Optimal Pairing of Signal Components Separated by Blind Techniques

Petr Tichavský, *Member, IEEE,* and Zbyněk Koldovský, *Student Member, IEEE*

***Abstract*—In this letter, the problem of optimal pairing of signal components separated by blind techniques in different time-windows or in different frequency bins is addressed. The optimum pairing is defined as the one which minimizes the sum of some distances (criteria of dissimilarity) of the to-be-assigned signal components. It is shown that the optimal pairing can be achieved by the Kuhn–Munkres algorithm known in graph theory as a solution to the optimal assignment problem. An advantage of the proposed pairing method is shown on data from electroencephalogram, which are blindly separated using the FastICA algorithm in a sliding time-window with the aim to study the time evolution of elements of the estimated mixing matrix.**

***Index Terms*—Electroencephalography, independent component analysis (ICA).**

I. Introduction

BLIND source separation (BSS) has raised much interest in the signal processing community since 1980s. Its aim is to recover original sources (*original signals*) from their mixtures (*measured signals*). The term "blind" is used because neither the original signals nor the mixing system are known. BSS has a multitude of interesting applications in telecommunications, speech recognition, medical signal processing, and others. Independent component analysis (ICA) is one of many methods for BSS. Its main assumptions are mutual independence and non-Gaussianity of original signals. Separation is achieved by finding such a transformation of measured signals that either the transformed signals are independent or their mutual dependence is as low as possible. Many ICA algorithms have been proposed until now, e.g., see [1] or [2].

The simplest and the most frequently considered mixing model is the linear one

$$x(t) = As(t) \tag{1}$$

where $s(t) = [s_1(t), \dots, s_n(t)]^T$ is a vector of n original signals, A is unknown $n \times n$ regular mixing matrix, and $x(t)$ is a vector denoting measured signals. Despite the simplicity of this model, there are the following indeterminacies: scales, signs, and a permutation of original signals. These parameters cannot be recovered without any *a priori* knowledge. We can assume without any loss of generality that the original signals have unit variances. Indeterminacy of the signs and the permutation of the signals (called *permutation problem*) is related to indeterminacy of signs and mutual order of columns in the mixing matrix A.

In many situations in practice, the permutation and signs of separated signals are important. If performance of some ICA algorithm is studied by Monte Carlo simulations, the estimated components should be somehow compared with the original ones. A widely used *performance index* introduced in [3] can serve as a measure of entire separation quality, but it does not provide a solution of the indeterminacies. Another application, where the permutation problem needs to be solved, is blind separation of convolutive mixtures in the frequency domain. The separation consists in estimating a set of instantaneous models for several frequency bins [6], [7]. Since the order of components in each frequency bin can be quite arbitrary, the correct reshuffling of the components is of essential importance.

Another example of application presented in this letter is separation of the system (1) at different time instants (time-windowing). Here, estimated signals or separation matrices cannot be compared, and stationarity of the ICA model cannot be studied without a solution of the permutation problem.

Assume that a measure of similarity of the separated components in two consequent time-windows or similarity of corresponding columns of the mixing matrices is given. A correct permutation of the components in the latter window is the one that minimizes the sum of distances from the components computed in the former window.

In this letter, the problem is interpreted in terms of graph theory, and it is shown to be equivalent to finding optimal matching in a complete bipartite graph. The optimal matching can be found by the Kuhn–Munkres algorithm [4], [5], which is also known under the name *Hungarian*. It is shown later in the letter that the traditional matching method, commonly used in the ICA applications and known in the graph theory as the *greedy* algorithm, does not find the optimal matching in general.

The permutation problem is formulated in Section II, and the greedy algorithm is presented in Section III. The Kuhn–Munkres algorithm is described in Section IV. An advantage of the algorithm compared to the traditional pairing method is shown in Section V on an example of a 19-channel data block from electroencephalogram (EEG), where the columns of the estimated mixing matrix are reshuffled in each time-window to achieve smooth tracking of the matrix.

II. Problem Formulation

Let s be the original signals and A the original mixing matrix, and let $\hat{s}$, $\hat{A}$ denote their estimates. In different scenarios, s can represent independent components estimated in one data

Manuscript received November 29, 2002; revised February 26, 2003. This work was supported by the Grant Agency of the Czech Republic under Grant 102/01/0021. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dimitris A. Pados.

The authors are with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, CZ-182 08 Prague, Czech Republic.

Digital Object Identifier 10.1109/LSP.2003.821658

1070-9908/04$20.00 © 2004 IEEE

window or in one frequency bin, and $\hat{s}$ independent components in a neighborhood time-window or frequency bin, and A, $\hat{A}$ the mixing matrices.

The problem to be solved is to find such a permutation of the signals $\hat{s}$, and if needed, also their signs, phases, or scales, so that the corresponding mixing matrix is as close to the original mixing matrix A as possible. Note that each change in order, signs, phases, or scales of the signals corresponds to a similar change in terms of columns of the mixing matrix.

It appears that the main problem here is to find a proper permutation, while the sign, phase, and possibly scale ambiguity can be solved using a convenient metric (measure of dissimilarity) between columns of A, $\hat{A}$. Let $d(\cdot,\cdot)$ denote such a metric, for example

$$d(A_i,\hat{A}_k)=\|A_{\cdot i}-\hat{A}_{\cdot k}\|^2=\sum_{\ell=1}^{n}|A_{\ell i}-\hat{A}_{\ell k}|^2 \qquad (2)$$

but the exact functional form is not essential for a further development. Let the main criterion to be minimized by a proper reordering, re-signing, and/or rescaling of the signals $\hat{s}$ be

$$c(A,\hat{A})=\sum_{i=1}^{n} d(A_i,\hat{A}_i). \qquad (3)$$

Note that if the distance $d(\cdot,\cdot)$ has the form (2), then $c(A,\hat{A})$ equals to the Frobenius norm of the matrix difference squared, $\|A-\hat{A}\|_F^2$.

Now, to cope with the sign ambiguity, which is encountered in blind separation of real-valued signals, define an auxiliary distance

$$d_1(A_i,\hat{A}_k)=\min\{d(A_i,\hat{A}_k),d(A_i,-\hat{A}_k)\}. \qquad (4)$$

For processing of complex-valued signals, where a phase ambiguity is encountered, define another auxiliary distance

$$d_2(A_i,\hat{A}_k)=\min_{\alpha\in(-\pi,\pi)} d(A_i,e^{j\alpha}\hat{A}_k). \qquad (5)$$

If, in addition, the signals $\hat{s}$ are not constrained to have a unit norm, but have a scale ambiguity, one can define

$$d_3(A_i,\hat{A}_k)=\min_{a\in\mathbb{C}} d(A_i,a\hat{A}_k) \qquad (6)$$

where $\mathbb{C}$ stands for the set of complex numbers. The permutation problem is equivalent to search for a permutation matrix

$$\hat{P}=\arg\min_{P\in\mathcal{P}}\sum_{i=1}^{n} d_m(A_{\cdot i},(\hat{A}P)_{\cdot i}) \qquad (7)$$

where $m = 1$, 2, or 3. Once the optimum solution of the problem (7) is found, the correct sign, phase, or scale, respectively, for each column of $\hat{A}$ can be selected according to the terms in (4), the phase angles α in (5), or the scale factors a in (6), where the minima are attained. Note that if $d(\cdot,\cdot)$ has the form (2), the minimum in (5) is achieved for $\alpha=\mathrm{Arg}[\hat{A}_k^H A_i]$ where "H" denotes the Hermitian transpose, and $d_2(A_i,\hat{A}_k)=\|A_i\|^2+\|\hat{A}_k\|^2-2|A_i^H\hat{A}_k|$. Similarly, the minimum in (6) is achieved for $a=\hat{A}_k^H A_i/\|\hat{A}_k\|^2$ and $d_3(A_i,\hat{A}_k)=\|A_i\|^2-|\hat{A}_k^H A_i|^2/\|\hat{A}_k\|^2$.

The permutation problem (7) can be formulated as the maximum matching problem in terms of graph theory as follows.

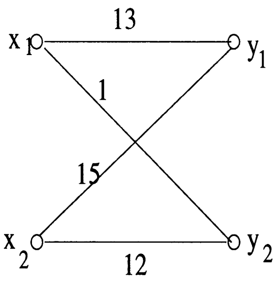

Fig. 1. Example of a complete bipartite graph with weighted edges where the optimal matching is different from the one found by the greedy algorithm.

Columns of the mixing matrices A and $\hat{A}$ are represented by vertices $X=\{x_1,\ldots,x_n\}$ and $Y=\{y_1,\ldots,y_n\}$, respectively, in a complete bipartite graph $G=(X\cup Y, X\times Y)$. Each edge $(x_i y_k)$ has the weight $w(x_i y_k)=-d_m(A_{\cdot i},\hat{A}_{\cdot k})$ for all $x_i\in X$, $y_k\in Y$, and $m = 1$, 2, or 3. A matching in G is such a subgraph in G that each vertex from X is joined with at most one vertex from Y and *vice versa*. The *weight of the matching* is defined as a sum of weights of the edges in the matching. Now, finding the matrix $\hat{P}$ from (7) is equivalent to finding the matching in G that has the maximum weight. This is also called the *optimal assignment* problem.

III. Traditional Matching (Greedy Algorithm)

The commonly used method of matching used in the ICA applications, e.g., [6], known in the graph theory as the *greedy* algorithm, runs recursively, so that in each step it selects the edge with the maximum weight among those vertices that have not yet been matched. The advantage of this matching method is its low computational complexity. It is easy to see that the complexity is $O(n^2)$, where n is the number of nodes in X and Y. Unfortunately, the optimal assignment may not, in general, be attained in this way. An example is shown in Fig. 1. While the optimum (maximum) assignment is $M=\{(x_1y_1),(x_2y_2)\}$ and has the total weight of 25, the greedy algorithm results in $M=\{(x_2y_1),(x_1y_2)\}$ with the total weight of only 16.

IV. Proposed Method

The Kuhn–Munkres algorithm is based on the notion of a *feasible vertex labeling*. A feasible vertex labeling in G is a function $l: X\cup Y\mapsto\mathbb{R}$ such that for all $x\in X$ and $y\in Y$

$$l(x)+l(y)\geq w(xy)$$

where xy is edge between x and y. Subgraph G_l of G, called *equality subgraph* for l, is defined as a subgraph containing those edges $xy\in X\times Y$, where $l(x)+l(y)=w(xy)$. Notice that the maximal matching in G_l is also optimal in G_l. From this, it readily follows that if M is a complete matching in G and $M\subseteq G_l$, then M is the optimal assignment of X to Y.

Thus, the problem of finding an optimal assignment is reduced to the problem of finding a feasible vertex labeling whose equality subgraph contains a complete matching of X to Y. The algorithm starts with a trivial but feasible vertex labeling $l(x)=\max_{y\in Y} w(xy)$ for $x\in X$ and $l(y)=0$ for $y\in Y$. The maximal matching in the initial graph G_l is $M=\arg\max_{(xy)\in G} w(xy)$. The labeling is sequentially updated in order to expand the corresponding equality subgraph G_l and consequently update the maximum matching M in G_l, until M is a complete matching for G. Summary of the algorithm is given below [4], [5], [8], [9].[1]

Kuhn–Munkres Algorithm: Start with an arbitrary feasible vertex labeling l and choose an arbitrary matching M in G_l.

Step 1) If M is a complete matching for G, then M is optimal. Stop. Otherwise find a vertex $x_0 \in X$ that has not yet been matched in M. Set $S = \{x_0\}$ and $T = \{\}$.

Step 2) If $\{y \in Y; \exists x \in S; (xy) \in G_l\} \neq T$ go to step 3). Otherwise put

$$\alpha_l = \min_{x \in S, y \in Y-T} \{l(x) + l(y) - w(xy)\} \tag{8}$$

and define a new vertex labeling by

$$l'(v) = \begin{cases} l(v) - \alpha_l, & \text{for } v \in S \\ l(v) + \alpha_l, & \text{for } v \in T \\ l(v), & \text{otherwise.} \end{cases}$$

It can be seen that $\alpha_l > 0$, $G_{l'} - G_l \neq 0$. Put $l = l'$, $G_l = G_{l'}$.

Step 3) Find $y \in (Y - T)$ such that $(xy) \in G_l$ for some $x \in S$. If y is matched in M, e.g., $(zy) \in M$ for some z, then replace S by $S \cup \{z\}$ and T by $T = T \cup \{y\}$ and go to step 2). Otherwise find a sequence of vertices $(x_0 y_0 x_1 y_1, \ldots, x_k y_k)$ such that $x_j \in X$, $y_j \in Y$, $y_k = y$, $(x_j y_j) \in (G_l - M)$ for $j = 0, \ldots, k$, and $(x_j y_{j-1}) \in M$ for $j = 1, \ldots, k$ (such sequence is called an *alternating path*). Define a new matching in G_l by

$$\begin{aligned} M' = & M \cup \{(x_j y_j), j = 0, \ldots, k\} \\ & - \{(x_j y_{j-1}), j = 1, \ldots, k\} \end{aligned}$$

and replace M by M'. Go to step 1).

Example: Search for the maximum matching in graph in Fig. 1 proceeds as follows.

1) Initial vertex labeling is $\mathbf{l} = [l(x_1), l(x_2), l(y_1), l(y_2)] = [13, 15, 0, 0]$, $G_l = \{(x_1 y_1), (x_2 y_1)\}$, $M = \{(x_2 y_1)\}$. An unmatched vertex in X is $x_0 = x_1$. After the sets S and T are updated to $S = \{x_1, x_2\}$ and $T = \{y_1\}$, (8) gives $\alpha_l = 3$.
2) New vertex labeling is $\mathbf{l} = [10, 12, 3, 0]$, $G_l = \{(x_1 y_1), (x_2 y_1), (x_2 y_2)\}$. In this G_l, there is an alternating path $(x_1 y_1 x_2 y_2)$, which leads to a new matching $M = \{(x_1 y_1), (x_2 y_2)\}$. This matching is already complete in G.

■

Computational complexity of the Kuhn–Munkres algorithm is determined by the minimization in (8), which requires $O(n^2)$ operations and has to be repeated $O(n^2)$ times. The resultant total complexity is $O(n^4)$ operations. However, if the minima $\min_{x \in S} \{l(x) + l(y) - w(xy)\}$ are computed once at the beginning for each $y \in Y - T$, then stored and updated together with each update of G_l, complexity of the second and other minimizations in (8) can be reduced to $O(n)$ operations. Thus, the total complexity of the whole algorithm is reduced to $O(n^3)$ [4], [5], [9]. This complexity is still higher than the complexity of the greedy algorithm, but in the ICA application it is quite acceptable, as the complexity is mostly negligible compared to the complexity of the ICA algorithm itself. Note that the later complexity is at least $O(n^2 N)$, where N is the length of the data $(N \gg n)$.

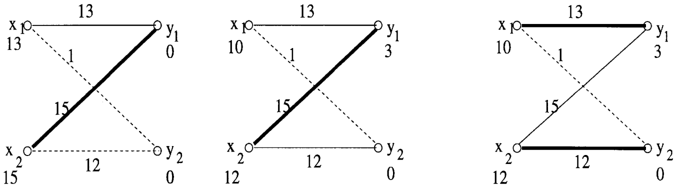

Fig. 2. Finding the maximum matching in graph in Fig. 1 via the Kuhn–Munkres algorithm. (Solid line) Edges in G_l. (Bold solid line) Matching M in G_l.

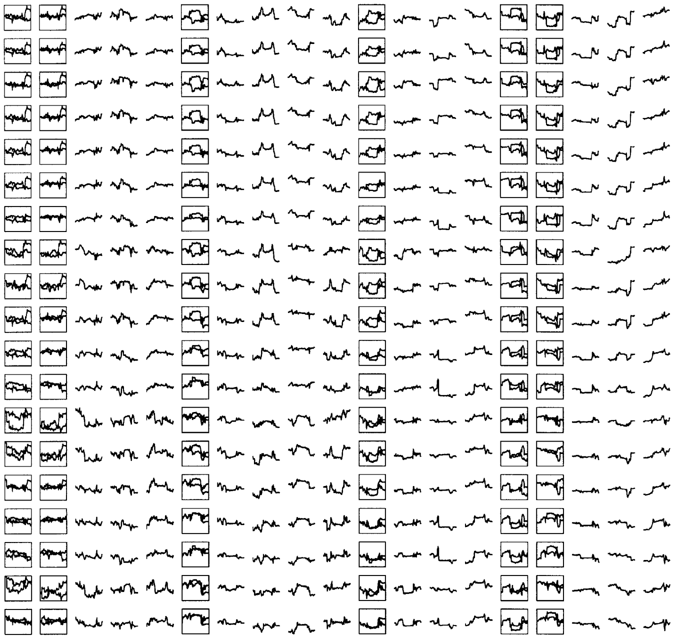

Fig. 3. The 19×19 graphs containing the evolution of elements of an estimated mixing matrix obtained by processing of a 19-channel EEG data block by the FastICA algorithm in a sliding window of the length of 6000 samples. In each step, columns of the estimated mixing matrix were reshuffled 1) by the greedy algorithm and 2) by the Kuhn–Munkres method. The results are distinct in the diagrams denoted by frames. The latter results are more stationary.

V. Experimental Results

Time evolution of the mixing matrix A in a linear ICA model (1) of EEG data[2] with $n = 19$ channels was studied using a sliding time-window. The whole dataset has $21\,000 \times 19$ samples and length of time-window was $N = 6000$. Sampling frequency was 128 Hz. In each of the $p = 15\,000$ windows, the data were processed by FastICA algorithm [10], using symmetric approach and $\tanh$ nonlinearity to estimate independent components and a mixing matrix A_k, $k = 1, \ldots, p$. In the second and all the consequent windows, the algorithm was initialized using

[1] See http://www.utia.cas.cz/user_data/scientific/SI_ dept/Tichavsky.html for a Matlab implementation.

[2] The data were provided by courtesy of V. Krajča (Department of Neurology, Bulovka Hospital, Prague, Czech Republic)

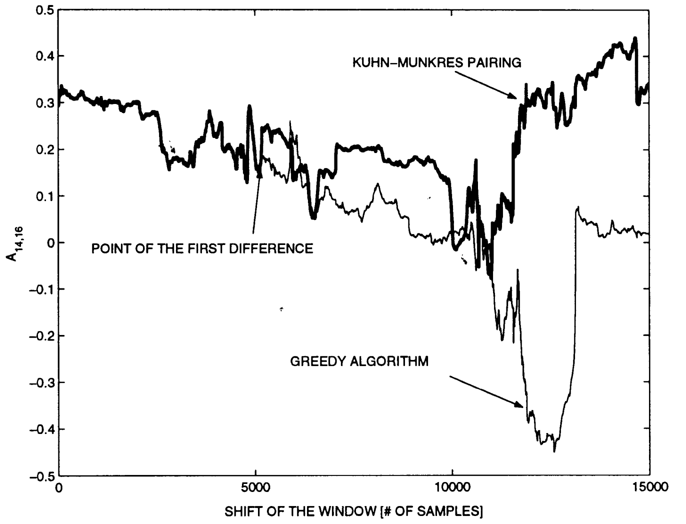

Fig. 4. The (14,16)th diagram in Fig. 3.

the mixing matrix computed in the previous window. It was observed that with this initialization the algorithm needed typically only two iterations to converge. However, the order of the estimated independent components changed in each step. Hence, the columns of matrices $A_2, \ldots, A_p$ were reshuffled, and the signs of columns were corrected, so that A_k is close to A_{k-1} for $k = 1, \ldots, p$. The reshuffling was done for (1) by the greedy method and (2) by the Kuhn–Munkres algorithm.

The resultant mixing matrix elements are shown as functions of time in Fig. 3, results for one of the elements is in Fig. 4. The figure consists of 19 × 19 graphs, each matrix element has a separate diagram. Note that for some matrix elements the estimated time evolution is identical for both matching methods. However, some matrix elements (marked with frames) have a different evolution, depending on the matching method used. The graphs are distinct always in whole columns. A closer look at the data showed that the two matching methods gave different ordering of components in three of the 14 999 time steps. In these three steps, the matrix distance $||A - \hat{A}||_F^2$ equals to 1.563, 1.961, and 0.966, respectively, for the Kuhn–Munkres algorithm, and to 1.988, 1.969, and 1.255, respectively, for the greedy algorithm.

After each of these changes, the new ordering was preserved in the further windows (and, therefore, the graphs do not differ in a few points only). It can be observed that when the Kuhn–Munkres algorithm is used, the estimated mixing matrix elements look more stationary, and also the total variance of the estimated matrix elements is lower compared to the case when the greedy algorithm is applied. While the Kuhn–Munkres algorithm finds always the optimal assignment, the greedy method does not.

VI. Conclusion

The Kuhn–Munkres algorithm is shown to be adequate for pairing of sources (signals) separated by BSS methods in different time-windows or in different frequency bins.

The experiment with the EEG data shows that the optimal pairing makes a difference and allows a smoother tracking of the mixing matrix elements. Computational complexity of the proposed pairing technique is $O(n^3)$ compared to $O(n^2)$ of the commonly used greedy algorithm.

References

[1] T. W. Lee, *Independent Component Analysis, Theory and Applications*. Boston, MA: Kluwer, 1998.

[2] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley, 2001.

[3] E. Moreau and O. Macchi, "High-order contrasts for self adaptive source separation," *Int. J. Adaptive Contr. Signal Process.*, vol. 10, no. 1, pp. 19–46, 1996.

[4] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*. Amsterdam, The Netherlands: Elsevier, 1976.

[5] L. Lovász and M. D. Plummer, *Matching Theory*. Amsterdam, The Netherlands: North-Holland, 1986.

[6] A. Dapena, M. F. Bugallo, and L. Castedo, "Separation of convolutive mixtures of temporally-white signals: A novel frequency-domain approach," in *Proc. 3rd Int. Conf. Independent Component Analysis and Blind Signal Separation*, San Diego, CA, Dec. 9–12, 2001, pp. 179–184.

[7] S. Ikeda and N. Murata, "A method of blind separation based on temporal structure of signals," in *Proc. ICONIP'98*, 1998, pp. 737–742.

[8] A. M. Daves. The optimal assignment problem. [Online]. Available: http://www.math.uwo.ca/~mdawes/courses/344/kuhn-munkres.pdf

[9] S. Khuller. Design and analysis of algorithms: Course notes. [Online]. Available: http://www.cs.umd.edu/users/samir/cs651.ps

[10] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Comput.*, vol. 6, pp. 1483–1492, 1997.