CorrIndex: a permutation invariant performance index

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

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 scale ambiguity 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. The existing methods and measures 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 the previous methods and measures, it has the lowest computational cost. By providing two theorems and a table of comparisons, we will show these advantages of CorrIndex compared to other measures.

Keywords: Assignment, Bipartite graph, Blind Source Separation, ICA, Permutation ambiguity, Tensor decomposition, Hungarian algorithm, Optimal transport, Performance index

1. Introduction

Permutation and scale ambiguity are relevant issues in some applications such as tensor decomposition [1] and source separation [2]. Firstly these two

*Corresponding author
Email addresses: elaheh.sobhani@gipsa-lab.grenoble-inp.fr,sobhani.es@gmail.com (Elaheh Sobhani), pierre.comon@gipsa-lab.grenoble-inp.fr (Pierre Comon), christian.jutten@gipsa-lab.grenoble-inp.fr (Christian Jutten), mbzadeh@sharif.edu (Massoud Babaie-Zadeh)

ambiguities are inherent in tensor representations, by definition of tensors [1]. Secondly, in Blind Source Separation, statistical independence is not affected by scaling or permutation of the sources [2]. A mixing matrix can then only be obtained 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, and the only remaining ambiguity is the permutation. The existing approaches to measure the performances of algorithms of source separation and tensor decomposition can be classified in three main categories: "greedy approaches", "graph-based methods" and "invariant measures". 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 based on the achieved permutation. Although these methods return back the estimated permutation as well as 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, especially when the size of the correlation matrix is larger than 10×10.

However, the viewpoint of a third category, namely invariant measures [12, 13, 14], differs from the latter approaches. These *invariant* indices measure the performance regardless of permutation, and yield an index that can directly be

used to compare algorithms. The reported indices of [12, 13, 14] are invariant to permutation, and the index of [12] will guarantee a zero gap up to permutation between estimated and original variables, if the obtained index is zero. Nevertheless, the methods of [12, 13, 14] are in the literature of source separation, and the indices introduced therein utilize the inverse (or pseudo-inverse) of the mixing matrix, which involves a high computational burden, and becomes intractable for large scale data. In addition, the index of [12] is not bounded from above.

In this paper, we introduce a new performance index, called *CorrIndex*, which can be considered to belong to the category of "invariant measures". CorrIndex is based on some correlation matrix (in fact scalar products), and because of that it does not lead to a high computational cost since matrix inversion is avoided. In addition, not only CorrIndex is invariant to permutation, but also it will guarantee a zero gap up to the permutation if CorrIndex = 0. Therefore, compared to greedy methods, CorrIndex is more reliable. Moreover, compared to graph-based methods and other invariant measures, it requires the lowest computational cost.

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

2. Problem formulation and Related works

Let $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N] \in \mathbb{R}_+^{M\times N}$ and $\widehat{\boldsymbol{A}} = [\hat{\boldsymbol{a}}_1, \hat{\boldsymbol{a}}_2, \ldots, \hat{\boldsymbol{a}}_N] \in \mathbb{R}_+^{M\times N}$ be the original and estimated matrices respectively. Let denote the group of permutations of N elements by $\mathrm{Perm}(N)$, and denote by $\boldsymbol{P}_\sigma$ the matrix associated with the permutation $\sigma \in \mathrm{Perm}(N)$.

Assume $\widehat{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{P}_\sigma + \boldsymbol{W}$, where the columns of $\boldsymbol{A}$ and $\widehat{\boldsymbol{A}}$ are normalized and $\boldsymbol{W}$ is an additive noise. The goal is to measure the gap between $\widehat{\boldsymbol{A}}$ and $\boldsymbol{A}$ up to a permutation indeterminacy. This measuring can be done with or

without estimating $\boldsymbol{P}_\sigma$. Seeking the optimal σ, can be written as the following optimization problem:

$$\underset{\sigma}{\operatorname{argmin}} \frac{1}{2} \sum_{n=1}^{N} \|\boldsymbol{a}_n - \hat{\boldsymbol{a}}_{\sigma(n)}\|_2^2 = \underset{\sigma}{\operatorname{argmax}} \sum_{n=1}^{N} \boldsymbol{a}_n^T \hat{\boldsymbol{a}}_{\sigma(n)}. \tag{1}$$

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

2.1. Methods based on correlation matrix

In this class of methods, the goal is to associate each column of $\widehat{\boldsymbol{A}}$ with the most correlated one in $\boldsymbol{A}$. In other words, the association is based on $\boldsymbol{C}$. According to the manner of selecting these pairs of columns, different approaches have been developed.

2.1.1. Maximum of each column [3]

$\hat{\boldsymbol{a}}_j$ is assigned to $\boldsymbol{a}_i$ if C_{ij} has the maximum value in the j^{th} column of $\boldsymbol{C}$. 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. On the other hand, the delivered measure is not reliable, since, even if the measure is zero, one cannot guarantee $\widehat{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{P}_\sigma$. The following toy numerical example illustrates well this problem.

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

$$\boldsymbol{C} = \begin{bmatrix} 0.8 & 0.3 & 0.1 \\ 0.85 & 0.9 & 0.5 \\ 0.5 & 0.2 & 0.7 \end{bmatrix}. \tag{2}$$

The concluded assignment by this method is $(\hat{\boldsymbol{a}}_1, \boldsymbol{a}_2)$, $(\hat{\boldsymbol{a}}_2, \boldsymbol{a}_2)$, $(\hat{\boldsymbol{a}}_3, \boldsymbol{a}_3)$ which is obviously not acceptable because column $\boldsymbol{a}_2$ is selected twice. Computing the square error via $\frac{1}{2}\sum_{n=1}^{3} \|\boldsymbol{a}_n - \hat{\boldsymbol{a}}_{\sigma(n)}\|_2^2$ by considering the assumption of normalized $\boldsymbol{a}_n$ and $\hat{\boldsymbol{a}}_{\sigma(n)}$ with respect to L_2 norm and by substituting the values

of $\boldsymbol{a}_i^T \hat{\boldsymbol{a}}_j$ from C_{ij}, one obtains $3 - 0.85 - 0.9 - 0.7 = 0.55$, which is less than the 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 measure can be proved to be always *optimistic* since it searches over a set of assignments larger than Perm(N). Therefore, the reported error based on its delivered assignment is always less than or equal to the exact error based on the optimal assignment.

2.1.2. Greedy approach [4, 5]

In order to avoid a non-acceptable assignment, after detecting the maximum value of each column of $\boldsymbol{C}$, its row and column can be removed for the rest of the algorithm. In other words, if in j^{th} column of matrix $\boldsymbol{C}$, C_{ij} is the maximum value, then the i^{th} row and j^{th} column of $\boldsymbol{C}$ will be ignored in the search of the 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 $\boldsymbol{C}$ expressed in (2), the resulted assignment will be $(\hat{\boldsymbol{a}}_1, \boldsymbol{a}_2)$, $(\hat{\boldsymbol{a}}_2, \boldsymbol{a}_1)$, $(\hat{\boldsymbol{a}}_3, \boldsymbol{a}_3)$ provided that the columns are swept from left to right. However, if the columns are swept in the opposite way, the assignment will be $(\hat{\boldsymbol{a}}_1, \boldsymbol{a}_1)$, $(\hat{\boldsymbol{a}}_2, \boldsymbol{a}_2)$, $(\hat{\boldsymbol{a}}_3, \boldsymbol{a}_3)$. Compared to the optimistic measure, 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.

This measure can be proved to be always *pessimistic*, since, by imposing a column ordering, this greedy approach searches over a set of assignments smaller than Perm(N). Therefore, the reported error based on its delivered assignment 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* [15], is customized for tensors and is applied to evaluate the performance of a tensor decomposition in terms

of estimating all the loading matrices together. For instance, this measure for a third order tensor $\boldsymbol{\mathcal{T}} = [\![\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]$ is calculated based on the correlation matrix $\boldsymbol{C} = \boldsymbol{C}_1 \boxdot \boldsymbol{C}_2 \boxdot \boldsymbol{C}_3$, where $\boxdot$ is the *Hadamard* product (element-wise product) and $\boldsymbol{C}_i \triangleq \boldsymbol{A}_i^T \widehat{\boldsymbol{A}}_i, i = 1, 2, 3$. This measure is also greedy, since the assignment is concluded based on the maximum values of $\boldsymbol{C}$, which have been chosen in a way explained in 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]. Harold Kuhn in 1955 developed and published the first polynomial time algorithm for optimal assignment problems, and he named it "Hungarian method" [7]. In 1957, James Munkres completed the algorithm by providing a constructive procedure for the Hungarian method [9], and the complexity of the algorithm was $\mathcal{O}(N^4)$.

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]. Actually, the optimal assignment problem can be modeled by a *bipartite* [8] graph which includes two disjoint and independent sets of vertices without any odd-length cycle, and every edge connects a vertex from one set to another. To be more precise, two sets of columns of $\boldsymbol{A}$ and $\widehat{\boldsymbol{A}}$ can be considered as sets of vertices of a bipartite graph which are fully connected to each other. Matrix $\boldsymbol{C}$ contains the weights of each edge in this graph. For the special case of bipartite graphs, as is the case in the above optimal transport problem, there are also several more efficient algorithms [8, 10]. The cost of such algorithms depends on the number of vertices, v, and edges, e. A naive implementation costs $\mathcal{O}(ev^2)$ [8, 16], or, after refinements, two other algorithms cost $\mathcal{O}(v^3)$ and $\mathcal{O}(ev \log v)$ [10, 17, 16]. There are some approximate algorithms as well which cost $\mathcal{O}(e\sqrt{v})$ or $\mathcal{O}(e)$ [18, 10]. For the corresponding bipartite graph of the optimal assignment problem described at the beginning of Section 2, $v = 2N$ and $e = N^2$. Therefore, the best exact

and approximate MWM algorithm cost $\mathcal{O}(8N^3)$ and $\mathcal{O}(N^2)$ respectively.

2.3. Methods based on optimal permutation

Suppose that instead of searching for the optimal permutation (described at the beginning of Section 2), we consider the following optimization problem [11]:

$$\boldsymbol{P}^\star = \underset{\boldsymbol{P}\in\mathbb{R}_+^{N\times N}}{\operatorname{argmin}} \sum_{i,j} D_{ij}P_{ij} \quad \text{s.t.} \quad \boldsymbol{P}\mathbb{1}_N = \boldsymbol{P}^T\mathbb{1}_N = \mathbb{1}_N, \tag{3}$$

where $\boldsymbol{D} = -\boldsymbol{C}$, $\mathbb{1}_N$ 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 [19]. By vectorizing (concatenating columns) $\boldsymbol{P}$ and $\boldsymbol{D}$, (3) can be rewritten in the standard form of linear program [11, Sec. 3.1]:

$$\boldsymbol{p}^\star = \underset{\boldsymbol{p}\in\mathbb{R}_+^{N^2}}{\operatorname{argmin}} \boldsymbol{d}^T\boldsymbol{p} \quad \text{s.t.} \quad \boldsymbol{Q}\boldsymbol{p} = \mathbb{1}_{2N}, \tag{4}$$

where $\boldsymbol{Q} = [\mathbb{1}_N^T \boxtimes \boldsymbol{I}_N, \boldsymbol{I}_N \boxtimes \mathbb{1}_N^T]^T \in \mathbb{R}^{2N\times N^2}$, $\boldsymbol{I}_N$ 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 polyhedron[1] whose vertices are permutations [20, Theorem 8.7.1]. On the other hand, a fundamental theorem of linear programming [21, Theorem 2.7] tells that the minimum of a linear 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 (3) or (4): in fact, looking for the best bistochastic matrix will eventually yield a permutation.

For example, the optimal permutation in experiment (2) is the identity matrix, and therefore, the exact error according to $\frac{1}{2}\sum_{n=1}^{3}\|\boldsymbol{a}_n - \hat{\boldsymbol{a}}_{\sigma(n)}\|_2^2$ by considering the assumption of normalized $\boldsymbol{a}_n$ and $\hat{\boldsymbol{a}}_{\sigma(n)}$ with respect to L_2 norm and by substituting the values of $\boldsymbol{a}_i^T\hat{\boldsymbol{a}}_j$ from C_{ij} is $3 - 0.8 - 0.9 - 0.7 = 0.6$.

[1] convex by definition.

In 1947, George B. Dantzig invented the simplex method [22] which finds an optimal solution by traversing the edges between vertices on a polyhedron set. Besides the simplex method, some other interior point methods have been introduced which move through the interior of the feasible region. In 1979, Ellipsoid method [23] of this family has been developed which costs $\mathcal{O}(q^6)$, where q is the size of unknown vector in the linear programming problem. Recently, in 2019 and 2020, by improving the needed matrix multiplications, the running time has been reduced to $\mathcal{O}(q^{2+\frac{1}{6}})$ [24] and $\mathcal{O}(q^{2+\frac{1}{18}})$ [25]. Therefore, as $q = N^2$ in (4), the lowest complexity 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, Auction algorithm [11, 26] provides a sub-optimal solution which costs $\mathcal{O}(N^2)$ [26].

2.4. Measures invariant to permutation

Methods described in Section 2.1 aim at estimating the permutation first, and then measure the error based on the estimated permutation. As it is mentioned in Section 2.1, these kinds of methods may return non-acceptable results, and there is no guarantee that $\widehat{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{P}_\sigma$, even if the measures they output are zero. In addition, the algorithms of Section 2.2 behave better but 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 between original and estimated mixing matrices without needing to find the corresponding permutation [12, 13, 14]. Moreover, the measure of [12] guarantees that the gap is zero if and only if $\widehat{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{P}_\sigma$. The details of these measures are as follows.

2.4.1. Comon measure [12]

If $\boldsymbol{A}$ is a square matrix, by defining $\boldsymbol{S} = \boldsymbol{A}^{-1}\widehat{\boldsymbol{A}}$, Comon measure is calculated as:

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

In [12], it has been proved that ϵ_1 is invariant to permutation, *i.e.* $\epsilon_1(\boldsymbol{A}, \widehat{\boldsymbol{A}}) = \epsilon_1(\boldsymbol{A}, \widehat{\boldsymbol{A}}\boldsymbol{P})$. Moreover, it has been shown that $\epsilon_1(\boldsymbol{A}, \widehat{\boldsymbol{A}}) = 0$ if and only if $\widehat{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{P}_\sigma$, where σ is the optimal permutation. Note that one could replace $\boldsymbol{A}^{-1}$ by $\boldsymbol{A}^\dagger$ when the mixing matrix $\boldsymbol{A}$ is not square.

2.4.2. Moreau-Macchi measure [13]

This index measures the distance between matrix $\boldsymbol{S} = \boldsymbol{A}^{-1}\widehat{\boldsymbol{A}}$ (or $\boldsymbol{S} = \boldsymbol{A}^\dagger\widehat{\boldsymbol{A}}$ for non-square $\boldsymbol{A}$) and a permutation matrix. It is defined as:

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

2.4.3. Amari measure [14]

This performance index is also based on $\boldsymbol{S} = \boldsymbol{A}^{-1}\widehat{\boldsymbol{A}}$ (or $\boldsymbol{S} = \boldsymbol{A}^\dagger\widehat{\boldsymbol{A}}$ for non-square $\boldsymbol{A}$) and takes the form:

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

The only difference between Amari and Moreau-Macchi measure is the power 2 which exists in ϵ_2.

An accurate investigation of measures reviewed in this section reveals that not only calculating these indices are costly, but also computing $\boldsymbol{A}^{-1}$ or $\boldsymbol{A}^\dagger$ is computationally expensive, since they become intractable in high dimension and their accuracy depends on the conditioning of $\boldsymbol{A}$. In addition, $\epsilon_2 = 0$ and $\epsilon_3 = 0$ do not guarantee the zero gap between $\boldsymbol{A}$ and $\widehat{\boldsymbol{A}}$, *i.e.* $\widehat{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{P}_\sigma$. Furthermore,

ϵ_1 is not bounded from above. One may think of replacing $\boldsymbol{S} = \boldsymbol{A}^\dagger \widehat{\boldsymbol{A}}$ by $\boldsymbol{C} = \boldsymbol{A}^T \widehat{\boldsymbol{A}}$, but in this case the second property of ϵ_1, which guarantees that $\epsilon_1 = 0$ is equivalent to $\widehat{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{P}_\sigma$, does not hold anymore.

3. Our proposed measure: CorrIndex

In this section, we introduce our index, "*CorrIndex*", which is based on correlation matrix $\boldsymbol{C} = \boldsymbol{A}^T \widehat{\boldsymbol{A}}$, where $\boldsymbol{A} \in \mathbb{R}_+^{M\times N}$ and $\widehat{\boldsymbol{A}} \in \mathbb{R}_+^{M\times N}$. In addition, assume that the columns of $\boldsymbol{A}$ and $\widehat{\boldsymbol{A}}$ are normalized by their L_2 norms. CorrIndex is defined as follows:

$$\begin{aligned}\text{CorrIndex}(\boldsymbol{C}) =& \frac{1}{2N}\left[\sum_{i=1}^{N} |\max_k |C_{ik}| - 1| \right.\\ &\left. + \sum_{j=1}^{N} |\max_k |C_{kj}| - 1|\right].\end{aligned} \tag{5}$$

In the following, it will be shown that CorrIndex is invariant to permutation, *i.e.* $\text{CorrIndex}(\boldsymbol{A}, \widehat{\boldsymbol{A}}) = \text{CorrIndex}(\boldsymbol{A}, \widehat{\boldsymbol{A}}\boldsymbol{P})$. Moreover, it will be shown that $\text{CorrIndex}(\boldsymbol{C}) = 0$ if and only if $\widehat{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{P}_\sigma$. It can be also observed that CorrIndex is bounded: $0 \leq \text{CorrIndex} \leq 1$, where the upper bound is achieved when all the columns of $\boldsymbol{A}$ and $\widehat{\boldsymbol{A}}$ are orthogonal to each other.

Theorem 1. *CorrIndex is invariant to permutation:*

$$\textit{CorrIndex}(\boldsymbol{A}, \widehat{\boldsymbol{A}}) = \textit{CorrIndex}(\boldsymbol{A}\boldsymbol{P}, \widehat{\boldsymbol{A}}) = \textit{CorrIndex}(\boldsymbol{A}, \widehat{\boldsymbol{A}}\boldsymbol{P}). \tag{6}$$

Proof. Assume that $\boldsymbol{C}_1 = \boldsymbol{A}^T \widehat{\boldsymbol{A}}$ and $\boldsymbol{C}_2 = (\boldsymbol{A}\boldsymbol{P})^T \widehat{\boldsymbol{A}}$. It is obvious that $\boldsymbol{C}_2 = \boldsymbol{P}^T \boldsymbol{C}_1$, and since CorrIndex is invariant to the row permutation according to (5), the proof is complete. The same proof applies to $\boldsymbol{C}_3 = \boldsymbol{A}^T \widehat{\boldsymbol{A}}\boldsymbol{P}$, because of the invariance of CorrIndex to column permutation. □

Theorem 2. *Suppose that* $\boldsymbol{A} \in \mathbb{R}_+^{M\times N}$ *and* $\widehat{\boldsymbol{A}} \in \mathbb{R}_+^{M\times N}$. *CorrIndex*$(\boldsymbol{A}, \widehat{\boldsymbol{A}}) = 0$ *if and only if* $\widehat{\boldsymbol{A}}$ *can be written as a permuted version of* $\boldsymbol{A}$*:*

$$\textit{CorrIndex}(\boldsymbol{A}, \widehat{\boldsymbol{A}}) = 0 \iff \widehat{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{P}. \tag{7}$$

Proof. First, if $\widehat{\boldsymbol{A}} = \boldsymbol{AP}$, then $\max_k C_{ik} = 1, \forall i$ and $\max_k C_{kj} = 1, \forall j$. Thus $\text{CorrIndex}(\boldsymbol{A}, \widehat{\boldsymbol{A}}) = 0$. Second, we prove the converse. If $\text{CorrIndex}(\boldsymbol{A}, \widehat{\boldsymbol{A}}) = 0$, 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 $\boldsymbol{C}$. Let us assume $C_{ij} = \boldsymbol{a}_i^T \hat{\boldsymbol{a}}_j = 1$. According to the Cauchy–Schwarz inequality and the assumption of normalized columns of $\boldsymbol{A}$ and $\widehat{\boldsymbol{A}}$, we have $|\boldsymbol{a}_i^T \hat{\boldsymbol{a}}_j| \leq \|\boldsymbol{a}_i\| \|\hat{\boldsymbol{a}}_j\|$, where the equality of two sides occurs if and only if $\boldsymbol{a}_i = \hat{\boldsymbol{a}}_j$. Since such a conclusion holds for all other associated pairs of columns of $\boldsymbol{A}$ and $\widehat{\boldsymbol{A}}$, therefore, $\widehat{\boldsymbol{A}} = \boldsymbol{AP}_\sigma$. □

4. Discussion and computer results

A multi-aspect comparison between CorrIndex and other reviewed methods has been done 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 ($\boldsymbol{C} = \boldsymbol{A}^T \widehat{\boldsymbol{A}}$ or $\boldsymbol{S} = \boldsymbol{A}^\dagger \widehat{\boldsymbol{A}}$), estimating the permutation and computing the index, is reported. In addition, the "Guarantee" part in Table 1 indicates whether the method provides some mathematical theorems of optimality and invariance or not. 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. Moreover, CorrIndex is invariant to permutation, and CorrIndex = 0 guarantees zero gap up to permutation between $\boldsymbol{A}$ and $\widehat{\boldsymbol{A}}$.

The index and computation time of each measure 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 Mojav and MATLAB 2019a. In order to show the drawbacks of greedy methods, this experiment is done on some matrices, $\boldsymbol{A} \in \mathbb{R}_+^{M \times N}$, whose columns are highly correlated. For this purpose, a correlation matrix, $\boldsymbol{R}^{N \times N}$, of the columns of $\boldsymbol{A}$ is designed such that its diagonal and non-diagonal entries are 1 and m respectively, where m is an arbitrary mutual coherence constant

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

Method	$\boldsymbol{C}$ or $\boldsymbol{S}$	Permutation	Index	Guarantee
Greedy	$\mathcal{O}(N^2M)$	0	1	✗
Graph	$\mathcal{O}(N^2M)$	$\mathcal{O}(8N^3)$	1	✓
Linprog	$\mathcal{O}(N^2M)$	$\mathcal{O}(N^4)$	1	✓
ϵ_1	$\mathcal{O}(11N^3)$	-	$\mathcal{O}(2N^2)$	✓
ϵ_2	$\mathcal{O}(11N^3)$	-	$\mathcal{O}(2N^2)$	✗
ϵ_3	$\mathcal{O}(11N^3)$	-	$\mathcal{O}(2N)$	✗
CorrIndex	$\mathcal{O}(N^2M)$	-	1	✓

among the columns of $\boldsymbol{A}$. Then, by considering the Cholesky decomposition of $\boldsymbol{R}$, *i.e.* $\boldsymbol{R} = \boldsymbol{L}^T\boldsymbol{L}$, and a random orthogonal matrix $\boldsymbol{U}^{M\times N}$ ($\boldsymbol{U}$ can be obtained by the QR decomposition of a random matrix), we set $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{L}$. $\widehat{\boldsymbol{A}}$ is earned by permuting the columns of $\boldsymbol{A}$ and adding Gaussian noise with standard deviation δ. At the end, the columns of $\boldsymbol{A}$ and $\widehat{\boldsymbol{A}}$ are normalized. In the experiment of Table 2, $M = 150, N = 100, m = 0.75, \delta = 0.1$, and $\boldsymbol{U}$ is the first N left-singular vectors of a random matrix whose entries are chosen randomly from uniform distribution on $(0, 1)$. The reported values are averaged over 50 realizations.

The reported measures by greedy methods 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 (0.36) or larger (1.04) error than the exact measure (0.73). As can be seen in Table 2, not only CorrIndex is the fastest measure, but also it reports the closest index to the exact measure (Hungarian and Linprog). Moreover, although the problem is ill-conditioned due to the high coherence of the columns of $\boldsymbol{A}$ and to additive noise, CorrIndex is more reliable compared to MWM which is considered as an optimal measure.

Table 2: A numerical comparison on methods measuring the gap between $\boldsymbol{A}^{150\times 100}$ and its permuted noisy version $\widehat{\boldsymbol{A}}$

Method	Index	Computing time
Greedy of [3]	0.36	0.0013
Greedy of [4, 5]	1.04	0.0056
Hungarian [9]	0.73	0.0048
MWM [10]	0.85	0.0030
Linprog [11]	0.73	1.35
Comon [12]	1.8e4	0.0032
Moreau-Macchi [13]	882.84	0.0027
Amari [14]	3.2e3	0.0028
CorrIndex	0.72	0.0004

5. Conclusion

In this paper, the problem of permutation ambiguity after tensor decomposition or blind source separation is addressed. It is mentioned that the previous methods and measures can be classified in three main categories: "greedy methods", "graph-based methods" and "invariant measures". These methods are reviewed, and it is inferred that greedy methods are not reliable especially in noisy situations. In addition, graph-based methods and invariant measures are computationally expensive. We propose a new performance index belonging to the class of invariant measures, called *CorrIndex*, which is not only reliable, but also according to the Table 1, has the lowest computational cost. We also provide proofs that CorrIndex is invariant to permutation, and that CorrIndex $= 0$ if and only if $\boldsymbol{A}$ and $\widehat{\boldsymbol{A}}$ differ by a permutation of columns.

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 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 Transactions 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] 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.

[16] E. W. Dijkstra, et al., A note on two problems in connexion with graphs, Numerische mathematik 1 (1) (1959) 269–271.

[17] E. Dinic, M. Kronrod, An algorithm for the solution of the assignment problem, in: Soviet Math. Dokl, Vol. 10, 1969, pp. 1324–1326.

[18] H. N. Gabow, R. E. Tarjan, Faster scaling algorithms for network problems, SIAM Journal on Computing 18 (5) (1989) 1013–1036.

[19] A. W. Marshall, I. Olkin, B. C. Arnold, Inequalities: theory of majorization and its applications, Vol. 143, Springer, 1979.

[20] R. A. Horn, C. R. Johnson, Matrix analysis, Cambridge university press, 1999.

[21] D. Bertsimas, J. N. Tsitsiklis, Introduction to linear optimization, Vol. 6, Athena Scientific Belmont, MA, 1997.

[22] G. B. Dantzig, M. N. Thapa, Linear programming, Springer, 1997.

[23] L. G. Khachiyan, A polynomial algorithm in linear programming, in: Doklady Akademii Nauk, Vol. 244, Russian Academy of Sciences, 1979, pp. 1093–1096.

[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.

[26] M. Khosla, A. Anand, Revisiting the auction algorithm for weighted bipartite perfect matchings, arXiv preprint arXiv:2101.07155.