorrIndex as a
 240 performance on estimating loading matrices. However, if we report CorrIndex

on each loading matrix separately, it would be an optimistic index, since implicitly a different permutation would be permitted for each loading matrix.

In order to have a more reliable performance index, we can apply CorrIndex to a matrix \mathbf{X} , built upon loading matrices stacked one below the other. In other words, for the tensor described in (5), CorrIndex($\mathbf{X}, \widehat{\mathbf{X}}$) applies to:

$$\mathbf{X} = \begin{bmatrix} \mathbf{A}^{(1)} \\ \mathbf{A}^{(2)} \\ \mathbf{A}^{(3)} \end{bmatrix}, \quad \widehat{\mathbf{X}} = \begin{bmatrix} \widehat{\mathbf{A}}^{(1)} \\ \widehat{\mathbf{A}}^{(2)} \\ \widehat{\mathbf{A}}^{(3)} \end{bmatrix}.$$

4. Discussion and computer results

245 A multi-aspect comparison between CorrIndex and other reviewed methods has been carried out, and is reported in Table 1, where the methods of Section 2.1 and 2.2 are referred by “Greedy” and “Graph”, respectively. The number of multiplications of each stage, *i.e.* computing the input matrix ($\mathbf{C} = |\mathbf{A}^H \widehat{\mathbf{A}}|$ or $\mathbf{S} = \mathbf{A}^\dagger \widehat{\mathbf{A}}$), estimating the permutation and computing the index, are reported. 250 In addition, the last column of Table 1 (“Significance of upper bound”) indicates if the upper bound makes sense, *i.e.* returning the maximum index value for the largest distance between \mathbf{A} and $\widehat{\mathbf{A}}$. According to Table 1, it is inferred that CorrIndex has the lowest computational complexity compared to the others in terms of the number of multiplications besides its theoretical guarantee, its 255 invariance to permutation and scaling ambiguity and its meaningful bounds.

In the rest of this section, we report either the relative error (for the greedy and graph-based methods) which is defined in (13) or the indices ϵ_i (Moreau-Macchi, Amari, ...). As greedy and graph-based methods estimate the permutation, we report the relative error between normalized \mathbf{A} and $\widehat{\mathbf{A}}$, by means of estimated matrix \mathbf{P} , as follows:

$$\text{relative error} = \frac{\|\mathbf{A} - \widehat{\mathbf{A}}\mathbf{P}\|_F}{\|\mathbf{A}\|_F}, \quad (13)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Remind that \mathbf{P} is aimed at being a permutation matrix, but that it might not be (see Section 2.1.1).

Table 1: Numbers of multiplications of computing each stage of CorrIndex and other methods

Method	\mathbf{C} or \mathbf{S}	Permutation	Index	Significance of upper bound
Greedy	$\mathcal{O}(N^2M)$	0	$\mathcal{O}(2MN)$	No
Graph	$\mathcal{O}(N^2M)$	$\mathcal{O}(8N^3)$	$\mathcal{O}(2MN)$	Yes
Linprog	$\mathcal{O}(N^2M)$	$\mathcal{O}(N^4)$	$\mathcal{O}(2MN)$	Yes
Comon [12]	$\mathcal{O}(11N^3)$	-	$\mathcal{O}(2N^2)$	No
Moreau-Macchi [13]	$\mathcal{O}(11N^3)$	-	$\mathcal{O}(2N^2)$	No
Amari [14]	$\mathcal{O}(11N^3)$	-	$\mathcal{O}(2N)$	No
CorrIndex	$\mathcal{O}(N^2M)$	-	1	Yes

The index and computation time of each index in a numerical experiment is reported in Table 2 to evaluate the methods practically. This experiment is executed on a laptop with a processor of 3.1 GHz Intel Core i5, 16 GB RAM, running macOS Mojave and MATLAB 2019a.

In order to show the drawbacks of greedy methods, this experiment is done on some matrices, $\mathbf{A} \in \mathbb{R}^{M \times N}$, whose columns are highly correlated. For this purpose, a correlation matrix, $\mathbf{R}^{N \times N}$, of the columns of \mathbf{A} is designed such that its diagonal and off-diagonal entries are 1 and γ , respectively, where γ is an arbitrary mutual coherence constant among the columns of \mathbf{A} . Then, by considering the Cholesky decomposition of \mathbf{R} , *i.e.* $\mathbf{R} = \mathbf{L}^T \mathbf{L}$, and a random orthogonal matrix $\mathbf{U}^{M \times N}$ (\mathbf{U} can be obtained by the QR decomposition of a random matrix), we set $\mathbf{A} = \mathbf{U}\mathbf{L}$.

$\hat{\mathbf{A}}$ is generated by permuting randomly the columns of \mathbf{A} and adding a noise matrix, \mathbf{W} , of the same size as \mathbf{A} with i.i.d. entries of Gaussian distribution with zero mean and unit variance, and weighted by the parameter δ . The variance δ^2 of the additive noise is adjusted such that we reach a desired Signal to Noise Ratio (SNR) defined as:

$$\text{SNR} = 10 \log_{10} \frac{\sum_{i,j} \mathbf{A}(i,j)^2}{\sum_{i,j} \delta^2 \mathbf{W}(i,j)^2}. \quad (14)$$

At the end, the columns of \mathbf{A} and $\hat{\mathbf{A}}$ are normalized.

Table 2: A numerical comparison on methods measuring the distance between $\mathbf{A}^{150 \times 100}$ with the mutual coherence constant $\gamma = 0.75$ and its permuted noisy version $\widehat{\mathbf{A}}$ with SNR = -1.76 dB averaged over 50 realizations. The index in five first rows of the table is the relative error defined in (13). On the other hand, the four last indices of the table are defined differently and are hence not comparable.

Method	Index	Computing time (ms)	Significance of upper bound
Greedy of [3]	0.37	0.9	No
Greedy of [4, 5]	1.05	5.3	No
Hungarian [9]	0.86	4.5	Yes
MWM [10]	0.86	2.9	Yes
Linprog [11]	0.86	1310	Yes
Comon [12]	1.8e4	3.3	No
Moreau-Macchi [13]	897.91	3.3	No
Amari [14]	3.2e3	2.9	No
CorrIndex	0.36	0.4	Yes

In the experiment of Table 2, $M = 150$, $N = 100$, with the mutual coherence constant $\gamma = 0.75$, $\delta = 0.1$ (which is equivalent to SNR = -1.76 dB), and \mathbf{U} is an orthogonal matrix obtained by concatenating the first N left-singular vectors of a random matrix whose entries are chosen randomly from a uniform distribution on $(0, 1)$. The reported values are averaged over 50 realizations.

The indices obtained by greedy methods and reported in Table 2 explicitly show the effect of coherence of input matrix on these types of methods. For instance, according to the performed experiment, greedy methods either report less (37%) or larger (105%) error than the exact index (86%). Note that as greedy methods try first to estimate permutation \mathbf{P} , and then calculate the error between matrices \mathbf{AP} and $\widehat{\mathbf{A}}$; hence, the indices computed by greedy methods may be compared to the exact error computed by graph-based methods. However, comparing other indices such as CorrIndex with the exact error does not make sense, since these indices are intrinsically different from (13).

As can be seen in Table 2, CorrIndex is the fastest index. In addition,

Table 3: A numerical comparison on methods measuring the distance between $\mathbf{A}^{150 \times 100}$ with the mutual coherence constant $\gamma = 0.95$ and its permuted noiseless version $\widehat{\mathbf{A}}$ averaged over 50 realizations. The index in five first rows of the table is the relative error defined in (13). The four last indices of the table are defined differently and are hence not comparable.

Method	Index	Computing time (ms)	Significance of upper bound
Greedy of [3]	$3.82e - 17$	0.89	No
Greedy of [4, 5]	0.10	5.2	No
Hungarian [9]	0	2.7	Yes
MWM [10]	0	4.0	Yes
Linprog [11]	0	1630	Yes
Comon [12]	$2.14e - 13$	3.3	No
Moreau-Macchi [13]	0	4.5	No
Amari [14]	$3.35e - 16$	3.8	No
CorrIndex	$2.37e - 16$	0.59	Yes

contrary to the indices of Comon, Moreau-Macchi and Amari, CorrIndex returns a value in the bounded range $[0, 1]$. One could normalize the Moreau-Macchi and Amari indices in order to obtain bounded values, but the signification of such upper bounds on these indices has never been investigated.

290 The experiment of Table 2 corresponds to an inaccurate estimation of \mathbf{A} (*i.e.* SNR = -1.76), and all the performance indices (perhaps except the greedy method of [3] which yields 37%) demonstrate convincingly the fact that estimation is not accurate. In order to evaluate the indices in the opposite situation (accurate estimation), we perform another experiment with the same setting as
295 that of Table 2 except that $\widehat{\mathbf{A}}$ is a permuted noiseless version of $\mathbf{A}^{150 \times 100}$ with $\gamma = 0.95$. The result of this experiment is reported in Table 3.

As it can be interpreted from the reported indices in Table 3, the greedy method of [4, 5] does not report zero relative error between \mathbf{A} and $\widehat{\mathbf{A}}$ which is not correct. However, in spite of highly correlated columns of \mathbf{A} , all indices
300 (except the greedy method of [4, 5]) demonstrate zero distance between \mathbf{A} and $\widehat{\mathbf{A}}$

which is true. Comparing Table 2 and Table 3 reveals that the greedy methods are much more sensitive to the correlation of the columns than other indices.

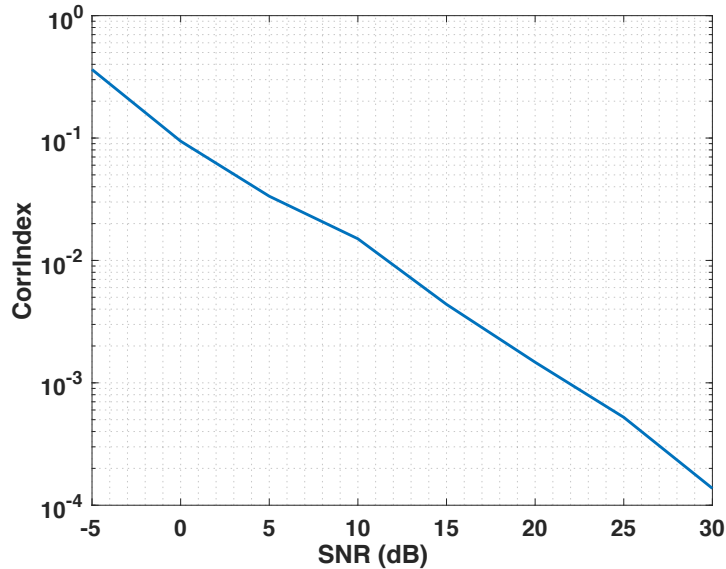


Figure 1: CorrIndex and noise. CorrIndex of a random matrix $\mathbf{A}^{6 \times 4}$ and its permuted noisy version $\hat{\mathbf{A}}$. This figure confirms the fact that the larger ϵ_0 , the larger CorrIndex.

As mentioned before, CorrIndex is based on (2) which tries to minimize the least square error between \mathbf{A} and $\hat{\mathbf{A}}$. Therefore, if the distance between \mathbf{A} and $\hat{\mathbf{A}}$ increases due to the additive noise in $\hat{\mathbf{A}}$, CorrIndex will return a larger value. To show this fact in practice, we performed an experiment whose result is depicted in Fig. 1. Generating a random matrix \mathbf{A} of dimension 6×4 , $\hat{\mathbf{A}}$ is obtained by permuting its columns and by adding a noise matrix, \mathbf{W} , of the same size as \mathbf{A} with independent and identically distributed (i.i.d.) entries of Gaussian distribution with zero mean and unit variance, and weighted by the parameter δ . The variance δ^2 of the additive noise is adjusted such that we reach a desired SNR as described in (14).

Figure 1 confirms the fact that the larger ϵ_0 , the larger CorrIndex. Therefore, in evaluating different decomposition methods, the one with the least CorrIndex would perform the best.

In Table 2, we show the effect of mutual coherence and noise on the results of each index. However, Fig. 2 investigates the effect of noise intensity, which is measured by SNR. In this experiment, as in Table 2, we generate a random matrix \mathbf{A} of dimension 150×100 with the mutual coherence constant $\gamma = 0.75$ and averaged the results over 50 realizations. Then, the matrix $\hat{\mathbf{A}}$ is obtained by permuting randomly the columns of \mathbf{A} and by adding a noise matrix according to each SNR value. The goal of this experiment is to show the drawbacks of greedy methods, and to do this, we compare the result of the greedy methods of [3, 4, 5] by one of the graph-based method (*i.e.* MWM [10]) which outputs the exact error. Therefore, we can simply conclude the inaccuracy of [3, 4, 5].

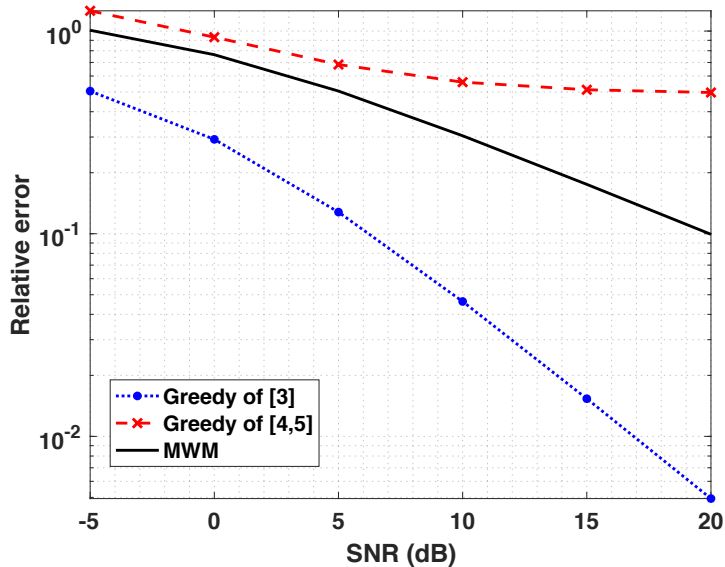


Figure 2: Drawbacks of greedy methods of [3, 4, 5]. Compare the relative error between a random matrix $\mathbf{A}^{150 \times 100}$ with the mutual coherence constant $\gamma = 0.75$ and its permuted noisy version $\hat{\mathbf{A}}$ versus SNR reported by greedy methods of [3, 4, 5] and by one of the exact indices, *i.e.* MWM averaged over 50 realizations.

Figure 2 shows the relative error (13) between $\hat{\mathbf{A}}$ as an estimation of \mathbf{A} . As the error output by MWM is exact, the difference between errors output by the greedy methods of [3, 4, 5] and the one by MWM show the inaccuracy

of [3, 4, 5]. As it is expected, the relative error by greedy method [3] (resp. [4, 5])
330 is optimistic (resp. pessimistic), since its reported error is smaller (resp. larger)
than the exact error. In addition, as SNR increases, this error gets larger, which
demonstrates that by decreasing the additive noise, the influence of mutual
coherence becomes more effective on the result of greedy methods.

5. Conclusion

335 In this paper, the problem of computing the distance between two matrices
up to permutation and scaling ambiguities is addressed. This problem occurs
for instance in tensor decompositions or in blind source separation. Existing
performance indices are classified in three main categories: “greedy methods”,
“graph-based methods” and “invariant indices”. These methods are reviewed,
340 and it is inferred that greedy methods are not reliable especially in noisy situa-
tions (they are either optimistic or pessimistic). In addition, graph-based meth-
ods and invariant indices are computationally expensive. We propose a new
performance index belonging to the class of invariant indices, called *CorrIndex*,
whose upper and lower bounds are easy to interpret, while being computationally
345 ally cheap.

References

- [1] P. Comon, Tensors: a brief introduction, *IEEE Signal Processing Magazine*
31 (3) (2014) 44–53.
- [2] P. Comon, C. Jutten, *Handbook of Blind Source Separation: Independent*
350 *component analysis and applications*, Academic press, 2010.
- [3] X. Fu, S. Ibrahim, H.-T. Wai, C. Gao, K. Huang, Block-randomized
stochastic proximal gradient for low-rank tensor factorization, *IEEE Trans-*
actions on Signal Processing 68 (2020) 2170–2185.

- [4] P. Comon, X. Luciani, A. L. De Almeida, Tensor decompositions, alternating least squares and other tales, *Journal of Chemometrics: A Journal of the Chemometrics Society* 23 (7-8) (2009) 393–405.
- [5] J. Coloigner, A. Karfoul, L. Albera, P. Comon, Line search and trust region strategies for canonical decomposition of semi-nonnegative semi-symmetric 3rd order tensors, *Linear Algebra and its applications* 450 (2014) 334–374.
- [6] C. Battaglino, G. Ballard, T. G. Kolda, A practical randomized cp tensor decomposition, *SIAM Journal on Matrix Analysis and Applications* 39 (2) (2018) 876–901.
- [7] H. W. Kuhn, The hungarian method for the assignment problem, *Naval research logistics quarterly* 2 (1-2) (1955) 83–97.
- [8] Z. Galil, Efficient algorithms for finding maximal matching in graphs, in: *Colloquium on Trees in Algebra and Programming*, Springer, 1983, pp. 90–113.
- [9] J. Munkres, Algorithms for the assignment and transportation problems, *Journal of the society for industrial and applied mathematics* 5 (1) (1957) 32–38.
- [10] R. Duan, S. Pettie, Linear-time approximation for maximum weight matching, *Journal of the ACM (JACM)* 61 (1) (2014) 1–23.
- [11] G. Peyré, M. Cuturi, et al., Computational optimal transport: With applications to data science, *Foundations and Trends® in Machine Learning* 11 (5-6) (2019) 355–607.
- [12] P. Comon, Independent component analysis, a new concept?, *Signal processing* 36 (3) (1994) 287–314.
- [13] E. Moreau, O. Macchi, A one stage self-adaptive algorithm for source separation, in: *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 3, IEEE, 1994, pp. III–49.

- [14] S.-I. Amari, A. Cichocki, H. H. Yang, et al., A new learning algorithm for blind signal separation, in: *Advances in neural information processing systems*, Morgan Kaufmann Publishers, 1996, pp. 757–763.
- [15] S. Sahnoun, P. Comon, Joint source estimation and localization, *IEEE Trans. Sig. Proc.* 63 (10) (2015) 2485–2495, hal-01005352.
- [16] A. Stegeman, Using the simultaneous generalized schur decomposition as a candecomp/parafac algorithm for ill-conditioned data, *Journal of Chemometrics: A Journal of the Chemometrics Society* 23 (7-8) (2009) 385–392.
- [17] Tichavsky, Petr and Koldovsky, Zbynek, Optimal pairing of signal components separated by blind techniques, *IEEE Signal Processing Letters* 11 (2) (2004) 119–122.
- [18] E. Dinic, M. Kronrod, An algorithm for the solution of the assignment problem, in: *Soviet Math. Dokl.*, Vol. 10, 1969, pp. 1324–1326.
- [19] E. W. Dijkstra, et al., A note on two problems in connexion with graphs, *Numerische mathematik* 1 (1) (1959) 269–271.
- [20] H. N. Gabow, R. E. Tarjan, Faster scaling algorithms for network problems, *SIAM Journal on Computing* 18 (5) (1989) 1013–1036.
- [21] A. W. Marshall, I. Olkin, B. C. Arnold, *Inequalities: theory of majorization and its applications*, Vol. 143, Springer, 1979.
- [22] R. A. Horn, C. R. Johnson, *Matrix analysis*, Cambridge university press, 1999.
- [23] D. Bertsimas, J. N. Tsitsiklis, *Introduction to linear optimization*, Vol. 6, Athena Scientific Belmont, MA, 1997.
- [24] M. B. Cohen, Y. T. Lee, Z. Song, Solving linear programs in the current matrix multiplication time, *Journal of the ACM (JACM)* 68 (1) (2021) 1–39.

- [25] Y. T. Lee, Z. Song, Q. Zhang, Solving empirical risk minimization in the current matrix multiplication time, in: Conference on Learning Theory, PMLR, 2019, pp. 2140–2157.
- 410 [26] M. Khosla, A. Anand, Revisiting the auction algorithm for weighted bipartite perfect matchings, arXiv preprint arXiv:2101.07155.