REVIEW

Blind Source Separation and Independent Component Analysis: A Review

Seungjin Choi

Department of Computer Science
Pohang University of Science and Technology
San 31, Hyoja-dong, Nam-gu, Pohang, Gyungbuk 790-784, Korea
E-mail: seungjin@postech.ac.kr

Andrzej Cichocki

RIKEN, Brain Science Institute, 2-1 Hirosawa, Wako, Saitama 351-0198, Japan
Warsaw University of Technology, Poland
E-mail: cia@bsp.brian.riken.go.jp

Hyung-Min Park and Soo-Young Lee

Department of BioSystems, Department of Electrical Engineering and Computer Science, and
CHUNG Moon Soul Center for BioInformation and BioElectronics,
Korea Advanced Institute of Science and Technology
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Korea
E-mail: {hmpark, sylee}@kaist.ac.kr

(Submitted on October 20, 2004)

Abstract - Blind source separation (BSS) and independent component analysis (ICA) are generally based on a wide class of unsupervised learning algorithms and they found potential applications in many areas from engineering to neuroscience. A recent trend in BSS is to consider problems in the framework of matrix factorization or more general signals decomposition with probabilistic generative and tree structured graphical models and exploit a *priori* knowledge about true nature and structure of latent (hidden) variables or sources such as spatio-temporal decorrelation, statistical independence, sparseness, smoothness or lowest complexity in the sense e.g., of best predictability. The possible goal of such decomposition can be considered as the estimation of sources not necessary statistically independent and parameters of a mixing system or more generally as finding a new reduced or hierarchical and structured representation for the observed (sensor) data that can be interpreted as physically meaningful coding or blind source estimation. The key issue is to find a such transformation or coding (linear or nonlinear) which has true physical meaning and interpretation. We present a review of BSS and ICA, including various algorithms for static and dynamic models and their applications. The paper mainly consists of three parts: (1) BSS algorithms for static models (instantaneous mixtures); (2) extension of BSS and ICA incorporating with sparseness or non-negativity constraints; (3) BSS algorithms for dynamic models (convolutive mixtures).

Keywords - Independent Component Analysis, Blind Source Separation, information theory, feature extraction

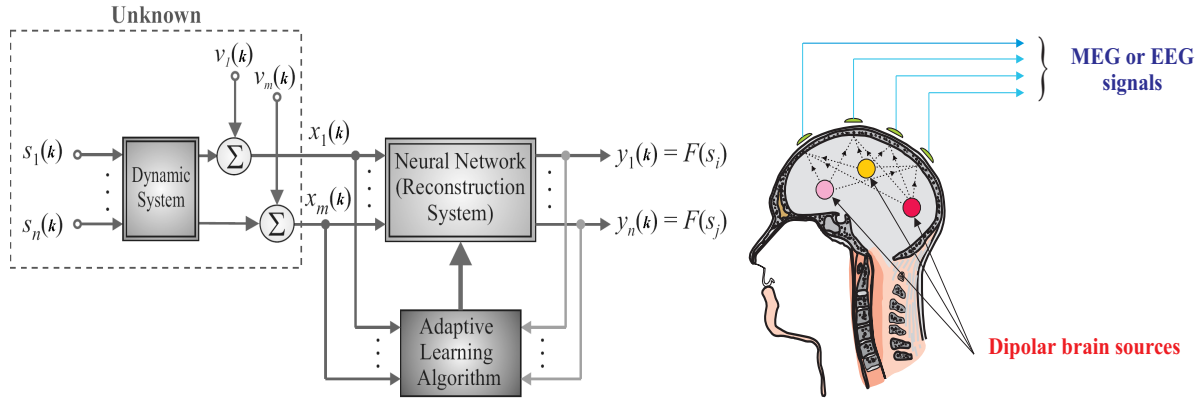


Figure 1. (a) General model illustrating blind source separation (BSS), (b) Such models are exploited in non-invasive multi-sensor recording of brain activity using EEG or MEG. It is assumed that the scalp sensors (electrodes, squids) picks up superposition neuronal brain sources and non-brain sources related to movements of eyes and muscles. Objective is to identify the individual signals coming from different areas of the brain.

1. Introduction

A fairly general blind signal separation (BSS) problem often referred as blind signal decomposition or blind source extraction (BSE) can be formulated as follows (see Figure 1 (a)).

We observe records of sensor signals $\boldsymbol{x}(k) = [x_1(k), \ldots, x_m(k)]^T$, where k is a discrete time and $(\cdot)^T$ means transpose of the vector, from an unknown MIMO (multiple-input/multiple-output) mixing and filtering system. The objective is to find an inverse system, sometimes termed a reconstruction system, neural network, or an adaptive inverse system, if it exists and is stable, in order to estimate the all primary source signals $\boldsymbol{s}(k) = [s_1(k), \ldots, s_n(k)]^T$ or only some of them with specific properties. This estimation is performed on the basis of only the output signals $\boldsymbol{y}(k) = [y_1(k), \ldots, y_n(k)]^T$ and the sensor signals. Preferably, the inverse (unmixing) system should be adaptive in such a way that it has some tracking capability in non-stationary environments. Instead of estimating the source signals directly, it is sometimes more convenient to identify an unknown mixing and filtering system first (e.g., when the inverse system does not exist, especially when system is overcomplete with the number of observations is less than the number of source signals, i.e., $m < n$) and then estimate source signals implicitly by exploiting some *a priori* information about the mixing system and applying a suitable optimization procedure. The problems of separating or extracting the original source waveforms from the sensor array, without knowing the transmission channel characteristics and the sources can be expressed briefly as a number of related BSS or blind signal decomposition problems such Independent Component Analysis (ICA) (and its extensions: Topographic ICA, Multidimensional ICA, Kernel ICA, Tree-dependent Component Analysis, Subband Decomposition -ICA), Sparse Component Analysis (SCA), Sparse PCA (SPCA), Non-negative Matrix Factorization (NMF), Smooth Component Analysis (SmoCA), Parallel Factor Analysis (PARAFAC), Time-Frequency Component Analyzer (TFCA) and Multichannel Blind Deconvolution (MBD) [4, 40, 78, 13, 90, 149, 150, 36, 95, 108].

There appears to be something magical about blind source separation; we are estimating the original source signals without knowing the parameters of mixing and/or filtering processes. It is difficult to imagine that one can estimate this at all. In fact, without some *a priori* knowledge, it is not possible to *uniquely* estimate the original source signals. However, one can usually estimate them up to certain indeterminacies. In mathematical terms these indeterminacies and ambiguities can be expressed as arbitrary scaling, permutation and delay of estimated source signals [138]. These indeterminacies preserve, however, the waveforms of original sources. Although these indeterminacies seem to be rather severe limitations, in a great number of applications these limitations are not essential, since the most relevant information about the source signals is contained in the temporal waveforms or time-frequency patterns of the source signals and usually not in their amplitudes or order in which they are arranged in the output of the system. For some dynamical models, however, there is no guarantee that the estimated or extracted signals have exactly the same waveforms as the source signals, and then the requirements must be

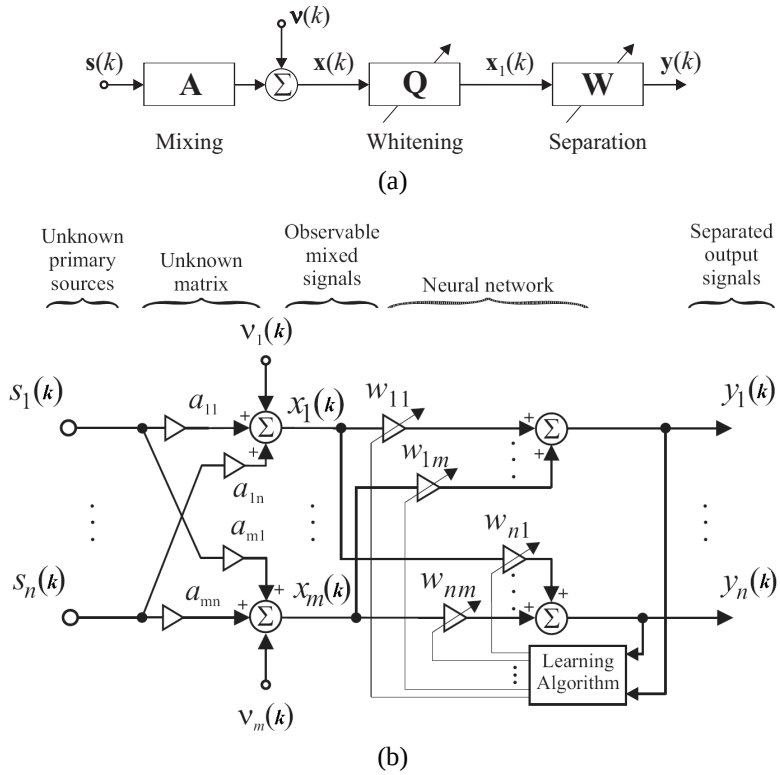


Figure 2. Block diagrams illustrating linear blind source separation or blind identification problem: (a) General schema with optional whitening, (b) Detailed model. For the overcomplete problem ($m < n$) the separating matrix $\boldsymbol{W}$ may not exist; in such cases we attempt to identify the mixing matrix $\boldsymbol{A}$ first and next to estimate sources by exploiting some *a priori* knowledge such as sparsity or independence of unknown sources.

sometimes further relaxed to the extent that the extracted waveforms are distorted (filtered or convolved) versions of the primary source signals (see Figure 1(a)).

The mixing and filtering processes of the unknown input sources s_j may have different mathematical or physical models, depending on the specific applications [78, 4, 39]. Most of linear BSS models in the simplest forms can be expressed algebraically as some specific problems of matrix factorization: Given observation (often called sensor or data) matrix $\boldsymbol{X} = [\boldsymbol{x}(1), \ldots, \boldsymbol{x}(N)] \in \mathbb{R}^{m \times N}$ perform the matrix factorization:

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{V}, \tag{1}$$

where N is the number of available samples, m is the number of observations, n is the number of sources, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ represents the unknown basis data matrix or mixing matrix (depending on applications), $\boldsymbol{V} \in \mathbb{R}^{m \times N}$ is an unknown matrix representing errors or noise and matrix, $\boldsymbol{S} = [\boldsymbol{s}(1), \ldots, \boldsymbol{s}(N)] \in \mathbb{R}^{n \times N}$ contains the corresponding latent (hidden) components that give the contribution of each basis vectors. Usually these latent components represent unknown source signals with specific statistical properties or temporal structures. The matrices have usually clear physical meanings. For example, the rows of matrix $\boldsymbol{S}$ that represent of sources should be as sparse as possible for SCA or statistically mutually independent as possible for ICA. Often it is required that estimated components are piecewise smooth (SmoCA) or take only non-negative values (NMF) or values with specific constraints [90, 44, 126].

Although some decompositions or matrix factorizations provide an exact reconstruction data (i.e., $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}$), we shall consider here decompositions which are approximative in nature. In fact, many problems in signal and image processing can be expressed in such terms of matrix factorization. Different cost functions and imposed constraints may lead to different types of matrix factorization. In many signal processing applications the data matrix $\boldsymbol{X} = [\boldsymbol{x}(1), \boldsymbol{x}(2) \ldots, \boldsymbol{x}(N)]$ is represented by vectors $\boldsymbol{x}(k)$ ($k = 1, 2, \ldots, N$) for a set of discrete time instants as multiple measurements or recordings, thus the compact aggregated matrix equation (1) can be written

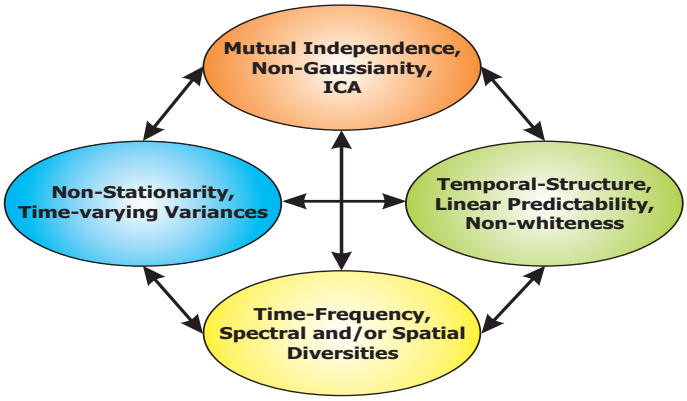


Figure 3. Basic approaches for blind source separation. Each approach exploits some *a priori* knowledge and specific properties of the source signals.

in a vector form as the system of linear equations:

$$\boldsymbol{x}(k) = \boldsymbol{A}\,\boldsymbol{s}(k) + \boldsymbol{v}(k), \qquad (k = 1, 2 \ldots, N) \tag{2}$$

where $\boldsymbol{x}(k) = [x_1(k), x_2(k), \ldots, x_m(k)]^T$ is vector of the observed signals at the discrete time instant k while $\boldsymbol{s}(k) = [s_1(k), s_2(k), \ldots, s_n(k)]^T$ is the vector of components at the same time instant[1].

The above formulated problems are related closely to linear inverse problem or more generally, to solving a large ill-conditioned system of linear equations (overdetermined or underdetermined depending on applications) where it is necessary to estimate reliably vectors $\boldsymbol{s}(k)$ and in some cases also to identify a matrix $\boldsymbol{A}$ for noisy data. Such systems of equations are often contaminated by noise or errors, thus the problem of finding an optimal and robust with respect noise solution arises. Wide classes of extrapolation, reconstruction, estimation, approximation, interpolation and inverse problems can be converted to minimum norm problems of solving underdetermined systems of linear equations (2) for $m < n$ [88, 40]. Generally speaking, in signal processing applications, the overdetermined ($m > n$) system of linear equations (2) describes filtering, enhancement, deconvolution and identification problems, while the underdetermined case describes inverse and extrapolation problems [51, 40]. In general, the number of source signals n is unknown. It is assumed that only the sensor vector $\boldsymbol{x}(k)$ is available and it is necessary to design a feed-forward or recurrent neural network and an associated adaptive learning algorithm that enables estimation of sources, identification of the mixing matrix $\boldsymbol{A}$ and/or separating matrix $\boldsymbol{W}$ with good tracking abilities. Often BSS is obtained by finding an $n \times m$, full rank, linear transformation (separating) matrix $\boldsymbol{W} = \hat{\boldsymbol{A}}^{+}$, where $\boldsymbol{A}^{+}$ means some well-defined pseudo-inverse of $\boldsymbol{A}$ such that the output signal vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]^T$, by $\boldsymbol{y} = \boldsymbol{W}\,\boldsymbol{x}$, contains components that are as independent as possible, as measured by an information-theoretic cost function such as the Kullback-Leibler divergence or other criteria like sparseness, smoothness or linear predictability [4, 40, 14, 24].

Although many different source separation algorithms are available, their principles can be summarized by the following four fundamental approaches (see Figure 3):

- The most popular approach exploits as the cost function some measure of signals statistical independence, non-Gaussianity or sparseness. When original sources are assumed to be statistically independent without a temporal structure, the higher-order statistics (HOS) are essential (implicitly or explicitly) to solve the BSS problem. In such a case, the method does not allow more than one Gaussian source.

- If sources have temporal structures, then each source has non-vanishing temporal correlation, and less restrictive conditions than statistical independence can be used, namely, second-order statistics (SOS) are often sufficient to estimate the mixing matrix and sources. Along this line, several methods have been developed [109, 19, 153, 37, 138, 42, 38]. Note that these SOS methods do not allow the separation of sources with identical power spectra shapes or i.i.d. (independent and identically distributed) sources.

[1]The data are often represented not in the time domain but rather in the complex frequency or the time frequency domain, so index k may have different meaning.

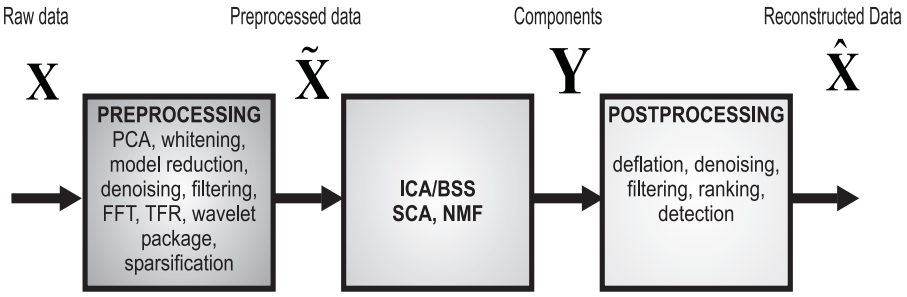


Figure 4. Fundamental three procedures implemented and exploited in the BSS/ICA for efficient decomposition and extraction of signals.

- The third approach exploits non-stationarity (NS) properties and second order statistics (SOS). Mainly, we are interested in the second-order non-stationarity in the sense that source variances vary in time. The non-stationarity was first taken into account by [107] and it was shown that a simple decorrelation technique is able for wide class of source signals to perform the BSS task. In contrast to other approaches, the non-stationarity information based methods allow the separation of colored Gaussian sources with identical power spectra shapes. However, they do not allow the separation of sources with identical non-stationarity properties. There are some recent works on non-stationary source separation [37, 36]. Methods that exploit either the temporal structure of sources (mainly second-order correlations) and/or the non-stationarity of sources, lead in the simplest scenario to the second-order statistics BSS methods. In contrast to BSS methods based on HOS, all the second-order statistics based methods do not have to infer the probability distributions of sources or nonlinear activation (score) functions (see next sections).

- The fourth approach exploits the various diversities[2] of signals, typically, time, frequency, (spectral or "time coherence") and/or time-frequency diversities, or more generally, joint space-time-frequency (STF) diversity. Such approach leads to concept of Time-Frequency Component Analyzer (TFCA) [20]. TFCA decomposes the signal into specific components in the time-frequency domain and computes the time-frequency representations (TFRs) of the individual components. Usually components are interpreted here as localized, sparse and structured signals in the time-frequency plain (spectrogram). In other words, in TFCA components are estimated by analyzing the time-frequency distribution of the observed signals. TFCA provides an elegant and promising solution to suppression of some artifacts and interference via masking and/or filtering of undesired - components.

More sophisticated or advanced approaches use combinations or integration of some of the above mentioned approaches: HOS, SOS, NS and STF (Space-Time-Frequency) diversity, in order to separate or extract sources with various statistical properties and to reduce the influence of noise and undesirable interferences.

The all above mentioned BSS methods belongs to a wide class of unsupervised learning algorithms. Unsupervised learning techniques try to discover a structure underlying a data set, extraction of meaningful features and finding useful representations of the given data [92] [81]. Since data can be always interpreted in many different ways, some knowledge is needed to determine which features or properties represent our true latent (hidden) components. For example, PCA finds a low-dimensional representation of the data that captures most of its variance. On the other hand SCA tries to explain data as mixture of sparse components (usually in time-frequency domain) and NMF seeks to explain data by parts-based localized additive representations (with non-negativity constraints).

BSS algorithms, e.g., ICA, SCA, NMF, STD and SmoCA, techniques are often considered as pure mathematical formulas, powerful, but rather mechanical procedures: There is illusion that there are not very much left for the user to do after the machinery has been optimally implemented. The successful and efficient use of the such tools strongly depends on *a priori* knowledge, common sense and appropriate use of the preprocessing and postprocessing tools. In other words, it is preprocessing of data and postprocessing of models where an expertise is truly needed in order to extract reliable, significant and physiologically meaningful components. Typical preprocessing tools include: Principal Component Analysis (PCA), Factor Analysis, (FA), model reduction, whitening, filtering, Fast Fourier Transform (FFT), Time Frequency Representation (TFR) and sparsification (wavelets package

[2]By diversities we mean usually different characteristics or features of the signals.

transformation) of data (see Figure 4). Postprocessing tools includes: Deflation and reconstruction ("cleaning") of original raw data by removing undesirable components, noise or artifacts. On the other hand, the assumed linear mixing models must be valid at least approximately and original sources signals should have specified statistical properties [40, 4, 41].

The problem of blind source separation (BSS) has received wide attention in various fields such as signal analysis and processing of speech, image, and biomedical signals (EEG, MEG, fMRI, PET), especially, signal extraction, enhancement, denoising, model reduction and classification problems [78, 40].

For many real-time applications one may need special hardwares, and both analogue [29] and digital [85] chips had been developed.

2. Blind Source Separation Based on Spatio-Temporal Decorrelation and Non-stationarity

Temporal, spatial and spatio-temporal[3] decorrelations play important roles in EEG/MEG data analysis. These techniques are based only on second-order statistics (SOS). They are the basis for modern subspace methods of spectrum analysis and array processing and are often used in a preprocessing stage in order to improve convergence properties of adaptive systems, to eliminate redundancy or to reduce noise. Spatial decorrelation or prewhitening is often considered a necessary (but not sufficient) condition for stronger stochastic independence criteria. After prewhitening, the BSS or ICA tasks usually become somewhat easier and well-posed (less ill-conditioned), because the subsequent separating (unmixing) system is described by an orthogonal matrix for real-valued signals and a unitary matrix for complex-valued signals and weights. Furthermore, spatio-temporal and time-delayed decorrelation can be used to identify the mixing matrix and to perform blind source separation of colored sources under certain weak conditions [40].

2.1 Robust Orthogonalization/Whitening

The whitening (or data sphering) is an important pre-processing step in a variety of BSS methods. The conventional whitening exploits the equal-time correlation matrix of the data $\boldsymbol{x}(k)$, so that the effect of additive noise can not be removed. The idea of a new whitening method lies in utilizing the time-delayed correlation matrices that are not sensitive to the white noise. A new whitening method is named as a *robust whitening*, motivated by the fact that it is not sensitive to the white noise. However, it is somewhat different from (Huber's) robust statistics.

The time-delayed correlation matrix of the observation data $\boldsymbol{x}(k)$ has the form

$$\begin{aligned} \boldsymbol{R}_x(\tau) &= E\{\boldsymbol{x}(k)\boldsymbol{x}^T(k-\tau)\} \\ &= \boldsymbol{A}\boldsymbol{R}_s(\tau)\boldsymbol{A}^T, \end{aligned} \tag{3}$$

for $\tau \neq 0$. One can easily see that the transformation $\boldsymbol{R}_x^{-\frac{1}{2}}(\tau)$ whiten the data $\boldsymbol{x}(k)$ without the effect of the noise vector $\boldsymbol{v}(k)$. It reduces the noise effect and project the data onto the signal subspace, in contrast to the conventional whitening transformation $\boldsymbol{R}_x^{-\frac{1}{2}}(0)$. Some source separation methods employed this robust whitening transformation [110, 33, 32, 18, 21].

In general, however, the matrix $\boldsymbol{R}_x(\tau)$ is not always positive definite, so the whitening transformation $\boldsymbol{R}_x^{-\frac{1}{2}}(\tau)$ may not be valid for some time-lag τ. The idea of the robust whitening is to consider a linear combination of several time-delayed correlation matrices, i.e.,

$$\boldsymbol{C}_x = \sum_{i=1}^{K} \alpha_i \boldsymbol{M}_x(\tau_i), \tag{4}$$

where

$$\boldsymbol{M}_x(\tau_i) = \frac{1}{2}\left\{\boldsymbol{R}_x(\tau_i) + \boldsymbol{R}_x^T(\tau_i)\right\}. \tag{5}$$

[3] Literally, space and time. Spatio-temporal data has both a spatial (i.e. location) and a temporal (i.e. time related) components.

A proper choice of $\{\alpha_i\}$ results in the positive definite matrix $\boldsymbol{C}_x$, as in the extended matrix pencil method. Once again, the FSGC method [137] can be used to find a set of coefficients $\{\alpha_i\}$ such that the matrix $\boldsymbol{C}_x$ is positive definite.

The matrix $\boldsymbol{C}_x$ has the eigen-decomposition

$$\boldsymbol{C}_x = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T, \tag{6}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]$ and

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}_1 & \\ & \mathbf{0} \end{bmatrix}, \tag{7}$$

where $\boldsymbol{D}_1 \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose diagonal elements are n principal eigenvalues of $\boldsymbol{C}_x$. Let $\boldsymbol{U}_1 = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$. Then the robust whitening transformation matrix is given by $\boldsymbol{Q} = \boldsymbol{D}_1^{-\frac{1}{2}}\boldsymbol{U}_1^T$. The transformation $\boldsymbol{Q}$ project the data onto n-dimensional signal subspace as well as carrying out whitening.

Let us denote the whitened n-dimensional data by $\overline{\mathbf{x}}(k)$

$$\begin{aligned} \overline{\mathbf{x}}(k) &= \boldsymbol{Q}\boldsymbol{x}(k) \\ &= \boldsymbol{B}\boldsymbol{s}(k) + \boldsymbol{Q}\boldsymbol{v}(k), \end{aligned} \tag{8}$$

where $\boldsymbol{B} \in \mathbb{R}^{n\times n}$. The whitened data $\overline{\mathbf{x}}(k)$ (in the sense that $\sum_{i=1}^{K} \alpha_i \boldsymbol{M}_z(\tau_i) = \boldsymbol{I}$) is a unitary mixture of sources with additive noise, i.e., $\boldsymbol{B}\boldsymbol{B}^T = \boldsymbol{I}$.

Algorithm Outline: Robust whitening

1. Estimate time-delayed correlation matrices and construct an $m \times mJ$ matrix

$$\mathcal{M} = [\boldsymbol{M}_x(\tau_1) \cdots \boldsymbol{M}_x(\tau_J)]. \tag{9}$$

Then compute the singular value decomposition (SVD) of $\mathcal{M}$, i.e.,

$$\mathcal{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{10}$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{V} \in \mathbb{R}^{mJ\times mJ}$ are orthogonal matrices, and $\boldsymbol{\Sigma}$ has nonzero entries at (i,i) position ($i = 1, \ldots, n$) and zeros elsewhere. The number of sources, n can be detected by inspecting the singular values. Define $\boldsymbol{U}_s$ by

$$\boldsymbol{U}_s = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_n], \tag{11}$$

where $\boldsymbol{u}_i$ is the ith column vector of the matrix $\boldsymbol{U}$ and $n \leq m$.

2. For $i = 1, \ldots, J$, compute

$$\boldsymbol{F}_i = \boldsymbol{U}_s^T \boldsymbol{M}_x(\tau_i)\boldsymbol{U}_s. \tag{12}$$

3. Choose any initial $\boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_J]^T$.

4. Compute

$$\boldsymbol{F} = \sum_{i=1}^{J} \alpha_i \boldsymbol{F}_i. \tag{13}$$

5. Compute a Schur decomposition of $\boldsymbol{F}$ and check if $\boldsymbol{F}$ is positive definite or not. If $\boldsymbol{F}$ is positive definite, the algorithm is terminated. Otherwise, go to Step 6.

6. Choose an eigenvector $\boldsymbol{u}$ corresponding to the smallest eigenvalue of $\boldsymbol{F}$ and update $\boldsymbol{\alpha}$ via replacing $\boldsymbol{\alpha}$ by $\boldsymbol{\alpha} + \boldsymbol{\delta}$ where

$$\boldsymbol{\delta} = \frac{\left[\boldsymbol{u}^T \boldsymbol{F}_1 \boldsymbol{u} \cdots \boldsymbol{u}^T \boldsymbol{F}_J \boldsymbol{u}\right]^T}{\|[\boldsymbol{u}^T \boldsymbol{F}_1 \boldsymbol{u} \cdots \boldsymbol{u}^T \boldsymbol{F}_J \boldsymbol{u}]\|}. \tag{14}$$

Go to step 4. This loop is terminated in a finite number of steps (see [137] for proof).

7. Compute

$$\boldsymbol{C}_x = \sum_{i=1}^{J} \alpha_i \boldsymbol{M}_x(\tau_i), \tag{15}$$

and perform an eigenvalue-decomposition of $\boldsymbol{C}_x$,

$$\boldsymbol{C}_x = [\boldsymbol{U}_{c1}, \boldsymbol{U}_{c2}] \begin{bmatrix} \boldsymbol{D}_1 & \\ & \boldsymbol{0} \end{bmatrix} [\boldsymbol{U}_{c1}, \boldsymbol{U}_{c2}]^T \tag{16}$$

where $\boldsymbol{U}_{c1}$ contains the eigenvectors associated with n principal singular values of $\boldsymbol{D}_1$.

8. The robust whitening transformation is performed by

$$\overline{\mathbf{x}}(k) = \boldsymbol{Q}\boldsymbol{x}(k), \tag{17}$$

where $\boldsymbol{Q} = \boldsymbol{D}_1^{-\frac{1}{2}} \boldsymbol{U}_{c1}^T$.

Note: In the case of $m = n$ (equal number of sources and sensors), steps 1 and 2 are not necessary. Simply we let $\boldsymbol{F}_i = \boldsymbol{M}_x(\tau_i)$.

2.2 AMUSE Algorithm and its Properties

AMUSE algorithm belongs to the group of the second-order statistics spatio-temporal decorrelation (SOS-STD) algorithms [40, 138, 48]. It provides identical or at least very similar decomposition of raw data as the well known and popular SOBI and TDSEP algorithms [19, 153]. This class algorithms are sometimes classified or referred as ICA algorithms. However, these algorithms do not exploit implicitly or explicitly statistical independence. Moreover, in the contrast to the standard higher order statistics ICA algorithms they are able to estimate colored Gaussian distributed sources and their performance in estimation of original sources is usually better if the sources have temporal structure.

AMUSE algorithm have some similarity with standard PCA. The main difference is that AMUSE employs PCA two times (in cascade way) in two separate steps: In the first step, standard PCA can be applied for whitening (sphering) data and in the second step SVD/PCA is applied for time delayed covariance matrix of the pre-whitened data. Mathematically AMUSE algorithm is the following two stage procedure: In the first step we apply a standard or robust prewhitening (sphering) as linear transformation $\boldsymbol{x}_1(k) = \boldsymbol{Q}\boldsymbol{x}(k)$ where $\boldsymbol{Q} = \boldsymbol{R}_x^{-1/2} = (\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T)^{-1/2} = \boldsymbol{V}(\boldsymbol{\Lambda})^{-1/2}\boldsymbol{V}^T$ of the standard covariance matrix $\boldsymbol{R}_{xx} = E\{\boldsymbol{x}(k)\boldsymbol{x}^T(k)\}$ and $\boldsymbol{x}(k)$ is a vector of observed data for time instant k. Next, (for pre-whitened data) the SVD is applied for time-delayed covariance matrix $\boldsymbol{R}_{x1x1} = E\{\boldsymbol{x}_1(k)\boldsymbol{x}_1^T(k-1)\} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, where $\boldsymbol{\Sigma}$ is diagonal matrix with decreasing singular values and $\boldsymbol{U}$, $\boldsymbol{V}$ are orthogonal matrices of left and right singular vectors. Then, an unmixing (separating) matrix is estimated as $\boldsymbol{W} = \boldsymbol{U}^T\boldsymbol{Q}$ [40].

The main advantage of AMUSE algorithm in comparison to other BSS/ICA algorithms is that it allows us automatically to order components due to application of SVD (singular value decomposition). In fact, the components are ordered according decreasing values of singular values of the time-delayed covariance matrix. In other words, AMUSE algorithm exploit a simple principle that the estimated components tends to be less complex or more precisely, they have better linear predictability than any mixture of those sources. It should be emphasized that all components estimated by AMUSE are uniquely defined and consistently ranked. Consistent ranking is due to the fact that these singular values are always ordered in decreasing order [48]. For real-world data probability that two singular values achieve the exactly same value is very small, so ordering is very consistent and unique.

The one disadvantage of AMUSE algorithm is that is relatively sensitive to additive noise since the algorithm exploits only one time delayed covariance matrix.

2.3 Robust SOBI Alagorithm

There is a current trend in ICA/BSS to investigate the "average eigen-structure" of a large set of data matrices formed as functions of available data (typically, covariance or cumulant matrices for different time delays). In other words, the objective is to extract reliable information (e.g., estimation of sources and/or the mixing matrix) from the eigen-structure of a possibly large set of data matrices [152, 27, 40]. However, since in practice we only have a finite number of samples of signals corrupted by noise, the data matrices do not exactly share the same eigen-structure. Furthermore, it should be noted that determining the eigen-structure on the basis of one or even two data matrices leads usually to poor or unsatisfactory results because such matrices, based usually on an arbitrary choice, may have some degenerate eigenvalues which leads to loss of information contained in other data matrices. Therefore, from a statistical point of view, in order to provide robustness and accuracy, it is necessary to consider the average eigen-structure by taking into account simultaneously a possibly large set of data matrices [40, 8, 152].

The average eigen-structure can be easily implemented via linear combination of several time-delayed covariance matrices and applying the standard EVD or SVD. An alternative approach to EVD/SVD is to apply the approximate joint diagonalization procedure (JAD). The objective of this procedure is to find the orthogonal matrix $\boldsymbol{U}$ which diagonalizes a set of matrices [152, 27]:

$$\boldsymbol{R}_x(p_i) = \boldsymbol{U}\boldsymbol{D}_i\boldsymbol{U}^T + \boldsymbol{\varepsilon}_i, \qquad (i = 1, 2, \ldots, L) \tag{18}$$

where $\boldsymbol{R}_x(p_i) \in \mathbb{R}^{n\times n}$ are data matrices (for example, time-delayed covariance matrices $\boldsymbol{R}_x(p_i) = E\{\boldsymbol{x}(k)\boldsymbol{x}^T(k-p_i)\}$ and/or cumulant matrices), the $\boldsymbol{D}_i$ are diagonal and real, and $\boldsymbol{\varepsilon}_i$ represent additive errors or noise matrix (as small as possible). If $L > 2$ for arbitrary matrices $\boldsymbol{R}_x(p_i)$, the problem becomes overdetermined and generally we can not find an exact diagonalizing matrix $\boldsymbol{U}$ with $\boldsymbol{\varepsilon}_i = \boldsymbol{0},\ \forall i$. An important advantage of the Joint Approximate Diagonalization (JAD) is that several numerically efficient algorithms exist for its computation, including Jacobi techniques (one sided and two sided), Alternating Least Squares (ALS), PARAFAC (Parallel Factor Analysis) and subspace fitting techniques employing the efficient Gauss-Newton optimization [152].

This idea has been implemented in robust SOBI algorithm which can be briefly outlined as follows:

1. Perform robust orthogonalization $\overline{\mathbf{x}}(k) = \boldsymbol{Q}\,\boldsymbol{x}(k)$ similar as in AMUSE algorithm.

2. Estimate the set of covariance matrices:

$$\widehat{\boldsymbol{R}}_{\overline{\mathbf{x}}}(p_i) = (1/N)\sum_{k=1}^{N} \overline{\mathbf{x}}(k)\,\overline{\mathbf{x}}^T(k-p_i) = \boldsymbol{Q}\,\widehat{\boldsymbol{R}}_{\boldsymbol{x}}(p_i)\,\boldsymbol{Q}^T \tag{19}$$

   for a preselected set of time lags $(p_1, p_2, \ldots, p_L)$ or band-pass filters B_i.

3. Perform JAD: $\boldsymbol{R}_{\overline{\mathbf{x}}}(p_i) = \boldsymbol{U}\boldsymbol{D}_i\boldsymbol{U}^T,\ \ \forall i$, i.e., estimate the orthogonal matrix $\boldsymbol{U}$ using one of the available numerical algorithm.

4. Estimate the source signals as $\widehat{\boldsymbol{s}}(k) = \boldsymbol{U}^T\,\boldsymbol{Q}\,\boldsymbol{x}(k)$ and the mixing matrix as $\widehat{\boldsymbol{A}} = \boldsymbol{Q}^+\,\boldsymbol{U}$.

The main advantage of the SOBI algorithm is its robustness in respect additive noise if number of covariance matrices is sufficiently large (typically more than 100).

2.4 SEONS: Incorporating with Non-stationarity

The SEcond-Order Non-stationary Source separation (SEONS) algorithm is illustrated here. The set of all matrices of the form $\boldsymbol{R}_1 - \lambda\boldsymbol{R}_2$ with $\lambda \in \mathbb{R}$ is said to be a *pencil*. Frequently we encounter into the case where $\boldsymbol{R}_1$ is symmetric and $\boldsymbol{R}_2$ is symmetric and positive definite. Pencils of this variety are referred to as *symmetric-definite pencils* [71].

Theorem 1 (pp. 468 in [71]) *If $\boldsymbol{R}_1 - \lambda \boldsymbol{R}_2$ is symmetric-definite, then there exists a nonsingular matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$ such that*

$$\boldsymbol{U}^T \boldsymbol{R}_1 \boldsymbol{U} = \mathit{diag}\left\{\gamma_1(\tau_1), \ldots, \gamma_n(\tau_1)\right\}, \tag{20}$$

$$\boldsymbol{U}^T \boldsymbol{R}_2 \boldsymbol{U} = \mathit{diag}\left\{\gamma_1(\tau_2), \ldots, \gamma_n(\tau_2)\right\}. \tag{21}$$

Moreover $\boldsymbol{R}_1 \boldsymbol{u}_i = \lambda_i \boldsymbol{R}_2 \boldsymbol{u}_i$ for $i = 1, \ldots, n$, and $\lambda_i = \frac{\gamma_i(\tau_1)}{\gamma_i(\tau_2)}$.

It is apparent from Theorem 1 that $\boldsymbol{R}_1$ should be symmetric and $\boldsymbol{R}_2$ should be symmetric and positive definite so that the generalized eigenvector $\boldsymbol{U}$ can be a valid solution if $\{\lambda_i\}$ are distinct. Unfortunately in [28], the symmetric-definite pencil was not considered, so we might have a numerical instability problem in the calculation of the generalized eigenvectors.

Now we explain how we construct a symmetric-definite pencil. Let us consider two time-delayed correlation matrices $\boldsymbol{R}_x(\tau_1)$ and $\boldsymbol{R}_x(\tau_2)$ for nonzero time-lags τ_1 and τ_2. For the requirement of symmetry, we replace $\boldsymbol{R}_x(\tau_1)$ and $\boldsymbol{R}_x(\tau_2)$ by $\boldsymbol{M}_x(\tau_1)$ and $\boldsymbol{M}_x(\tau_2)$ that are defined by

$$\boldsymbol{M}_x(\tau_1) = \frac{1}{2}\left\{\boldsymbol{R}_x(\tau_1) + \boldsymbol{R}_x^T(\tau_1)\right\}, \tag{22}$$

$$\boldsymbol{M}_x(\tau_2) = \frac{1}{2}\left\{\boldsymbol{R}_x(\tau_2) + \boldsymbol{R}_x^T(\tau_2)\right\}. \tag{23}$$

Then the pencil $\boldsymbol{M}_x(\tau_2) - \lambda \boldsymbol{M}_x(\tau_1)$ is a symmetric pencil. In general, the matrix $\boldsymbol{M}_x(\tau_1)$ is not positive definite for $\tau_1 \neq 0$. Thus instead of $\boldsymbol{M}_x(\tau_1)$, we consider a linear combination of several time-delayed correlation matrices, i.e.,

$$\boldsymbol{C}_1 = \sum_{i=1}^{J} \alpha_i \boldsymbol{M}_x(\tau_i). \tag{24}$$

The set of coefficients, $\{\alpha_i\}$, is chosen in such a way that the symmetric matrix $\boldsymbol{C}_1$ is positive definite. One simple way to do this is to use the finite step global convergence (FSGC) algorithm [137]. This method is referred to as *Extended Matrix Pencil Method* that is summarized below.

Algorithm Outline: Extended Matrix Pencil Method

1. Compute $\boldsymbol{M}_x(\tau_2)$ for some time-lag $\tau_2 \neq 0$ and calculate the matrix $\boldsymbol{C}_1 = \sum_{i=1}^{J} \alpha_i \boldsymbol{M}_x(\tau_i)$ by the FSGC method.
2. Find the generalized eigenvector matrix $\boldsymbol{V}$ of the pencil $\boldsymbol{M}_x(\tau_2) - \lambda \boldsymbol{C}_1$ which satisfies
$$\boldsymbol{M}_x(\tau_2)\boldsymbol{V} = \boldsymbol{C}_1 \boldsymbol{V} \boldsymbol{\Lambda}. \tag{25}$$
3. The unmixing matrix is given by $\boldsymbol{W} = \boldsymbol{V}^T$.

Now we consider the case where sources are second-order non-stationary and have non-vanishing temporal correlations. It follows from the assumptions (AS1)-(AS3) that we have

$$\boldsymbol{M}_x(k_r, \tau_i) = \boldsymbol{A}\boldsymbol{M}_s(k_r, \tau_i)\boldsymbol{A}^T, \tag{26}$$

for $\tau_i \neq 0$. In practice $\boldsymbol{M}_x(k_r, \tau_i)$ is computed using the samples in the rth time-windowed data frame, i.e.,

$$\boldsymbol{R}_x(k_r, \tau_i) = \frac{1}{N_r} \sum_{k \in \mathcal{N}_r} \boldsymbol{x}(k)\boldsymbol{x}^T(k - \tau_i),$$

$$\boldsymbol{M}_x(k_r, \tau_i) = \frac{1}{2}\left\{\boldsymbol{R}_x(k_r, \tau_i) + \boldsymbol{R}_x^T(k_r, \tau_i)\right\},$$

where $\mathcal{N}_r$ is a set of data points in the rth time-windowed frame and N_r is the number of data points in $\mathcal{N}_r$. It is straightforward to see that the extended matrix pencil method can be also applied to the case of non-stationary sources.

Algorithm Outline: Extended Matrix Pencil Method (non-stationary case)

1. We partition the observation data into two non-overlapping blocks, $\{\mathcal{N}_1, \mathcal{N}_2\}$.
2. Compute $\boldsymbol{M}_x(k_2, \tau_2)$ for some time-lag $\tau_2 \neq 0$ using the data points in $\mathcal{N}_2$.
3. Calculate the matrix $\boldsymbol{C}_1(k_1) = \sum_{i=1}^{J} \alpha_i \boldsymbol{M}_x(k_1, \tau_i)$ by the FSGC method using the data points in $\mathcal{N}_1$.
4. Find the generalized eigenvector matrix $\boldsymbol{V}$ of the pencil $\boldsymbol{M}_x(k_2, \tau_2) - \lambda \boldsymbol{C}_1(k_1)$ which satisfies

$$\boldsymbol{M}_x(k_2, \tau_2)\boldsymbol{V} = \boldsymbol{C}_1(k_1)\boldsymbol{V}\boldsymbol{\Lambda}. \tag{27}$$

3. The unmixing matrix is given by $\boldsymbol{W} = \boldsymbol{V}^T$.

Remarks: The method in [132] employed two matrices $\boldsymbol{R}_x(k_1, 0)$ and $\boldsymbol{R}(k_2, 0)$ to estimate the unmixing matrix.

In order to improve the statistical efficiency, we can employ the joint approximate diagonalization method [23] in our case, as in the JADE and SOBI. The joint approximate diagonalization method in [23] finds an unitary transformation that jointly diagonalizes several matrices (which do not have to be symmetric nor positive definite). The method SEONS is based on this joint approximate diagonalization. In this sense the SEONS includes the SOBI as its special case (if sources are stationary). The algorithm is summarized below.

Algorithm Outline: SEONS

---

1. The robust whitening method (described in Section **??**) is applied to obtain the whitened vector $\overline{\mathbf{x}}(k) = \boldsymbol{Q}\boldsymbol{x}(k)$. In the robust whitening step, we used the whole available data points.
2. Divide the whitened data $\{\overline{\mathbf{x}}(k)\}$ into K non-overlapping blocks and calculate $\boldsymbol{M}_{\overline{\mathbf{x}}}(k_r, \tau_j)$ for $r = 1, \ldots, K$ and $j = 1, \ldots, J$. In other words, at each time-windowed data frame, we compute J different time-delayed correlation matrices of $\overline{\mathbf{x}}(k)$.
3. Find a unitary joint diagonalizer $\boldsymbol{V}$ of $\{\boldsymbol{M}_{\overline{\mathbf{x}}}(k_r, \tau_j)\}$ using the joint approximate diagonalization method in [23], which satisfies

$$\boldsymbol{V}^T \boldsymbol{M}_{\overline{\mathbf{x}}}(k_r, \tau_j)\boldsymbol{V} = \boldsymbol{\Lambda}_{r,j}, \tag{28}$$

   where $\{\boldsymbol{\Lambda}_{r,j}\}$ is a set of diagonal matrices.

4 The unmixing matrix is computed as $\boldsymbol{W} = \boldsymbol{V}^T\boldsymbol{Q}$.

Recently Pham [121] developed a joint approximate diagonalization method where non-unitary joint diagonalizer of several Hermitian positive matrices is computed by a way similar to the classical Jacobi method. Second-order non-stationarity was also exploited in [122], but only noise-free data was considered. The following extended Pham-Cardoso method generalizes the method in [122]. One advantage of the extended Pham-Cardoso is the fact that it does not require the whitening step because the joint approximate diagonalization method in [122] finds a non-unitary joint diagonalizer. However, it requires that the set of matrices to be diagonalized should be Hermitian and positive definite, so we need to find a linear combination of time-delayed correlation matrices that is positive definite at each data frame, which increase the computational complexity.

Algorithm Outline: Extended Pham-Cardoso

---

1. Divide the data $\{\boldsymbol{x}(k)\}$ into K non-overlapping blocks and calculate $\boldsymbol{M}_x(k_r, \tau_j)$ for $r = 1, \ldots, K$ and $j = 1, \ldots, J$.

2. At each data frame, we compute

$$\boldsymbol{C}_r = \sum_{i=1}^{J} \alpha_i^{(k)} \boldsymbol{M}_x(k_r, \tau_i) \tag{29}$$

by the FSGC method for $r = 1, \ldots, K$. Note that $\{\boldsymbol{C}_r\}$ is symmetric and positive definite.

3. Find a non-unitary joint diagonalizer $\boldsymbol{V}$ of $\{\boldsymbol{C}_r\}$ using the joint approximate diagonalization method in [121], which satisfies

$$\boldsymbol{V}\boldsymbol{C}_r\boldsymbol{V}^T = \boldsymbol{\Lambda}_r, \tag{30}$$

where $\{\boldsymbol{\Lambda}_r\}$ is a set of diagonal matrices.

4. The unmixing matrix is computed as $\boldsymbol{W} = \boldsymbol{V}$.

---

3. Blind Source Extraction Using Linear Predictability and Adaptive Band-Pass Filters

There are two main approaches to solve the problem of blind separation and deconvolution. The first approach, which was mentioned briefly in the previous section, is to simultaneously separate all sources. In the second one, we extract sources sequentially in a blind fashion, one by one, rather than separating them all simultaneously. In many applications, a large number of sensors (electrodes, sensors, microphones or transducers) are available but only a very few source signals are subjects of interest. For example, in the modern EEG or MEG devices, we observe typically more than 100 sensor signals, but only a few source signals are interesting; the rest can be considered as interfering noise. In another example, the cocktail party problem, it is usually essential to extract the voices of specific persons rather than separate all the source signals of all speakers available (in mixing form) from an array of microphones. For such applications it is essential to develop and apply reliable, robust and effective learning algorithms which enable us to extract only a small number of source signals that are potentially interesting and contain useful information. The blind source extraction (BSE) approach may have several advantages over simultaneous blind separation/deconvolution, such as.

- Only "interesting" signals need to be extracted. For example, if the source signals are mixed with a large number of noise terms, we may extract only specific signals which possess some desired statistical properties.
- Signals can be extracted in a *specified order* according to the statistical features of the source signals, e.g., in the order determined by absolute values of generalized normalized kurtosis. Blind extraction of sources can be considered as a generalization of sequential extraction of principal components, where decorrelated output signals are extracted according to the decreasing order of their variances.
- The available learning algorithms for BSE are purely local, global stable and typically biologically plausible.

We can use two different models and criteria. The first criterion is based on higher order statistics (HOS) which assumes that the sources are mutually statistically independent and non-Gaussian (at most only one can be Gaussian). For independence criteria, we will use some measures of non-Gaussianity [40].

The second criterion, based on the concept of linear predictability and assumes that source signals have some temporal structure, i.e., the sources are colored with different autocorrelation functions or equivalently have different spectra shapes. In this approach, we exploit the temporal structure of signals rather than their statistical independence [49, 134]. Intuitively speaking, the source signals s_j have less complexity than the mixed sensor signals x_j. In other words, the degree of temporal predictability of any source signal is higher than (or equal to) that of any mixture. For example, waveforms of a mixture of two sine waves with different frequencies are more complex or less predictable than either of the original sine waves. This means that applying the standard linear predictor model and minimizing the mean squared error $E\{\varepsilon^2\}$, which is measure of predictability, we can separate or extract signals with different temporal structures. More precisely, by minimizing the error, we maximize a measure of temporal predictability for each recovered signal [50, 47].

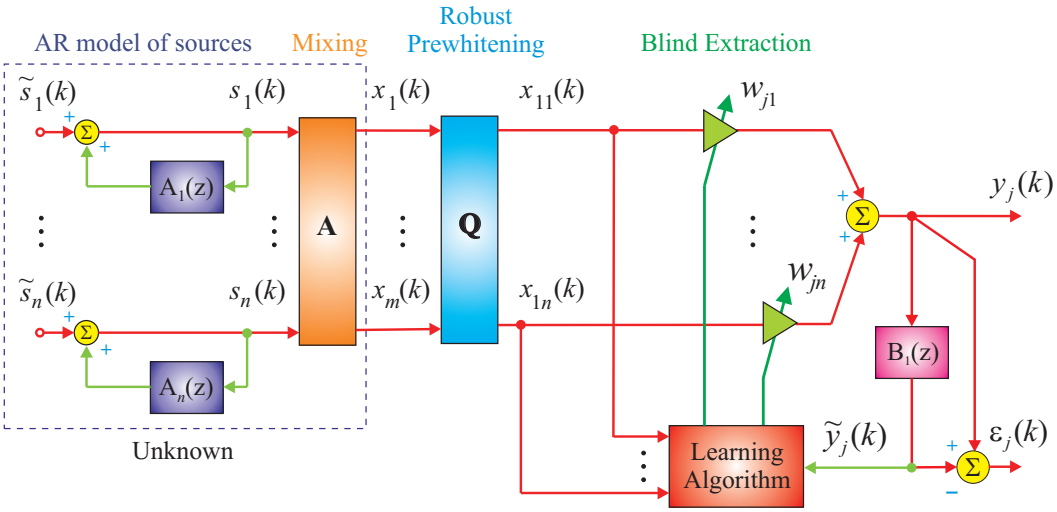


Figure 5. Block diagram illustrating implementation of learning algorithm for blind extraction of a temporally correlated source.

It is worth to note that two criteria used in BSE: temporal linear predictability and non-Gaussianity based on kurtosis may lead to different results. Temporal predictability forces the extracted signal to be smooth and possibly less complex while the non-Gaussianity measure forces the extracted signals to be as independent as possible with sparse representation for sources that have positive kurtosis.

Let us assume that temporally correlated source signals are modelled by autoregressive processes (AR) (see Figure 5) as

$$s_j(k) \;=\; \widetilde{s}_j(k) + \sum_{p=1}^{L} \widetilde{a}_{jp} s_j(k-p) = \widetilde{s}_j(k) + A_j(z) s_j(k), \tag{31}$$

where $A_j(z) = \sum_{p=1}^{L} \widetilde{a}_{jp}\, z^{-p}$, $z^{-p}s(k) = s(k-p)$ and $\widetilde{s}_j(k)$ are i.i.d. unknown innovative processes. In practice, the AR model can be extended to more general models such as the Auto Regressive Moving Average (ARMA) model or the Hidden Markov Model (HMM) [40, 78, 7].

For ill-conditioned problems (when a mixing matrix is ill-conditioned and/or source signals have different amplitudes), we can apply optional preprocessing (prewhitening) to the sensor signals $\mathbf{x}$ in the form

$$\mathbf{x}_1 = \mathbf{Q}\mathbf{x},$$

where $\boldsymbol{Q} \in \mathbb{R}^{n\times m}$ is a decorrelation matrix ensuring that the auto-correlation matrix $\mathbf{R}_{\mathbf{x}_1\mathbf{x}_1} = E\{\mathbf{x}_1\mathbf{x}_1^T\} = \mathbf{I}_n$ is an identity matrix.

To model temporal structures of source signals, we consider a linear processing unit with an adaptive filter with the transfer function $B_1(z)$ (which estimates one $A_j(z)$) as illustrated in Figure 5.

Let us assume for simplicity, that we want to extract only one source signal, e.g. $s_j(k)$, from the available sensor vector $\boldsymbol{x}(k)$. For this purpose, we employ a single processing unit described as (see Figure 6):

$$y_1\,(k) \;=\; \boldsymbol{w}_1^T \boldsymbol{x}\,(k) = \sum_{i=1}^{m} w_{1i}\; x_i\,(k)\,, \tag{32}$$

$$\boldsymbol{\varepsilon}_1\,(k) \;=\; y_1\,(k) - \sum_{p=1}^{L} b_{1p}\; y_1\,(k-p) = \boldsymbol{w}_1^T \boldsymbol{x}\,(k) - \boldsymbol{b}_1^T \bar{\boldsymbol{y}}_1(k), \tag{33}$$

where $\boldsymbol{w}_1 = [w_{11}, w_{12}, \ldots, w_{1m}]^T$, $\bar{\boldsymbol{y}}_1(k) = [y_1\,(k-1)\,, y_1(k-2), \ldots, y_1\,(k-L)]^T$, $\boldsymbol{b}_1 = [b_{11}, b_{12}, \ldots, b_{1L}]^T$ and $B_1\,(z) = \sum_{p=1}^{L} b_{1p} z^{-p}$ is the transfer function of the corresponding FIR filter. It

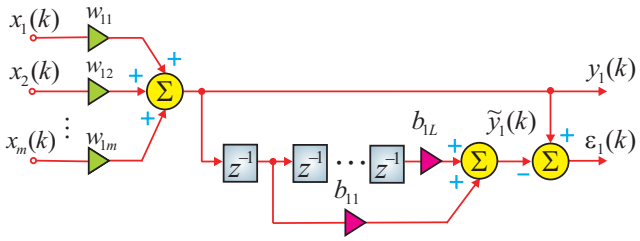


Figure 6. The neural network structure of single extraction unit using a linear predictor.

should be noted that the FIR filter can have a sparse representation. In particular, only one single processing unit, e.g. with delay p and $b_{1p} \neq 0$ can be used instead of L parameters. The processing unit has two outputs: $y_1(k)$ which estimates the extracted source signals, and $\varepsilon_1(k)$, which represents a linear prediction error or estimation of the innovation, after passing the output signal $y_1(k)$ through FIR filter.

Our objective is to estimate optimal values of vectors $\boldsymbol{w}_1$ and $\boldsymbol{b}_1$, in such a way that the processing unit successfully extracts one of the sources. This is achieved if the global vector defined as $\boldsymbol{g}_1 = \boldsymbol{A}^T \boldsymbol{w}_1 = \left(\boldsymbol{w}_1^T \boldsymbol{A}\right)^T = c_j \boldsymbol{e}_j$ contains only one nonzero element, e.g. in the j-th row, such that $y_1(k) = c_j s_j$, where c_j is an arbitrary nonzero scaling factor. For this purpose, we reformulate the problem as a minimization of the cost function

$$\mathcal{J}(\boldsymbol{w}_1, \boldsymbol{b}_1) = E\left\{\varepsilon_1^2\right\}. \tag{34}$$

The main motivation for applying such a cost function is the assumption that primary source signals (signals of interest) have temporal structures and can be modelled, e.g., by an autoregressive model [40, 16, 67, 82].

According to the AR model of source signals, the filter output can be represented as $\varepsilon_1(k) = y_1(k) - \tilde{y}_1(k)$, where $\tilde{y}_1(k) = \sum_{p=1}^{L} b_{1p} y_1(k-p)$ is defined as an error or estimator of the innovation source $\tilde{s}_j(k)$. The mean squared error $E\{\varepsilon_1^2(k)\}$ achieves a minimum $c_1^2 E\{\tilde{s}_j^2(k)\}$, where c_1 is a positive scaling constant, if and only if $y_1 = \pm c_1 s_j$ for any $j \in \{1, 2, \ldots, m\}$ or $y_1 = 0$ holds.

Let us consider the processing unit shown in Figure 6. The associated cost function (34) can be evaluated as follows:

$$E\left\{\varepsilon_1^2\right\} = \boldsymbol{w}_1^T \widehat{\boldsymbol{R}}_{\boldsymbol{x}_1 \boldsymbol{x}_1} \boldsymbol{w}_1 - 2\boldsymbol{w}_1^T \widehat{\boldsymbol{R}}_{\boldsymbol{x}_1 \bar{\boldsymbol{y}}_1} \boldsymbol{b}_1 + \boldsymbol{b}_1^T \widehat{\boldsymbol{R}}_{\bar{\boldsymbol{y}}_1 \bar{\boldsymbol{y}}_1} \boldsymbol{b}_1, \tag{35}$$

where $\widehat{\boldsymbol{R}}_{\boldsymbol{x}_1 \boldsymbol{x}_1} \approx E\{\boldsymbol{x}_1 \boldsymbol{x}_1^T\}$, $\widehat{\boldsymbol{R}}_{\boldsymbol{x}_1 \bar{\boldsymbol{y}}_1} \approx E\{\boldsymbol{x}_1 \bar{\boldsymbol{y}}_1^T\}$ and $\widehat{\boldsymbol{R}}_{\bar{\boldsymbol{y}}_1 \bar{\boldsymbol{y}}_1} \approx E\{\bar{\boldsymbol{y}}_1 \bar{\boldsymbol{y}}_1^T\}$, are estimators of true values of correlation and cross-correlation matrices: $\boldsymbol{R}_{\boldsymbol{x}_1 \boldsymbol{x}_1}, \boldsymbol{R}_{\boldsymbol{x}_1 \bar{\boldsymbol{y}}_1}, \boldsymbol{R}_{\bar{\boldsymbol{y}}_1 \bar{\boldsymbol{y}}_1}$, respectively. In order to estimate vectors $\boldsymbol{w}_1$ and $\boldsymbol{b}_1$, we evaluate gradients of the cost function and equalize them to zero as follows:

$$\frac{\partial \mathcal{J}_1(\boldsymbol{w}_1, \boldsymbol{b}_1)}{\partial \boldsymbol{w}_1} = 2\widehat{\boldsymbol{R}}_{\boldsymbol{x}_1 \boldsymbol{x}_1} \boldsymbol{w}_1 - 2\widehat{\boldsymbol{R}}_{\boldsymbol{x}_1 \bar{\boldsymbol{y}}_1} \boldsymbol{b}_1 = \boldsymbol{0}, \tag{36}$$

$$\frac{\partial \mathcal{J}_1(\boldsymbol{w}_1, \boldsymbol{b}_1)}{\partial \boldsymbol{b}_1} = 2\widehat{\boldsymbol{R}}_{\bar{\boldsymbol{y}}_1 \bar{\boldsymbol{y}}_1} \boldsymbol{b}_1 - 2\widehat{\boldsymbol{R}}_{\bar{\boldsymbol{y}}_1 \boldsymbol{x}_1} \boldsymbol{w}_1 = \boldsymbol{0}. \tag{37}$$

Solving the above matrix equations, we obtain a simple iterative algorithm:

$$\tilde{\boldsymbol{w}}_1 = \widehat{\boldsymbol{R}}_{\boldsymbol{x}_1 \boldsymbol{x}_1}^{-1} \widehat{\boldsymbol{R}}_{\boldsymbol{x}_1 \bar{\boldsymbol{y}}_1} \boldsymbol{b}_1, \quad \boldsymbol{w}_1 = \frac{\tilde{\boldsymbol{w}}_1}{||\tilde{\boldsymbol{w}}_1||_2}, \tag{38}$$

$$\boldsymbol{b}_1 = \widehat{\boldsymbol{R}}_{\bar{\boldsymbol{y}}_1 \bar{\boldsymbol{y}}_1}^{-1} \widehat{\boldsymbol{R}}_{\bar{\boldsymbol{y}}_1 \boldsymbol{x}_1} \boldsymbol{w}_1 = \widehat{\boldsymbol{R}}_{\bar{\boldsymbol{y}}_1 \bar{\boldsymbol{y}}_1}^{-1} \widehat{\boldsymbol{R}}_{\bar{\boldsymbol{y}}_1 y_1}, \tag{39}$$

where the matrices $\widehat{\boldsymbol{R}}_{\bar{\boldsymbol{y}}_1 \bar{\boldsymbol{y}}_1}$ and $\widehat{\boldsymbol{R}}_{\bar{\boldsymbol{y}}_1 y_1}$ are estimated based on the parameters $\boldsymbol{w}_1$ obtained in the previous iteration step. In order to avoid the trivial solution $\boldsymbol{w}_1 = \boldsymbol{0}$, we normalize the vector $\boldsymbol{w}_1$ to unit length in each iteration step as $\boldsymbol{w}_1(l+1) = \tilde{\boldsymbol{w}}_1(l+1) / \|\tilde{\boldsymbol{w}}_1(l+1)\|_2$ (which ensures that $E\{y_1^2\} = 1$).

It is worth to note here that in our derivation matrices $\widehat{\boldsymbol{R}}_{\bar{\boldsymbol{y}}_1 \bar{\boldsymbol{y}}_1}$ and $\widehat{\boldsymbol{R}}_{\bar{\boldsymbol{y}}_1 y_1}$ are assumed to be independent of the vector $\boldsymbol{w}_1(l+1)$, i.e., they are estimated based on $\boldsymbol{w}_1(l)$ in the previous iteration step. This two-phase

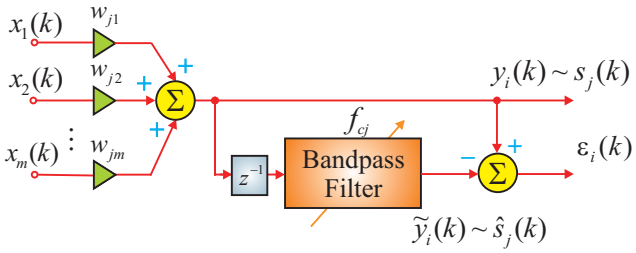


Figure 7. The conceptual model of single processing unit for extraction of sources using an adaptive band-pass filter.

procedure is similar to the expectation maximization (EM) scheme: (i) Freeze the correlation and cross-correlation matrices and learn the parameters of the processing unit $(\boldsymbol{w}_1, \boldsymbol{b}_1)$; (ii) freeze $\boldsymbol{w}_1$ and $\boldsymbol{b}_1$ and learn new statistics (i.e., matrices $\widehat{\boldsymbol{R}}_{\bar{\boldsymbol{y}}_1 y_1}$ and $\boldsymbol{R}_{\bar{\boldsymbol{y}}_1 \bar{\boldsymbol{y}}_1}$) of the estimated source signal, then go back to (i) and repeat. Hence, in phase (i), our algorithm extracts a source signal, whereas in phase (ii) it learns the statistics of the source.

The above algorithm can be further considerably simplified. It should be noted that in order to avoid inversion of the autocorrelation matrix $\boldsymbol{R}_{\boldsymbol{x}_1 \boldsymbol{x}_1}$ in each iteration step, we can perform the standard prewhitening or standard PCA as a preprocessing step and then normalize the sensor signals to unit variance. In such cases, $\widehat{\boldsymbol{R}}_{\boldsymbol{x}_1 \boldsymbol{x}_1} = \boldsymbol{I}_n$ and the algorithm is simplified to [16]

$$\tilde{\boldsymbol{w}}_1 = \widehat{\boldsymbol{R}}_{\boldsymbol{x}_1 \bar{\boldsymbol{y}}_1} \boldsymbol{b}_1 = \widehat{\boldsymbol{R}}_{\boldsymbol{x}_1 \tilde{y}_1}, \quad \boldsymbol{w}_1 = \frac{\tilde{\boldsymbol{w}}_1}{||\tilde{\boldsymbol{w}}_1||_2}, \tag{40}$$

where $\widehat{\boldsymbol{R}}_{\boldsymbol{x}_1 \tilde{y}_1} = \frac{1}{N} \sum_{k=1}^{N} \boldsymbol{x}_1(k)\, \tilde{y}_1(k)$.

It is interesting to note that the algorithm can be formulated in an equivalent form as

$$\boldsymbol{w}_1(l+1) = \frac{\langle \boldsymbol{x}_1(k) \tilde{y}_1(k) \rangle}{\langle y_1^2(k) \rangle}. \tag{41}$$

From (40)-(41) it follows that our algorithm is similar to the power method for finding the eigenvector $\boldsymbol{w}_1$ associated with the maximal eigenvalue of the matrix $\boldsymbol{R}_{\boldsymbol{x}_1}(\boldsymbol{b}_1) = E\{\sum_{p=1}^{L} b_{1p} \boldsymbol{x}_1(k) \boldsymbol{x}_1^T(k-p)\}$. This observation suggests that it is not necessary to minimize the cost function with respect to parameters $\{b_{1p}\}$ but it is enough to choose an arbitrary set of them for which the largest eigenvalue is unique (single). More generally, if all eigenvalues of the generalized covariance matrix $\boldsymbol{R}_{\boldsymbol{x}_1}(\boldsymbol{b}_1)$ are distinct, then we can extract all sources simultaneously by estimating principal eigenvectors of $\boldsymbol{R}_{\boldsymbol{x}_1}(\boldsymbol{b}_1)$.

For noisy data, instead of linear predictor, we can use a band-pass filter (or in a parallel way several processing units with a bank of band-pass filters) with fixed or adjustable center frequency and a band-pass bandwidth [42, 47, 68]. The approach is illustrated in Figure 7. By minimizing the cost function $\mathcal{J}(\boldsymbol{w}_j) = E\{\varepsilon_j^2\}$ subject to the constraint $\|\boldsymbol{w}_j\|_2 = 1$, we obtain the on-line learning rule (for prewhitened data):

$$\tilde{\boldsymbol{w}}_j(l+1) = \langle \boldsymbol{x}_1(k) \tilde{y}_j(k) \rangle = \frac{1}{N} \sum_{k=1}^{N} \boldsymbol{x}_1(k) \tilde{y}_j(k), \tag{42}$$

$$\boldsymbol{w}_j(l+1) = \frac{\tilde{\boldsymbol{w}}_j(l+1)}{||\tilde{\boldsymbol{w}}_j(l+1)||_2}, \tag{43}$$

where $y_j(k) = \boldsymbol{w}_j^T(l)\, \boldsymbol{x}_1(k)$, $\tilde{y}_j(k) = B_j(z) y_j(k) = \boldsymbol{w}_j^T(l) \tilde{\boldsymbol{x}}_1^T(k)$, $\tilde{\boldsymbol{x}}_1(k) = B_j(z) \boldsymbol{x}_1(k)$. The above algorithm can extract a source successfully if the cross covariance matrix $\boldsymbol{R}_{\boldsymbol{x}_1 \tilde{\boldsymbol{x}}_1} = E\{\boldsymbol{x}_1 \tilde{\boldsymbol{x}}_1\}$ has a unique (single) maximum eigenvalue.

The proposed algorithm (42)-(43) is insensitive to white noise and/or arbitrary distributed zero-mean noise which is beyond the bandwidth of the band-pass filter. Moreover, the processing unit is able to extract the filtered

from noise version of a source signal if it is a narrow band signal.

Summarizing, the method employed adaptive band-pass filter has several advantages:

- The method does not need a deflation procedure. One processing unit can extract all desired narrow-band sources sequentially one-by-one by adjusting the center frequency and bandwidth of the band-pass filter. Parallel extraction of an arbitrary group of sources is also possible by employing several band-pass filters with different characteristics.
- The algorithm is computationally very simple and efficient.
- The proposed algorithm is robust to additive noise, both white and narrow band colored noise. In contrast to other methods, the covariance matrix of noise does not need to be estimated or modelled.

4. Independent Component Analysis (ICA)

ICA can be defined as follows: The ICA of a random vector $\boldsymbol{x}(k) \in \mathbb{R}^m$ is obtained by finding an $n \times m$, (with $m \geq n$), full rank separating (transformation) matrix $\boldsymbol{W}$ such that the output signal vector $\boldsymbol{y}(k) = [y_1(k), y_2(k), \ldots, y_n(k)]^T$ (components) estimated by

$$\boldsymbol{y}(k) = \boldsymbol{W}\,\boldsymbol{x}(k), \tag{44}$$

are as independent as possible evaluated by an information-theoretic cost function such as minima of Kullback-Leibler divergence or maximization of cumulants [78, 38, 24, 59].

Independence of random variables is a more general concept than decorrelation. Roughly speaking, we say that random variables y_i and y_j are statistically independent if knowledge of the values of y_i provides no information about the values of y_j. Mathematically, the independence of y_i and y_j can be expressed by the relationship

$$p(y_i, y_j) = p(y_i)p(y_j), \tag{45}$$

where $p(y)$ denotes the probability density function (pdf) of the random variable y. In other words, signals are independent if their joint pdf can be factorized.

If independent signals are zero-mean, then the generalized covariance matrix of $f(y_i)$ and $g(y_j)$, where $f(y)$ and $g(y)$ are different, odd nonlinear activation functions (e.g., $f(y) = \tanh(y)$ and $g(y) = y$ for super-Gaussian sources) is a non-singular diagonal matrix:

$$\boldsymbol{R_{fg}} = E\{\boldsymbol{f}(\boldsymbol{y})\boldsymbol{g}^T(\boldsymbol{y})\} = \begin{bmatrix} E\{f(y_1)g(y_1)\} & & 0 \\ & \ddots & \\ 0 & & E\{f(y_n)g(y_n)\} \end{bmatrix}, \tag{46}$$

i.e., the covariances $E\{f(y_i)g(y_j)\}$ are all zero. It should be noted that for odd $f(y)$ and $g(y)$, if the probability density function of each zero-mean source signal is even, then the terms of the form $E\{f(y_i)\}E\{g(y_i)\}$ equal zero. The true general condition for statistical independence of signals is the vanishing of high-order cross-cumulants [53, 52, 43].

The above diagonalization principle can be expressed as

$$\boldsymbol{R}_{fg}^{-1} = \boldsymbol{\Lambda}^{-1}, \tag{47}$$

where $\boldsymbol{\Lambda}$ is any diagonal positive definite matrix (typically, $\boldsymbol{\Lambda} = \boldsymbol{I}$, By pre-multiplying the above equation by separating matrix $\boldsymbol{W}$ and $\boldsymbol{\Lambda}$, we obtain:

$$\boldsymbol{\Lambda}\boldsymbol{R}_{fg}^{-1}\boldsymbol{W} = \boldsymbol{W}, \tag{48}$$

which suggest the following iterative multiplicative learning algorithm [64]

$$\tilde{\boldsymbol{W}}(l+1) = \boldsymbol{\Lambda}\boldsymbol{R}_{fg}^{-1}\boldsymbol{W}(l), \tag{49}$$

$$\boldsymbol{W}(l+1) = \tilde{\boldsymbol{W}}(l+1)\left[\tilde{\boldsymbol{W}}^T(l+1)\tilde{\boldsymbol{W}}(l+1)\right]^{-1/2}, \tag{50}$$

where the last equation represents the symmetric orthogonalization to keep algorithm stable. The above algorithm is simple and fast but need prewhitening the data.

In fact, a wide class of algorithms for ICA can be expressed in general form as (see Table 1) [40]

$$\nabla \boldsymbol{W}(l) = \boldsymbol{W}(l+1) - \boldsymbol{W}(l) = \eta \boldsymbol{F}(\boldsymbol{y}) \boldsymbol{W}(l), \tag{51}$$

where $\boldsymbol{y}(k) = \boldsymbol{W}(l)\boldsymbol{x}(k)$ and the matrix $\boldsymbol{F}(\boldsymbol{y})$ can take different forms, for example $\boldsymbol{F}(\boldsymbol{y}) = \boldsymbol{\Lambda}_n - \boldsymbol{f}(\boldsymbol{y})\boldsymbol{g}^T(\boldsymbol{y})$ with suitably chosen nonlinearities $\boldsymbol{f}(\boldsymbol{y}) = [f(y_1), ..., f(y_n)]$ and $\boldsymbol{g}(\boldsymbol{y}) = [g(y_1), ..., g(y_n)]$ [53, 57, 40, 64].

Assuming prior knowledge of the source distributions $p_i(y_i)$, we can estimate $\boldsymbol{W}$ using maximum likelihood (ML):

$$J(\boldsymbol{W}, \boldsymbol{y}) = -\frac{1}{2} \log |\det(\boldsymbol{W}\boldsymbol{W}^T)| - \sum_{i=1}^{n} \log(p_i(y_i) \tag{52}$$

Using natural gradient descent to increase likelihood we get:

$$\boldsymbol{W}(l+1) = \eta \ \left[\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y})\boldsymbol{y}^T\right] \boldsymbol{W}(l), \tag{53}$$

where $\boldsymbol{f}(\boldsymbol{y}) = [f_1(y_1), f_2(y_2), \ldots, f_n(y_n)]^T$ is an entry-wise nonlinear score function defined by:

$$f_i(y_i) = -\frac{p_i'(y_i)}{p_i(y_i)} = -\frac{d \log(p_i(y_i)}{d(y_i)} \tag{54}$$

Alternatively, for signals corrupted by additive Gaussian noise, we can use higher order matrix cumulants. As illustrative example, let us consider the following cost function which is measure of independence [57, 56]:

$$J(\boldsymbol{W}, \boldsymbol{y}) = -\frac{1}{2} \log |\det(\boldsymbol{W}\boldsymbol{W}^T)| - \frac{1}{1+q} \sum_{i=1}^{n} |C_{1+q}(y_i)|, \tag{55}$$

where we use the following notations: $C_q(y_1)$ denotes the q-order cumulants of the signal y_i and $\mathbf{C}_{p,q}(\boldsymbol{y}, \boldsymbol{y})$ denotes the cross-cumulant matrix whose elements are $[\mathbf{C}_{pq}(\boldsymbol{y}, \boldsymbol{y})]_{ij} = Cum(\underbrace{y_i, y_i, \ldots, y_i}_{p}, \underbrace{y_j, y_j, \ldots, y_j}_{q})$.

The first term in (55) assures that the determinant of the global matrix will not approach zero. By including this term, we avoid the trivial solution $y_i = 0, \ \forall i$. The second terms force the output signals to be as far as possible from Gaussianity, since the higher order cumulants are a natural measure of non-Gaussianity and they will vanish for Gaussian signals. It can be shown that for such a cost function, we can derive the following equivariant and robust in respect to Gaussian noise algorithm [57, 56, 58]

$$\boldsymbol{\Delta W}(l) = \boldsymbol{W}(l+1) - \boldsymbol{W}(l) = \eta_l \left[\mathbf{I} - \mathbf{C}_{1,q}(\boldsymbol{y}, \boldsymbol{y}) \, \mathbf{S}_{q+1}(\boldsymbol{y})\right] \boldsymbol{W}(l), \tag{56}$$

where $\mathbf{S}_{q+1}(\boldsymbol{y}) = \mathrm{sign}(\mathrm{diag}(\mathbf{C}_{1,q}(\boldsymbol{y}, \boldsymbol{y})))$ and $\boldsymbol{F}(\boldsymbol{y}) = \mathbf{I} - \mathbf{C}_{1,q}(\boldsymbol{y}, \boldsymbol{y}) \, \mathbf{S}_{q+1}(\boldsymbol{y})$.

It should be noted that ICA can perform blind source separation, i.e., enable to estimate true sources only if they are all statistically independent and non-Gaussian (except possibly of one) [40, 17].

4.1 Subband Decomposition – Independent Component Analysis (SD-ICA)

Despite the success of using standard ICA in many applications, the basic assumptions of ICA may not hold for some kind of signals hence some caution should be taken when using standard ICA to analyze real world problems, especially in biomedical signal processing. In fact, by definition, the standard ICA algorithms are not able to estimate statistically dependent original sources, that is, when the independence assumption is violated. In this section, we will present a natural extension and generalization of ICA called Subband Decomposition ICA (SD-ICA) which relaxes considerably the assumption regarding mutual independence of primarily sources [44, 135, 46]. The key idea in this approach is the assumption that the unknown wide-band source signals can be dependent, however some their narrow band sub-components are independent. In other words, we assume that each unknown source can be modelled or represented as a sum (or linear combinations) of narrow-band sub-signals (sub-components):

$$s_i(k) = s_{i1}(k) + s_{i2}(k) + \cdots + s_{iK}(k). \tag{57}$$

Table 1. Basic equivariant adaptive learning algorithms for ICA. Some of these algorithms require prewhitening.

No.	Learning Algorithm	References
1.	$\boldsymbol{\Delta W} = \eta\Big[\boldsymbol{\Lambda} - \langle \boldsymbol{f}(\boldsymbol{y})\, \boldsymbol{g}^T(\boldsymbol{y})\rangle\Big]\boldsymbol{W}$	Cichocki, Unbehauen, Rummert (1994)
	$\boldsymbol{\Lambda}$ is a diagonal matrix with non-negative elements λ_{ii}	
	$\boldsymbol{W}(l+1) = \Big[\boldsymbol{I} \mp \eta\, [\boldsymbol{I} - \langle \boldsymbol{f}(\boldsymbol{y})\, \boldsymbol{g}^T(\boldsymbol{y})\rangle]\Big]^{\mp 1}\, \boldsymbol{W}(l)$	Cruces, Cichocki, Castedo (2000)
2.	$\boldsymbol{\Delta W} = \eta\Big[\boldsymbol{\Lambda} - \langle \boldsymbol{f}(\boldsymbol{y})\, \boldsymbol{y}^T\rangle\Big]\boldsymbol{W}, \quad f(y_i) = -p'(y_i)/p(y_i)$	Bell, Sejnowski (1995)
	$\lambda_{ii} = \langle f(y_i(k))y_i(k)\rangle \quad$ or $\quad \lambda_{ii} = 1, \ \ \forall i$	Amari, Cichocki, Yang (1995)
		Choi, Cichocki, Amari (1999)
3.	$\boldsymbol{\Delta W} = \eta\Big[\boldsymbol{I} - \langle \boldsymbol{y}\, \boldsymbol{y}^T\rangle - \langle \boldsymbol{f}(\boldsymbol{y})\, \boldsymbol{y}^T\rangle + \langle \boldsymbol{y}\, \boldsymbol{f}^T(\boldsymbol{y})\rangle\Big]\boldsymbol{W}$	Cardoso, Laheld, (1996)
4.	$\boldsymbol{\Delta W} = \eta\Big[\boldsymbol{I} - \langle \boldsymbol{y}\, \boldsymbol{y}^T\rangle - \langle \boldsymbol{f}(\boldsymbol{y})\, \boldsymbol{y}^T\rangle + \langle \boldsymbol{f}(\boldsymbol{y})\, \boldsymbol{f}^T(\boldsymbol{y})\rangle\Big]\boldsymbol{W}$	Karhunen, Pajunen (1997)
5.	$\tilde{\boldsymbol{W}} = \boldsymbol{W} + \eta\Big[\boldsymbol{\Lambda} - \langle \boldsymbol{f}(\boldsymbol{y})\, \boldsymbol{y}^T\rangle\Big]\boldsymbol{W}, \ \ \lambda_{ii} = \langle f(y_i)\, y_i\rangle$	Hyvärinen, Oja (1999)
	$\eta_{ii} = [\lambda_{ii} + \langle f'(y_i)\rangle]^{-1}; \quad \boldsymbol{W} = (\tilde{\boldsymbol{W}}\tilde{\boldsymbol{W}}^T)^{-1/2}\, \tilde{\boldsymbol{W}}$	
6.	$\boldsymbol{\Delta W} = \eta\Big[\boldsymbol{I} - \boldsymbol{\Lambda}^{-1}\langle \boldsymbol{y}\, \boldsymbol{y}^T\rangle\Big]\boldsymbol{W}$	Choi, Cichocki, Amari (2000)
	$\lambda_{ii}(k) = \langle y_i^2(k))\rangle$	Amari, Cichocki (1998)
7.	$\boldsymbol{\Delta W} = \eta\Big[\boldsymbol{I} - \boldsymbol{C}_{1,q}(\boldsymbol{y}, \boldsymbol{y})\, \boldsymbol{S}_{q+1}(\boldsymbol{y})\Big]\boldsymbol{W}$	Cruces, Castedo, Cichocki (2002)
	$C_{1,q}(y_i, y_j) = Cum(y_i, \underbrace{y_j, \ldots, y_j}_{q})$	
8.	$\boldsymbol{W}(l+1) = \exp(\eta\, \boldsymbol{F}[\boldsymbol{y}])\, \boldsymbol{W}(l)$	Nishimori, Fiori(1999,2003)
	$\boldsymbol{F}(\boldsymbol{y}) = \boldsymbol{\Lambda} - \langle \boldsymbol{y}\, \boldsymbol{y}^T\rangle - \langle \boldsymbol{f}(\boldsymbol{y})\, \boldsymbol{y}^T\rangle + \langle \boldsymbol{y}\, \boldsymbol{f}^T(\boldsymbol{y})\rangle$	Cichocki, Georgiev (2002)
9.	$\tilde{\boldsymbol{W}} = \boldsymbol{\Lambda}\, \boldsymbol{R}_{fg}^{-1}\boldsymbol{W}$	Fiori (2003)
	$\boldsymbol{W} = (\tilde{\boldsymbol{W}}\, \tilde{\boldsymbol{W}}^T)^{-1/2}\tilde{\boldsymbol{W}}$	

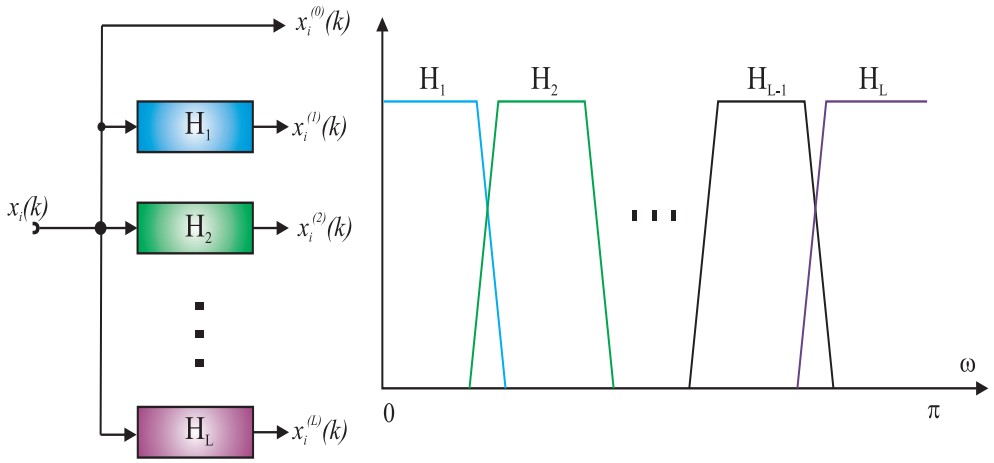


Figure 8. Bank of band-pass filters employed in preprocessing stage for SD-ICA with typical frequency bands. For each sensor signal we employ the identical set of filters.

For example, in the simplest case, source signals can be modelled or decomposed into their low- and high-frequency sub-components:

$$s_i(k) = s_{iL}(k) + s_{iH}(k) \qquad (i = 1, 2, \ldots, n). \tag{58}$$

In practice, the high-frequency sub-components $s_{iH}(k)$ are often found to be mutually independent. In such a case, we can use a High-Pass Filter (HPF) to extract high frequency sub-components and then apply any standard ICA algorithm to such preprocessed sensor (observed) signals. We have implemented these concepts in our ICALAB software and extensively tested for some experimental data [40, 41].

The basic concept in Subband Decomposition ICA is to divide the sensor signal spectra into their subspectra or subbands, and then to treat those subspectra individually for the purpose at hand. The subband signals can be ranked and processed independently. Let us assume that only a certain set of sub-components are independent. Provided that for some of the frequency subbands (at least one) all sub-components, say $\{s_{ij}(k)\}_{i=1}^{n}$, are mutually independent or temporally decorrelated, then we can easily estimate the mixing or separating system under condition that these subbands can be identified by some *a priori* knowledge or detected by some self-adaptive process. For this purpose, we simply apply any standard ICA algorithm, however not for all available raw sensor data but only for suitably pre-processed (e.g., subband filtered) sensor signals.

Such explanation can be summarized as follows. The SD-ICA (Subband Decomposition ICA) can be formulated as a task of estimation of the separating matrix $\boldsymbol{W}$ and/or the estimating mixing matrix $\hat{\boldsymbol{A}}$ on the basis of suitable subband decomposition of sensor signals and by applying a classical ICA (instead for raw sensor data) for one or several preselected subbands for which source sub-components are independent.

By applying any standard ICA/BSS algorithm for specific subbands and raw sensor data, we obtain sequence of separating matrices $\boldsymbol{W}_0, \boldsymbol{W}_1, \ldots, \boldsymbol{W}_L$, where $\boldsymbol{W}_0$ is the separating matrix from the original data $\boldsymbol{x}$ and $\boldsymbol{W}_j$ is the separating matrix from preprocessing sensor data $\boldsymbol{x}_j$ in j-th subband. In order to identify for which subbands corresponding source sub-components are independent, we propose to compute the global (mixing-separating) matrices $\boldsymbol{G}_{jq} = \boldsymbol{W}_j \boldsymbol{W}_q^{-1}$, $\forall j \neq q$, where $\boldsymbol{W}_q$ is estimating separating matrix for q-th subband. If sub-components are mutually independent for at least two subbands, say for the subband j and subband q, then the global matrix $\boldsymbol{W}_j \boldsymbol{W}_q^{-1} = \boldsymbol{P}_{jq}$ will be generalized permutation matrix with only one nonzero (or dominated) element in each row and each column. This follows from the simple observation that in such case the both matrices $\boldsymbol{W}_j$ and $\boldsymbol{W}_q$ represent inverses (for $m = n$) of the same mixing matrix $\boldsymbol{A}$ (neglecting nonessential scaling and permutation ambiguities). In this way, we can blindly identify essential information for which frequency subbands the source sub-components are independent and we can easily identify correctly the mixing matrix. Furthermore, the same concept can be used to estimate blindly the performance index and to compare performance of various ICA algorithms, especially for large scale problems.

In the preprocessing stage we can use any linear transforms, especially, more sophisticated methods, such as

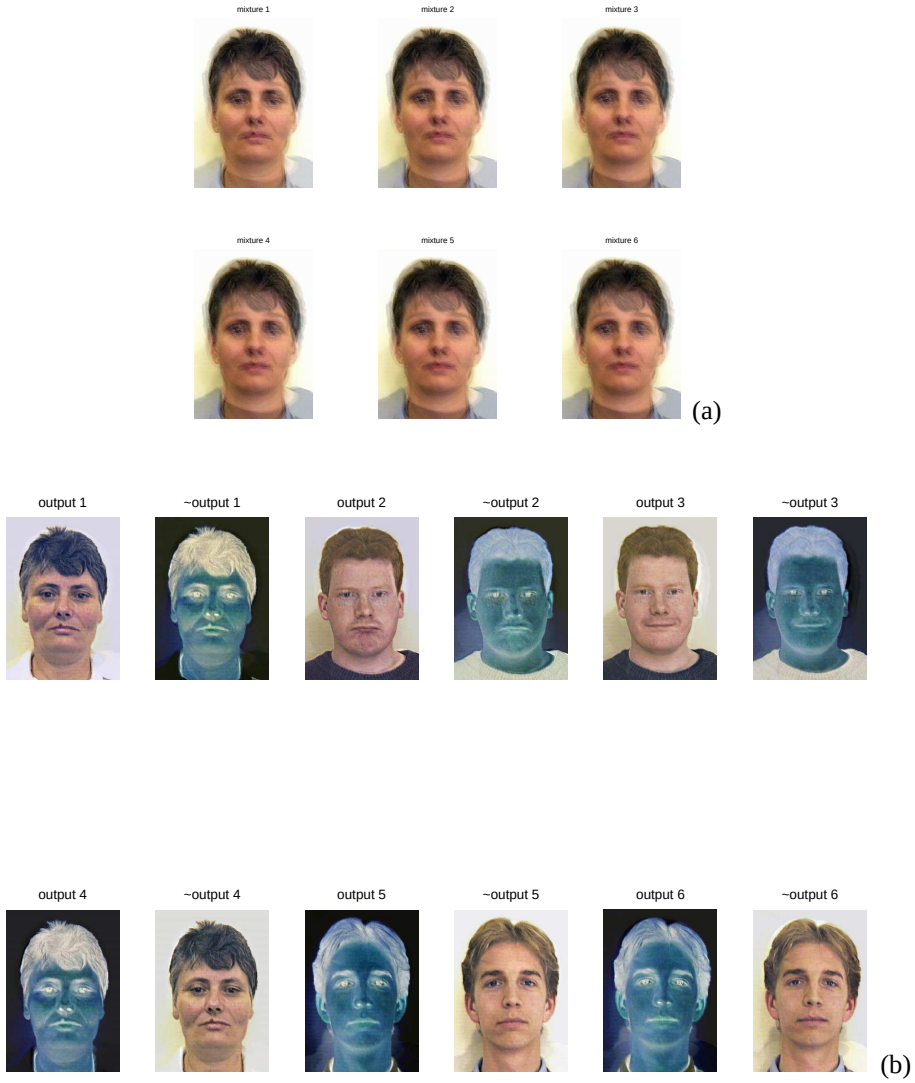


Figure 9. Example 1: (a) Observed overlapped images. (b) Reconstructed original images using SD-ICA with subband decomposition preprocessing.

block transforms, multirate subband filter bank or wavelet transforms, can be applied. We can extend and generalize further this concept by performing the decomposition of sensor signals in a composite time-frequency domain rather than in frequency subbands as such. This naturally leads to the concept of wavelets packets (subband hierarchical trees) and to block transform packets [40, 151, 13]. Such preprocessing techniques has been implemented in ICALAB [41].

Simulation illustrative example: In this experiment 6 human faces of three persons are mixed by using the random generated ill-conditioned mixing matrix $\boldsymbol{A}$ (assumed to be unknown). The mixing images shown in Figure 9(b) are linear superposition of strongly correlated faces, thus any classical ICA algorithm failed to separate them. In order to reconstruct original images we applied in the preprocessing stage 10 subbands filters for the observed images. Using the method described above, we have identified 3 subbands for which the sub-components are completely independent. For such preprocessed mixed images we have applied the standard natural gradient ICA learning algorithm [40]. The estimated original images are shown in Figure 9 (c). It is interesting to note that original images are reconstructed almost perfectly although some of them are strongly dependent. The same principle can be applied for any real world EEG/MEG fMRI signals.

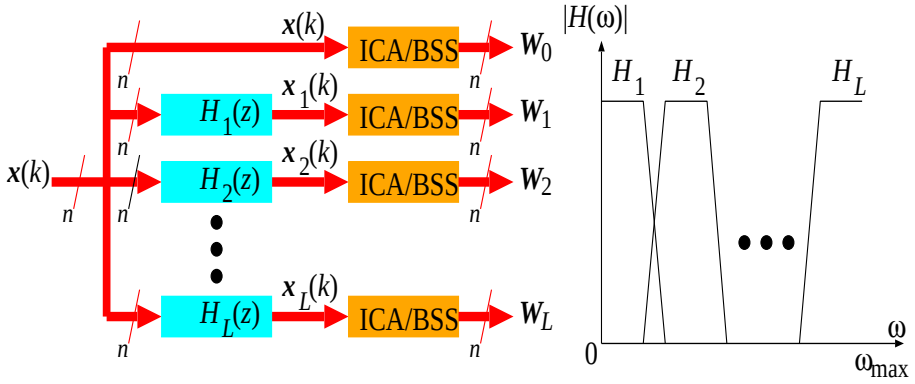


Figure 10. Bank of filters employed in preprocessing stage for investigating validity and reliability of any ICA/BSS algorithms. The subbands can be overlapped or not and have more complex subbands forms. Furthermore, the coefficients of transfer functions FIR filters can be suitably designed or even randomly chosen.

5. Validity of ICA-based BSS Algorithms for Real World Data

One of the fundamental question in BSS is problem whether the obtained results of the specific BSS/ICA algorithm is reliable and represent inherent properties of the model and data or it is just a random, purely mathematical, decomposition of data without physical meaning. In fact, since most of BSS algorithms are stochastic in nature, their results may be could be somewhat different in different runs even for the same algorithm. Thus, the results obtained in a single run or for single set of data of any BSS algorithm should be interpreted with reserve and reliability of estimated sources should be analyzed by investigating the spread of the obtained estimates for many runs [75]. Such an analysis can performed, for example, by using resampling or bootstrapping method in which the available data is randomly changed by producing surrogate data sets from the original data [104]. The specific ICA/BSS algorithm is then run many times with bootstrapped samples that are somewhat different from each other. Alternative approach called ICASSO has been developed by [75] which is based on running the specific BSS algorithm many times for various different initial conditions and parameters and visualizing the clustering structure of the estimated sources (components) in the signal subspace. In order to estimate algorithmic reliability it was suggested to run the BSS algorithm many times using different initial conditions and assessing which of the components are found in almost all run. For this purpose the estimated components are clustered and classified. The reliable components corresponds to small and well separated clusters from the rest of components, while unreliable components usually do not belong to any cluster [75, 104].

It is worth to note that the concept of MSD-ICA described in the previous section can be extended easily to more general and flexible multi-dimensional models for checking validity and reliability of ICA (or more generally BSS) algorithms with the number sensors equal to or larger than the number of unknown sources (see Figure 10). In this model we can use a bank of stable filters with transfer functions $H_i(z)$, for example, set of FIR (finite impulse response filters). The parameters (coefficients) of such FIR filters can be randomly generated. In this case the proposed method has some similarity with resampling or bootstrap approach proposed by [104]. Similarly to MSD-ICA, we run any BSS algorithm for sufficiently large number of filters and generate set of separating matrices: $\{\boldsymbol{W}_0, \boldsymbol{W}_1, \ldots, \boldsymbol{W}_L\}$ or alternatively set of estimated mix matrices: $\{\hat{\boldsymbol{A}}_0, \hat{\boldsymbol{A}}_1, \ldots, \hat{\boldsymbol{A}}_L\}$.[4] In the next step we estimate the global mixing-separating matrices $\boldsymbol{G}_{pq} = \boldsymbol{W}_p \boldsymbol{W}_q^+$ for any $p \neq q$.

The performance of blind separation can be characterized by one single performance index (sometimes referred as Amari's performance index) which we refer as blind performance index (since we do not know true mixing matrix)

$$BPI_i = \frac{1}{n}\sum_{j=1}^{n}\left(\frac{\sum_{i=1}^{n}|g_{ij}|^2}{max_i\ |g_{ij}|^2} - 1\right) + \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\sum_{j=1}^{n}|g_{ij}|^2}{max_j\ |g_{ij}|^2} - 1\right), \tag{59}$$

where g_{ij} is ij-th element of the matrix $\boldsymbol{G}_{pq}$. In many cases, we are not able to achieve perfect separation for some

[4] The set of matrices can be further extended if data will be bootstrapped and/or initial conditions will be changed for each run.

sources or we are able to extract only some sources (not of all them). In such cases instead of using one global performance index we can define local performance index as

$$BPI_i = \left(\frac{\sum_{j=1}^{n} |g_{ij}|^2}{max_j \ |g_{ij}|^2} - 1 \right) \tag{60}$$

Alternatively, we can use the following performance index for each component i [104]

$$e_i = \arccos \left(\frac{g_{ii}}{\sqrt{\sum_{j=1}^{n} |g_{ij}|^2}} \right) \tag{61}$$

If the performance index BPI_i or e_i for specific index i and bands p, q is close to zero this means that with high probability this component is successfully extracted. In order to asses significant components the all estimated components should be clustered according their mutual similarities. These similarities can be searched in time domain or frequency domain. The natural measure of similarity between of the estimated components can be absolute value of their mutual correlation coefficients $|r_{ij}|$ for $i \neq j$ which are elements of the similarity matrix [75]

$$\boldsymbol{R} = \overline{\boldsymbol{W}} \ \boldsymbol{R}_{xx} \ \overline{\boldsymbol{W}}^T, \tag{62}$$

where $\overline{\boldsymbol{W}} = [\boldsymbol{W}_0, \boldsymbol{W}_1, \ldots, \boldsymbol{W}_l]$ and $\boldsymbol{R}_{xx} = E\{\boldsymbol{x}\boldsymbol{x}^T\} = \boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^T$ is covariance matrix of observations under assumption that the covariance matrix of sources $\boldsymbol{R}_{ss} = E\{\boldsymbol{s}\boldsymbol{s}^T\}$ is a diagonal matrix and separating matrices $\boldsymbol{W}_p$ are normalized (e.g., to unit length vectors).

ICA decompostion, $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}$, has inherently duality. Considering the data matrix $\boldsymbol{X} \in \mathbb{R}^{m \times N}$ where its each row is assumed to be a time course of an attribute, ICA decomposition produces n independent time courses. On the other hand, regarding the data matrix in the form of $\boldsymbol{X}^T$, ICA decomposition leads to n independent patterns (for instance, images in fMRI or arrays in DNA microarray data).

The standard ICA (where $\boldsymbol{X}$ is considered) is treated as *temporal ICA* (tICA). Its dual decomposition (regarding $\boldsymbol{X}^T$) is known as *spatial ICA* (sICA). Combining these two ideas, leads to *spatio-temporal ICA* (stICA). These variations of ICA, were first investigated in [133]. Spatial ICA or spatio-temporal ICA were shown to be useful in fMRI image analysis [133] and gene expression data analysis [103, 86].

Suppose that the singular value decomposition (SVD) of $\boldsymbol{X}$ is given by

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T = \left(\boldsymbol{U}\boldsymbol{D}^{1/2}\right)\left(\boldsymbol{V}\boldsymbol{D}^{1/2}\right)^T = \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{V}}^T, \tag{63}$$

where $\boldsymbol{U} \in \mathbb{R}^{m \times n}$, $\boldsymbol{D} \in \mathbb{R}^{n \times n}$, and $\boldsymbol{V} \in \mathbb{R}^{N \times n}$ for $n \leq \min(m, N)$.

Temporal ICA Temporal ICA finds a set of independent time courses and a corresponding set of dual unconstrained spatial patterns. It embodies the assumption that each row vector in $\widetilde{\boldsymbol{V}}^T$ consists of a linear combination of n independent sequences, i.e., $\widetilde{\boldsymbol{V}}^T = \widetilde{\boldsymbol{A}}_T\boldsymbol{S}_T$, where $\boldsymbol{S}_T \in \mathbb{R}^{n \times N}$ has a set of n independent temporal sequences of length N and $\widetilde{\boldsymbol{A}}_T \in \mathbb{R}^{n \times n}$ is an associated mixing matrix.

Unmixing by $\boldsymbol{Y}_T = \boldsymbol{W}_T\widetilde{\boldsymbol{V}}^T$ where $\boldsymbol{W}_T = \boldsymbol{P}\widetilde{\boldsymbol{A}}_T^{-1}$, leads us to recover the n dual patterns $\boldsymbol{A}_T$ associated with the n independent time courses, by calculating $\boldsymbol{A}_T = \widetilde{\boldsymbol{U}}\boldsymbol{W}_T^{-1}$, which is a consequence of $\widetilde{\boldsymbol{X}} = \boldsymbol{A}_T\boldsymbol{Y}_T = \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{V}}^T = \widetilde{\boldsymbol{U}}\boldsymbol{W}_T^{-1}\boldsymbol{Y}_T$.

Spatial ICA

Spatial ICA seeks a set of independent spatial patterns $\boldsymbol{S}_S$ and a corresponding set of dual unconstrained time courses A_S. It embodies the assumption that each row vector in $\widetilde{\boldsymbol{U}}^T$ is composed of a linear combination of n independent spatial patterns, i.e., $\widetilde{\boldsymbol{U}}^T = \widetilde{\boldsymbol{A}}_S\boldsymbol{S}_S$, where $\boldsymbol{S}_S \in \mathbb{R}^{n \times m}$ contains a set of n independent m-dimensional patterns and $\widetilde{\boldsymbol{A}}_S \in \mathbb{R}^{n \times n}$ is an encoding variable matrix (mixing matrix).

Define $\boldsymbol{Y}_S = \boldsymbol{W}_S \widetilde{\boldsymbol{U}}^T$ where $\boldsymbol{W}_S$ is a permuted version of $\widetilde{\boldsymbol{A}}_S^{-1}$. With this definition, the n dual time courses $\boldsymbol{A}_S \in \mathbb{R}^{N \times n}$ associated with the n independent patterns, is computed by $\boldsymbol{A}_S = \widetilde{\boldsymbol{V}} \boldsymbol{W}_S^{-1}$, since $\widetilde{\boldsymbol{X}}^T = \boldsymbol{A}_S \boldsymbol{Y}_S = \widetilde{\boldsymbol{U}} \widetilde{\boldsymbol{V}}^T = \widetilde{\boldsymbol{V}} \boldsymbol{W}_S^{-1} \boldsymbol{Y}_S$. Each column vector of $\boldsymbol{A}_S$ corresponds to a temporal mode.

Spatio-Temporal ICA

In linear decomposition, sICA enforces independence constraints over space, to find a set of independent spatial patterns, whereas tICA embodies independence constraints over time, to seek a set of independent time courses. Spatio-temporal ICA finds a linear decomposition, by maximizing the degree of independence over space as well as over time, without necessarily producing independence in either space or time. In fact it allows a trade-off between the independence of arrays and the independence of time courses.

Given $\widetilde{\boldsymbol{X}} = \widetilde{\boldsymbol{U}} \widetilde{\boldsymbol{V}}^T$, stICA finds the following decomposition:

$$\widetilde{\boldsymbol{X}} = \boldsymbol{S}_S^T \Lambda \boldsymbol{S}_T, \tag{64}$$

where $\boldsymbol{S}_S \in \mathbb{R}^{n \times m}$ contains a set of n independent m-dimensional patterns, $\boldsymbol{S}_T \in \mathbb{R}^{n \times N}$ has a set of n independent temporal sequences of length N, and $\boldsymbol{\Lambda}$ is a diagonal scaling matrix. There exist two $n \times n$ mixing matrices, $\boldsymbol{W}_S$ and $\boldsymbol{W}_T$ such that $\boldsymbol{S}_S = \boldsymbol{W}_S \widetilde{\boldsymbol{U}}^T$ and $\boldsymbol{S}_T = \boldsymbol{W}_T \widetilde{\boldsymbol{V}}^T$. The following relation

$$\begin{aligned} \widetilde{\boldsymbol{X}} &= \boldsymbol{S}_S^T \boldsymbol{\Lambda} \boldsymbol{S}_T \\ &= \widetilde{\boldsymbol{U}} \boldsymbol{W}_S^T \boldsymbol{\Lambda} \boldsymbol{W}_T \widetilde{\boldsymbol{V}}^T \\ &= \widetilde{\boldsymbol{U}} \widetilde{\boldsymbol{V}}^T, \end{aligned} \tag{65}$$

implies that $\boldsymbol{W}_S^T \boldsymbol{\Lambda} \boldsymbol{W}_T = \boldsymbol{I}$, which leads to

$$\boldsymbol{W}_T = \boldsymbol{W}_S^{-T} \boldsymbol{\Lambda}^{-1}. \tag{66}$$

Linear transforms, $\boldsymbol{W}_S$ and $\boldsymbol{W}_T$, are found by jointly optimizing objective functions associated with sICA and tICA. That is, the objective function for stICA has the form

$$\mathcal{J}_{stICA} = \alpha \mathcal{J}_{sICA} + (1 - \alpha) \mathcal{J}_{tICA}, \tag{67}$$

where $\mathcal{J}_{sICA}$ and $\mathcal{J}_{tICA}$ could be infomax criteria or log-likelihood functions and α defines the relative weighting for spatial independence and temporal independence. More details on stICA can be found in [133].

6. Sparse Component Analysis and Sparse Signal Representations

Sparse Component Analysis (SCA) and sparse signals representations (SSR) arise in many scientific problems, especially, where we wish to represent signals of interest by using a small (or sparse) number of basis signals from a much larger set of signals, often called dictionary. Such problems arise also in many applications such as electro-magnetic and biomagnetic inverse problems (EEG/MEG), feature extraction, filtering, wavelet denoising, time-frequency representation, neural and speech coding, spectral estimation, direction of arrival estimation, failure diagnosis and speed-up processing [40, 102, 101].

In opposite to ICA where the mixing matrix and source signals are estimated simultaneously the SCA is usually a multi stage procedure. In first stage we need to find a suitable linear transformation which guarantee that sources in the transformed domain are sufficiently sparse. Typically, we represent the observed data in the time-frequency domain using wavelets package [101]. In the next step, we estimate the columns $\boldsymbol{a}_i$ of the mixing matrix $\boldsymbol{A}$ using a sophisticated hierarchical clustering technique. This step is the most difficult and challenging task since it requires to identify precisely intersections of all hyperplanes on which observed data are located [66, 136]. In the last step, we estimate sparse sources using, for example, a modified robust linear programming (LP), quadratic programming (QP) or semi-definite programming (SDP) optimization. The big advantage of SCA is its ability to reconstruct of original sources even if the number of observations (sensors) is smaller than number of sources under certain weak conditions [101, 46, 65].

We can state the subset selection sub-problem as follows: Find an optimal subset of $r << n$ columns from the matrix $\boldsymbol{A}$ which we denote by $\boldsymbol{A}_r \in \mathbb{R}^{m \times r}$ such that $\boldsymbol{A}_r \boldsymbol{s}_{r*} \cong \boldsymbol{x}$, or equivalently $\boldsymbol{A}\boldsymbol{s}_* + \boldsymbol{e}_r = \boldsymbol{x}$, where $\boldsymbol{e}_r$ represents some residual error vector which norm should below some threshold. The problem consists often not only in estimating the sparse vector $\boldsymbol{s}_*$ but also correct or optimal sparsity profile that is the sparsity index r, that is detection the number of sources.

Usually, we have interest in sparsest and unique representation, i.e., it is necessary to find solution having the smallest possible number of nonzero-components. The problem can be reformulated as the following robust optimization problem:

$$(P_\rho) \quad J_\rho(\boldsymbol{s}) = \|\boldsymbol{s}\|_\rho = \sum_{j=1}^{n} \rho(s_j) \quad \text{s. t.} \qquad \boldsymbol{A}\,\boldsymbol{s} = \boldsymbol{x}, \tag{68}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, (usually with $n >> m$) and $\|\boldsymbol{s}\|_\rho$ suitably chosen function which measures the sparsity of the vector $\boldsymbol{s}$. It should be noted the sparsity measure does not need be necessary a norm, although we use such notation. For example, we can apply Shannon, Gauss or Renyi entropy or normalized kurtosis as measure of the sparsity [40, 88, 151]. In the standard form, we use l_p-norm with $0 \leq p \leq 1$. Especially, l_0 quasi-norm attract a lot of attention since it ensures sparsest representation [61, 102, 101] . Unfortunately, such formulated problem (68) for l_p-norm with $p < 1$ is rather very difficult, especially for $p = 0$ it is NP-hard, so for a large scale problem it is numerically untractable. For this reason, we often use Basis Pursuit (BP) or standard Linear Programming (LP) for $\|\boldsymbol{s}\|_\rho = \|\boldsymbol{s}\|_1$, with $\rho = p = 1$.

In practice, due to noise and other uncertainty (e.g., measurement errors) the system of linear underdetermined equations should not be satisfied precisely but with some prescribed tolerance (i.e., $\boldsymbol{A}\,\boldsymbol{s} \cong \boldsymbol{x}$ in the sense that $\|\boldsymbol{x} - \boldsymbol{A}\,\boldsymbol{s}\|_q \leq \boldsymbol{\varepsilon}$). From the practical point of view as well as from a statistical point of view, it is convenient and quite natural to replace the exact constraints $\boldsymbol{x} = \boldsymbol{A}\,\boldsymbol{s}$ by the constraint $\|\boldsymbol{x} - \boldsymbol{A}\,\boldsymbol{s}\|_q \leq \boldsymbol{\varepsilon}$, where choice of l_q-norm depends on distribution of noise and specific applications. For noisy and uncertain data we should to use a more flexible and robust cost function (in comparison to the standard (P_ρ) problem) which will be referred as Extended Basis Pursuit Denoising ($EBPD$)[46]:

$$(EBPD) \qquad J_{q,\rho}(\boldsymbol{s}) = \|\boldsymbol{x} - \boldsymbol{A}\,\boldsymbol{s}\|_q^q + \alpha\,\|\boldsymbol{s}\|_\rho, \tag{69}$$

There are several possible basic choices for l_q and sparsity criteria ($\|\boldsymbol{s}\|_\rho = \|\boldsymbol{s}\|_p$) For example, for the uniform (Laplacian) distributed noise we should choose l_∞-Chebyshev norm (l_1-norm). Some basic choices of ρ (for $l_q = 2$) are $\rho = 0$ (minimum l_0 quasi norm or atomic decomposition related with the matching pursuit (MP) and FOCUSS algorithm), $\rho = 1$ (basis pursuit denoising) and $\rho = 2$ (ridge regression) [88, 151, 61]. The optimal choice of ρ norms depends on distribution of noise in sparse components. For example, for noisy components, we can use robust norms such as Huber function defined as $\|\boldsymbol{s}\|_{\rho_H} = \sum_i \rho_H(s_i)$, where $\rho_H(s_i) = s_i^2/2$ if $|s_i| \leq \beta$ and $\rho_H(s_i) = \beta\,|s_i| - \beta^2/2$ if $|s_i| > \beta$, and/or epsilon norm defined as $\|\boldsymbol{s}\|_{\boldsymbol{\varepsilon}} = \sum_j |s_j|_{\boldsymbol{\varepsilon}}$ where $|s_j|_{\boldsymbol{\varepsilon}} = \max\{0,\ (|s_j| - \boldsymbol{\varepsilon})\}$.

The practical importance of the $EBPD$ approach in comparison to the standard LP or BP approach is that the $EPBD$ allows for treating the presence of noise or errors due to mismodeling. Moreover, using the $EBPD$ approach, we have possibility to adjust the sparsity profile (i.e., adjust the number of nonzero components) by tuning the parameter α. In contrast, by using the LP approach we do not have such option. Furthermore, the method can be applied both for undercomplete and/or overcomplete models (i.e., when the number of sources is larger or less than the number of sensors.

The practical importance of the extended quadratic programming approach in contrast to the linear programming or standard Basis Pursuit approach is that the (QP) allows for treating the presence of noise or errors due to mismodeling. In practice, in the presence of noise the true model is: $\boldsymbol{x}(k) = \boldsymbol{A}\,\boldsymbol{s}(k) + \boldsymbol{v}(k)$.

7. Non-negative Matrix Factorization and Sparse Coding with Non-negativity Constraints

7.1 Blind Separation of Independent Sources with Non-negativity Constraints

In many applications such as computer tomography and biomedical image processing non-negative constraints are imposed for entries ($a_{ij} \geq 0$) of the mixing matrix $\boldsymbol{A}$ and/or estimated source signals ($s_j(k) \geq 0$)

(2)) [40, 118, 76]. Moreover, recently several authors suggested that a decomposition of a observation $\boldsymbol{X} = \boldsymbol{AS}$ into non-negative factors or Non-negative Matrix Factorization (NMF), is able to produce useful and meaningful representation of real- world data, especially in image analysis, hyperspectral data processing, biological modeling and sparse coding [118, 90, 123, 76].

In this section, we present very simple and practical technique for estimation of non-negative independent sources and entries of the mixing matrix $\boldsymbol{A}$ using standard ICA approach and suitable postprocessing. In other words, we will show that by simple modifications of existing ICA or BSS algorithms, we are able to satisfy non-negativity constraints of sources and simultaneously impose they are sparse or independent as possible. Without loss of generality, we assume that all sources are non-negative, i.e., $s_j(k) = \widetilde{s}_j(k) + c_j \geq 0 \quad \forall j, k$. Moreover, we assume that zero mean sub-component $\widetilde{s}_j(k)$ are mutually statistically independent[5].

Furthermore, we may assume if necessary that the entries of nonsingular mixing matrix $\boldsymbol{A}$ are also non-negative i.e, $a_{ij} \geq 0 \quad \forall i, j$ and optionally that columns of the mixing matrix are normalized vectors to have 1-norm equals unity [118, 40].

We propose two stage procedure. In the first stage, we can apply any standard ICA or BSS algorithm for zero-mean (pre-processed) sensor signals without any constraints in order to estimate the separating matrix $\boldsymbol{W}$ up to an arbitrary scaling and permutation and estimate waveforms of original sources by projecting (nonzero mean) raw sensor signals $s_j(k)$ via the estimated separating matrix $(\widehat{\boldsymbol{s}}(k) = \boldsymbol{W}\boldsymbol{x}(k))$.

It should be noted that since the global mixing-unmixing matrix defined as $\boldsymbol{G} = \boldsymbol{WA}$ after successful extraction of sources is a generalized permutation matrix containing only one nonzero (negative or positive) element in each row and each column, thus the each estimated source in the first stage will be either non-negative or non-positive for every time instant.

In the second stage in order to recover original waveform of sources with correct sign all estimated non positive sources should be inverted, i.e. multiplied by -1. It should be noted that this procedure is valid for an arbitrary nonsingular mixing matrix with both positive and negative elements.

If the original mixing matrix $\boldsymbol{A}$ has non-negative entries then in order to identify it the corresponding vectors of the estimating matrix $\widehat{\boldsymbol{A}} = \boldsymbol{W}^{-1}$ should be multiplied by the factor -1. In this way, we can estimate original sources and blindly identify the mixing matrix satisfying non-negativity constraints. Furthermore, if necessary, we can redefine $\widehat{\boldsymbol{A}}$ and $\hat{\boldsymbol{s}}$ as follows: $\widehat{\hat{a}}_{kj} = \hat{a}_{kj} / \sum_{i=1}^{n} \hat{a}_{ij}$ and $\widehat{\hat{s}}_j = \hat{s}_j (\sum_{i=1}^{n} \hat{a}_{ij})$. After such transformation, the new estimated mixing matrix $\widehat{\widehat{\boldsymbol{A}}}$ has column sums equal to one and the vector $\boldsymbol{x} = \widehat{\widehat{\boldsymbol{A}}}\, \widehat{\widehat{\boldsymbol{s}}}$ is unchanged.

There are several known procedures which are not biased by white or Gaussian noise. In the second stage we can easily identify the mixing matrix $\widehat{\boldsymbol{A}} = \boldsymbol{W}^{-1}$ It should be noted that since the global mixing-umnixing matrix $\boldsymbol{G} = \boldsymbol{WA}$ is generalized matrix containing only one nonzero (negative or positive) element in each row and each column so the estimated sources

Summarizing, from this simple explanation it follows that it is not necessary to develop any special kind of algorithms for BSS with non-negativity constraints (see for example, [90, 123, 118]). Any standard ICA algorithm (batch or on-line) can be applied first for zero-mean signals and waveforms of original sources and desired mixing matrix with non-negativity constraints can be estimated exploiting basic properties of the assumed model, however, under rather strong assumption that original sources are mutually independent.

7.2 Non-negative Matrix Factorization Using Multiplicative Algorithms

The method based on standard ICA approach presented in the previous section enables to estimate the mixing matrix $\boldsymbol{A}$ and non-negative components $s_j(k) = s_{jk} \geq 0$ ($\forall j, k$ only under assumption that original sources are independent.

The NMF (Non-negative Matrix Factorization) introduced by Lee and Seung [90], sometimes called also PMF (Positive Matrix Factorization) which was first introduced by Paatero does not assume explicitly or implicitly sparseness or the mutual statistical independence of components however usually provides sparse decomposition. The NMF found wide applications in spectroscopy, chemometrics and environmental science where the matrices have clear physical meanings and some normalization are imposed to them (for example, the matrix $\boldsymbol{A}$ has columns normalized to unit length).

[5]It should be noted that non negative sources $s_j(k) = \tilde{s}_j(k) + c_j$ are non independent, even zero mean sub-components $\widetilde{s}_j(k)$ are independent, since dc (constant) sub-components c_j are dependent. Due to this reason we refer the problem as non-negative blind source separation rather non-negative ICA [123].

NMF decomposes the data matrix $\boldsymbol{X}$ as a product of two matrices $\boldsymbol{A}$ and $\boldsymbol{S}$ having only non-negative elements. This results in reduced representation of the original data. In the reduced data set, each feature is a linear combination of the original attribute set. The NMF has low computational cost and the ability to deal with both dense and sparse data sets.

The NMF method is designed to capture alternative structures inherent in the data and, possibly to provide more biological insight. Lee and Seung introduced NMF in its modern formulation as a method to decompose images. For example, in this context, NMF yielded a decomposition of human faces into parts reminiscent of features such as lips eyes, nose, etc. By contrast to other factorization methods, such as ICA or PCA, to image data yielded components with no obvious visual interpretation. When applied to text, NMF gave some interesting evidence of differentiating meanings of the same word depending on context (semantic polysemy). Here, we attempt to employ NMF to extract hidden interesting components from spectra and/or spectrograms of EEG data.

NMF does not allow negative entries in the matrix factors $\boldsymbol{A}$ and $\boldsymbol{S}$ in the model $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}$. Unlike the other matrix factorization these non-negativity constraints permit the combination of multiple basis signals to represent original signals. But only additive combinations are allowed, because the nonzero elements of $\boldsymbol{A}$ and $\boldsymbol{S}$ are all positive. Thus in such decomposition no subtractions can occur. For these reasons, the non-negativity constraints are compatible with the intuitive notion of combining components to form a whole signal or image, which is how NMF learns a parts-based representation [90].

Whereas the original application of NMF focused on grouping elements of images into parts (using the matrix $\boldsymbol{A}$), we take the dual viewpoint by focusing primarily on grouping samples into components representing by the matrix $\boldsymbol{S}$.

In this section, we overview several adaptive algorithms for the NMF.

Let us consider the following cost function (which is optimal for Gaussian distributed noise):

$$\begin{aligned} & J_1(\boldsymbol{A}, \boldsymbol{S}) = \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}\|^2 = \sum_{i,k} |[\boldsymbol{X}]_{ik} - [\boldsymbol{A}\boldsymbol{S}]_{ik}|^2 \\ & \text{s. t. } \; a_{ij} \geq 0, \quad s_j(k) = s_{jk} \geq 0 \quad \forall\, i, j, k, \end{aligned} \tag{70}$$

where the mixing matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is unknown corresponding to the basis matrix in previous sections, the matrix $\boldsymbol{S} \in \mathbb{R}^{n \times N}$ is composed of the n unknown non-negative sources, the only observable $\boldsymbol{X} \in \mathbb{R}^{m \times N}$ is a data matrix with its rows being mixtures of sources. We assume that matrices $\boldsymbol{X}$, $\boldsymbol{A}$ and $\boldsymbol{S}$ are non-negative. Based only on the observable mixture matrix $\boldsymbol{X}$, we will estimates unknown matrices $\boldsymbol{A}$ and $\boldsymbol{S}$ by using the optimization approach.

For any rank n, the NMF algorithms group the available data into classes or clusters of components. The key open issue is to find whether a given rank n decomposes the data into "meaningful" components. Typically n is chosen so that $(m + N)n < mN$. In general, the NMF algorithms may or may not converge to the same meaningful solutions on each run, depending on the random initial conditions and the kind of the algorithm we use. If a clustering into n classes is strong, we would expect that sample assignment to clusters would vary little from run to run. Although NMF is pure algebraic factorization, it was shown that as the rank n increases the method may uncover some structure or substructures, whose robustness can be evaluated by ran algorithm for gradually increasing n. In fact, NMF may reveal hierarchical structure when it exists but does not force such structure on the data like SCA or ICA does. Thus, NMF may have some advantages in exposing meaningful components and discover fine substructures.

Using the gradient descent approach for this cost function in respect of elements a_{ij} of $\boldsymbol{A}$, we obtain

$$\begin{aligned} & \boldsymbol{\Delta} a_{ij}(l+1) = a_{ij}(l+1) - a_{ij}(l) \\ = \; & -\eta_{ij} \frac{\partial J_1}{\partial a_{ij}} = -\eta_{ij} \left([\boldsymbol{X}\, \boldsymbol{S}^T]_{ij} - [\boldsymbol{A}\, \boldsymbol{S}\, \boldsymbol{S}^T]_{ij} \right). \end{aligned} \tag{71}$$

Analogously, assuming that elements a_{ij} are fixed, we obtain additive update rule for elements s_{jk} of $\boldsymbol{S}$

$$\begin{aligned} & \boldsymbol{\Delta} s_{jk} = s_{jk}(l+1) - s_{jk}(l) \\ = \; & \tilde{\eta}_{jk} \frac{\partial J}{\partial s_{jk}} = -\tilde{\eta}_{jk} \left([\boldsymbol{A}\, \boldsymbol{X}^T]_{jk} - [\boldsymbol{A}^T \boldsymbol{A}\, \boldsymbol{S}]_{jk} \right), \end{aligned} \tag{72}$$

where $s_{jk} = s_j()$ and $x_{ik} = x_i(k)$. The above additive learning rules do not ensure automatically non-negativity

constraints, however, if use the so-called Exponential Gradient (EO) :

$$\boldsymbol{s}_j \leftarrow \boldsymbol{s}_j \exp\left(-\eta \frac{\partial J}{\partial \boldsymbol{s}_j}\right) \tag{73}$$

then non-negativity constraints are automatically preserved, because the elements of exponential gradients are always positive. Multiplicative learning rules such as EO typically lead to faster convergence than additive updates if the solution of the optimization is sparse, containing a large number of zero elements.

Lee and Seung proposed to choose in (71)-(72) specific learning rates [90, 126]

$$\eta_{ij} = \frac{a_{ij}}{[\boldsymbol{A}\boldsymbol{S}\boldsymbol{S}^T]_{ij}} \tag{74}$$

and

$$\tilde{\eta}_{ij} = \frac{s_{jk}}{[\boldsymbol{A}^T\boldsymbol{A}\,\boldsymbol{S}]_{jk}} \tag{75}$$

what leads to simple multiplicative update rules:

$$a_{ij} \leftarrow a_{ij}\,\frac{[\boldsymbol{X}\,\boldsymbol{S}^T]_{ij}}{[\boldsymbol{A}\,\boldsymbol{S}\,\boldsymbol{S}^T]_{ij}} \tag{76}$$

$$s_{jk} \leftarrow s_{jk}\,\frac{[\boldsymbol{A}^T\,\boldsymbol{X}]_{jk}}{[\boldsymbol{A}^T\boldsymbol{A}\,\boldsymbol{S}]_{jk}} \tag{77}$$

that guarantee the positivity constraints assuming initial conditions are positive[6] and local convergence.

The above learning rules provides usually sparse non-negative representation of data, although they do not guarantee the sparsest possible solution (i.e. not necessary representation which contain the highest possible number of zero elements of $\boldsymbol{S}$ and/or $\boldsymbol{A}$). Moreover, solutions are not unique and algorithms may stuck in local minima. Usually, the better performance (in the sense that it have less probability that it stuck in local minima) has algorithm (84)-(86).

Although standard NMF (without any auxiliary constraints) provides sparseness of its component, we can achieve some control this sparsity by imposing additional constraints to natural non-negativity constraints. In fact, as we already mentioned, we can incorporate sparsity constraints in several ways. For example, in order to ensure a sparse non-negative solution for the cost function (70) it can be modified in quite general form as follows:

$$J_\alpha(\boldsymbol{A},\boldsymbol{S}) = \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}\|^2 + \alpha_A \sum_{ij} f(a_{ij}) + \alpha_S \sum_{jk} f(s_{jk}) \tag{78}$$
$$\text{s. t. } a_{ij} \geq 0, \quad s_j() = s_{jk} \geq 0 \;\; \forall\, i,j,k,$$

where $\alpha_A \geq 0$ and $\alpha_S \geq 0$ are regularization parameters which control tradeoff between sparsity of $\boldsymbol{A}$ and $\boldsymbol{S}$, respectively and accuracy on the NMF ($\boldsymbol{X} \approx \boldsymbol{A}\,\boldsymbol{S}$) and $f(\cdot)$ is suitably chosen function which is measure of sparsity. In order to achieve sparse representation we usually chose $f(s) = |s|$ or simply $f(s_j) = s_j$ or $f(s_j) = s_j \log(s_j)$ with constraints $s_j \geq 0$. Note, that we treat both matrices $\boldsymbol{A}$ and $\boldsymbol{S}$ in symmetric way. By using the gradient descent approach with 1-norm constraints the multiplicative learning rules (76) -(77) can be modified as follows:

$$a_{ij} \leftarrow a_{ij}\,\frac{\left[[\boldsymbol{X}\,\boldsymbol{S}^T]_{ij} - \alpha_A\right]_+}{[\boldsymbol{A}\,\boldsymbol{S}\,\boldsymbol{S}^T]_{ij}} \tag{79}$$

$$s_{jk} \leftarrow s_{jk}\,\frac{\left[[\boldsymbol{A}^T\,\boldsymbol{X}]_{jk} - \alpha_s\right]_+}{[\boldsymbol{A}^T\boldsymbol{A}\,\boldsymbol{S}]_{jk}}, \tag{80}$$

[6]Note that the multiplicative learning rules can not set the some values of a_{ij} and s_{jk} exactly zero, therefore in practice we enforce by introducing a threshold constraints, e.g., s_{ik} equals zero or very small positive value δ if a actual value of $s_{ik} \leq \boldsymbol{\varepsilon}_t$ where the threshold $\boldsymbol{\varepsilon}_t$ determines the noise floor.

where nonlinear operator $[x]_+ = \max\{0, x\}$ is introduced to ensure non-negativity constraints.

Alternative modification has bee proposed by Hoyer [76], which can be expressed in somewhat more general form as

$$a_{ij} \leftarrow a_{ij} \frac{[\boldsymbol{X}\,\boldsymbol{S}^T]_{ij}}{[\boldsymbol{A}\,\boldsymbol{S}\,\boldsymbol{S}^T]_{ij} + \alpha_A} \tag{81}$$

$$s_{jk} \leftarrow s_{jk} \frac{[\boldsymbol{A}^T\,\boldsymbol{X}]_{jk}}{[\boldsymbol{A}^T\boldsymbol{A}\,\boldsymbol{S}]_{jk} + \alpha_S} \tag{82}$$

The sparse NMF procedure can be implemented as follows.

Algorithm Outline: Non-negative Matrix Factorization with Sparsity Constraints

1. Initialize elements of $\boldsymbol{A}(0)$ and $\boldsymbol{S}(0)$ to random non-negative values. For example, choose random $\boldsymbol{S}$ and constrained non-negative least squares (NLS) for $\boldsymbol{A}$. Normalize each column of $\boldsymbol{A}(0)$ to unit 2-norm. Set $l = 0$.

2. Update matrix $\boldsymbol{A}$ using the multiplicative learning rule, e.g. (79).

   Alternatively, if no sparseness constraints are imposed on $\boldsymbol{A}$, we can use [76]

   $$\tilde{\boldsymbol{A}} = \boldsymbol{A}(l) - \eta\,(\boldsymbol{A}(l)\,\boldsymbol{S}(l) - \boldsymbol{X})\;\boldsymbol{S}^T(l).$$

   Force small values of $\boldsymbol{A}$ to be approximately zero, i.e., any values of $\boldsymbol{A}$ smaller than ε are set to zero or very small value δ. Normalize each column of $\boldsymbol{A}$ to the unit norm ($\boldsymbol{A}(l+1) = \tilde{\boldsymbol{A}}$).

3. Update matrix $\boldsymbol{S}$ using modified multiplicative learning rule, e.g. (80). Force small values of $\boldsymbol{S}$ to be approximately zero.

4. Iterate (back to Step 2) till convergence is achieved.

Alternative cost function which intrinsically ensures non-negativity constraints and it is related the Poisson likelihood is a functional based on Kullback-Leibler divergence [90, 126]:

$$\begin{aligned} J_2(\boldsymbol{A}, \boldsymbol{S}) &= D(\boldsymbol{X} \,||\, [\boldsymbol{A}\boldsymbol{S}]) \\ &= \sum_{i,k} \left(x_{ij} \log \frac{x_{ik}}{[\boldsymbol{A}\boldsymbol{S}]_{ik}} - x_{ik} + [\boldsymbol{A}\boldsymbol{S}]_{ik} \right) + \alpha_A \sum_{ij} a_{ij} + \alpha_S \sum_{ik} s_{ik}, \end{aligned} \tag{83}$$

where two optional additional terms are introduced in order to impose sparseness of the components. The non-negative coefficients $\alpha_A \geq 0$ and $\alpha_S \geq 0$ control the sparsity profiles of the matrix $\boldsymbol{A}$ and $\boldsymbol{S}$, respectively. The minimization of this cost function leads to multiplicative learning rules:

$$\tilde{a}_{ij} \leftarrow a_{ij} \frac{\sum_{k=1}^{N} s_{jk}\,(x_{ik}/[\boldsymbol{A}\,\boldsymbol{S}]_{ik})}{(1+\alpha_A)\sum_{p=1}^{N} s_{jp}}, \tag{84}$$

$$s_{jk} \leftarrow s_{jk} \frac{\sum_{i=1}^{m} a_{ij}\,(x_{ik}/[\boldsymbol{A}\,\boldsymbol{S}]_{ik})}{(1+\alpha_S)\sum_{q=1}^{m} a_{qj}}, \tag{85}$$

$$a_{ij} \leftarrow \frac{\tilde{a}_{ij}}{\sum_i \tilde{a}_{ij}} \tag{86}$$

During the above updates, we should update the matrices $\boldsymbol{A}$ and $\boldsymbol{S}$ alternatively. However, it should be noted that we do not need to update the whole matrices. Instead, after updating one row of $\boldsymbol{A}$, we need to update the corresponding column of $\boldsymbol{S}$ and so on, since we only need one row (or column) of corresponding matrices occurring in the learning rules. Due to some physical constraints and also in order to achieve a unique solution it is necessary usually to normalize in each iteration the columns of $\boldsymbol{A}$ or rows of $\boldsymbol{S}$ to unity or fixed norm.

Above NMF multiplicative algorithm is closely related to the SMART (Simultaneous Multiplicative Algebraic Reconstruction Technique) developed and analyzed by Byrne in 1997 [40]

$$\begin{aligned} s_{jk} &\leftarrow s_{jk} \exp\left(\sum_{i=1}^{m} \bar{a}_{ij} \log(\frac{x_{ik}}{\bar{\boldsymbol{a}}_i^T \boldsymbol{s}(k)})\right) \\ &= s_{jk} \prod_{i=1}^{m} \left(\frac{x_{ik}}{\bar{\boldsymbol{a}}_i^T \boldsymbol{s}(k)}\right)^{a_{ij}}, \end{aligned} \tag{87}$$

where $\bar{\boldsymbol{a}}_i$ is the i-th row of the normalized matrix $\boldsymbol{A}$ to unity 1-norm of columns.

7.3 Local NMF Algorithm

The standard NMF models does not impose any constraints on the described above is not able reveal local features in the data $\boldsymbol{X}$. In order to learn more local features the alternative cost function has been proposed [102]:

$$J_3(\boldsymbol{A}, \boldsymbol{S}) = \sum_{i,k} \left(x_{ij} \log \frac{x_{ik}}{[\boldsymbol{AS}]_{ik}} - x_{ik} + [\boldsymbol{AS}]_{ik} + \beta_A \, u_{ik} \right) - \beta_S \, \sum_i v_{ii},$$

where $\boldsymbol{U} = [u_{ik}] = \boldsymbol{A}^T \boldsymbol{A}$ and $\boldsymbol{V} = [v_{ik}] = \boldsymbol{S}\boldsymbol{S}^T$ and β_A, β_S are some non-negative coefficients. In the above cost two additional terms have been introduced in order impose the following constraints:

- To make basis vectors $\boldsymbol{a}_j$ as orthogonal as possible in order to minimize redundancies. This is accomplished by minimizing the terms $\sum_{i \neq k} u_{ik}$
- To minimize the number of basis components (column $\boldsymbol{a}_j$ of $\boldsymbol{A}$) required to represent $\boldsymbol{X}$. In other words, we wish that the basis vectors contains as many nonzero elements as possible. This is accomplished by minimizing $\sum_i u_{ii}$.
- To retain the components that give the most (information) variance. This is equivalent to maximizing $\sum_i v_{ii}$.

The multiplicative update rules for local NMF take the following form:

$$s_{jk} \leftarrow \sqrt{s_{jk} \sum_{i=1}^{m} a_{ij} \, \frac{x_{ik}}{[\boldsymbol{A}\,\boldsymbol{S}]_{ik}}}, \tag{88}$$

$$\tilde{a}_{ij} \leftarrow a_{ij} \, \frac{\sum_{k=1}^{N} s_{jk} \, (x_{ik}/[\boldsymbol{A}\,\boldsymbol{S}]_{ik})}{\sum_{p=1}^{N} s_{jp}}, \tag{89}$$

$$a_{ij} \leftarrow \frac{\tilde{a}_{ij}}{\sum_i \tilde{a}_{ij}} \tag{90}$$

It should be noted that for all multiplicative learning rules ensures non-negativity of matrices if initial matrices was also non-negative. Usually we start form arbitrary non-negative matrices (for example, elements of matrices are uniformly distributed from 0 to1). Iteration should be continued until the RMS error change will be negligible small (say, less than 0.01%). Since all presented algorithms are based on gradient descent approach they only guarantee to achieve only local minima. To address this limitation, we can repeat the procedure several times starting form different initial matrices. The NMF factorizations leading to lowest RMS error should be used in further analysis.

An essential feature of the NMF approach is that it reduces the data set from its full data space to lower dimensional NMF space determined by rank n (typically, $n < (mN)/(m+N)$).

The utility of NMF for estimating latent (hidden) components and their clusters or classes from EEG data (represented in the frequency or time frequency domain) stems from its non-negativity constraints, which facilitates the detection of sharp boundaries among classes. These components are typically sparse, localized, and relatively independent, which makes a natural signals decomposition suitable for flexible and promising interpretation. Despite its promising features, NMF has the limitation due to of non-uniqueness of solutions and difficulties chose optimal dimensions of matrices $\boldsymbol{A}$ and $\boldsymbol{S}$, as well as interpretation of some components.

In summary, NMF is a powerful technique for extracting, clustering and classifying of latent components. Recently NMF was generalized to a multilayer generative network, which leads to *multiplicative up-propagation* learning [2]. However, the challenge that still remains is to provide a meaningful physiological interpretation to some of NMF discovered latent components or classes of components when the structures of the true sources are completely unknown.

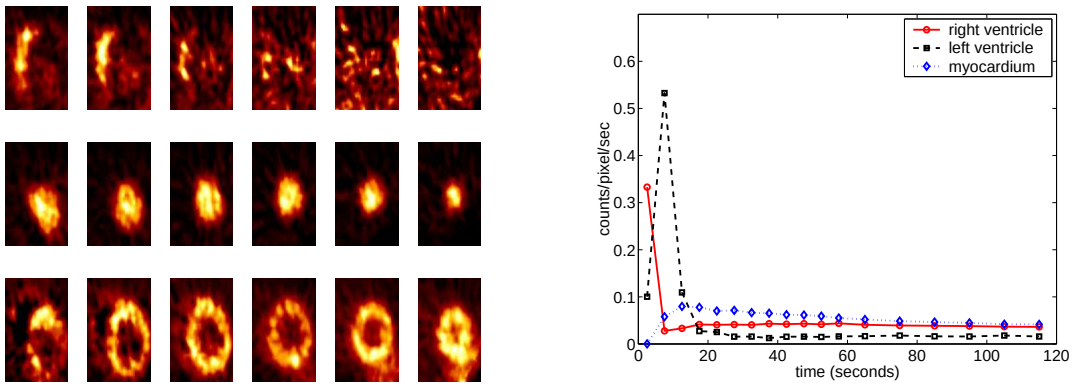
7.4 Application of NMF to PET Image Analysis

An interesting application of NMF to dynamic PET image analysis, appeared in [94, 93].

In [94, 93], they performed $H_2^{15}O$ PET scans on seven dogs at rest and after pharmacological stress using Adnenosine or Dipyridamole. All the scans were acquired with an ECAT EXACT 47 scanner (Simens-CTI, Knoxville, USA) which has an intrinsic resolution of 5.2 mm FWHM (full width at half maximum) and images 47 contiguous planes with thickness of 3.4 mm simultaneously for a longitudinal field of view of 16.2 cm. Before $H_2^{15}O$ administration, transmission scanning was performed using three Ge-68 rod sources for attenuation correction. Dynamic emission scans (5 sec$\times$12, 10 sec$\times$9, 30 sec$\times$3) were initiated simultaneously with the injection of 555-740 MBq $H_2^{15}O$. Transaxial images were reconstructed by means of a filtered back-projection algorithms as $128 \times 128 \times 47$ matrices with a size of $2.1 \times 2.1 \times 3.4$ mm

The initial eighteen frames (two minutes) of PET images were used for analysis. The dynamic PET images were re-oriented to short axis and were re-sampled to produce 1-cm-thick slices in order to increase the signal to noise ratio. Only the cardiac regions were then masked to remove extra cardiac components and to reduce the quantity of data and hence the burden of computation. The resulting masked images with dimension of $32 \times 32 \times 6 \times 18$ (pixel $\times$ pixel $\times$ plane $\times$ frame) were reformulated to 18×6144 (frame $\times$ pixel) data matrix $\boldsymbol{X} = \boldsymbol{AS}$.

Each row of the matrix $\boldsymbol{S}$ corresponds to basis image which represent cardiac component. Figure 11 shows the basis images obtained using NMF. Three cardiac components (right ventricle, left ventricle, myocardium) were successfully extracted. Each column vector of the matrix $\boldsymbol{A}$ represent the time activity curve (TAC) which is useful to calculate blood flow estimation [91]. Figure 11 (b) shows the TAC, where two peaks at each of right ventricle and left ventricle, more dispersion in left ventricle and mycardium.



(a) (b)

Figure 11. Basis images (a) and time activity curves (b), computed by NMF, are shown.

7.5 EEG/MEG Applications and Open Problems

A great challenge in neurophysiology is to non-invasively asses the physiological changes occurring in different parts of the brain. These activations can be modelled and measured often as neuronal brain source signals that indicate the function or malfunction of various physiological sub-systems. To extract the relevant information for diagnosis and therapy, expert knowledge not only in medicine and neuroscience but also in statistical signal processing are required.

To understand human neurophysiology, we currently rely on several types of non-invasive neuroimaging techniques. These techniques include electroencephalography (EEG) and magnetoencephalography (MEG)

[106, 80]. Brain source signals are extremely weak, non-stationary signals and distorted by noise, interference and on-going activity of the brain. Moreover, they are mutually superimposed and low-passed filtered by EEG/MEG recording systems (see Figure 1 (b)). Besides classical signal analysis tools (such as adaptive supervised filtering, parametric or non-parametric spectral estimation, time-frequency analysis, and higher-order statistics), intelligent blind signal processing techniques (IBSP) can be used for preprocessing, noise and artifact reduction, enhancement, detection and estimation of neuronal brain source signals.

In recent years, a great interest has been in applying high density array EEG systems to analyze patterns and imaging of the human brain, where EEG has desirable property of excellent time resolution. This property combined with other systems such as eye tracking and EMG (electromyography) systems with relatively low cost of instrumentations makes it attractive for investigating the higher cognitive mechanisms in the brain and opens a unique window to investigate the dynamics of human brain functions as they are able to follow changes in neural activity on a millisecond time-scale. In comparison, the other functional imaging modalities (positrontomography (PET) and functional magnetic resonance imaging (fMRI)) are limited in temporal resolution to time scales on the order of, at best, one second by physiological and signal-to-noise considerations.

Determining active regions of the brain, given EEG/MEG measurements on the scalp is an important problem. A more accurate and reliable solution to such a problem can give information about higher brain functions and patient-specific cortical activity. However, estimating the location and distribution of electric current sources within the brain from EEG/MEG recording is an ill-posed problem, since there is no unique solution and the solution does not depend continuously on the data. The ill-posedness of the problem and distortion of sensor signals by large noise sources makes finding a correct solution a challenging analytic and computational problem.

If one knows the positions and orientations of the sources in the brain, one can calculate the patterns of electric potentials or magnetic fields on the surface of the head. This is called the forward problem. If otherwise one has only the patterns of electric potential or magnetic fields on the scalp level, then one needs to calculate the locations and orientations of the sources. This is called the inverse problem. Inverse problems are notoriously more difficult to solve than forward problems. In this case, given only the electric potentials and magnetic fields on the scalp surface, there is no unique solution to the problem. The only hope is that there is some additional information available that can be used to constrain the infinite set of possible solutions to a single unique solution. This is where intelligent blind source separation can be used [39].

Every EEG electrode montage acts as a some kind of spatial filters of cortical brain activity and the BSS procedure can be considered also as spatial filter which attempt to cancel the effect of superposition of various brain activities, and estimated components represent physiologically different processes [106, 105, 60].

BSS and its related methods like PARAFAC or SPCA are promising approaches for the elimination of artifacts and noise from EEG/MEG data and enhancement of neuronal brain sources. In fact, for these applications, ICA/BSS techniques have been successfully applied to remove artifacts and noise including background brain activity, electrical activity of the heart, eye-blink and other muscle activity, and environmental noise efficiently [84, 83, 54, 141, 80, 108]. However, most of the methods require manual detection, classification of interference components and the estimation of the cross-correlation between independent components and the reference signals corresponding to specific artifacts [84, 105, 54].

One important problem is how to automatically detect, extract and eliminate noise and artifacts. Another relevant problem is how to enhance extract and classify the "brain sources".

A conceptual model for the elimination of noise and other undesirable components from multi-sensory data is depicted in Figure 12. First, BSS is performed using suitably chosen robust (with respect to the noise) algorithm by a linear transformation of sensory data as $\mathbf{y}(k) = \mathbf{W}\mathbf{x}(k)$, where the vector $\mathbf{y}(k)$ represents the specific components (e.g., sparse, smooth, spatio-temporally decorrelated or statistically independent components). Then, the projection of interesting or useful components (e.g., spatio-temporal decorrelated or independent activation maps) $\tilde{y}_j(k)$ back onto the sensors (electrodes). The corrected or "cleaned" sensor signals are obtained by linear transformation $\hat{\mathbf{x}}(k) = \mathbf{W}^{+}\tilde{\mathbf{y}}(k)$, where $\mathbf{W}^{+}$ is some pseudo-inverse of the unmixing matrix $\mathbf{W}$ and $\tilde{\boldsymbol{y}}(k)$ is the vector obtained from the vector $\boldsymbol{y}(k)$ after removal of all the undesirable components (i.e., by replacing them with zeros). The entries of estimated attenuation matrix $\hat{\boldsymbol{A}} = \boldsymbol{W}^{+}$ indicate how strongly each electrode picks up each individual component. Back projection of some significant components $\hat{\mathbf{x}}(k) = \mathbf{W}^{+}\tilde{\mathbf{y}}(k)$ allows us not only remove some artifacts and noise but also to enhance EEG data. In many cases the estimated components must be at first filtered or smoothed in order to identify all significant components. In addition to the denoising and artifacts removal, BSS techniques can be used to decompose EEG/MEG data into individual components, each representing

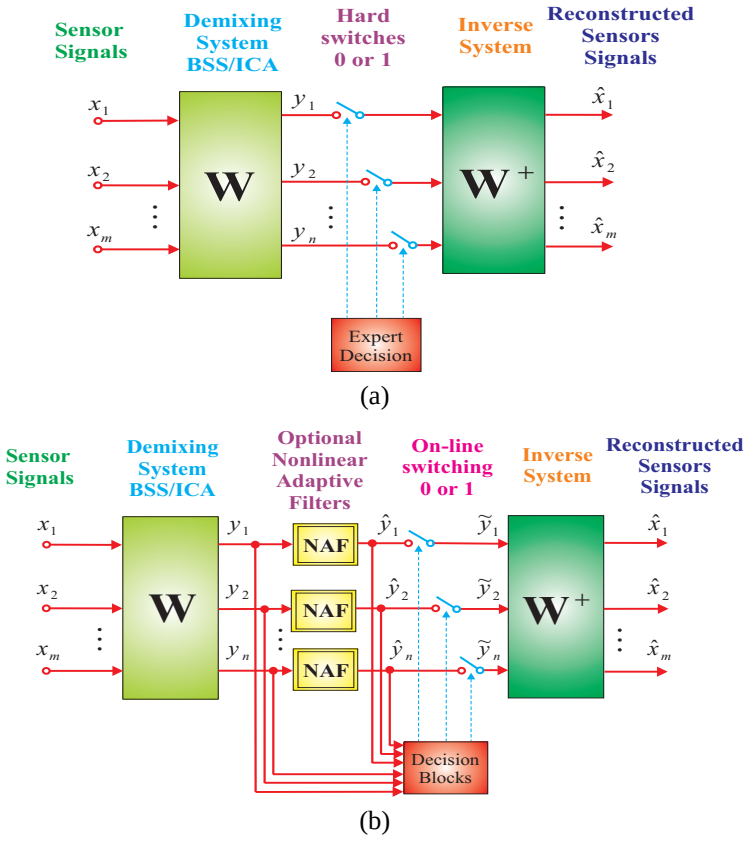


Figure 12. Basic models for removing undesirable components like noise and artifacts and enhancing multi-sensory (e.g., EEG/MEG) data: (a) Using expert decision and hard switches, (b) using auxiliary nonlinear adaptive filters to smooth the components and hard switches. Often the estimated components are also normalized, ranked, ordered and clustered in order to identify significant and physiological meaningful sources or artifacts.

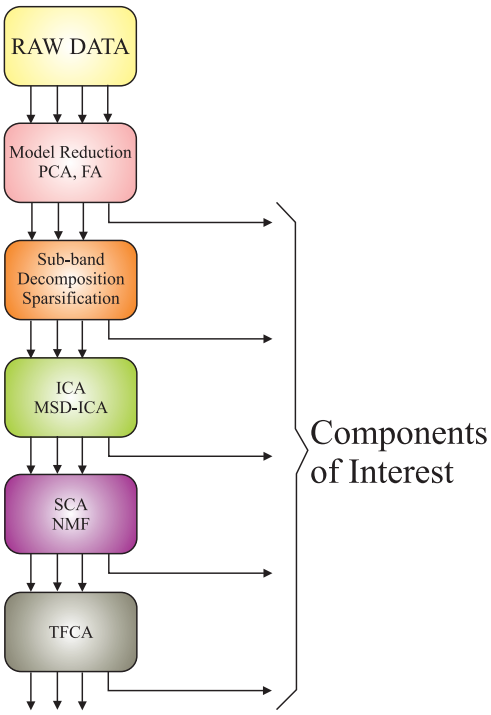


Figure 13. Conceptual model of sequential blind sources extraction. In each stage different criterion can be used.

a physiologically distinct process or brain source. The main idea here is to apply localization and imaging methods to each of these components in turn. The decomposition is usually based on the underlying assumption of sparsity and/or statistical independence between the activation of different cell assemblies involved. An alternative criteria for the decomposition are spatio-temporal decorrelation, temporal predictability or smoothness of components.

The BSS or more general BSP approaches are promising methods for the blind extraction of useful signals from the EEG/MEG data. The EEG/MEG data can be first decomposed into useful signal and noise subspaces using standard techniques PCA, SPCA or Factor Analysis (FA) and standard filtering. Next, we apply BSS algorithms to decompose the observed signals (signal subspace) into specific components. The BSS approaches enable us to project each component (localized "brain source") onto an activation map at the skull level. For each activation map, we can apply an EEG/MEG source localization procedure, looking only for a single dipole (or brain source) per map. By localizing multiple sources independently, we can dramatically reduce the computational complexity and increase the likelihood of efficiently converging to the correct and reliable solution.

One of the biggest strength of BSS approach is that it offers a variety of powerful and efficient algorithms that are able to estimate various kind of sources (sparse, independent, spatio-temporally decorrelated, smooth etc.). Some of the algorithms, e.g., AMUSE or TICA [40, 48, 58, 59, 34], are able to automatically rank and order the component according to their complexity or sparseness measures. Some algorithms are very robust in respect to noise (e.g., SOBI or SONS) [36, 37, 35, 45, 68]. In some cases, it is recommended to use algorithms in cascade (multiple) or parallel mode in order to extract components with various features and statistical properties [40]. In real world scenario latent (hidden) components (e.g., brain sources) have various complex properties and features. In other words, true unknown sources are seldom all sparse or only all statistically independent, or all spatio-temporally decorrelated. Thus, if we apply only one single technique like ICA or SCA or STD we usually fail to extract all hidden components. We need rather to apply fusion strategy or combination of several criteria and associated algorithms to extract all desired sources. We may apply here two possible approaches. The most promising approach is a sequential blind extraction (see Figure 13) in which we extract components one by one in each stage applying different criterion (e.g., statistical independence, sparseness. smoothness etc). In this way, we can extract sequentially different components with various properties.

In alternative approach, after suitable preprocessing, we perform simultaneously (in parallel way) several BSS methods (ICA, SCA, STD, TFCA). Next the estimated components are normalized, ranked, clustered and

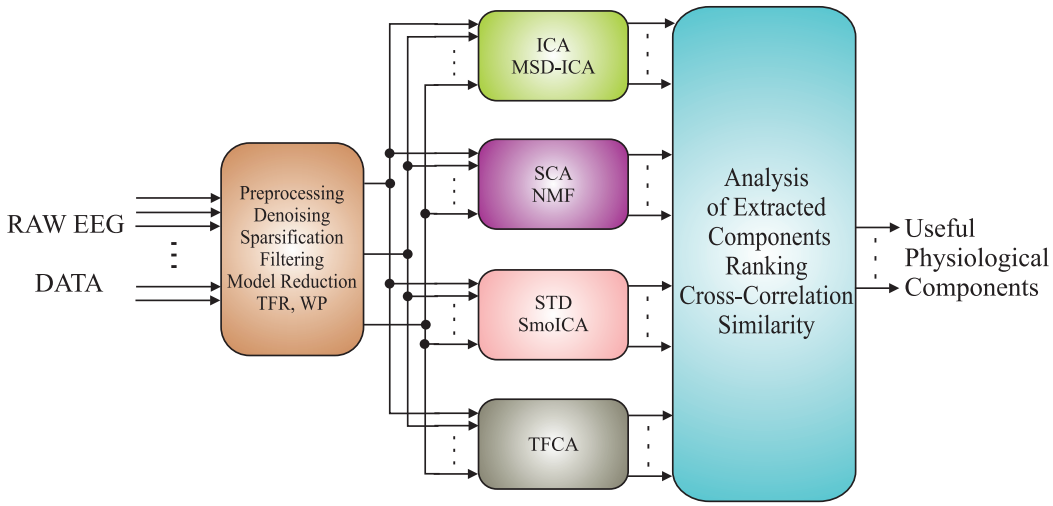


Figure 14. Parallel model employing fusion strategy of BSS algorithms for estimation of physiologically meaningful event-related brain sources. The reliability of estimated sources or components should be analyzed by investigating the spread of the obtained components for many trials and possibly many subjects. Usually, the useful and significant components corresponds to small and well separated clusters from the rest of components, while unreliable components usually do not belong to any cluster.

compared to each other using some similarity measures (see Figure 14). Furthermore, the components are back projected to scalp level and brain sources are localized on basis of clusters of sub-components. In this way, on basis of *a priori* knowledge (e.g., information about external stimuli for event related brain sources), we can identify components with some electrophysiological meaning and specific localizations.

In summary, blind source separation and generalized component analysis (BSS/GCA) algorithms allows [39, 67, 105]:

1. Extract and remove artifacts and noise from raw EEG/MEG data.
2. Recover neuronal brain sources activated in cortex (especially, in auditory, visual, somatosensory, motoric and olfactory cortex).
3. Improve the signal-to-noise ratio (SNR) of evoked potentials (EP's), especially AEP, VEP and SEP.
4. Improve spatial resolution of EEG and reduce level of subjectivity involved in the brain source localization.
5. Extract features and hidden brain patterns and classify them.

Applications of BSS show special promise in the areas of non-invasive human brain imaging techniques to delineate the neural processes that underlie human cognition and sensoro-motor functions. These approaches lead to interesting and exciting new ways of investigating and analyzing brain data and develop new hypotheses how the neural assemblies communicate and process information. This is actually an extensive and potentially promising research area. However, these techniques and methods still remain to be validated at least experimentally to obtain full gain of the presented approach.

The fundamental problems here are: What are the system's real properties and how can we get information about them? What is valuable information in the observed data and what is only noise or interference? How can the observed (sensor) data be transformed into features characterizing the brain sources in a reasonable way?

8. Feature Extraction from Speech Signals

ICA can be used for finding statistically efficient representations of speech and natural sounds [92, 100]. ICA finds a linear transform of multivariate data which minimizes mutual information among the data. Therefore, ICA

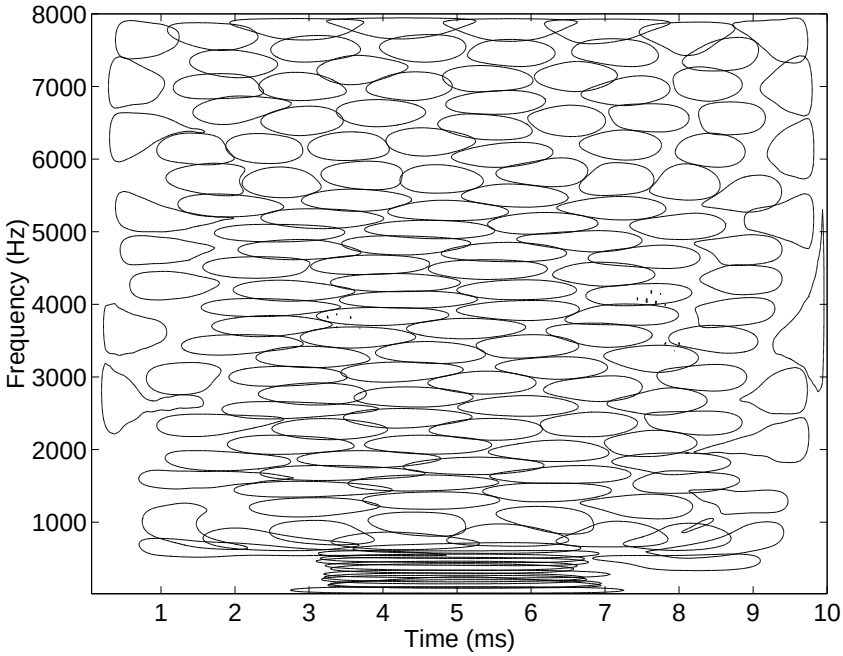


Figure 15. Contour lines of Wigner-Ville distributions for 160 learned basis vectors. Each contour line represents the locus of a half o f the maximum peak amplitude of WVD.

can function as a computational algorithm for sensory information processing such that the redundancy among the input signals is reduced. In addition, to learn efficient representations, they set sparseness constraint on the distribution [92]. Since a sparse distribution has a small percentage of informative values (nonzero values) in the tails and most of the values are around zero, one can encode and decode the data with a small number of the coefficients. The learned speech features (basis vectors) are localized in both time and frequency. Time-frequency analysis of basis vectors shows the property similar with the critical bandwidth of human auditory system as shown in Figure 15.

In order to obtain more complex speech features, an ICA-based computational model also has been developed [87]. After generating speech features at the inner-hair-cells by the existing model of cochlea, they applied an ICA algorithm with topology-preserving mapping. Figure 16 shows the learned speech features, and the features represent complex signal characteristics at the auditory cortex such as onset/offset and frequency modulation in time.

9. Convolutive Source Separation: Problem Formulation

In this section, we formulate a more general model where mixing involves convolution and time-delays. Let us consider a set of unknown independent components, $\boldsymbol{s}(k) = [s_1(k),\ s_2(k),\ \cdots,\ s_n(k)]^T$, such that the components $s_i(k)$ are zero-mean and mutually independent. The independent components are transmitted through channels and mixed to give observations $x_i(k)$. Therefore, the mixtures are linear combinations of delayed and filtered versions of the independent components. One of them can be expressed as

$$x_i(k) = \sum_{j=1}^{n} \sum_{p=0}^{L_m-1} a_{ij}(p) s_j(k-p), \tag{91}$$

where $a_{ij}(p)$ denotes a mixing filter coefficient.

The task is to estimate the independent components from the observations without resort to *a priori* knowledge about the mixing system. As in ICA of instantaneous mixtures, ICA of convolutive mixtures also needs certain assumptions about the independent components such as approximate distributions and statistics. Furthermore, since ICA of convolutive mixtures has indeterminacy of the estimated independent components up to permutation

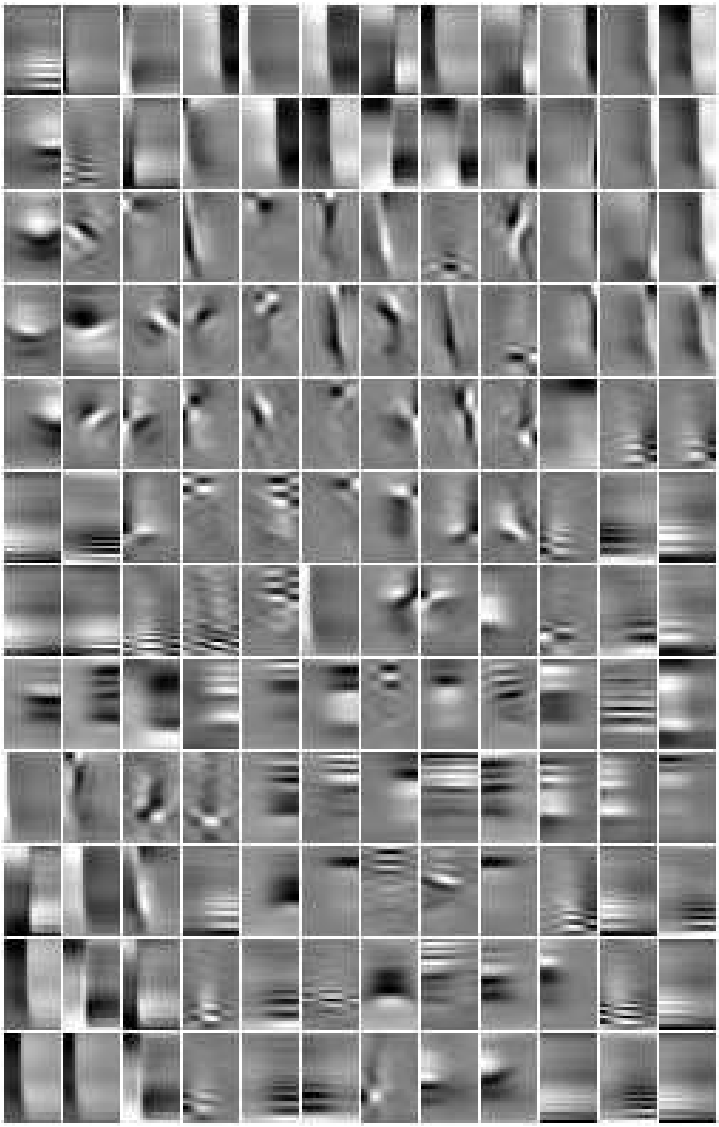


Figure 16. Spectra for 144 learned complex speech features.

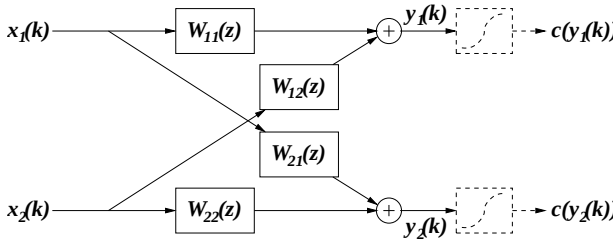


Figure 17. A 2×2 feedforward network for ICA of convolutive mixtures.

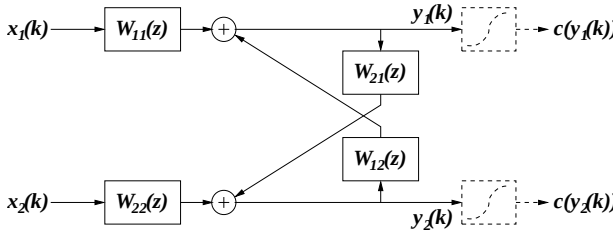


Figure 18. A 2×2 feedback network for ICA of convolutive mixtures.

and arbitrary filtering, some algorithms attempt to make the estimated signals temporally whitened. The whitening may degrade outputs in many applications such as separation of natural signals and can be avoided by forcing some constraints [139] or post-processing [73]. In order to perform ICA for convolutive mixtures, most methods try to apply their algorithms to the mixtures in the time domain [139, 112, 6] and the frequency domain [130, 115, 12]. In addition, some papers have recently proposed the methods which use filter banks or subbands [117, 116, 77, 62].

10. Time-Domain Methods

10.1 Architectures: Feedforward vs. Feedback

To obtain the independent components from observations, one can consider two types of networks in the time domain. One of them is a feedforward architecture which can be expressed as

$$y_i(k) = \sum_{j=1}^{m} \sum_{p=0}^{L_a} w_{ij}(p) x_j(k-p), \tag{92}$$

where adaptive filters $w_{ij}(p)$ force outputs $y_i(k)$ to reproduce the original independent components $s_i(k)$ [139]. Figure 17 illustrates a 2×2 feedforward network.

On the other hand, a feedback architecture can be constructed for the inverse system and expressed as [139]

$$y_i(k) = \sum_{p=0}^{L_a} w_{ii}(p) x_i(k-p) + \sum_{j=1, j\neq i}^{n} \sum_{p=1}^{L_a} w_{ij}(p) y_j(k-p). \tag{93}$$

The architecture consists of three different filter coefficients: zero-delay weights in direct filters $w_{ii}(0)$, other weights in direct filters $w_{ii}(p), p \neq 0$, and weights in feedback cross-filters $w_{ij}(p), i \neq j$. A 2×2 feedback network is shown in Figure 18.

Advantage of the feedforward system is that it can learn a more general inverse system since it can approximate a solution for ICA of nonminimum-phase mixing systems [96, 97, 74]. A nonminimum-phase filter can be expressed as a product of a minimum-phase filter with an all-pass filter. The minimum-phase filter has all of its poles and zeros inside the unit circle, and the all-pass filter represents the time-delay of the nonminimum-phase filter with a unit frequency magnitude response. Thus, the inverse of the nonminimum-phase filter is a product of the

inverse of a minimum-phase filter, which is a stable causal filter, with the inverse of an all-pass filter, which is time-advance. The resulting filter is a non-causal stable filter. By imposing an appropriate time-delay, a feedforward system can be realized and used for inverting a nonminimum-phase mixing system.

The distortion of the estimated independent components should also be considered. Let us model a situation in the z-transform domain such that two independent components are mixed to give two observations as follows:

$$\begin{aligned} X_1(z) &= A_{11}(z)S_1(z) + A_{12}(z)S_2(z), \\ X_2(z) &= A_{21}(z)S_1(z) + A_{22}(z)S_2(z), \end{aligned} \tag{94}$$

where the upper cases denote the z-transforms of the corresponding lower cases in (91). Since ICA of convolutive mixtures has indeterminacy up to arbitrary filtering, the estimated independent components will be distorted by filtering depending on the ICA algorithm used. If the original independent components are i.i.d. signals, $S_1(z)$ and $S_2(z)$ can be recovered by temporal whitening.

However, many independent components including natural signals are not i.i.d. In this case, several methods first estimate the innovation processes of the independent components and then build a post-processing filter which will try to artificially color the signal [140]. One of the desired results may be $A_{11}(z)S_1(z)$ and $A_{22}(z)S_2(z)$, which is what each observation would obtain in the absence of the interfering source, because the result is not affected by any other distortion except mixing process. By including whitening filters in a separating structure, a method has been proposed which directly extracted colored components in one step [1]. If the feedback architecture is used, the result can be obtained by forcing direct filters $w_{ii}(p)$ to scaling factors [139]. Here, note that ICA can be achieved when $A_{11}(z)$ and $A_{22}(z)$, in addition to the mixing system, have stable inverses.

10.2 Basic Algorithms ICA algorithms for convolutive mixtures in the time domain are not as various as those for instantaneous mixtures because of the complexity of architectures to perform ICA. In this paper, we go through two major algorithms: infomax algorithm and decorrelation algorithm.

Let us pass outputs of the architectures through bounded nonlinear functions, which approximate the cumulative density functions (cdfs) of the original independent components, to give $c(y_i(k))$. If outputs $y_i(k)$ are desired independent components, $c(y_i(k))$ follow a uniform density which has the largest entropy among distributions of bounded variables. Infomax algorithm performs ICA for convolutive mixtures by maximizing the entropy of $y_i(k)$ [139, 17].

For a feedforward architecture, infomax algorithm provides learning rules of the adaptive filter coefficients as follows [139]:

$$\begin{aligned} \Delta \boldsymbol{W}(0) &\propto [\boldsymbol{W}^T(0)]^{-1} - \boldsymbol{f}(\boldsymbol{y}(k))\boldsymbol{x}^T(k), \\ \Delta w_{ij}(p) &\propto -f_i(y_i(k))x_j(k-p), \quad p \neq 0, \quad f_i(y_i(k)) = -\frac{\frac{\partial p_i(y_i(k))}{\partial y_i(k)}}{p_i(y_i(k))}, \end{aligned} \tag{95}$$

where $\boldsymbol{W}(0)$ is the matrix composed by zero-delay weights, and $\boldsymbol{y}(k)$ and $\boldsymbol{x}(k)$ denote a set of estimated independent components and the observation vector, respectively. In addition, $f_i(\cdot)$ is called a score function and $p_i(y_i)$ denotes the pdf of y_i. Learning rules for a feedback architecture are [139]

$$\begin{aligned} \Delta w_{ii}(0) &\propto 1/w_{ii}(0) - f_i(y_i(k))x_i(k), \\ \Delta w_{ii}(p) &\propto -f_i(y_i(k))x_i(k-p), \quad p \neq 0, \\ \Delta w_{ij}(p) &\propto -f_i(y_i(k))y_j(k-p), \quad i \neq j. \end{aligned} \tag{96}$$

On the other hand, the second-order statistics can be used for ICA of convolutive mixtures if the original signals are non-stationary. A non-negative cost function can be given as [112]

$$Q = \frac{1}{2B}\sum_{b=1}^{B}\left(\log\det\left\langle \mathrm{diag}(\boldsymbol{y}(k)\boldsymbol{y}(k)^T)\right\rangle_b - \log\det\left\langle \boldsymbol{y}(k)\boldsymbol{y}(k)^T\right\rangle_b\right), \tag{97}$$

where $\langle\cdot\rangle_b$ denotes the time-averaging operator for the bth local analysis block, and B is the number of the local analysis blocks. Note that the cost function takes the minimum value only when the second-order cross-correlation

becomes zero. A gradient learning rule can be obtained by minimizing the cost function with respect to the adaptive parameters.

10.3 Applying the Natural Gradient to ICA Networks

The ordinary gradient has provided most of the popular learning algorithms in various optimization frameworks [73]. However, the parameter space is not Euclidean in many cases. In those cases, the steepest direction of a function is not given by the ordinary gradient, but by the natural gradient [5, 3]. Therefore, it is commonly known that the natural gradient improves convergence speed significantly [3].

By applying the natural gradient to the infomax algorithm, a learning rule for a feedforward architecture can be derived as [6, 31]

$$\Delta \boldsymbol{W}(p) \propto \boldsymbol{W}(p) - \boldsymbol{f}(\boldsymbol{y}(k))\boldsymbol{r}^T(p), \quad f_i(y_i(k)) = -\frac{\frac{\partial p_i(y_i(k))}{\partial y_i(k)}}{p_i(y_i(k))}, \tag{98}$$

where $\boldsymbol{W}(p)$ is the matrix composed by the pth delay filter coefficients, and $\boldsymbol{r}(p) = \sum_{l=0}^{L_a} \boldsymbol{W}^T(l)\boldsymbol{y}(k-p+l)$. Unfortunately, the natural gradient learning rule shows that the update of $\boldsymbol{W}(p)$ depends on future outputs $\boldsymbol{y}(k-p+l),\ p-l<0$, through $\boldsymbol{r}(p)$. In addition, it involves very intensive computation to compute all $\boldsymbol{r}(p),\ p=0,\cdots,L_a$, at each time step. Practically, the algorithm is modified by imposing a L_a sample delay to remove the non-causal terms and reusing past results. With this modification, the algorithm is approximated as

$$\Delta \boldsymbol{W}(p) \propto \boldsymbol{W}(p) - \boldsymbol{f}(\boldsymbol{y}(k-L_a))\boldsymbol{r}^T(k-p), \tag{99}$$

where $\boldsymbol{r}(k) = \sum_{l=0}^{L_a} \boldsymbol{W}^T(L_a-l)\boldsymbol{y}(k-l)$.

More generally, the natural gradient can be given in the z-transform domain by [3]

$$\tilde{\bigtriangledown} Q = \bigtriangledown Q(z)\boldsymbol{W}^T(z^{-1})\boldsymbol{W}(z), \tag{100}$$

where $\bigtriangledown Q$ denotes the ordinary gradient. In addition, the natural gradient can be applied to a feedback architecture [30].

11. Frequency-Domain Methods

11.1 Overall Flow

When a mixing environment is quite complex, filters of the ICA network may require thousands of taps to appropriately invert the mixing. In such cases, the time domain methods have a large computational load to compute convolution of long filters and amounts to update filter coefficients. The methods can be implemented in the frequency domain using FFT in order to decrease the computational load because the convolution operation in the time domain can be performed by element-wise multiplication in the frequency domain.

Learning rules can be simply formulated in the frequency domain by the FIR polynominal matrix algebra which extends the algebra of scalar matrices to the algebra of matrices of filters or polynomials [89]. For example, the natural gradient infomax rule for a feedforward architecture can be expressed as

$$\Delta \underline{\boldsymbol{W}} \propto [\underline{\boldsymbol{I}} - \mathrm{fft}(\boldsymbol{f}(\boldsymbol{y}))\{\mathrm{fft}(\boldsymbol{y})\}^H]\underline{\boldsymbol{W}}, \tag{101}$$

where $\underline{\boldsymbol{W}}$ denotes a matrix composed of filters of the feedforward architecture in the frequency domain. A fast implementation of the adaptive filters in the frequency domain can be achieved by employing the overlap and save block technique [96, 114, 63].

However, the learning rule is just the efficient implementation using FFT of the time domain algorithm. In a more real sense, the frequency domain method means performing ICA for instantaneous mixtures in every frequency bins. Note that the convolutive mixtures can be expressed as

$$\boldsymbol{x}(f,k) = \boldsymbol{A}(f)\boldsymbol{s}(f,k), \quad \forall f. \tag{102}$$

Here, $\boldsymbol{x}(f,k)$ and $\boldsymbol{s}(f,k)$ are vectors, which are the frequency components of mixtures and the independent components at frequency f, respectively. $\boldsymbol{A}(f)$ denotes a matrix containing elements of the frequency transforms of

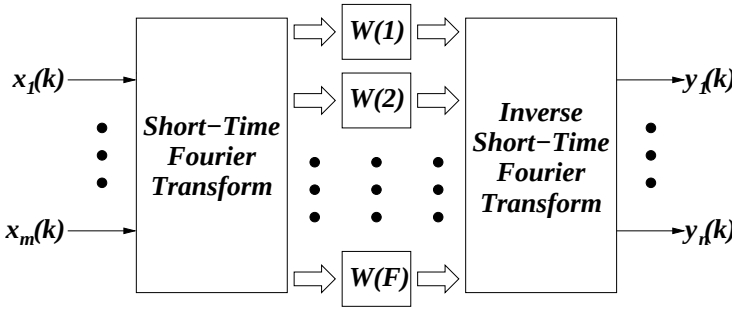


Figure 19. A frequency domain method for ICA of convolutive mixtures.

mixing filters at frequency f. From (102), one can reason that convolutive mixtures can be represented by a set of instantaneous mixtures in the frequency domain. Thus, the independent components can be recovered by applying ICA for instantaneous mixtures at each frequency bin and transforming the results in the time domain as shown in Figure 19. In this figure, $\boldsymbol{W}(f)$ denotes an unmixing matrix at frequency f.

11.2 Basic Algorithms

Various ICA algorithms for instantaneous mixtures can be applied to ICA for convolutive mixtures in the frequency domain. Among these algorithms, we go through several popular algorithms for ICA of convolutive mixtures. Most of the algorithms can be traditionally categorized into two groups. One takes some aspects of higher order statistics into account explicitly [26], and the other does it implicitly through nonlinear functions of outputs [96, 17].

As one of the most prevailing methods for the latter group, infomax algorithm can be considered. The algorithm can be derived by the same manner as explained in the time domain method. Applying the natural gradient [5, 25], infomax learning rule at each frequency bin is

$$\Delta \boldsymbol{W}(f) \propto [\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y}(f,k))\boldsymbol{y}^H(f,k)]\boldsymbol{W}(f). \tag{103}$$

Contrary to the time domain method, the input signals $\boldsymbol{x}(f,k)$ are complex numbers. In order to deal with complex-valued data, score function $\boldsymbol{f}(\cdot)$ should also be changed [128, 131]. It is worth noting that the infomax algorithm provides almost same formulation as maximum likelihood estimation [120, 24], negentropy maximization [70], Bussgang algorithm [98], and minimizing mutual information [5, 59].

In order to perform ICA by computing higher order statistics explicitly, the fourth order cross-cumulants are usually considered. For zero-mean random variables $x_i,\ x_j,\ x_k,\ x_l$, the cross-cumulant is defined as [78]

$$\begin{aligned} cum(x_i, x_j, x_k, x_l) &= E[x_i x_j x_k x_l] - E[x_i x_j]E[x_k x_l] \\ &\quad -E[x_i x_k]E[x_j x_l] - E[x_i x_l]E[x_j x_k]. \end{aligned} \tag{104}$$

Since the cross-cumulants of independent signals are zero, one can obtain desired independent components by minimizing a cost function using the fourth order cross-cumulants or jointly diagonalizing eigenmatrices of the cross-cumulant tensor [26, 78].

It is already known that the diagonalization of the simple cross-covariance matrix provides just the decorrelated components and does not contain sufficient information to estimate the independent components. Therefore, most of the ICA algorithms consider mutual information or higher order statistics. However, ICA can also be performed with the second-order statistics only by adding covariances with time-lags if the original independent components have time-dependencies.

As a simple method, the time-delayed decorrelation algorithm is as follows: The covariance matrix $\boldsymbol{R}_{\boldsymbol{xx}}(k, k-p)$ of $\boldsymbol{x}(k)$ with time-lag p is

$$\boldsymbol{R}_{\boldsymbol{xx}}(k, k-p) = E[\boldsymbol{x}(k)\boldsymbol{x}^H(k-p)] = \boldsymbol{A}\boldsymbol{\Lambda}(k, k-p)\boldsymbol{A}^H, \tag{105}$$

where $\boldsymbol{A}$ denotes the mixing matrix and $\boldsymbol{\Lambda}(k, k-p) = E[\boldsymbol{s}(k)\boldsymbol{s}^H(k-p)]$ is a diagonal matrix. Using the covariance matrices with no time-lag $p = 0$ and a given time-lag, one can construct an eigenvalue problem as

$$\boldsymbol{R_{xx}}(k)\boldsymbol{R_{xx}}^{-1}(k, k-p)\boldsymbol{A} = \boldsymbol{A}\boldsymbol{\Lambda}(k)\boldsymbol{\Lambda}^{-1}(k, k-p), \tag{106}$$

where $\boldsymbol{R_{xx}}(k) = \boldsymbol{R_{xx}}(k, k)$ for simplicity. Therefore, independent components can be obtained by diagonalizing the covariance matrices at the same time [62, 109, 111, 99]. In addition, the number of used covariance matrices can be increased [19].

In addition, non-stationarity of independent components also allows to perform ICA with the second-order statistics. If the independent components are non-stationary, $\boldsymbol{\Lambda}(k) \neq \boldsymbol{\Lambda}(k+K)$ for $K \neq 0$. Therefore, multiple different equations for the covariance matrices such as (105) can be obtained for different choices of K to provide successful estimation of the independent components [119, 142].

Beamforming may be combined with the frequency domain ICA to improve the performance [127]. The proposed system integrated the frequency domain ICA and null beamforming based on the estimated direction-of-arrival (DOA) information.

11.3 Resolving Indeterminacy of Frequency-Domain ICA Algorithms

Using the short-time Fourier transform, frequency domain ICA algorithms regard convolutive mixtures as a set of instantaneous mixtures. Even though an ICA algorithm for instantaneous mixtures precisely estimates an unmixing matrix at each frequency bin, the algorithm will still have indeterminacy of scaling and permutation at each frequency bin. This indeterminacy may deteriorate the performance of the ICA algorithm. Therefore, the permutation and scaling problem should be resolved to reconstruct the desired independent components.

One of the algorithms to solve the permutation and scaling problem makes use of the envelopes of frequency spectra assuming that the independent components have time-varying statistical properties [111, 9]. In this algorithm, first, decomposition of frequency spectra is performed by

$$\boldsymbol{v}(f, k; i) = \boldsymbol{W}(f)^{-1} \begin{bmatrix} 0 \\ \vdots \\ y_i(f, k) \\ \vdots \\ 0 \end{bmatrix}, \tag{107}$$

where $y_i(f, k)$ denotes the ith element of $\boldsymbol{y}(f, k)$. Then, the permutation problem is solved with the envelopes of frequency spectra, each of which is

$$\xi[\boldsymbol{v}(f, k; i)] = \frac{1}{2T+1} \sum_{k'=k-K}^{k+K} \sum_{j=1}^{n} |v_j(f, k'; i)|, \tag{108}$$

where K is a positive constant and $v_j(f, k; i)$ denotes the jth element of $\boldsymbol{v}(f, k; i)$. With the definition of similarity given by

$$\mathrm{sim}(f) \equiv \sum_{i \neq j} r\{\xi[\boldsymbol{v}(f, k; i)],\ \xi[\boldsymbol{v}(f, k; j)]\}, \tag{109}$$

this algorithm sorts frequency bins in order of weakness of similarity among the independent components, so that

$$\mathrm{sim}(f_1) \leq \mathrm{sim}(f_2) \leq \cdots \leq \mathrm{sim}(f_F). \tag{110}$$

Here, F is the number of frequency bins and r denotes a normalized correlation estimated as

$$r\{\alpha(k),\ \beta(k)\} = \frac{\frac{1}{L_k}\sum_k \alpha(k)\beta(k)}{\sqrt{\frac{1}{L_k}\sum_k \alpha^2(k) \cdot \frac{1}{L_k}\sum_k \beta^2(k)}}, \tag{111}$$

where L_k denotes the length of $\alpha(k)$ and $\beta(k)$. For the frequency bin f_1, which has the smallest correlation, its independent components are assigned to specific outputs $\boldsymbol{y}'(f_1, k; i) = \boldsymbol{v}(f_1, k; i)$. Then, for the frequency bins

$\{f_l,\ l = 2, \cdots, F\}$ sorted in the increasing order of the correlation, the independent components are assigned to outputs that have more correlation between the envelopes of the frequency bins. That is,

$$\boldsymbol{y}'(f_l, k; i) = \boldsymbol{v}(f_l, k; \sigma(i)), \tag{112}$$

where the permutation is given as

$$\sigma(i) = \arg\max_{\sigma(i)} \sum_{i=1}^{n} r\{\xi[\boldsymbol{v}(f_l, k; \sigma(i))],\ \sum_{j=1}^{l-1} \xi[\boldsymbol{y}'(f_j, k; i)]\}. \tag{113}$$

The process is repeated in turns until all the frequency bins are covered.

Another well-known algorithm to fix arbitrary permutations is to limit effective length of the unmixing filters after estimating unmixing matrices at sufficiently many frequency bins [119, 146]. The constraint on the effective length links the frequencies and provides a solution for the permutation problem by restricting the unmixing matrices to be continuous or smooth in the frequency domain.

Also, DOA estimation can be used for solving the permutation problem. Assuming linearly arranged and closely spaced sensors and a plain wavefront with no reverberation, the frequency response of a mixing filter $a_{jl}(p)$ is approximated as

$$\boldsymbol{A}_{jl}(f) = \exp(j2\pi f \frac{d_j \cos\theta_l}{c}), \tag{114}$$

where c, d_j, and θ_l denote the propagation velocity, the position of sensor x_j, and the direction of source s_l, respectively. Thus, the frequency response of the overall system can be expressed as

$$\boldsymbol{T}_{ik}(f) = \sum_{j=1}^{n} \boldsymbol{W}_{ij}(f)\boldsymbol{A}_{jk}(f) = \sum_{j=1}^{n} \boldsymbol{W}_{ij}(f) \exp(j2\pi f \frac{d_j \cos\theta_l}{c}). \tag{115}$$

Regarding θ_l as a variable θ provides a directivity pattern, and the directivity patterns can estimate source directions to align permutations [79].

In the method using the envelopes of frequency spectra, a misalignment at a frequency bin may cause consecutive misalignments. However, the DOA method fixes the permutations of a frequency bin regardless of other frequency bins. Since the DOA is computed by an approximation of a mixing system, the DOA method is not precise. In order to exploit the advantages of the two methods, a method fixed the permutations at some frequency bins where the confidence of the DOA method was sufficiently high, and then decided the permutations for the remaining frequency bins by the envelopes of frequency spectra [129].

12. Filter Bank Methods

12.1 Overall Flow

Time domain methods regard convolutive mixtures as results of a big and complicated system and try to estimate an unmixing network all at once which often requires a great many parameters. On the other hand, frequency domain methods decompose the convolutive mixtures so minutely that the decomposed mixtures can be modelled as instantaneous mixtures rather than convolutive mixtures.

For heavily reverberant environments, the performances of the frequency domain ICA methods are seriously degraded because of insufficiency of data to learn the unmixing matrices with a large number of frequency bins. On the other hand, the time domain ICA methods should adapt very long unmixing filters and usually show slow convergence especially for colored inputs. Therefore, the results from a frequency domain ICA method can be regarded as the inputs for a time domain ICA method in order for the time domain ICA to remove the residual components of the frequency domain ICA [112, 113].

Instead of combining the two ICA methods, one can make a compromise between the two extreme cases to give filter bank methods [**?**, 116, 77, 10]. Figure 20 shows a 2×2 network for a filter bank method to perform ICA. In the filter bank methods, the input mixtures are splitted into subband signals by analysis filters. The resulting subband signal is band-limited and can be subsampled. Although the input mixtures are split into subband signals, each subband still covers a somewhat broad frequency band. Moreover, when the subband signal is subsampled,

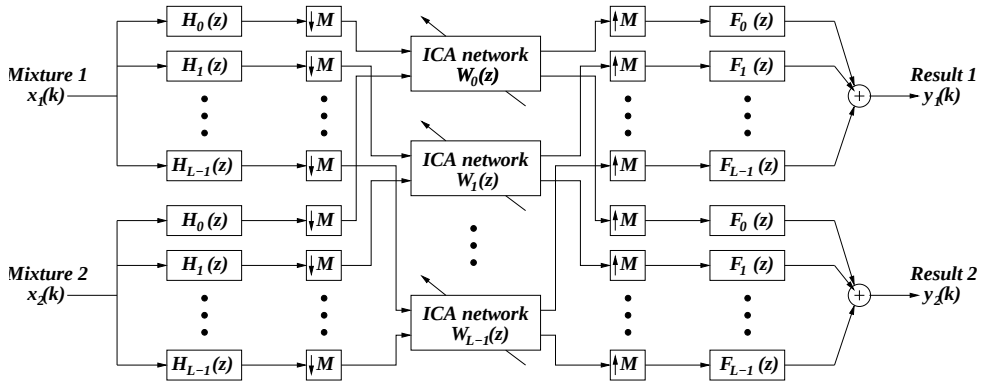


Figure 20. A 2×2 network for a filter bank method to perform ICA.

the decimation factor is usually much smaller than the length of mixing filters. Therefore, the subsampled signals are still convolutive mixtures, but their effective mixing filters decrease by the decimation factor. A typical ICA algorithm for convolutive mixtures can be used to obtain the independent components from the subsampled signals in each subband, and the unmixing filter length is much shorter than that of full-band time domain methods. Each output signal from the unmixing network is expanded, and desired independent components can be reconstructed from the subband output signals through synthesis filters after fixing permutation and scaling.

If critically sampled filter banks are used for analysis and synthesis filter banks, cross adaptive filters between adjacent bands are required to compensate for the distortion caused by aliasing [69], or spectral gaps are required in order not to have aliasing [147]. However, the cross adaptive filters introduce additional adaptive parameters and may cause slow convergence speed and poor performance. On the other hand, the spectral gaps distort reconstructed signals.

However, alias-free property and perfect reconstruction are very essential in order to use filter banks without any side-effects because they limit the performance of the methods primarily apart from capability of ICA algorithms in subbands. With oversampled filter banks, in which the decimation factor is smaller than the number of analysis filters, aliasing can be neglected with each filter having a high stopband attenuation. An oversampled filter bank can be implemented by a uniform complex-valued filter bank [72]. In the filter bank, analysis filters $h_l(k)$ are obtained from a real-valued low-pass prototype filter $q(k)$ by a generalized discrete Fourier transform (GDFT) [55],

$$h_l(k) = e^{j\frac{2\pi}{L}(l+1/2)(k-(L_q-1)/2)} \cdot q(k), \quad l = 0,\ 1,\ \cdots,\ L-1, \quad k = 0,\ 1,\ \cdots,\ L_q-1, \tag{116}$$

where L_q is the length of $q(k)$. Complex-conjugate and time-reversed versions of the analysis filters are selected for synthesis filters

$$f_l(k) = \tilde{h}_l(k) = h_l^*(L_q - k - 1). \tag{117}$$

The prototype filter can be designed by iterative least-squares algorithm with a cost function that considers reconstructiveness and stopband attenuation [72]. In addition, the filter bank can be efficiently implemented by employing polyphase representation of the analysis and synthesis filters [145, 144].

12.2 Performing ICA in subbands and resolving indeterminacy of filter bank methods

When one performs ICA in the oversampled filter bank, adaptive filter coefficients in each subband can be adjusted without any information of the other subbands because of the negligible aliasing of filter bank [145, 144, 143]. Since the subband signals are convolutive mixtures rather than instantaneous mixtures, the ICA algorithm in each subband may be the basically same as time domain methods in Section **??**. To perform ICA with the complex-valued filter banks, subband signals are complex-valued data, and the learning rules of the adaptive filter coefficients are changed to deal with complex-valued data. Using the infomax algorithm for a feedback network

in each subband, the learning rules are

$$\begin{aligned} \Delta w_{ii}(0) &\propto 1/w_{ii}^*(0) - f_i(y_i(k))x_i^*(k), \\ \Delta w_{ii}(p) &\propto -f_i(y_i(k))x_i^*(k-p), \quad p \neq 0, \\ \Delta w_{ij}(p) &\propto -f_i(y_i(k))y_j^*(k-p), \quad i \neq j. \end{aligned} \tag{118}$$

The second-order statistics can also be used for estimating unmixing networks [10]. As in the case of time domain methods, forcing direct filters to scaling factors in each subband does not make the recovered outputs whitened.

Filter bank methods have an unmixing network in each subband and adapt filter coefficients of the network independently of the other subbands. Therefore, the filter bank methods have the same permutation and scaling problem as frequency domain methods. In order to fix this problem, the algorithm in [111] can be applied to the filter bank methods after necessary modifications [116]. When the algorithm is used in the frequency domain, the algorithm multiplies each recovered independent component by inversed unmixing matrix at each frequency bin in order not to have an ambiguity of scaling as in (107). In the filter bank methods, however, the ambiguity of scaling can be avoided by normalizing filters by the corresponding scaling factors used as the direct filters or fixing the direct filters to specific scales in each subband. Other procedures follow the algorithm for frequency domain methods in similar manner [111]. In addition, it is reported that using null beamformers as the initial value of an unmixing system relaxes the permutation problem [10].

13. Comparison of Three Methods

If the length of unmixing filter is very long, time domain methods have a large computational load to compute convolution of long filters and amounts to update filter coefficients. In addition, they show slow convergence speed, especially for colored input signals such as speech signals.

The computational load can be reduced by frequency domain methods in which multiplication at each frequency bin replaces convolution operation in the time domain. Since adaptation of an unmixing matrix does not interfere with others, the frequency domain methods can improve convergence. However, a long frame size is required to cover long mixing filters. To maintain computational efficiency and obtain data which are not much overlapped with those from adjacent frames, the frame shift has to increase as the frame size increases. Therefore, the number of data at each frequency bin decreases. Since this causes insufficiency of data to learn the unmixing matrices, the performance will be degraded [12]. In addition to the performance of ICA algorithms at each frequency bin, the permutation and scaling problem has to be settled to obtain desired outputs because the unmixing matrix is adapted by ICA algorithms which have permutation and scaling indeterminacy.

Filter bank methods do not have performance limitation unlike frequency domain methods since ICA algorithms in each subband are based on time domain methods. In addition, computational complexity is considerably reduced for long adaptive filter length because a simplified ICA network can be used to process decimated input signals at the subsampled rate in each subband. Filter bank methods are also appropriate for parallel processing because each subband can independently compute subband output signals and adapt the filter coefficients of the unmixing network without other subbands. Additionally, methods are able to choose the number of subbands regardless of complexity of mixing environments, and they improve convergence of the adaptive filter coefficients because they use subband input signals which are much more whitened by decimation than time domain methods.

If a mixing environment is complex, frequency domain methods require a great many frequency bins and also very large frame shift. Thus, the envelope of each frequency spectrum can not be estimated exactly to fix permutation. However, the number of subbands in filter bank methods is usually much smaller than the required number of frequency bins in frequency domain methods. Therefore, one can resolve the permutation problem much easily because each subband can have a sufficiently broad band to exactly estimate the envelope.

For example, we have compared the three methods through simulations on blind separation of speech mixtures. Two real-recorded speech data were used as the source signals. Each signal had 5 second length at $16kHz$ sampling rate. For each category, we have chosen the infomax algorithm to learn adaptive parameters because of its popularity and simplicity. It is known that speech signal approximately follows Laplacian distribution. Therefore, $\mathrm{sgn}(\cdot)$ was used as the score function. We have mixed the two speech data with 4 room impulse responses from 2 speakers to 2 microphones which had been measured in a normal office room as shown in Figure 21.

Experimental results were compared in terms of signal-to-interference ratio (SIR). For a 2×2 mixing/unmixing

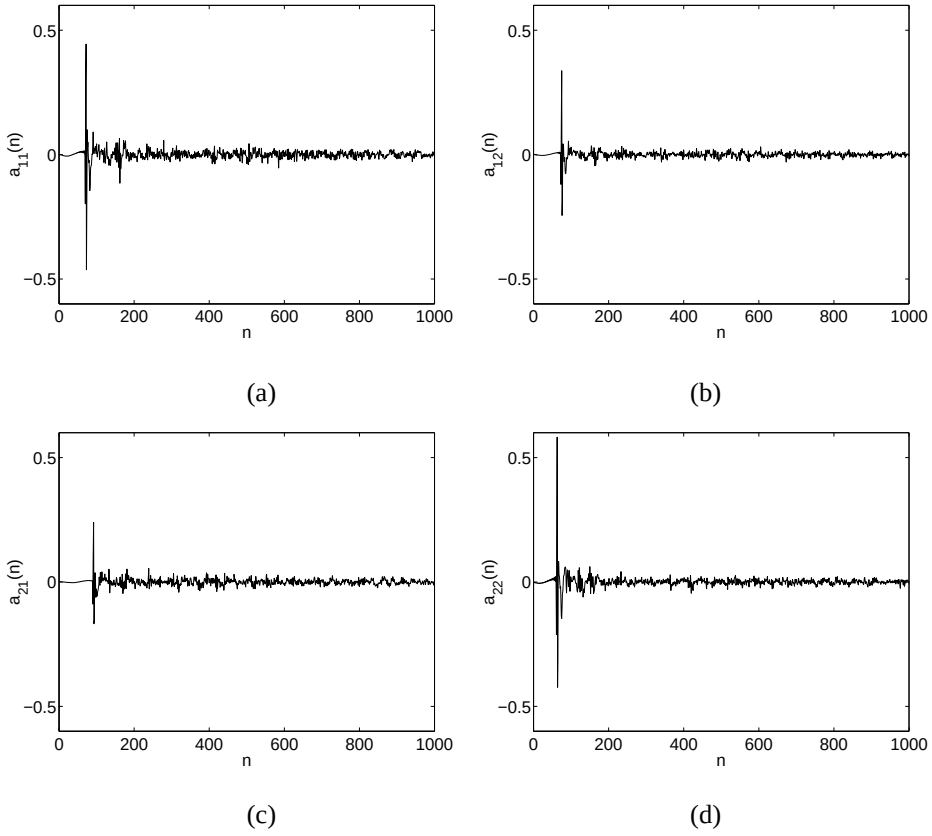


Figure 21. Room impulse reponses of the mixing system

system, the SIR is defined as a ratio of the signal power to the interference power at the outputs,

$$\mathrm{SIR}(dB) = \frac{1}{2} \cdot \left| 10 \log \left(\frac{\langle (y_{1,s_1}(k))^2 \rangle}{\langle (y_{1,s_2}(k))^2 \rangle} \cdot \frac{\langle (y_{2,s_2}(k))^2 \rangle}{\langle (y_{2,s_1}(k))^2 \rangle} \right) \right|. \tag{119}$$

In (119), $y_{j,s_i}(k)$ denotes the jth output of the cascaded mixing/unmixing system only when $s_i(k)$ is active.

In Figure 22, we have displayed learning curves of the three methods for the blind source separation problem. To perform the time domain method, we have used a feedback network, where each filter length was 2048 taps. In the frequency domain method, the frame size was 2048 samples, and the frame shift was a sixteenth of the frame size. In addition, we have designed a filter bank in order to perform the filter bank method. Figure 23 shows frequency response of analysis filters of a uniform oversampled filter bank using GDFT. The filter bank was designed for alias-free decimation by factor 10, and it was constructed from a prototype filter with 220 taps. For the separation network in each subband, we have used a feedback network in which the number of taps of each filter was 205.

SIRs of the frequency domain method were much smaller than those of the other two methods. It is because the frequency domain method has a performance limitation which comes from the contradiction between the long reverberation covering and the insufficient learning data. Moreover, the permutation problem severely degrades the performances. The learning curves in Figure 22 show that the filter bank method had much faster convergence speed than the time domain method since less colored signals by decimation were used in each subband for the filter bank approach. Contrary to the frequency domain approach, the permutation problem was successfully fixed in the filter bank approach.

14. Remarks on Underdetermined Problem

Some papers tackled the underdetermined case where the number of independent components are larger

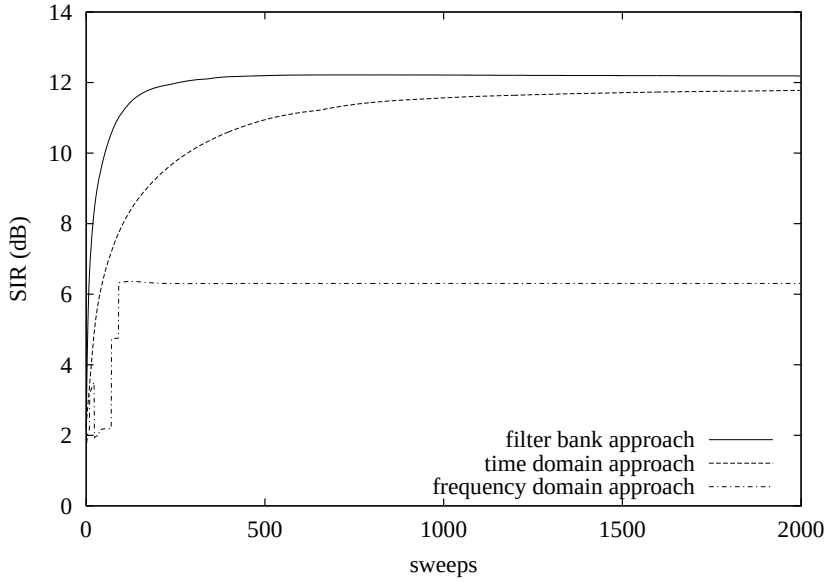


Figure 22. Learning curves of the three methods to blind source separation

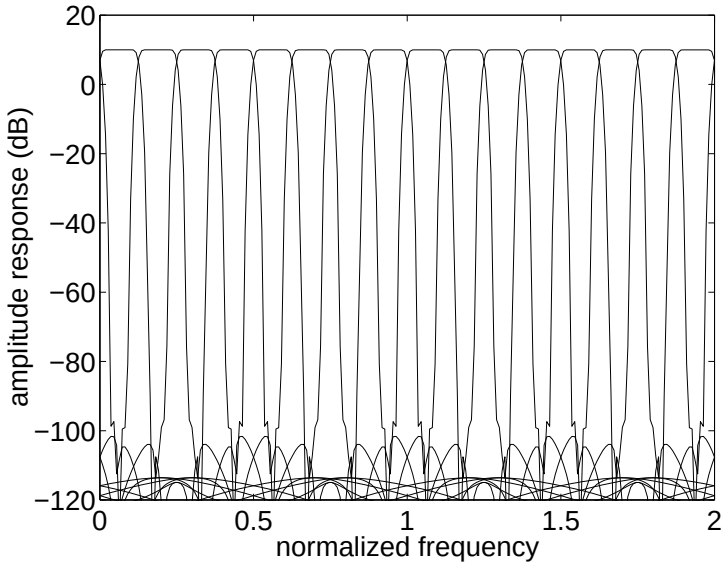


Figure 23. Frequency response of analysis filters of a uniform oversampled filter bank

than that of mixtures [22, 148, 11]. The problem is very challenging, and most of the papers modelled mixtures as delayed ones with attenuation.

In order to get more sparse signals, they employ linear transforms such as the short-time Fourier transform (STFT), and they assume that only one independent component is dominant at any given point in the time-frequency domain [148, 15, 124]. By obtaining 2 observations, the mixtures can be expressed as

$$x_1(k) = \sum_{j=1}^{n} s_j(k), \tag{120}$$

$$x_2(k) = \sum_{j=1}^{n} a_j s_j(k - d_j). \tag{121}$$

If only the jth source is nonzero for a given (f, k) in the time-frequency domain,

$$\begin{bmatrix} \boldsymbol{X}_1(f,k) \\ \boldsymbol{X}_2(f,k) \end{bmatrix} = \begin{bmatrix} 1 \\ a_j \exp(-j2\pi f d_j) \end{bmatrix} \boldsymbol{S}_j(f,k). \tag{122}$$

Therefore, the mixing parameters can be estimated by

$$\hat{a}(f,k) = \left| \frac{\boldsymbol{X}_2(f,k)}{\boldsymbol{X}_1(f,k)} \right|, \tag{123}$$

$$\hat{d}(f,k) = \frac{1}{2\pi f} \angle \left(\frac{\boldsymbol{X}_1(f,k)}{\boldsymbol{X}_2(f,k)} \right). \tag{124}$$

A two-dimensional histogram of amplitude-delay estimates can be used to determine the number of independent components and the mixing parameters. An independent component can be estimated by applying the corresponding time-frequency mask to the mixture [148, 124].

In the case where the number of dominant independent components is equal to the number of observations at any given point, conventional ICA methods can be used to estimate desired independent components [11]. In addition, making additional assumptions on the statistical properties of the independent components or maximizing the likelihood of a noisy mixing model can provide estimation of independent components or model parameters [22, 15, 125].

15. Discussion and Conclusions

In this paper we have discussed briefly several extensions and modifications of blind source separation and decomposition algorithms for spatio-temporal decorrelation, independent component analysis, sparse component analysis and non-negative matrix factorization where various criteria and constraints are imposed such linear predictability, smoothness, mutual independence, sparsity and non-negativity of extracted components. Especially, we described generalization and extension of ICA to SD-ICA which relaxes considerably the condition on independence of original sources. Using these concepts in many cases, we are able to reconstruct (recover) the original brain sources and to estimate mixing and separating matrices, even if the original sources are not independent and in fact they are strongly correlated. Moreover, we propose a simple method for checking validity and true performance of BSS separation by applying the bank of filers with various frequency characteristics.

We have also reviewed algorithms for ICA of convolved mixtures. The methods can be divided into three categories: time domain methods, frequency domain methods, and filter bank methods. We have gone through well-known algorithms for each category. In addition, we compared advantages and disadvantages among algorithms from the three categories.

Acknowledgment: S. Choi, H.-M. Park, and S.-Y. Lee were supported by Korean Ministry of Science and Technology as the Brain Neuroinformatics Research Program.

References

[1] F. Abrard and Y. Deville. Blind source separation in convolutive mixtures: a hybrid approach for colored sources. In *Proc. Int. Work-Conf. Artificial and Natural Neural Networks*, pages 802–809, June 2001.

[2] J. -H. Ahn, S. Choi, and J. -H. Oh. A multiplicative up-propagation algorithhm. In *Proc. Int. Conf. Machine Learning*, pages 17–24, Banff, Canada, 2004.

[3] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.

[4] S. Amari and A. Cichocki. Adaptive blind signal processing - neural network approaches. *Proceedings IEEE*, 86:1186–1187, 1998.

[5] S. Amari, A. Cichocki, and H.H. Yang. A new learning algorithm for blind signal separation. In Michakel C. Mozer David S. Touretzky and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems 1995*, volume 8, pages 757–763. MIT Press: Cambridge, MA, 1996.

[6] S. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang. Multichannel blind deconvolution and equalization using the natural gradient. In *IEEE International Workshop on Wireless Communication*, pages 101–104, 1997.

[7] S. Amari, A. Hyvärinen, S.-Y. Lee, T.-W. Lee, and V.D. Sanchez. Blind signal separation and independent component analysis. *Neurocomputing*, 49(12):1–5, 2002.

[8] S. Amari and J.-F.Cardoso. Blind source separation — semi-parametric statistical approach. *IEEE Trans. on Signal Processing*, 45(11):2692–2700, Dec. 1997.

[9] J. Anemüller and B. Kollmeier. Amplitude modulation decorrelation for convolutive blind source separation. In *Proc. Int. Conf. ICA and BSS*, pages 215–220, June 2000.

[10] S. Araki, S. Makino, R. Aichner, T. Nishikawa, and H. Saruwatari. Subband based blind source separation with appropriate processing for each frequency band. In *Proc. Int. Conf. ICA and BSS*, pages 499–504, April 2003.

[11] S. Araki, S. Makino, A. Blin, R. Mukai, and H. Sawada. Underdetermined blind separation for speech in real environments with sparseness and ICA. In *Proc. IEEE ICASSP*, pages 881–884, 2004.

[12] S. Araki, S. Makino, R. Mukai, T. Nishikawa, and H. Saruwatari. Fundamental limitation of frequency domain blind source separation for convolved mixtures of speech. In *Proc. Int. Conf. ICA and BSS*, pages 132–137, December 2001.

[13] F. Bach and M. Jordan. Beyond independent components: trees and clusters. *Journal of Machine Learning Research*, 4:1205–1233, 2003.

[14] U.-M. Bae, H.-M. Park, and S.-Y. Lee. Blind signal separation and independent component analysis. *Neurocomputing*, 49(12):315–327, 2002.

[15] R. Balan, J. Rosca, and S. Rickard. Scalable non-square blind source separation in the presence of noise. In *Proc. IEEE ICASSP*, pages 293–296, 2003.

[16] A. K. Barros and A. Cichocki. Extraction of specific signals with temporal structure. *Neural Computation*, 13(9):1995–2000, September 2001.

[17] A.J. Bell and T.J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7, no. 6:1129–1159, Nov 1995.

[18] A. Belouchrani. *Séparation Autodidacte de Sources: Algorithme, Performances et Application a des Signaux Expérimentaux*. PhD thesis, ENST, Telecom Paris, July 1995.

[19] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and É. Moulines. A blind source separation technique using second-order statistics. *IEEE Trans. Signal Processing*, 45(2):434–444, February 1997.

[20] A. Belouchrani and M.G. Amin. A new approach for blind source separation using time-frequency distributions. *Proc. SPIE*, 2846:193–203, 1996.

[21] A. Belouchrani and A. Cichocki. Robust whitening procedure in blind source separation context. *Electronics Letters*, 36(24):2050–2051, Nov. 2000.

[22] P. Bofill. Underdetermined blind separation of delayed sound sources in the frequency domain. *Neurocomputing*, 55:627–641, May 2002.

[23] J. -F. Cardoso. Jacobi angles for simultaneous diagonalization. *SIAM J. Mat. Anal. Appl.*, 17(1):161–164, 1996.

[24] J.-F. Cardoso. Infomax and maximum likelihood for blind source separation. *IEEE Signal Processing Letters*, 4(4):112–114, 1997.

[25] J.-F. Cardoso and B. Laheld. Equivariant adaptive source separation. *IEEE Trans. Signal Processing*, 44(12):3017–3030, 1996.

[26] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non-gaussian signals. *IEE Proc. F*, 140(6):360–370, 1993.

[27] J.-F. Cardoso and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM Journal Mat. Anal. Appl.*, 17(1):161–164, January 1996.

[28] C. Chang, Z. Ding, S. F. Yau, and F. H. Y. Chan. A matrix-pencil approach to blind separation of colored nonstationary signals. *IEEE Trans. Signal Processing*, 48(3):900–907, Mar. 2000.

[29] K.-S. Cho and S.-Y. Lee. Implementation of infomax ica algorithm with analog cmos circuits. In *Proc. International Conference on Independent Component Analysis and Blind Signal Separation*, pages 70 – 73. Vancuber, Canada, 2001.

[30] S. Choi, S. Amari, and A. Cichocki. Natural gradient learning for spatio-temporal decorrelation: recurrent network. *IEICE Trans. Fundamentals*, E83-A(12):2175–2722, 2000.

[31] S. Choi and A. Cichocki. An unsupervised hybrid network for blind separation of independent non-gaussian source signals in multipath environment. *Journal of communications and Networks*, 1(1):19–25, 1999.

[32] S. Choi and A. Cichocki. Blind separation of nonstationary and temporally correlated sources from noisy mixtures. In *Proc. IEEE Workshop on Neural Networks for Signal Processing*, pages 405–414, Sidney, Austrailia, 2000.

[33] S. Choi and A. Cichocki. Blind separation of nonstationary sources in noisy mixtures. *Electronics Letters*, 36(9):848–849, Apr. 2000.

[34] S. Choi, A. Cichocki, and S. Amari. Blind equalization of SIMO channels via spatio-temporal anti-Hebbian learning rule. In *Proc. of the 1998 IEEE Workshop on NNSP Cambridge*, pages 93–102, UK, 1998. IEEE Press, N.Y.

[35] S. Choi, A. Cichocki, and S. Amari. Flexible independent component analysis. In *Proc. of the 1998 IEEE Workshop on NNSP*, pages 83–92, Cambridge, UK, 1998.

[36] S. Choi, A. Cichocki, and S. Amari. Equivariant nonstationary source separation. *Neural Networks*, 15:121–130, 2002.

[37] S. Choi, A. Cichocki, and A. Belouchrani. Second order nonstationary source separation. *Journal of VLSI Signal Processing*, 32(1–2):93–104, August 2002.

[38] S. Choi, A. Cichocki, L. L. Zhang, and S. Amari. Approximate maximum likelihood source separation using the natural gradient. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E86-A(1):198–205, January 2003.

[39] A. Cichocki. Blind signal processing methods for analyzing multichannel brain signals. *International Journal of Bioelectromagtism*, 6(1), 2004.

[40] A. Cichocki and S. Amari. *Adaptive Blind Signal And Image Processing*. John Wiley, New York, 2003. New revised and improved edition.

[41] A. Cichocki, S. M. Amari, K. Siwek, T. Tanaka, and et al. ICALAB toolboxes for signal and image processing $www.bsp.brain.riken.go.jp$. JAPAN, 2004.

[42] A. Cichocki and A. Belouchrani. Sources separation of temporally correlated sources from noisy data using bank of band-pass filters. In *Third International Conference on Independent Component Analysis and Signal Separation (ICA-2001)*, pages 173–178, San Diego, USA, Dec. 9-13 2001.

[43] A. Cichocki, R.E. Bogner, L. Moszczyński, and K. Pope. Modified Hérault-Jutten algorithms for blind separation of sources. *Digital Signal Processing*, 7 No.2:80 – 93, April 1997.

[44] A. Cichocki and P. Georgiev. Blind source separation algorithms with matrix constraints. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E86-A(1):522–531, January 2003.

[45] A. Cichocki, R. R. Gharieb, and T. Hoya. Efficient extraction of evoked potentials by combination of Wiener filtering and subspace methods. In *Proceedings of 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-2001)*, volume 5, pages 3117–3120, Salt Lake City, Utah, USA, May 2001. IEEE, IEEE.

[46] A. Cichocki, Y. Li, P. G. Georgiev, and S. Amari. Beyond ICA: Robust sparse signal representations. In *Proceedings of 2004 IEEE International Symposium on Circuits and Systems (ISCAS2004)*, volume V, pages 684–687, Vancouver, Canada, May 2004.

[47] A. Cichocki, T. M. Rutkowski, and K. Siwek. Blind signal extraction of signals with specified frequency band. In *Neural Networks for Signal Processing XII: Proceedings of the 2002 IEEE Signal Processing Society Workshop*, pages 515–524, Martigny, Switzerland, September 2002. IEEE.

[48] A. Cichocki, S. Shishkin, T. Musha, Z. Leonowicz, T. Asada, and T. Kurachi. EEG filtering based on blind source separation improves detection of Alzheimer disease. *Clinical Neurphysiology*, 2005.

[49] A. Cichocki and R. Thawonmas. On-line algorithm for blind signal extraction of arbitrarily distributed, but temporally correlated sources using second order statistics. *Neural Processing Letters*, 12(1):91–98, August 2000.

[50] A. Cichocki, R. Thawonmas, and S. Amari. Sequential blind signal extraction in order specified by stochastic properties. *Electronics Letters*, 33(1):64–65, January 1997.

[51] A. Cichocki and R. Unbehauen. *Neural Networks for Optimization and Signal Processing*. John Wiley & Sons, New York, 1994. new revised and improved edition.

[52] A. Cichocki and R. Unbehauen. Robust neural networks with on-line learning for blind identification and blind separation of sources. *IEEE Trans. Circuits and Systems I : Fundamentals Theory and Applications*, 43(11):894–906, Nov. 1996.

[53] A. Cichocki, R. Unbehauen, and E. Rummert. Robust learning algorithm for blind separation of signals. *Electronics Letters*, 30(17):1386–1387, August 1994.

[54] A. Cichocki and S. Vorobyov. Application of ICA for automatic noise and interference cancellation in multisensory biomedical signals. In *Proceedings of the Second International Workshop on ICA and BSS, ICA'2000*, pages 621–626, Helsinki, Finland, 19-22 June 2000.

[55] R. E. Crochiere and L. R. Rabiner. *Multirate Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1983.

[56] S. Cruces, A. Cichocki, and L. Castedo. An iterative inversion approach to blind source separation. *IEEE Trans. on Neural Networks*, 11(6):1423–1437, 2000.

[57] S. A. Cruces, L. Castedo, and A. Cichocki. Robust blind source separation algorithms using cumulants. *Neurocomputing*, 49:87–118, December 2002.

[58] S. A. Cruces and A. Cichocki. Combining blind source extraction with joint approximate diagonalization: Thin algorithms for ICA. In *Proceedings of 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 463–468, Kyoto, Japan, April 2003. Riken, ICA.

[59] S. A. Cruces, A. Cichocki, and L. De Lathauwer. Thin QR and SVD factorizations for simultaneous blind signal extraction. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 217–220, Vienna, Austria, 2004. (ISBN: 3-200-00165-8).

[60] A. Delorme and S. Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics. *J. Neuroscience Methods, 134:9-21, 2004*, 134:9–21, 2004.

[61] D. L. Donoho and M. Elad. Representation via l1 minimization. *The Proc. National Academy of Science*, 100:2197–2202, March 2004.

[62] F. Ehlers and H. G. Schuster. Blind separation for convolutive mixtures and an application in automatic speech recongition in a noisy environment. *IEEE Trans. Signal Processing*, 45(10):2608–2612, October 1997.

[63] E. Ferrara. Fast implementation of LMS adaptive filters. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 28(4):474–478, 1980.

[64] S. Fiori. A fully multiplicative orthoghonal-group ICA neural algorithm. *Electronics Letters*, 39(24):1737–1738, 2003.

[65] P. G. Georgiev and A. Cichocki. Sparse component analysis of overcomplete mixtures by improved basis pursuit method. In *Proceedings of 2004 IEEE International Symposium on Circuits and Systems (ISCAS2004)*, volume V, pages 37–40, Vancouver, Canada, May 2004.

[66] P. G. Georgiev, F. J. Theis, and A. Cichocki. Blind source separation and sparse component analysis of overcomplete mixtures. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP2004)*, volume V, pages 493–496, Montreal, Canada, May 2004. IEEE Signal Processing Society.

[67] R. R. Gharieb and A. Cichocki. Noise reduction in brain evoked potentials based on third-order correlations. *IEEE Transactions on Biomedical Engineering*, 48:501–512, 2001.

[68] R. R. Gharieb and A. Cichocki. Second-order statistics based blind source separation using a bank of subband filters. *Digital Signal Processing*, 13:252–274, 2003.

[69] A. Gilloire and M. Vetterli. Adaptive filtering in subbands with critical sampling: analysis, experiments, and application to acoustic echo cancellation. *IEEE Trans. Signal Processing*, 40(8):1862–1875, August 1992.

[70] M. Girolami and C. Fyfe. Generalised independent component analysis through unsupervised learning with emergent bussgang properties. In *Proc. IEEE ICNN*, volume 3, pages 1788–1791, June 1997.

[71] G. H. Golub and C. F. Van Loan. *Matrix Computations, 2nd edition*. Johns Hopkins, 1993.

[72] M. Harteneck, S. Weiss, and R. W. Stewart. Design of near perfect reconstruction oversampled filter banks for subband adaptive filters. *IEEE Trans. Circuits Syst. II*, 46:1081–1085, August 1999.

[73] S. Haykin, editor. *Adaptive Filter Theory*. Prentice Hall, New Jersey, NJ, 1996.

[74] S. Haykin. *(Ed.) Unsupervised Adaptive Filtering Volume 2: Blind Deconvolution*, volume 2. John Wiley & Sons, February 2000.

[75] J. Himberg, A. Hyvärinen, and F. Esposito. Validating the independent components of neuroimaging time series via clustering and visualization. *NeuroImage*, 22(3):1214–1222, 2004.

[76] P.O. Hoyer and A. Hyvärinen. Independent component analysis applied to feature extraction from color and stereo images. *Network: Computation in Neural Systems*, 11(3):191–210, 2000.

[77] J. Huang, K.-C. Yen, and Y. Zhao. Subband-based adaptive decorrelation filtering for co-channel speech separation. *IEEE Trans. Speech and Audio Processing*, 8(4):402–406, July 2000.

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

[79] M. Z. Ikram and D. R. Morgan. A beamforming approach to permutation alignment for multichannel frequency-domain blind speech separation. In *Proc. IEEE ICASSP*, pages 881–884, May 2002.

[80] O. Jahn, A. Cichocki, A. Ioannides, and S. Amari. Identification and elimination of artifacts from MEG signals using efficient independent components analysis. In *Proc. of th 11th Int. Conference on Biomagentism BIOMAG-98*, pages 224–227, Sendai, Japan, 1999.

[81] H.-B. Jeon, J.-H. Lee, and S.-Y. Lee. On the center-frequency ordered speech feature extraction based on independent component analysis. In *Proceedings of International Conference on Neural Information Processing*, pages 1199 – 1203. Shanghai, China, 2001.

[82] H.-Y. Jung and S.-Y. Lee. On the temporal decorrelation of feature parameters for noise-robust speech recognition. *IEEE Transactions on Speech and Audio Processing*, 8(7):407–416, 2000.

[83] T.P. Jung, C. Humphries, T.-W. Lee, S. Makeig, M. McKeown, V. Iragui, and T. Sejnowski. ICA removes artifacts from electroencephalographic recordings. *Advances in Neural Information Processing Systems (NIPS)*, 10, 1997.

[84] T.P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M. McKeown, V. Iragui, and T.J. Sejnowski. Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, 37:167–178, 2000.

[85] C.M. Kim, H.M. Park, T. Kim, S.Y. Lee, and Y.K. Choi. FPGA implementation of ICA algorithm for blind signal separation and active noise canceling. *IEEE Transactions on Neural Networks*, 14(5):1038 – 1046, 2003.

[86] S. Kim and S. Choi. Independent arrays or independent time courese for gene expression data. In *Proc. IEEE Int'l Symp. Circuits and Systems*, Kobe, Japan, 2005. to appear.

[87] T. Kim and S.-Y. Lee. Learning self-organized topology-preserving complex speech features at primary auditory cortex. *Neurocomputing*, 2005. in press.

[88] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15(2):349–396, February 2003.

[89] R. H. Lambert. *Multichannel Blind Deconvolution: FIR Matrix Algebra and Separation of Multipath Mixtures*. PhD thesis, University of Southern California, Los Angeles, May 1996.

[90] D. D. Lee and H. S. Seung. Learning of the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[91] D. S. Lee, J. S. Lee, and J. Y. Ahn. Application of independent component analysis to dynamic $H_2^{15}O$ positron emission tomography. In *Prof. ICONIP*, pages 1221–1226, 2000.

[92] J.-H. Lee, T.-W. Lee, H.-Y. Jung, and S.-Y. Lee. On the efficient speech feature extraction based on independent component analysis. *Neural Processing Letters*, 15:235–245, 2002.

[93] J. S. Lee, D. D. Lee, S. Choi, and D. S. Lee. Application of non-negative matrix factorization to dynamic positron emission tomography. In *Proc. ICA*, pages 629–632, San Diego, California, 2001.

[94] J. S. Lee, D. D. Lee, S. Choi, K. S. Park, and D. S. Lee. Nonnegative matrix factorization of dynamic images in nuclear medicine. In *IEEE Medical Imaging Conference*, 2001.

[95] S.-Y. Lee. Auditory pathway model and its VLSI implementation for robust speech recognition in real-world noisy environment, , 2003, 5. International Joint Conference on Neural Networks and Signal Processing, Nanjing, China, 2003.

[96] T.-W. Lee. *Independent Component Analysis*. Kluwer Academic Publishers, Boston, MA, 1998.

[97] T.-W. Lee, A. J. Bell, and R. H. Lambert. Blind separation of delayed and convolved sources. In *Advances in Neural Information Processing Systems 9*, pages 758–764, Cambridge, MA, 1997. MIT Press.

[98] T-W. Lee, M. Girolami, A.J. Bell, and T.J. Sejnowski. A unifying information-theoretic framework for independent component analysis. *Computers & Mathematics with Applications*, 31(11):1–21, March 2000.

[99] T.-W. Lee, A. Ziehe, R. Orglmeister, and T. J. Sejnowski. Combining time-delayed decorrelation and ICA: Towards solving the cocktail party problem. In *Proc. IEEE ICASSP*, volume 2, pages 1249–1252, Seattle, WA, 1998.

[100] M. S. Lewicki. Efficient coding of natural sounds. *Nature Neuroscience*, 2002.

[101] Y. Li, A. Cichocki, and S. Amari. Analysis of sparse representation and blind source separation. *Neural Computation*, 16(6):1193–1204, June 2004.

[102] Y. Li, A. Cichocki, S. Amari, S. Shishkin, J. Cao, and F. Gu. Sparse representation and its applications in blind source separation. In *Seventeenth Annual Conference on Neural Information Processing Systems (NIPS-2003)*, Vancouver, December 2003.

[103] W. Liebermeister. Linear modes of gene expression determined by independent component analysis. *Bioinformatics*, 18(1):51–60, 2002.

[104] F. Mainecke, A. Ziehe, M. Kawanabe, and K.-R. Müller. A resampling approach to estimate the stability of one dimensional or multidimensional independent components. *NeuroImage*, 49(13):1514–1525, 2002.

[105] S. Makeig, S. Debener, J. Onton, and A. Delorme. Mining event-related brain dynamics. *Trends in Cognitive Science*, 8:204–210, 2004.

[106] S. Makeig, A. Delorme, M. Westerfield M., J. Townsend, E. Courchense, and T. Sejnowski. Electroencephalographic brain dynamics following visual targets requiring manual responses. *PLOS Biology*, page (in press), 2004.

[107] K. Matsuoka, M. Ohya, and M. Kawamoto. A neural net for blind separation of nonstationary signals. *Neural Networks*, 8(3):411–419, 1995.

[108] F. Miwakeichi, E. Martnez-Montes, P. A. Valds-Sosa, N. Nishiyama, H. Mizuhara, and Y. Yamaguchi. Decomposing EEG data into space-time-frequency components using Parallel Factor Analysis. *NeuroImage*, 22(3):1035–1045, 2004.

[109] L. Molgedey and H.G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72(23):3634–3637, 1994.

[110] K. R. Müller, P. Philips, and A. Ziehe. JADE$_{TD}$: Combining higher-order statistics and temporal information for blind source separation (with noise). In *Proc. ICA'99*, pages 87–92, Aussois, France, 1999.

[111] N. Murata, S. Ikeda, and A. Ziehe. An approach to blind source separation based on temporal structure of speech signals. *Neurocomputing*, 41(1/4):1–24, 2001.

[112] T. Nishikawa, H. Saruwatari, and K. Shikano. Blind source separation based on multi-stage ICA combining frequency-domain ICA and time-domain ICA. In *Proc. IEEE ICASSP*, volume 1, pages 917–920, May 2002.

[113] T. Nishikawa, H. Saruwatari, K. Shikano, S. Araki, and S. Makino. Multistage ICA for blind source separation of real acoustic convolutive mixture. In *Proc. Int. Conf. ICA and BSS*, pages 523–528, April 2003.

[114] A. Oppenheim and R. Schafer, editors. *Discrete-Time Signal Processing*. Prentice Hall, New Jersey, NJ, 1989.

[115] H.-M. Park, H.-Y. Jung, T.-W. Lee, and S.-Y. Lee. Subband-based blind signal separation for noisy speech recognition. *Electronics Letters*, 35(23):2011–2012, 1999.

[116] H.-M. Park, S.-H. Oh, and S.-Y. Lee. An oversampled filter bank approach to independent component analysis for convolved mixtures. In *Proc. Joint Int. Conf. Artificial Neural Networks and Neural Information Processing*, pages 354–357, June 2003.

[117] H.M. Park, S.H. Oh, and S.Y. Lee. A filter bank approach to independent component analysis and its application to adaptive noise cancelling. *Neurocomputing*, 55(3-4):755–759, 2003.

[118] L. Parra, K.R. Mueller, C. Spence, A. Ziehe, and P. Sajda. Unmixing hyperspectral data. In *In Advances in Neural Information Processing Systems, (NIPS2000)*, pages 942–948. Morgan Kaufmann, 2000.

[119] L. Parra and C. Spence. Convolutive blind separation of non-stationary sources. *IEEE Trans. Speech and Audio Processing*, 8(3):320–327, May 2000.

[120] B. Pearlmutter and L. Parra. Maximum likelihood blind source separation: a context-sensitive generalization of ICA. In *Advances in Neural Information Processing Systems 9*, pages 613–619, Cambridge, MA, 1997. MIT Press.

[121] D. T. Pham. Joint approximate diagonalization of positive definite hermitian matrices. Technical Report LMC/IMAG, University of Grenoble, France, 2000.

[122] D. T. Pham and J. -F. Cardoso. Blind separation of instantaneous mixtures of nonstationary sources. In *Proc. ICA*, pages 187–192, Helsinki, Finland, 2000.

[123] M. D. Plumbley. Algorithms for nonnegative independent component anlaysis. *IEEE Trans. Neural Networks*, 14(3):534–543, May 2003.

[124] S. Rickard, R. Balan, and J. Rosca. Real-time time-frequency based blind source separation. In *Proc. Int. Conf. ICA and BSS*, pages 651–656, December 2001.

[125] J. Rosca, C. Borss, and R. Balan. Generalized sparse signal mixing model and application to noisy blind source separation. In *Proc. IEEE ICASSP*, pages 877–880, 2004.

[126] P. Sajda, S. Du, and L. Parra. Recovery of constituent spectra using non-negative matrix factorization. In *Proceedings of SPIE – Volume 5207)*, pages 321–331. Wavelets: Applications in Signal and Image Processing, 2003.

[127] H. Saruwatari, S. Kurita, and K. Takeda. Blind source separation combining frequency-domain ICA and beamforming. In *Proc. IEEE ICASSP*, pages 2733–2736, 2001.

[128] H. Sawada, R. Mukai, S. Araki, and S. Makino. A polar-coordinate based activation function for frequency domain blind source separation. In *Proc. Int. Conf. ICA and BSS*, pages 663–668, December 2001.

[129] H. Sawada, R. Mukai, S. Araki, and S. Makino. A robust and precise method for solving the permutation problem of frequency-domain blind source separation. *IEEE Trans. Speech and Audio Processing*, 12:530–538, September 2004.

[130] P. Smaragdis. Information theoretic approaches to source separation. Master's thesis, MIT Media Arts and Sci. Dept., MIT, Boston, June 1997.

[131] P. Smaragdis. Blind separation of convolved mixtures in the frequency domain. *Neurocomputing*, 22:21–34, November 1998.

[132] A. Souloumiac. Blind source detection and separation using second order non-stationarity. In *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, pages 1912–1915, 1995.

[133] J. V. Stone, J. Porrill, N. R. Porter, and I. W. Wilkinson. Spatiotemporal independent component analysis of event-related fmri data using skewed probability density functions. *NeuroImage*, 15(2):407–421, 2002.

[134] J.V. Stone. Blind source separation using temporal predictability. *Neural Computation*, 13(7):1559–1574, 2001.

[135] T. Tanaka and A. Cichocki. Subband decomposition independent component analysis and new performance criteria. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP2004)*, volume V, pages 541–544, Montreal, Canada, May 2004.

[136] F. J. Theis, P. G. Georgiev, and A. Cichocki. Robust overcomplete matrix recovery for sparse sources using a generalized Hough transform. In *Proceedings of 12th European Symposium on Artificial Neural Networks (ESANN2004)*, pages 223–232, Bruges, Belgium, April 2004.

[137] L. Tong, Y. Inouye, and R. Liu. A finite-step global convergence algorithm for the parameter estimation of multichannel MA processes. *IEEE Trans. Signal Processing*, 40(10):2547–2558, Oct. 1992.

[138] L. Tong, R.-W. Liu, V.-C. Soon, and Y.-F. Huang. Indeterminacy and identifiability of blind identification. *IEEE Trans. on Circuits and Systems*, 38(5):499–509, May 1991.

[139] K. Torkkola. Blind separation of convolved sources based on information maximization. In *Proc. IEEE Int. Workshop on NNSP*, 1996.

[140] J. K. Tugnait. Identification and deconvolution of multichannel linear non-gaussian processes using higher order statistics and inverse filter criteria. *IEEE Trans. Signal Processing*, 45:658–672, March 1997.

[141] S. Vorobyov and A. Cichocki. Blind noise reduction for multisensory signals using ICA and subspace filtering, with application to EEG analysis. *Biological Cybernetics*, 86(4):293–303, April 2002.

[142] H. Wee and J. Principe. A criterion for BSS based on simultaneous diagonalization of time correlation matrices. In *Proc. IEEE Int. Workshop on NNSP*, pages 496–508, 1997.

[143] S. Weiss. *On Adaptive Filtering on Oversampled Subbands*. PhD thesis, Signal Processing Division, Univ. Strathclyde, Glasgow, May 1998.

[144] S. Weiss, L. Lampe, and R. W. Stewart. Efficient implementations of complex and real valued filter banks for comparative subband processing with an application to adaptive filtering. In *Proc. 1st Int. Symp. Commun. Systems and Digital Signal Processing*, pages 32–35, April 1998.

[145] S. Weiss, A. Stenger, R. W. Stewart, and R. Rabenstein. Steady-state performance limitations of subband adaptive filters. *IEEE Trans. Signal Processing*, 49(9):1982–1991, September 2001.

[146] H.-C. Wu and J. C. Principe. Simultaneous diagonalization in the frequency domain (SDIF) for source separation. In *Proc. Int. Conf. ICA and BSS*, pages 245–250, January 1999.

[147] Y. Yamada, H. Ochi, and H. Kiya. A subband adaptive filter allowing maximally decimation. *IEEE J. Select. Areas Commun.*, 12(9):1548–1552, September 1994.

[148] Ö. Yilmaz and S. Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Trans. Signal Processing*, 52:1830–1847, July 2004.

[149] L. Zhang, A. Cichocki, and S. Amari. Multichannel blind deconvolution of nonminimum-phase systems using filter decomposition. *IEEE Transactions on Signal Processing*, 52(5):1430–1442, 2004.

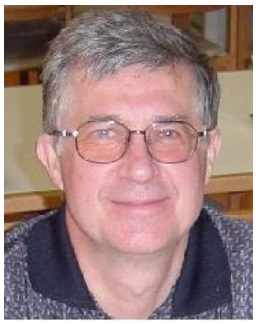
[150] L. Zhang, A. Cichocki, and S. Amari. Self-adaptive blind source separation based on activation functions adaptation. *IEEE Transactions on Neural Networks*, 15(2):233–244, 2004.

[151] M. Zibulevsky, P. Kisilev, Y.Y. Zeevi, and B.A. Pearlmutter. Blind source separation via multinode sparse representation. In *In Advances in Neural Information Processing Systems, (NIPS2001)*, pages 185–191. Morgan Kaufmann, 2002.

[152] A. Ziehe, M. Kawanabe, S. Harmeling, and K.-R. Müller. A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *Journal of Machine Learning Research*, 5:801–818, July 2004.

[153] A. Ziehe, K.-R.M¨uller, G. Nolte, B.-M. Mackert, and G. Curio. Artifact reduction in biomagnetic recordings based on time-delayed second order correlations. *IEEE Trans. on Biomedical Engineering*, 47:75–87, 2000.

Seungjin Choi was born in Seoul, Korea, in 1964. He received the B.S. and M.S. degrees in Electrical Engineering from Seoul National University, Korea, in 1987 and 1989, respectively and the Ph.D degree in Electrical Engineering from University of Notre Dame, Indiana, in 1996. He was a Visiting Assistant Professor in Department of Electrical Engineering at University of Notre Dame, Indiana during the Fall semester of 1996. He was with the Laboratory for Artificial Brain Systems, RIKEN, Japan in 1997 and was Assistant Professor in the School of Electrical and Electronics Engineering, Chungbuk National University from 1997 to 2000. Since 2001, he is Associate Professor of Computer Science at Pohang University of Science and Technology (POSTECH), Korea. He has also been an invited senior researcher at Laboratory for Advanced Brain Signal Processing, Brain Science Institute, RIKEN, Japan since 1998. His primary research interests include statistical machine learning, probabilistic graphical models, Bayesian learning, representational learning, and computational biology, as well as independent component analysis. Dr. Choi was IEEE Machine Learning for Signal Processing (MLSP) TC member and currently is IEEE Blind Signal Processing TC member.



Andrzej Cichocki was born in Poland. He received the M.Sc. (with honors), Ph.D., and Habilitate Doctorate (Dr.Sc.) degrees, all in electrical engineering, from Warsaw University of Technology (Poland) in 1972, 1975, and 1982, respectively. Since 1972, he has been with the Institute of Theory of Electrical Engineering and Electrical Measurements at the Warsaw University of Technology, where he became a full Professor in 1991. He is the co-author of three books: Adaptive Blind Signal and Image Processing, (John Wiley 2003 with Professor Shun-ichi Amari), MOS Switched-Capacitor and Continuous-Time Integrated Circuits and Systems (Springer, 1989) and Neural Networks for Optimization and Signal Processing (J. Wiley and Teubner Verlag, 1993/94, both with Professor Rolf Unbehauen) and co-author of more than one 150 papers. Two of his books have been translated to Chinese and other languages. He spent at University Erlangen-Nuernberg (Germany) a few years as Alexander Humboldt Research Fellow and Guest Professor working in the area VLSI of electronic circuits, artificial neural networks and optimization. He conducted and realized several successful research projects. In 1996-99 he has been working as a Team Leader of the Laboratory for Artificial Brain Systems, at the Frontier Research Program RIKEN (Japan), in the Brain Information Processing Group and since 1999 he is head of the laboratory for Advanced Brain Signal Processing in the Brain Science Institute RIKEN, Japan. He is also member of several international Scientific Committees and the associated Editor of IEEE Transaction on Neural Networks (since January 1998). His current research interests include biomedical signal and image processing (especially blind signal/image processing), neural networks and their applications, learning theory and robust algorithms, generalization and extensions of independent and principal component analysis, optimization problems and nonlinear circuits and systems theory and their applications.

Hyung-Min Park received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1997, 1999, and 2003, respectively. He is currently working in the Department of Biosystems, Korea Advanced Institute of Science and Technology, as a Post-Doc. His current research interests include the theory and applications of independent component analysis, blind signal separation, adaptive noise canceling, and noise-robust speech recognition.

Soo-Young Lee had graduated with BS from Seoul National University in 1975, MS from Korea Advanced Institute of Science and Technology in 1977, and Ph.D. in Electrophysics from Polytechnic Institute of New York in 1984. He joined the Korea Advanced Institute of Science and Technology as an Assistant Professor in 1986, and now is a Professor at the Department of BioSystems and also the Department of Electrical Engineering & Computer Science. Since 1998 Dr. Lee has been serving as the Director of the Brain Science Research Center, which is the main research organization of the Korean Brain Neuroinformatics Research Program. His current research interests include modeling, applications, and chip implementations of human auditory pathways for robust speech feature extraction, binaural processing, and top-down attention. Recently his research interests have been extended to the artificial brain with human-like perception and self-development capability.