ELSEVIER

Neurocomputing 24 (1999) 55–93

NEUROCOMPUTING

Neural networks for blind separation with unknown number of sources

Andrzej Cichocki[a,c,*], Juha Karhunen[b], Wlodzimierz Kasprzak[a,d], Ricardo Vigário[b]

[a] *Laboratory for Open Information Systems, Brain Science Institute Riken, 2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan*

[b] *Laboratory of Computer and Information Science, Helsinki University of Technology, P.O. Box 2200, FIN-02150 Espoo, Finland*

[c] *Department of Electrical Engineering, Warsaw University of Technology, Pl. Politechniki 1, PL-00-661 Warsaw, Poland*

[d] *Institute of Control and Computation Engineering, Warsaw University of Technology, Nowowiejska 15/19, PL-00-665 Warsaw, Poland*

Received 8 December 1996; accepted 2 September 1998

Abstract

Blind source separation problems have recently drawn a lot of attention in unsupervised neural learning. In the current approaches, the number of sources is typically assumed to be known in advance, but this does not usually hold in practical applications. In this paper, various neural network architectures and associated adaptive learning algorithms are discussed for handling the cases where the number of sources is unknown. These techniques include estimation of the number of sources, redundancy removal among the outputs of the networks, and extraction of the sources one at a time. Validity and performance of the described approaches are demonstrated by extensive computer simulations for natural image and magnetoencephalographic (MEG) data.

Keywords: Blind separation; Image processing; Neural networks; Unsupervised learning; Signal reconstruction

* Correspondence address: Laboratory for Open Information Systems, Brain Science Institute Riken, 2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan. Tel.: (+ 81) 48-467-9668; fax: (+ 81) 48-467-9686; e-mail: cia@brain.riken.go.jp

0925-2312/99/$ – see front matter
PII: S0925-2312(98)00091-5

Notation

$\boldsymbol{R}_{xx} = E\{\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\}$	covariance matrix of signal vector $\boldsymbol{x}(t)$
$\boldsymbol{u}_i$	the ith principal eigenvector of the matrix $\boldsymbol{R}_{xx}$
λ_i	the ith eigenvalue of the matrix $\boldsymbol{R}_{xx}$
$\hat{\kappa}_4$	the normalized kurtosis of a source signal
$\eta(t)$	learning rate
m	number of sources
n	number of sensors
$l \in [1, \ldots, n]$	number of outputs
$\boldsymbol{s}(t)$	m-dimensional vector of source signals
$\boldsymbol{x}(t)$	n-dimensional vector of mixed signals
$\boldsymbol{y}(t)$	n- or l-dimensional vector of separated (output) signals
$\boldsymbol{z}(t)$	n-dimensional vector of output signals after redundancy elimination
$\boldsymbol{n}(t)$	n-dimensional vector of noise signals
$\boldsymbol{v}(t)$	m- or l-dimensional vector of pre-whitened signals
$\boldsymbol{A}(t) = [a_{ij}]_{n \times m}$	(unknown) full rank mixing matrix
$\boldsymbol{V}(t) = [v_{ij}]_{l \times n}$	pre-whitening matrix
$\boldsymbol{W}(t) = [w_{ij}]_{l \times n}$	global de-mixing matrix
$\hat{\boldsymbol{W}}(t) = [\hat{w}_{ij}]_{l \times l}$	source separation matrix after pre-whitening
$\boldsymbol{W}^{(k)}(t)$	de-mixing matrix of the kth layer
$\boldsymbol{P}(t)$	generalized permutation matrix
$\boldsymbol{J}(\boldsymbol{y}, \boldsymbol{W})$	cost (risk) function

1. Introduction

In *blind source separation* (BSS), the goal is to extract statistically independent but otherwise unknown source signals from their linear mixtures without knowing the mixing coefficients [1–54]. This kind of blind techniques have applications in several areas, such as data communications, speech processing, and various biomedical signal processing problems (MEG/EEG data); see for example [34,46].

The study of BSS began about 10 years ago mainly in the area of statistical signal processing, even though the related single channel *blind deconvolution* problem has been studied already earlier. Quite recently, BSS has become a highly popular research topic in unsupervised neural learning. Neural network researchers have approached the BSS problem from different starting points, such as information theory [1,4,5] and nonlinear generalizations of Hebbian/anti-Hebbian learning rules [15–17,27,30,32,36,43]. Despite of recent advances in neural BSS, there still exist several open questions and possible extensions of the basic mixing model that have received only limited attention thus far [32].

Although many neural learning algorithms have been proposed for the BSS problem, in their corresponding models and network architectures it is usually assumed that the number of source signals is known a priori. Typically it should be

equal to the number of sensors and outputs. However, in practice, these assumptions do not often hold. The main objective of this paper is to study the behavior of various network structures for a BSS problem, where the number of sources is different from the number of outputs and where the number of sources is in general unknown. We shall propose several alternative solutions to these problems.

The paper is organized as follows. In Section 2 we first define the general BSS problem, and then briefly consider special but important cases that may appear in BSS problems. In Section 3 we discuss two alternative source separation approaches for solving the BSS problem. The first approach uses pre-whitening, while the second approach tries to separate and to determine the source number directly from the input data. The theoretical basics of proposed learning rules are given in an appendix. Computer simulation results are given in Section 4, and the last Section 5 contains discussion of the achieved results and some conclusions.

2. Problem formulation

2.1. The general blind source separation problem

Assume that there exist m zero mean source signals $s_1(t), \ldots, s_m(t)$ that are scalar-valued and mutually (spatially) statistically independent (or as independent as possible) at each time instant or index value t. The original sources $s_i(t)$ are unknown to the observer, which has to deal with n possibly noisy but different linear mixtures $x_1(t), \ldots, x_n(t)$ of the sources (usually for $n \geq m$). The mixing coefficients are some unknown constants. The task of blind source separation is to find the waveforms $\{s_i(t)\}$ of the sources, knowing only the mixtures $x_j(t)$ and usually the number m of sources.

Denote by $\boldsymbol{x}(t) = [x_1(t), \ldots, x_n(t)]^T$ the n-dimensional tth data vector made up of the mixtures at discrete index value (usually time) t. The BSS mixing (data) model can then be written in the vector form

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{n}(t) = \sum_{i=1}^{m} s_i(t)\boldsymbol{a}_i + \boldsymbol{n}(t). \tag{1}$$

Here $\boldsymbol{s}(t) = [s_1(t), \ldots, s_m(t)]^T$ is the source vector consisting of the m source signals at the index value t. Furthermore, each source signal $s_i(t)$ is assumed to be a stationary zero mean stochastic process. $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m]$ is a constant full-rank $n \times m$ mixing matrix whose elements are the unknown coefficients of the mixtures (for $n \geq m$). The vectors $\boldsymbol{a}_i$ are basis vectors of *independent component analysis* (ICA) [19,20,32].

Besides the above general case, we also discuss the noise-free simplified mixing model, where the additive noise $\boldsymbol{n}(t)$ is negligible so that it can be omitted from the considerations.

We assume further that in the general case the noise signal has a Gaussian distribution but none of the sources is Gaussian. In the simplified case at most one of the source signals $s_i(t)$ is allowed to have a Gaussian distribution. These assumptions

follow from the fact that it is impossible to separate several Gaussian sources from each other [6,48].

In standard neural and adaptive source separation approaches, an $m \times n$ separating matrix $\boldsymbol{W}(t)$ is updated so that the m-vector

$$\boldsymbol{y}(t) = \boldsymbol{W}(t)\boldsymbol{x}(t) \tag{2}$$

becomes an estimate $\boldsymbol{y}(t) = \hat{\boldsymbol{s}}(t)$ of the original independent source signals. Fig. 1 shows a schematic diagram of the mixing and source separation system. In neural realizations, $\boldsymbol{y}(t)$ is the output vector of the network and the matrix $\boldsymbol{W}(t)$ is the total weight matrix between the input and output layers. The estimate $\hat{s}_i(t)$ of the ith source signal may appear in any component $y_j(t)$ of $\boldsymbol{y}(t)$. The amplitudes of the source signals $s_i(t)$ and their estimates $y_j(t)$ are typically scaled so that they have unit variance. This ambiguity can be expressed mathematically as

$$\boldsymbol{y}(t) = \hat{\boldsymbol{s}}(t) = \boldsymbol{WAs}(t) = \boldsymbol{PDs}(t), \tag{3}$$

where $\boldsymbol{P}$ is a permutation matrix and $\boldsymbol{D}$ is a nonsingular scaling matrix.

With a neural realization in mind, it is desirable to choose the learning algorithms so that they are as simple as possible but yet provide sufficient performance. Many different neural separating algorithms have been proposed recently [2–6,9–18, 24,26–38,41–46,52,54]. Their performance usually strongly depends on stochastic properties of the source signals. These properties can be determined from higher-order statistics (cumulants) of the sources. Especially useful is a fourth-order cumulant called *kurtosis*. For the ith source signal $s_i(t)$, the *normalized kurtosis* is defined by

$$\hat{\kappa}_4[s_i(t)] = \frac{E\{s_i(t)^4\}}{E^2\{s_i(t)^2\}} - 3. \tag{4}$$

If $s_i(t)$ is Gaussian, its kurtosis $\hat{\kappa}_4[s_i(t)] = 0$. Source signals that have a negative kurtosis are often called sub-Gaussian ones. Typically, their probability distribution is "flatter" than the Gaussian distribution. Respectively, super-Gaussian sources (with a positive kurtosis) have usually a distribution which has a longer tail and sharper peak when compared with the Gaussian distribution.

If the sign of the kurtosis (4) is the same for all the sources $s_i(t)$, $i = 1, \ldots, m$, and the input vectors are pre-whitened, one can use a particularly simple separating criterion,

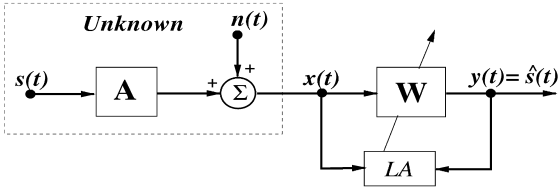


Fig. 1. Illustration of the mixing model and neural network for blind separation. LA means learning algorithm.

This is the sum of the fourth moments of the outputs [36,44]

$$J(\boldsymbol{y}) = \sum_{i=1}^{m} E\{y_i(t)^4\}, \tag{5}$$

usually subject to one of the constraints

$$E\{y_i^2\} = 1, \quad \forall i; \ \|\boldsymbol{w}_i\| = 1, \quad \forall i; \tag{6}$$

$$\text{or} \quad w_{ii} = 1, \ \forall i; \quad \text{or} \quad E\{f(y_i)\boldsymbol{x}(t) - \boldsymbol{f}(\boldsymbol{w})\|\boldsymbol{w}\|\} = 0. \tag{7}$$

Here we have assumed that the number l of outputs equals to the number m of the sources. A separating matrix $\boldsymbol{W}$ minimizes Eq. (5) for sub-Gaussian sources, and maximizes it for super-Gaussian sources [43]. The choice of nonlinear functions in neural separating algorithms depends on the sign of the normalized kurtoses of the sources. This is discussed briefly later on in this paper.

2.2. Separation with estimation of the number of sources

A standard assumption in BSS is that the number m of the sources should be *known* in advance. Like in most neural BSS approaches, we have assumed up to now that the number m of the sources and outputs l are *equal* in the separating network. Generally, both these assumptions may not hold in practice. In this paper we shall propose two different approaches for neural blind separation with simultaneous determination of the source number m. The only additional requirement in these approaches is that the number of available mixtures is greater than or equal to the true number of the sources, that is, $n \geq m$.

For completeness of our considerations, let us first briefly discuss the difficult case where there are less mixtures than sources: $n < m$. Then the $n \times m$ mixing matrix $\boldsymbol{A}$ in Eq. (1) has more columns than rows. In this case, complete separation is usually out of question. This is easy to understand by considering the much simpler situation where the mixing matrix $\boldsymbol{A}$ is *known* (recall that in BSS this does not hold), and there is no noise. Even then the set of linear equations (1) has an infinite number of solutions because there are more unknowns than equations, and the source vector $\boldsymbol{s}(t)$ cannot be determined for arbitrary distributed sources.

However, some kind of separation may still be achievable in special instances at least. This topic has recently been studied theoretically in [6]. The authors show that it is possible to separate the m sources into n disjoint groups if and only if $\boldsymbol{A}$ has n linearly independent column vectors, and the remaining $m - n$ column vectors satisfy the special condition that each of them is parallel to one of these n column vectors.

Before proceeding, we point out that it is not always necessary or even desirable in BSS problems to separate all the sources contained in the mixtures. This holds for example in situations where the number of sensors is large and only a few most powerful sources are of interest. In particular, Hyvärinen and Oja [27,46] have recently developed separating algorithms which estimate one source at a time. However, the sources are extracted in somewhat arbitrary order depending on the initial values, etc., though the first separated sources are usually among the most

powerful ones. Instead of neural gradient rules which converge somewhat slowly and may not be applicable to high-dimensional problems, one can use semi-neural fixed-point algorithms [26,28,48] for separating sources. An example of extracting one source at a time from auditory evoked fields in given in Section 4.

3. Two neural network approaches to BSS

3.1. Source separation with a pre-whitening layer

Fig. 2 shows a two-layer neural network for blind source separation, where the first layer performs *pre-whitening* (*sphering*) and the second one separation of sources. The respective weight matrices are denoted by $\boldsymbol{V}$ and $\hat{\boldsymbol{W}}$. The operation of the network is described by

$$\boldsymbol{y}(t) = \hat{\boldsymbol{W}}(t)\boldsymbol{v}(t) = \hat{\boldsymbol{W}}\boldsymbol{V}\boldsymbol{x}(t) = \boldsymbol{W}(t)\boldsymbol{x}(t), \tag{8}$$

where $\boldsymbol{W} \equiv \hat{\boldsymbol{W}}\boldsymbol{V}$ is the total separating matrix.

The network of Fig. 2 is useful in context with such BSS algorithms that require whitening of the input data for good performance. In *whitening* (*sphering*), the data vectors $\boldsymbol{x}(t)$ are pre-processed using a whitening transformation

$$\boldsymbol{v}(t) = \boldsymbol{V}(t)\boldsymbol{x}(t). \tag{9}$$

Here $\boldsymbol{v}(t)$ denotes the whitened vector, and $\boldsymbol{V}(t)$ is an $m \times n$ whitening matrix.

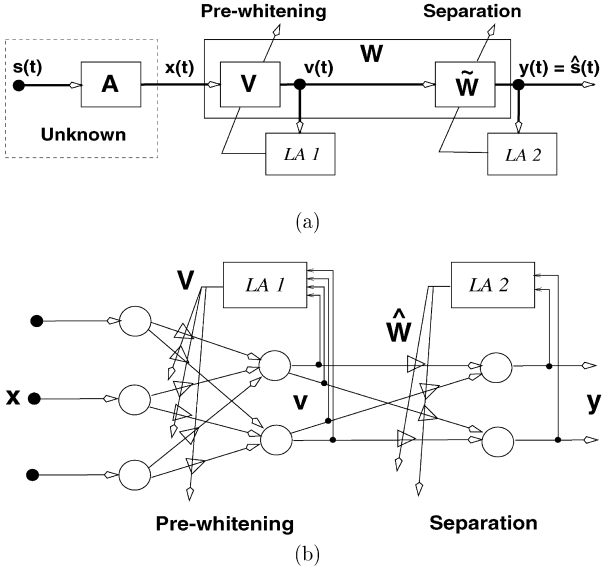


Fig. 2. The two-layer feed-forward network for pre-whitening and blind separation: (a) block diagram, (b) detailed neural network with signal reduction during pre-whitening.

If $n > m$, where m is known in advance, $\boldsymbol{V}(t)$ simultaneously reduces the dimension of the data vectors from n to m. In whitening, the matrix $\boldsymbol{V}(t)$ is chosen so that the covariance matrix $E\{\boldsymbol{v}(t)\boldsymbol{v}(t)^{\mathrm{T}}\}$ becomes the unit matrix $\boldsymbol{I}_m$. Thus, the components of the whitened vectors $\boldsymbol{v}(t)$ are mutually uncorrelated and they have unit variance. Uncorrelatedness is a necessary pre-requisite for the stronger independence condition. After pre-whitening the separation task usually becomes easier, because the subsequent separating matrix $\hat{\boldsymbol{W}}$ can be constrained to be orthogonal [36,46]:

$$\hat{\boldsymbol{W}}\hat{\boldsymbol{W}}^{\mathrm{T}} = \boldsymbol{I}_m, \tag{10}$$

where $\boldsymbol{I}_m$ is the $m \times m$ unit matrix. Whitening seems to be especially helpful in large-scale problems, where separation of sources can sometimes be impossible in practice without resorting to it.

3.1.1. Neural learning rules for pre-whitening

There exist many solutions for whitening the input data [15,32,36,47]. The simplest adaptive, on-line learning rules for pre-whitening have the following matrix forms [22]:

$$\boldsymbol{V}(t+1) = \boldsymbol{V}(t) + \eta(t)[\boldsymbol{I} - \boldsymbol{v}(t)\boldsymbol{v}^{\mathrm{T}}(t)] \tag{11}$$

or [10,47]

$$\boldsymbol{V}(t+1) = \boldsymbol{V}(t) + \eta(t)[\boldsymbol{I} - \boldsymbol{v}(t)\boldsymbol{v}^{\mathrm{T}}(t)]\boldsymbol{V}(t). \tag{12}$$

The first algorithm is a local one, in the sense that the update of every weight v_{ij} is made on the basis of two neurons i and j only. The second algorithm is a robust one with *equivariant property* [10] as the global system (in the sense that the update of every synaptic weight v_{ij} depends on outputs of all neurons), described by combined mixing and de-correlation process

$$\boldsymbol{P}(t+1) \overset{\mathrm{df}}{=} \boldsymbol{V}(t+1)\boldsymbol{A} = \boldsymbol{P}(t) + \eta(t)[\boldsymbol{I} - \boldsymbol{P}(t)\boldsymbol{s}(t)\boldsymbol{s}^{\mathrm{T}}(t)\boldsymbol{P}^{\mathrm{T}}(t)]\boldsymbol{P}(t) \tag{13}$$

is completely independent of the mixing matrix $\boldsymbol{A}$. Both these pre-whitening rules can be used in context with neural separating algorithms. The first rule (11) seems to be more reliable than Eq. (12) if a large number of iterations or tracking of mildly non-stationary sources is required. In these instances, the latter algorithm (12) may sometimes suffer from stability problems.

3.1.2. Nonlinear principal subspace learning rule for the separation layer

The *nonlinear PCA subspace rule* developed and studied by Oja, Karhunen, Xu and their collaborators (see [35,38,45]) employs the following update rule for the orthogonal separating matrix $\hat{\boldsymbol{W}}$:

$$\hat{\boldsymbol{W}}(t+1) = \hat{\boldsymbol{W}}(t) + \eta(t)\boldsymbol{g}[\boldsymbol{y}(t)][\boldsymbol{v}(t) - \hat{\boldsymbol{W}}^{\mathrm{T}}(t)\boldsymbol{g}[\boldsymbol{y}(t)]]^{\mathrm{T}}, \tag{14}$$

where $\boldsymbol{v}(t) = \boldsymbol{V}(t)\boldsymbol{x}(t)$, $\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t)$, and $\boldsymbol{y}(t) = \hat{\boldsymbol{W}}(t)\boldsymbol{v}(t)$. Here and later on, $\boldsymbol{g}[\boldsymbol{y}(t)]$ denotes the column vector whose ith component is $g_i[y_i(t)]$, where $g_i(t)$ is usually an odd and

monotonically increasing nonlinear activation function. The learning rate $\eta(t)$ must be positive for stability reasons.

A major advantage of the learning rule (14) is that it can be realized using a simple modification of one-layer standard symmetric PCA network, allowing a relatively simple neural implementation [35,36]. The separation properties of Eq. (14) have been analyzed mathematically in simple cases in [45]. In a recent paper [38] it is shown that the Nonlinear PCA rule (14) is related to several other ICA and BSS approaches and contrast functions. Efficient recursive least-squares type algorithms for minimizing the nonlinear PCA criterion in blind separation have been developed in [37,38]. They provide a clearly faster convergence than the stochastic gradient rule (14) at the expense of somewhat greater computational load.

3.2. Signal number reduction by pre-whitening

The first class of approaches for source number determination in the BSS problem is based on the natural compression ability of the pre-whitening layer. If standard Principal Component Analysis (PCA) is used for pre-whitening, one can then simultaneously compress information optimally in the mean-square error sense and filter the possible noise [15,36]. In fact the PCA whitening matrix $\boldsymbol{V}$ can be computed as

$$\boldsymbol{V} = \sqrt{\boldsymbol{R}_{xx}^{-1}} = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{U}^{\mathrm{T}}, \tag{15}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix of the eigenvalues and $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n]$ is the orthogonal matrix of the associated eigenvectors of the covariance matrix $\boldsymbol{R}_{xx} = E[\boldsymbol{x}(t)\boldsymbol{x}^{\mathrm{T}}(t)] = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}}$.

If there are more mixtures than sources ($n > m$), it is possible to use the PCA approach for estimating the number m of the sources. If m is estimated correctly and the input vectors $\boldsymbol{x}(t)$ are compressed to m-dimensional vectors $\boldsymbol{v}(t)$ in the whitening stage using the network structure in Fig. 2b, then there are usually no specific problems in the subsequent separation stage.

In practice, the source number is determined by first estimating the eigenvalues λ_i of the data covariance matrix $E\{\boldsymbol{x}(t)\boldsymbol{x}(t)^{\mathrm{T}}\}$. Let us denote these ordered eigenvalues by

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0. \tag{16}$$

In the ideal case where the noise term $\boldsymbol{n}(t)$ in Eq. (1) is zero, only the m largest "signal" eigenvalues $\lambda_1, \ldots, \lambda_m$ are nonzero, and the rest $n - m$ "noise" eigenvalues of the data covariance matrix are zero. If the powers of the sources are much larger than the power of noise, the m largest signal eigenvalues are still clearly larger than noise eigenvalues, and it is straightforward to determine m from the breakpoint. However, if some of the sources are weak or the power of the noise is not small, it is generally hard to see what is the correct number m of sources just by inspecting the eigenvalues. In [33] it is demonstrated that two well-known information-theoretic criteria, MDL and AIC, yield in practice good estimates of the number of sources for noisy mixtures on certain conditions.

We have also considered a modified network structure, where the possible data compression takes place in the separation layer instead of the pre-whitening layer. The nonlinear PCA subspace rule (14) can well be used for learning the separating matrix $\hat{\boldsymbol{W}}$, because this algorithm has originally been designed for situations where data compression takes place simultaneously with learning of the weight matrix $\hat{\boldsymbol{W}}$ [35,52]. If the number n of mixtures equals to the number m of sources, and the goal is to extract only some sources, so that the number of outputs $l < m$, this alternative structure seems to perform better. On the other hand, if $n > m$ (the number of mixtures is larger than that of sources), and $l = m$, the quality of the separation results was in our experiments slightly better when the data compression from n to m took place in the whitening stage instead of the separation stage.

Generally, this modified network structure is not recommendable if the power of the noise is not small or the number of mixtures n is larger than the number m of the sources. This is easy to understand, because in this case whitening without data compression tends to amplify the noise by making the variances of n components of the whitened vectors $\boldsymbol{v}(t)$ all equal to unity.

3.3. Source separation without pre-whitening

Whitening has some disadvantages, too. The most notable of these is that for ill-conditioned mixing matrices and weak sources the separation results may be poor. Therefore, some other neural algorithms have been developed that learn the separating matrix $\boldsymbol{W}$ directly. A single layer performs the linear transformation

$$\boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t), \tag{17}$$

where $\boldsymbol{W}$ is an $n \times n$ square nonsingular matrix of synaptic weights updated according to some on-line learning rule. In this section we discuss simple neural network models and associated adaptive learning algorithms, which do not require any pre-processing.

3.3.1. General (robust) global rule

The whitening algorithms discussed so far can be easily generalized for the blind source separation problem. For example, a general form of the learning rule (12) was proposed in [17,18], as

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \eta(t)\{\boldsymbol{I} - \boldsymbol{f}[\boldsymbol{y}(t)]\boldsymbol{g}[\boldsymbol{y}^{\mathrm{T}}(t)]\}\boldsymbol{W}(t), \tag{18}$$

which can be written in scalar form as

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)\left[w_{ij}(t) - f_i[y_i(t)] \sum_{k=1}^{m} w_{kj}(t) g_k[y_k(t)] \right], \tag{19}$$

where $\eta(t) > 0$ is the adaptive learning rate and $\boldsymbol{I}$ is the $n \times n$ identity matrix. $\boldsymbol{f}(\boldsymbol{y}) = [f(y_1), \ldots, f(y_n)]^{\mathrm{T}}$ and $\boldsymbol{g}(\boldsymbol{y}^{\mathrm{T}}) = [g(y_1), \ldots, g(y_n)]$ are vectors of nonlinear activation functions, where $f(y), g(y)$ is a pair of suitably chosen nonlinear functions.

These nonlinear functions are used in the above rule mainly for introducing higher-order statistics or cross-cumulants into computations. The rule tries to cancel these higher-order statistics, leading to at least approximate separation of sources (or independent components). The choice of the activation functions $f(y), g(y)$ depends on the statistical distribution of the source signals (this problem is discussed in the appendix).

The above rule is derived more rigorously in the appendix by using the concept of *Kullback–Leibler* divergence (or mutual information) and the natural gradient concept developed by Amari [1,4].

3.3.2. Simplified (nearly local) rule

The learning rule (18) can be simplified by applying another generalized gradient form [11, 12]:

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) \mp \eta(t) \frac{\partial J}{\partial \boldsymbol{W}} \boldsymbol{W}^{\mathrm{T}}(t). \tag{20}$$

In this case we obtain a relatively simple self-normalized local learning rule [12,16]:

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) \pm \eta(t)\{\boldsymbol{I} - \boldsymbol{f}[\boldsymbol{y}(t)]\boldsymbol{y}^{\mathrm{T}}(t)\}. \tag{21}$$

This learning rule which can be written in scalar form as $w_{ij}(t+1) = w_{ij}(t) \pm \eta_i(t)[\delta_{ij} - f_i(y_i(t))y_j(t)]$, is stable for both signs $+$ and $-$ under zero initial conditions. The local learning rule (21) can be regarded as a generalization of the local whitening rule (11). Furthermore, this is the simplest on-line learning rule for the BSS problem that to our knowledge has been proposed thus far.

3.3.3. Equivariant property

It is very interesting to observe that the learning rule (18) has a so-called *equivariant* property [3,4,10,17]. This means that its performance is independent of the scaling factors and/or mixing matrix $\boldsymbol{A}$. Therefore, the algorithm is able to extract extremely weak signals mixed with strong ones provided that there is no noise. Moreover, the condition number of mixing matrix can then be even 10^{15}, and it depends only on the precision of the calculations [17,18].

The simplified local learning rule (21) does not have the equivariant property. Hence, a single layer neural network with this learning rule may sometimes fail to separate signals, especially if the problem is ill-conditioned. However, we have discovered that by applying a multi-layer structure (feed-forward or recurrent) this algorithm is also able to solve very ill-conditioned separation problems [11–13]. In such a case we apply the same simple local learning rule (21) for each layer, as illustrated by Fig. 3. However, for each layer we can use different nonlinear functions for introducing different higher-order statistics, which usually improves the quality of separation.

3.4. Noise-free redundancy reduction

The separation algorithms (18) and (21) presented so far for the complete (determined) source case ($m = n$) can be applied in the more general (over-determined) case, when the number of sources is unknown, but not larger than the number of sensors, that is if $n \geq m$. In this case we assume that the dimension of matrix $\boldsymbol{W}(t)$ is still $n \times n$. If $n > m$ there appears a redundancy among the separated signal set, meaning that one or more signals are extracted in more than one channel. If additive noise exist in each sensor channel, then they appear on the redundant outputs. But consider the noise-free case. Then some separated signals appear in different channels with different scaling factors.

In [11] we have proposed to add a post-processing layer to the separation network for the elimination of redundant signals. Thus the applied neural network consists of two or more layers (Fig. 4), where the first sub-network (a single layer or a multi-layer) simultaneously separates the sources and the last (post-processing) layer eliminates redundant signals. The post-processing layer determines the number of active sources in the case where the number of sensors (mixtures) n is greater than the number of the primary sources m. Such a layer is described by the linear transformation $\boldsymbol{z}(t) = \tilde{\boldsymbol{W}}(t)\boldsymbol{y}(t)$, where the synaptic weights (elements of the matrix $\tilde{\boldsymbol{W}}(t)$) are updated using the following adaptive local learning algorithm:

$$\begin{aligned} &\tilde{w}_{ii}(t) = 1, \quad \forall t \ \forall i, \\ &\Delta\tilde{w}_{ij}(t) = -\eta(t) f[z_i(t)] g[z_j(t)], \quad i \neq j, \end{aligned} \tag{22}$$

where $g(z)$ is a nonlinear odd activation function (e.g. $g(z) = \tanh(\alpha z)$) and $f(z)$ is either a linear or slightly nonlinear odd function.

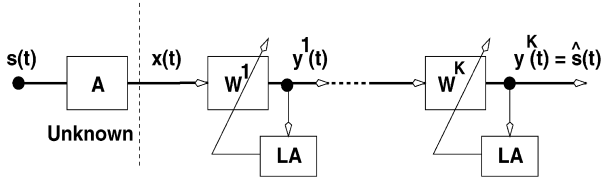


Fig. 3. A multi-layer feed-forward neural network architecture for blind source separation without pre-whitening.

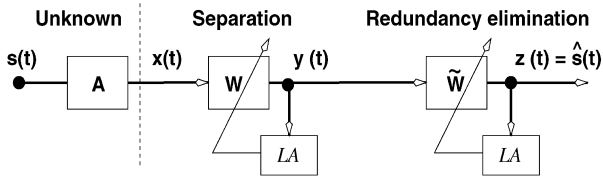


Fig. 4. The scheme of a two-layer neural network for blind separation and redundancy elimination.

Constraints imposed on the matrix $\tilde{\boldsymbol{W}}(t)$ ensure mutual de-correlation of the output signals, which eliminates the redundancy of the output signals and may also improve the separation results. However, we have found that this performance strongly depends on the statistical distributions of the sources. If the source signals are not completely independent, for example when the sources consist of two or more natural images, the post-processing layer should be used for redundancy elimination only.

The learning rules for redundancy elimination given above can be derived using the same optimization criterion which was used for source separation (Eqs. (18) and (21)), but with some constraints for the elements of the matrix $\tilde{\boldsymbol{W}}(t)$, e.g. $\tilde{w}_{ii}(t) = 1, \forall i$. It should be noted that the rule (22) is similar to the well-known Herault–Jutten rule [30], but it is now applied to a feed-forward network with different activation functions.

4. Computer simulation results

4.1. Experimental arrangements

In this section some illustrative experiments are presented using the proposed approaches, in particular the three separation algorithms summarized in Table 1. In order to estimate the quality of separation, we use in our simulations known original source signals (images) and a known mixing matrix. Of course, these quantities are unknown to the learning algorithms that are being tested. The separation results are best inspected by comparing images showing the true sources and the separated sources. This gives a good qualitative assessment of the achieved performance.

Different types of image sources are applied in a single experiment – sources with both positive or negative kurtosis and a Gaussian noise image are mixed together (compare Table 2). By scanning them, they can easily be transformed to 1-D signals (see Fig. 5). It should be noted that the stochastic characteristics of a 1-D signal corresponding to some natural image is frequently changing. Hence, in order to achieve convergence of the weights during the learning process, we apply a descending learning rate.

Table 1
Three separation algorithms considered in the paper

Pos.	Separation rule	Description
1	$\Delta \boldsymbol{W} = \eta[\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y})\boldsymbol{g}(\boldsymbol{y})^{\mathrm{T}}]\boldsymbol{W}$	Global algorithm with equivariant property
2	$\Delta \boldsymbol{W} = \pm \eta[\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y})\boldsymbol{g}(\boldsymbol{y})^{\mathrm{T}}]$	Simple local algorithm
3	$\Delta \boldsymbol{W} = \eta \boldsymbol{f}(\boldsymbol{y})[\boldsymbol{v}^{\mathrm{T}} - \boldsymbol{f}(\boldsymbol{y})^{\mathrm{T}}\boldsymbol{W}] \simeq \eta[\boldsymbol{f}(\boldsymbol{y})\boldsymbol{y}^{\mathrm{T}} - \boldsymbol{y}\boldsymbol{f}(\boldsymbol{y})^{\mathrm{T}}]\boldsymbol{W}$, $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{v}, \quad \boldsymbol{v} = \boldsymbol{V}\boldsymbol{x} = \boldsymbol{V}\boldsymbol{A}\boldsymbol{s}$	Nonlinear PCA with pre-whitening

Table 2
Statistical characteristics of normalized kurtosis $\hat{\kappa}_4$ of source images used in experiments in Section 4

In experiments in Section 4.1							
Source	Cichocki	Karhunen	Kasprzak	Vigário	Random	Sinusoid	Nature
$\hat{\kappa}_4$	− 1.110	− 1.129	− 1.011	− 0.595	− 0.602	0.410	0.171
In experiments in Section 4.2							
Source	Flowers	Model	Waterfall	Bark	Blocks	Marmor	
$\hat{\kappa}_4$	− 0.759	− 1.131	0.090	− 0.909	− 0.871	− 0.836	
In experiments in Section 4.3							
Source	Miss 1	Miss 2	Miss 3	Noise			
$\hat{\kappa}_4$	− 0.909	− 1.473	− 0.604	− 0.001			

In most experiments, natural or synthetic grey-scale images are used; their size is equal to 256×384 (Section 4.1) or 256×256 (Sections 4.2 and 4.3). The images have always 256 grey levels. Before the start of the learning procedure the image signals should be transformed to zero-mean signals, and for compatibility with the learning rate and initial weights they are also scaled to the interval $[-1.0,1.0]$. For presentation, the resulting signals are mapped back to the grey-level interval of [0, 255]. Zero signals having small amplitudes around 0.0 correspond to uniformly grey images, and are represented by a grey image with all pixel values equal to 127.

The obtained results can be assessed quantitatively by using suitable mathematical measure. Examples of such measures are: *PSNR* (peak signal-to-noise ratio) between each reconstructed source and the corresponding original source, and an error index *EI* for the whole set of separated sources. These measures are defined as follows.

1. *PSNR* (peak signal-to-noise ratio):

$$PSNR = 10 \log_{10}\left(\frac{A^2}{MSE}\right). \tag{23}$$

Here *MSE* is the mean square error of the separated source:

$$MSE = \frac{1}{N} \sum_{k=1}^{N} (\hat{s}_{jk} - s_{jk})^2, \tag{24}$$

and $A = s_{\max} - s_{\min}$ is the amplitude peak of the source signal.

The *PSNR* factor is computed for each pair consisting of an output signal and a source. For a given signal the highest *PSNR* value determines the best corresponding source.

2. For the whole set of separated sources, one can calculate an average error index *EI*, which is defined by

$$EI = \frac{1}{m}\left[\sum_{i=1}^{n}\left(\sum_{j=1}^{m}\frac{|p_{ij}|^2}{\max_i |p_{ij}|^2} - 1\right)\right] + \frac{1}{n}\left[\sum_{j=1}^{m}\left(\sum_{i=1}^{n}\frac{|p_{ij}|^2}{\max_j |p_{ij}|^2} - 1\right)\right]. \tag{25}$$

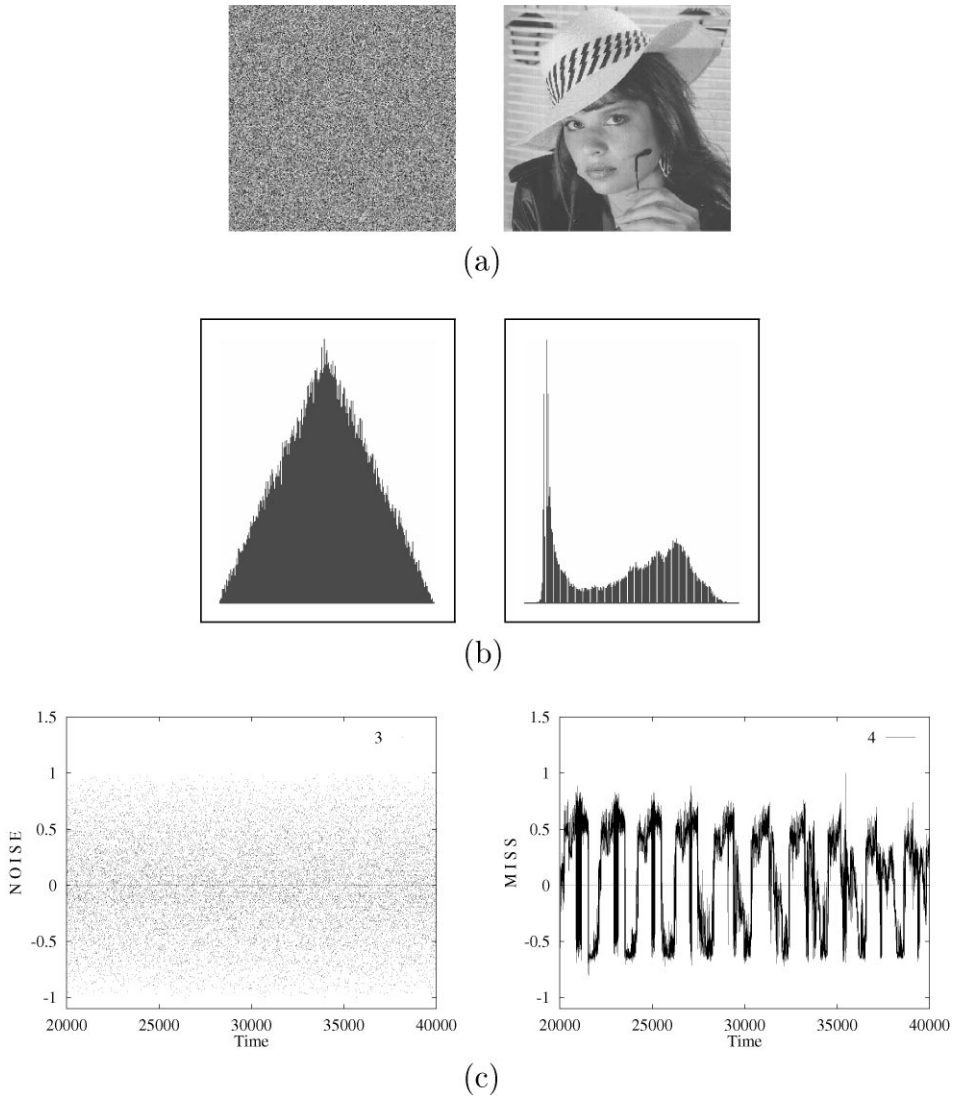


Fig. 5. Example of a noise image and a natural face image (assumed to be completely unknown to the neural net): (a) the two source images, (b) their image histograms, (c) the source signals, i.e. the source images after normalization and transformation to zero-mean 1-D signals.

The numbers p_{ij} are entries of a normalized matrix $\boldsymbol{P}(t)$, derived from $\hat{\boldsymbol{P}}(t) \in R^{n \times m}$:

$$\hat{\boldsymbol{P}} = \tilde{\boldsymbol{W}}\boldsymbol{W}^{(k)} \ldots \boldsymbol{W}^{(1)}\boldsymbol{V}\boldsymbol{A},$$

by normalizing every non-zero row $i = 1, \ldots, n$ of the matrix $\hat{\boldsymbol{P}}$ in such a way that $\max_i |p_{ij}| = 1$. The first component of EI gives the error of the output signals, averaged over the number of sources. The second part adds an additional penalty if

the same source appears multiple times in the output set. This component is averaged over the number of outputs. The row-like normalization of the matrix $\hat{\boldsymbol{P}}$ is a necessary condition for proper estimation of the penalty value. In the ideal case of perfect separation, the matrix $\boldsymbol{P}$ becomes a permutation matrix. Then only one of the elements on each row and column equals to unity, and all the other elements are zero. In this ideal case EI attains its minimum possible value zero.

The step-size $\eta(t)$ depends on the expected signal amplitude and the initial values of $\boldsymbol{W}$. We use a descending $\eta(t)$ which is the largest possible, providing a fast learning and convergence of the algorithm. Usually for signal amplitudes in the interval $[-1,1]$ and initial weight values < 1, η is below 0.1. The initial matrix $\boldsymbol{W}(0)$ is a non-zero random matrix with elements scaled to the interval $[-1,1]$.

4.2. Basic source separation

In the first experiment we tested the separation ability of the pre-whitening rule and the three separation rules, proposed in Section 3, for the following BSS problem: the mixing takes place without additive noise (although one of the source signals is itself a noise signal) and the number of sources and sensors is equal (but more than two sources are mixed). As shown in Fig. 6, seven images have been mixed by using randomly generated, ill-conditional mixing matrices $\boldsymbol{A}_{5\times 5}$ and $\boldsymbol{A}_{7\times 7}$, respectively:

$$\boldsymbol{A}_{5\times 5} = \begin{Bmatrix} 2.00 & 4.82 & 3.47 & 1.65 & 35.2 \\ 0.55 & 1.20 & 3.79 & 1.82 & 48.0 \\ 0.91 & 1.15 & 4.35 & 1.61 & 19.3 \\ 0.46 & 1.18 & 5.61 & 4.98 & 30.6 \\ 0.76 & 1.38 & 3.31 & 1.21 & 22.3 \end{Bmatrix}, \tag{26}$$

$$\boldsymbol{A}_{7\times 7} = \begin{Bmatrix} 0.560 & 0.930 & 0.300 & 0.950 & 0.750 & 2.900 & 0.380 \\ 0.520 & 0.620 & 0.150 & 0.830 & 0.410 & 3.290 & 0.180 \\ 0.915 & 0.420 & 0.680 & 0.340 & 0.900 & 3.180 & 0.700 \\ 0.510 & 0.720 & 0.410 & 0.890 & 0.910 & 3.520 & 0.110 \\ 0.700 & 0.960 & 0.340 & 0.900 & 0.920 & 2.900 & 0.740 \\ 0.410 & 0.210 & 0.150 & 0.830 & 0.210 & 3.170 & 0.550 \\ 0.930 & 0.180 & 0.660 & 0.310 & 0.230 & 2.880 & 0.260 \end{Bmatrix}. \tag{27}$$

The condition numbers of these matrices are $cond(\boldsymbol{A}_{5\times 5}) = 1359.1$,$cond(\boldsymbol{A}_{7\times 7}) = 443.0$.

The source set consists of four natural face images with negative normalized kurtosis κ_4, one noise image with negative κ_4, one synthetic image with positive κ_4 and one natural image with slightly positive κ_4 (see Table 2). For the computation of the signal moments μ_2,μ_3,κ we use the full scan of each image.

Among the compared rules are: the pre-whitening algorithm (11), the two-layer nonlinear PCA subspace rule (14), the one-layer global rule (18), and the multi-layer

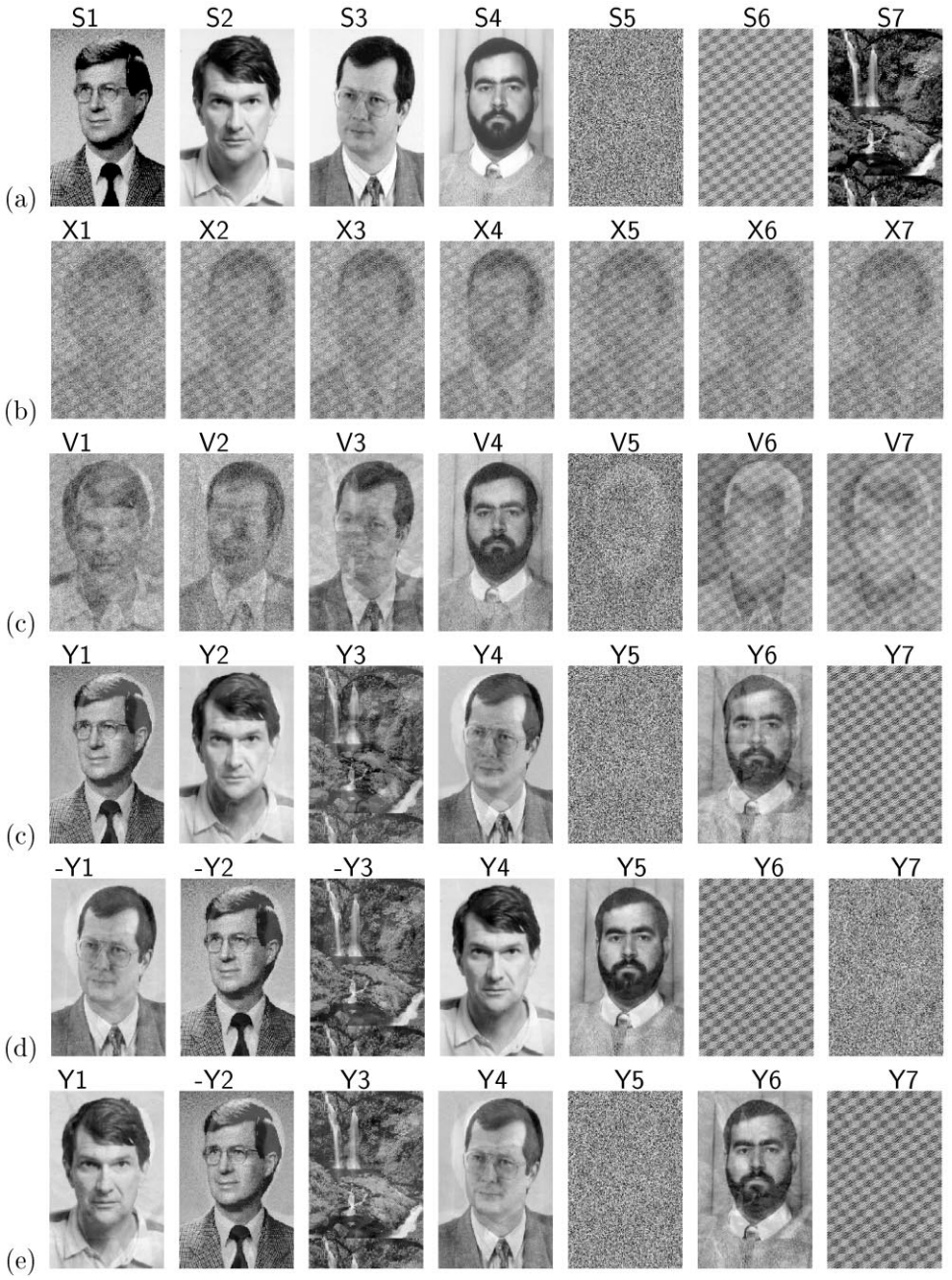


Fig. 6. Testing the separation ability for sources with different sign of the kurtosis. There are five sub-Gaussian and two super-Gaussian sources (a). They are mixed together to seven input images for separation (b). Results after: (c) pre-whitening and nonlinear PCA subspace layer, (d) one-layer global rule separation, (e) five layers of local rule separation.

local learning rule (21). The performance factors of result signals are given in Table 3. In every experiment the final weights $W_{\lim}$ are taken for computation of the quantitative results.

The source images and mixtures used in these experiments are rather demanding for separation algorithms, because for them the assumptions made on the data model (1) are actually not valid. This is because the applied face images are clearly correlated (see Table 4), with locally changing stochastic properties. Thus the source signals are not independent, but they are non-stationary also – some signal samples are more and some less correlated to each other. During the learning process we can select less correlated output samples than the overall correlation factor is, thus, concentrating the learning process on approximately half amount of signal samples.

It is remarkable that we are able to achieve sufficiently good quality separation for practical purposes, especially by the algorithms (18) and (21) which do not use pre-whitening. Generally, algorithms applying pre-whitening are not able to separate the clearly correlated face images. This follows from the fact that the pre-whitened data vectors $\boldsymbol{v}(t)$ are uncorrelated: $E\{\boldsymbol{v}(t)\boldsymbol{v}(t)^{\mathrm{T}}\} = \boldsymbol{I}$. In the separation phase, this property is preserved by requiring that also the output vectors $\boldsymbol{y}(t) = \hat{\boldsymbol{W}}(t)\boldsymbol{v}(t)$ must be uncorrelated: $E\{\boldsymbol{y}(t)\boldsymbol{y}(t)^{\mathrm{T}}\} = \boldsymbol{I}$, leading to the orthogonality condition $\hat{\boldsymbol{W}}(t)\hat{\boldsymbol{W}}(t)^{\mathrm{T}} = \boldsymbol{I}$ for the separating matrix $\hat{\boldsymbol{W}}$. It is just this strict requirement of uncorrelatedness that

Table 3
Error index and quality factors for the separation tests with different separation rules: (a) for five sub-Gaussian sources, (b) for five sub-Gaussian and two super-Gaussian sources

Signal	*EI*	PSNR [dB]						
		S1	S2	S3	S4	S5	S6	S7
(a)								
Separation with nonlinear PCA rule								
$\boldsymbol{v}$ whitening	1.5727	14.17	–	–	17.77	18.04		
$\boldsymbol{y}$ separation	0.2436	17.69	19.61	18.02	17.90	45.07		
Multi-layer separation with local rule								
$\boldsymbol{y}^{(5)}$ fifth layer	0.1900	21.23	23.37	15.26	20.65	32.56		
Separation with global rule								
$\boldsymbol{y}$ separation	0.0457	23.12	36.32	20.32	22.97	36.65		
(b)								
Separation with nonlinear PCA rule								
$\boldsymbol{v}$ whitening	2.032	13.50	–	–	21.53	18.93	16.89	14.50
$\boldsymbol{y}$ separation	0.209	19.12	21.80	17.88	16.98	38.96	44.02	19.49
Multi-layer separation with local rule								
$\boldsymbol{y}^{(5)}$ fifth layer	0.146	19.23	22.73	17.55	18.82	42.80	30.57	25.77
Separation with global rule								
$\boldsymbol{y}$ separation	0.136	20.97	27.90	17.54	21.07	39.58	21.96	23.75

makes good quality separation impossible for algorithms employing pre-whitening if the original sources are markedly correlated. The nonlinear PCA subspace rule (14) is a notable exception among learning algorithms employing pre-whitening, because it can provide a separating matrix $\hat{\boldsymbol{W}}(t)$ that is not orthogonal [32]. This is actually a benefit in the case of correlated sources, allowing the algorithm to adopt to such situations in a certain degree.

But also the output images found by the algorithms (18) and (21) are in fact more independent than the original correlated sources, in any case they are less correlated than the original sources (see Table 4). For the global algorithm (18), an obvious explanation is that it can be derived by minimizing the Kullback–Leibler divergence (see the appendix); the minimum is achieved for independent outputs. It is especially noteworthy that the same derivation holds irrespective of whether true independent components exist or not. This is because the product of distributions of individual outputs, corresponding to the situation where the outputs are truly independent, is the "target" distribution to which the "distance" of the true joint distribution is minimized. The distance or difference measure between these two distribution is the Kullback–Leibler divergence (see the appendix for details). A preliminary conclusion on these considerations is that the global rule (18) in fact tries to provide a best approximate solution to the ICA problem, in the sense of minimizing the Kullback–Leibler divergence.

On the other hand, after convergence the algorithm (18) tries to satisfy the condition

$$E\{\boldsymbol{f}[\boldsymbol{y}(t)]\boldsymbol{g}[\boldsymbol{y}^{\mathrm{T}}(t)]\} = \boldsymbol{I}. \tag{28}$$

This can be derived by setting $\boldsymbol{W}(t+1) = \boldsymbol{W}(t) = \boldsymbol{W}$ and taking expectations from the both sides of Eq. (18). Eq. (28) is a generalized decorrelating condition which does not force uncorrelated outputs, $E[\boldsymbol{y}(t)\boldsymbol{y}(t)^{\mathrm{T}}] = \boldsymbol{I}$ (unless both $\boldsymbol{f}(t)$ and $\boldsymbol{g}(t)$ are linear functions in which case separation is impossible). If for example $\boldsymbol{f}[\boldsymbol{y}(t)] = \tanh[\boldsymbol{y}(t)]$ and $\boldsymbol{g}[\boldsymbol{y}^{\mathrm{T}}(t)] = \boldsymbol{y}^{\mathrm{T}}(t)$, one can see by inserting the Taylor series expansion $\tanh(t) = t - t^3/3 + 2t^5/15 \ldots$ that Eq. (28) leads to a condition which tries to make the sum of the correlation matrix $\mathrm{E}[\boldsymbol{y}(t)\boldsymbol{y}(t)^{\mathrm{T}}]$ and higher-order moment matrices of $\boldsymbol{y}(t)$ equal to the unit matrix $\boldsymbol{I}$. Note that the same condition (28) (with $\boldsymbol{g}[\boldsymbol{y}^{\mathrm{T}}(t)] = \boldsymbol{y}^{\mathrm{T}}(t)$) is valid for the local algorithm (21), too, explaining why it also is able to roughly separate moderately correlated sources.

Even though the algorithms discussed here provide in practice good separation for correlated sources, they do not separate the original correlated sources perfectly. This can be seen both by inspecting the correlation values in Table 4 and the Fig. 6. Some errors appear especially around the heads in face images. If the sources are even more correlated than in Fig. 6, the errors become more pronounced. The basic reason seems to be that for correlated source signals, solutions to the BSS and best approximate ICA problems are in fact different.

The choice of the activation function (nonlinearity) depends on the sign of *normalized kurtosis* (4) of the source signals. It has recently been shown [9,10] that this is sufficient for successful separation, though the knowledge of the probability densities of the sources would help to achieve a better accuracy (see the appendix). In blind separation, these densities are usually unknown.

Table 4
Correlation values ($\times 100$) between pairs of sources, i.e. $E\{s_i s_j\}$, and between pairs of signals after mixing, pre-whitening and separation in the experiments in Fig. 6

Signals	Signal-pair correlations ($\times 100$)									
Sources										
	S1-2	S1-3	S1-4	S1-5	S2-3	S2-4	S2-5	S3-4	S3-5	S4-5
$\boldsymbol{s}$	21.3	44.9	32.7	1.27	35.5	21.5	0.75	41.5	0.39	0.84
Sensor (mixed) signals										
	X1-2	X1-3	X1-4	X1-5	X2-3	X2-4	X2-5	X3-4	X3-5	X4-5
$\boldsymbol{x}$	98.6	98.8	98.4	99.6	97.3	97.5	99.2	99.6	99.4	99.2
Separated (output) signals										
	y1-2	y1-3	y1-4	y1-5	y2-3	y2-4	y2-5	y3-4	y3-5	y4-5
After pre-whitening and nonlinear PCA separation										
$\boldsymbol{y}$	−0.26	10.0	−0.28	1.04	−1.60	−1.68	−1.78	−2.84	−9.593	−9.36
After local rule separation										
$\boldsymbol{y}^{(5)}$	−7.84	5.37	−2.52	6.47	2.99	3.99	−0.93	−2.70	−3.17	−3.88
After global rule separation										
$\boldsymbol{y}$	10.2	14.3	13.0	−0.61	12.1	10.2	0.25	19.7	−0.98	−0.24

If the source signals are expected to have negative kurtosis values, that is, they are *sub-Gaussian* signals, we choose in the global algorithm (18)

$$f(y_j) = y_j^3 \quad \text{and} \quad g(y_j) = y_j, \tag{29}$$

or

$$f(y_j) = y_j^3 \quad \text{and} \quad g(y_j) = \tanh(\alpha y_j). \tag{30}$$

On the other hand, for *super-Gaussian* sources with positive kurtosis, we choose

$$f(y_j) = \tanh(\alpha y_j) \quad \text{and} \quad g(y_j) = y_j, \tag{31}$$

or

$$f(y_j) = \tanh(\alpha y_j) \quad \text{and} \quad g(y_j) = y_j^3, \tag{32}$$

for obtaining successful separation (see the appendix).

When the source signals have both positive and negative kurtosis values, a combination of above functions can be applied. If the signs of the kurtosises are unknown, one can estimate them adaptively fairly easily in context with all separation algorithms, see for example [25]. The global rule may separate the sources on different outputs than it is expected by the kind of applied activation function.

In the case of local rule, the first choices of the nonlinearities given above are applied for both types of sources. For the nonlinear PCA rule (14), $g_i[y_i] = \tanh(\alpha y_i)$

for sub-Gaussian sources, and $g_i[y_i] = y_i + \tanh(\alpha y_i)$ (or $g_i(y_i) = y_i^3$) for super-Gaussian sources [36,38].

4.3. Data compression in the pre-whitening stage

An example of using the two-layer NN of Fig. 2b for BSS with source number estimation is shown in Fig. 7. There are six source images shown on the first row: S1–S3 are natural scenes and S4–S6 are textures. All the sources were sub-Gaussian except S3 which had a small positive kurtosis value (compare Table 2). The images labeled X1–X8 show eight mixtures formed of these sources, and V1–V6 are the six pre-whitened images. The separated sources Y1–Y6 are shown on the bottom row – they are very close to the original. In this experiment, the nonlinear PCA subspace rule was used for learning the orthogonal separating matrix $\hat{\boldsymbol{W}}$. In this noiseless case, the correct number of sources is obtained directly as a by-product of the PCA-based whitening, and it equals to the number of nonzero eigenvalues of the data covariance matrix.

However, in practical situations it may happen that we estimate m incorrectly. In another experiment the same source images S1–S6 as in Fig. 7 were used, the number of different mixtures was $n = m = 6$, but the number of pre-whitened signals p was smaller than the number of sources, $p < m$. Conceptually, in the separation layer this situation corresponds to the difficult case, where there are less mixtures than sources. Again the nonlinear PCA subspace rule (14) was applied in the separation layer. In the case of five outputs the outputs were still fairly close to original sources, but one of them was already missing. When the underestimation of m becomes more severe ($p = 4$, 3, or 2), the outputs were usually some mixtures of the source images S1–S6, and it seems also that some of the sources were lost almost completely.

For comparison, Fig. 8 shows the results for the same source images S1–S6 as before, but when the data compression takes place in the separating layer instead of the pre-whitening layer. In the simulation of Fig. 8, the number of mixtures n was equal to the number of sources $m = 6$. It can be seen that the network always yielded output signals, which were very close to some original sources. However, the particular sources separated in this experiment depend on the chosen initial values, mixing matrix, and learning parameters.

In spite of these fairly good results, the basic structure (Fig. 2b) where the data compression takes place already in the pre-whitening layer, yields better results if the number of mixtures $n > m$, and $l = m$ sources are separated.

4.4. Separation and redundancy reduction

We present here two experiments for the second class of architectures which do not use pre-whitening. In these networks, either the global single-layer rule (18) or the local multi-layer rule (21) are used for over-determined separation. Moreover, in the noise-free case an additional post-processing layer, using the learning rule (22), is applied for redundancy reduction.

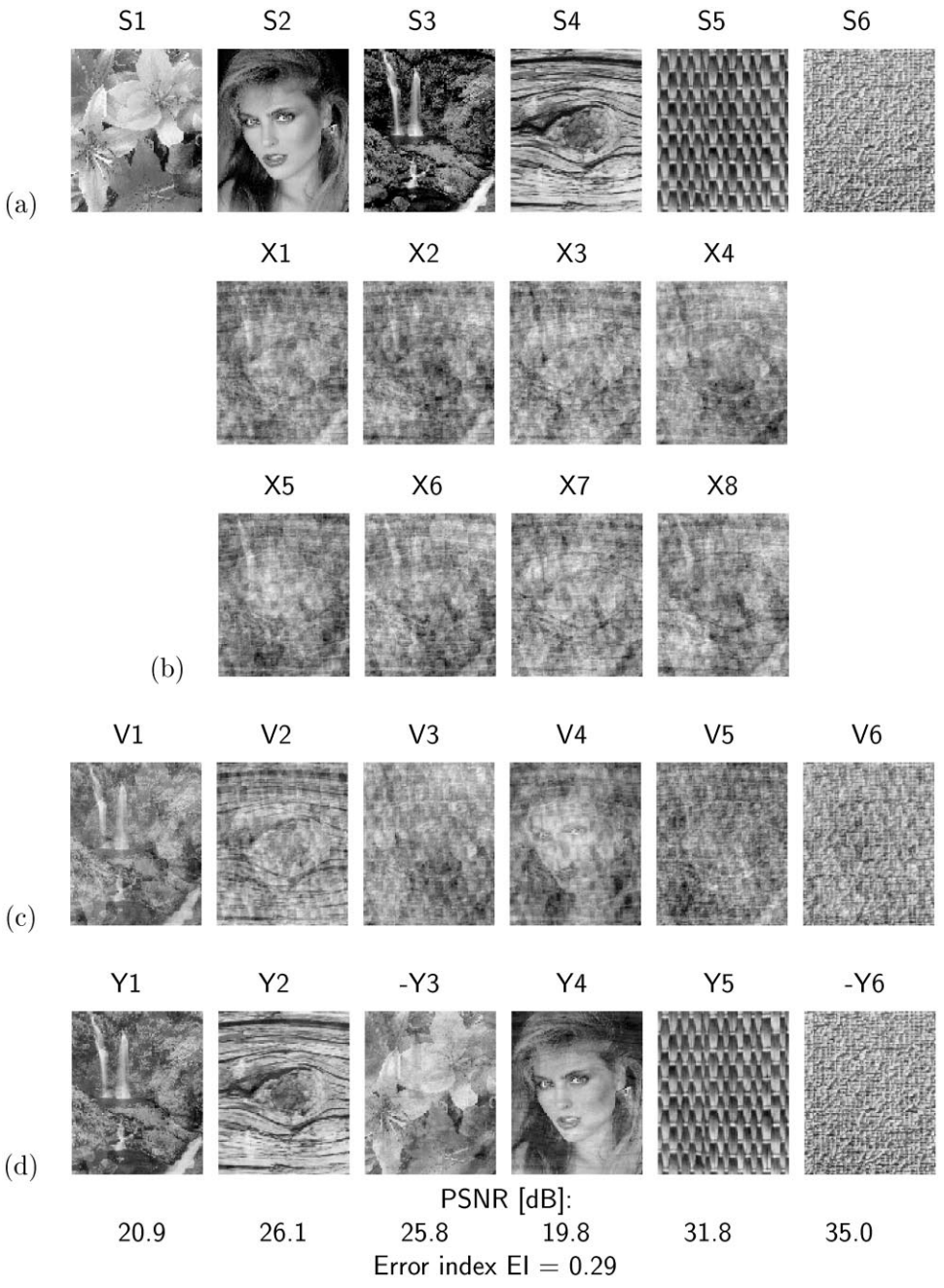


Fig. 7. Example of correct source number determination in context with PCA based pre-whitening: (a) six sources S1–S6, (b) eight mixtures X1–X8, (c) six uncorrelated (pre-whitened) signals V1–V6 and (d) six separated signals Y1–Y6.

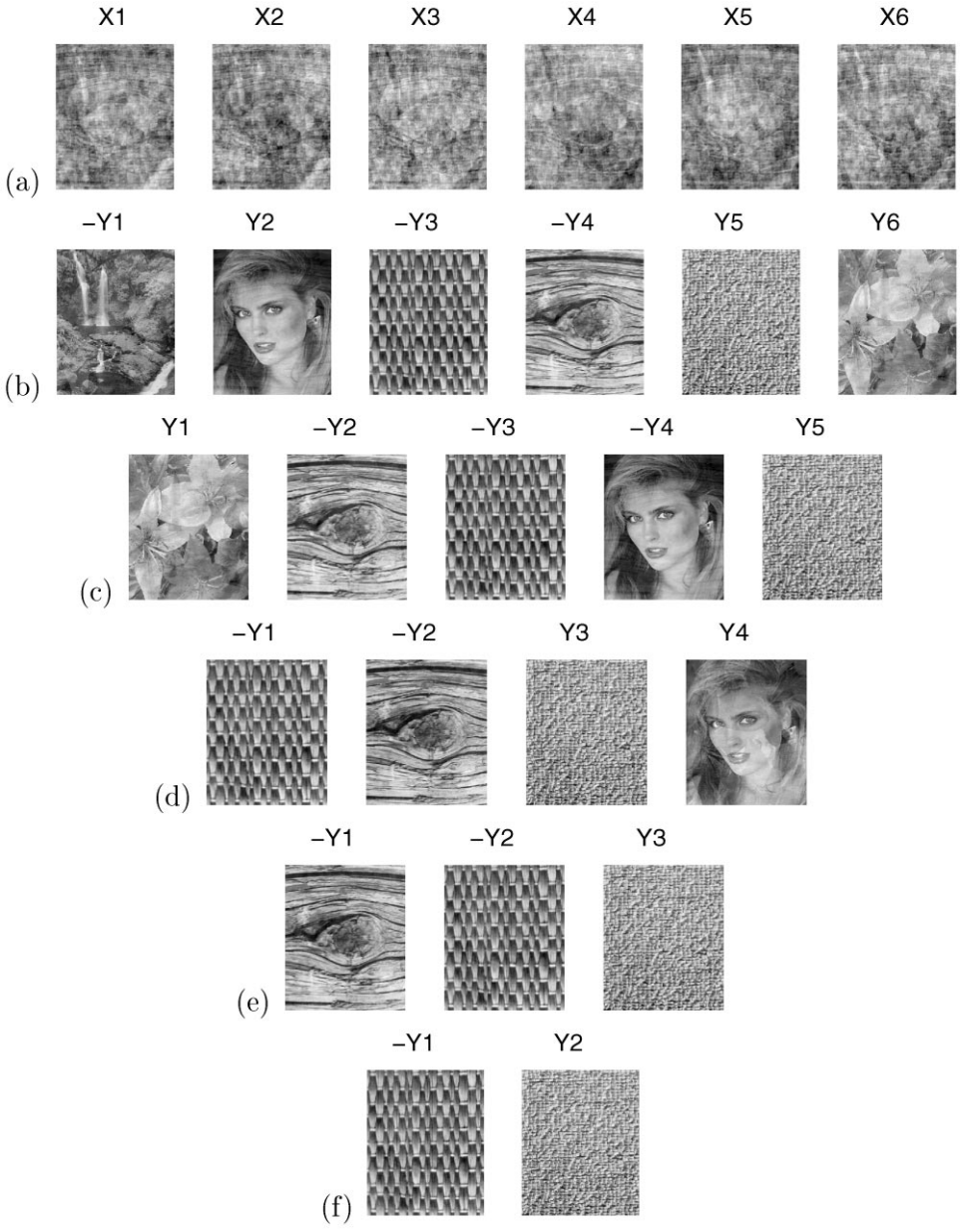


Fig. 8. Examples of source separation using a modified network with non-square matrix $\boldsymbol{W}$. There are both six mixtures and pre-whitened signals. Data compression takes place in the separation layer: (a) six mixtures, (b–f) separation signals for different reduction ratios, i.e. if there are 5, 4, 3 or 2 output signals, respectively.

Three original images were mixed by a randomly chosen matrix, which is assumed to be completely unknown to the separation network. This matrix $\boldsymbol{A}_{5\times 3}$ has a condition number of 43.1:

$$\boldsymbol{A}_{5\times 5} = \begin{Bmatrix} 1.11 & 1.12 & 1.09 \\ 1.55 & 1.11 & 1.20 \\ 1.10 & 1.088 & 1.31 \\ 0.99 & 0.980 & 0.97 \\ 1.20 & 0.960 & 1.30 \end{Bmatrix}. \tag{33}$$

In the first experiment no noise was assumed, whereas in the second case additional additive noise were added to every sensor image (mixture). The results of processing the first set of mixtures (noise-free) are given in Fig. 9, whereas the results for the noisy case are provided in Fig. 10.

The local learning rule used in the multi-layer network structure estimates the sources in a sequential order – the first source after one processing cycle, the second source after the second processing cycle, etc.

The global learning rule determines all the sources simultaneously using a single-layer network. If there are more outputs and mixtures than sources ($n > m$), the separation quality is usually slightly worse than in the case where the number of mixtures is correct ($n = m$). The final redundancy elimination layer suppresses redundant signals and does not switch between channel signals.

In the noisy case even no redundancy among the sources occur. On the "free" output channels the noise signals appear instead (compare the bottom row in Fig. 10). The separated images are already of high quality. Hence, the redundancy elimination layer may even lead to slight decrease in the quality of separation. Then it is better to choose as outputs of the network those signals $y_i(t)$ from the separation layer which correspond to non-suppressed channels in the reduction layer.

4.5. Sequential separation of sources

In this last example, we show that it is also possible to extract one independent component or source signal at a time from the available mixtures. This technique has turned out to be very useful in practical applications where the number of sources or independent components is completely unknown.

The real-world data used in these experiments consisted of auditory and somatosensory evoked fields (AEFs and SEFs, respectively), measured by means of magnetoencephalography (MEG). MEG is a non-invasive brain mapping technique, related to the electroencephalography (EEG), sensitive to the net magnetic flux arising from the post-synaptic currents of thousands of neurons, acting synchronously. We used a 122-channel whole-scalp Neuromag-122^{TM} neuromagnetometer, thus the original data vector sequence was 122-dimensional. The AEFs and SEFs are cortical

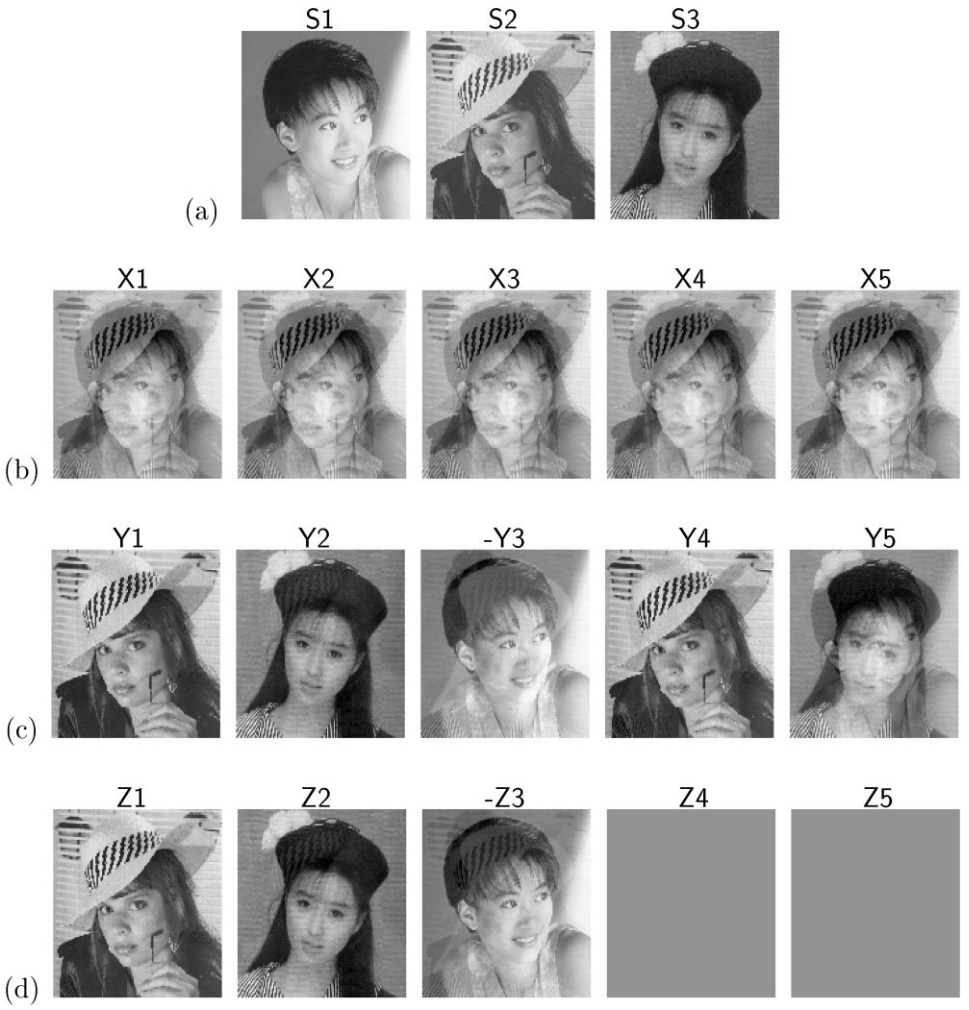


Fig. 9. Example of blind separation with redundancy reduction using the model in Fig. 4 (no noise). There are five mixtures (b) of three unknown sources (a) available. After a single layer with global rule (c) two redundant images always appear. After applying the post-processing layer both these signals are suppressed (d), as required.

responses to auditory and somatosensory simulation, time-locked to the respective stimuli, presenting minimal inter-individual differences to a particular set of stimulus parameters (see [29,51] for more detailed description).

For practical extraction of the most powerful independent components, we used a computationally efficient fixed-point algorithm [28,26,46]. One iteration of the generalized fixed-point algorithm for finding a row vector $\boldsymbol{w}_i^{\mathrm{T}}$ of the orthogonal

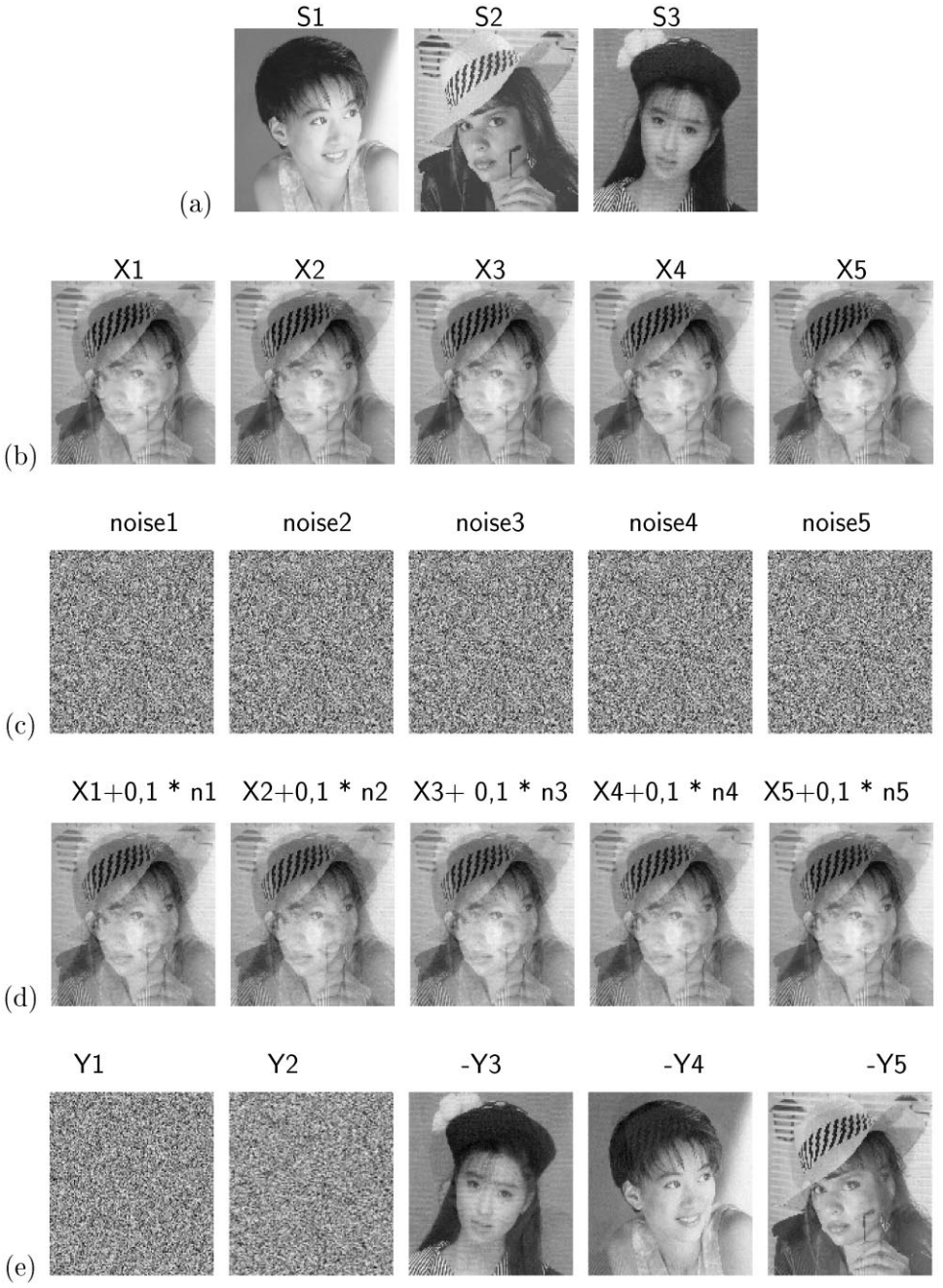


Fig. 10. Example of blind separation with more sensors than sources and with additive noise (app. 10%) – the first stage of the model in Fig. 4 is applied. At first three sources (a) are mixed to five sensor images (b). Then five convolutive noise signals of the noise image (c) are added to the mixed images – 10% of one noise signal to one sensor image (d). After a single layer with global rule (e) two noise images appear, but the three others correspond clearly to the three source images. In this case the redundancy reduction layer need not to be applied.

separating matrix $\hat{\boldsymbol{W}}$ after pre-whitening is [26,46]

$$\begin{aligned} \boldsymbol{w}_i^* &= E\{\boldsymbol{v}g(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{v})\} - E\{g'(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{v})\}\boldsymbol{w}_i, \\ \boldsymbol{w}_i &= \boldsymbol{w}_i^*/\|\boldsymbol{w}_i^*\|. \end{aligned} \tag{34}$$

Here $g(y)$ is again a suitable nonlinearity, typically $g(y) = y^3$ or $g(y) = \tanh(y)$, and $g'(y)$ is its derivative. If the cubic nonlinearity $g(y) = y^3$ is used, $E\{g'(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{v})\} = 3\|\boldsymbol{w}\|^2$. This choice yields the standard fixed-point algorithm [28], which is somewhat simpler and was used in these experiments. The expectations are in practice replaced by their sample means. Hence, the fixed-point algorithm is not a truly neural adaptive algorithm. However, we want to emphasize that neural separating algorithms could have been used instead of the fixed-point algorithm here, too. The vectors $\boldsymbol{w}_i$ must be orthogonalized against each other; this can be done either sequentially or symmetrically [26]. Usually the algorithm (34) converges after 5–20 iterations.

It should be added that a suitable data compression, made during the whitening process as discussed in previous sections, may be required in order to avoid overfitting, typical of ICA methods. Choosing a mild compression rate, or no compression at all, may lead to solutions that are practically zero almost everywhere, except at the point of a single spike or bump.

As seen earlier, ICA solutions are defined up to a scaling and permutation. Nevertheless, it is expectable that solutions corresponding to the most powerful

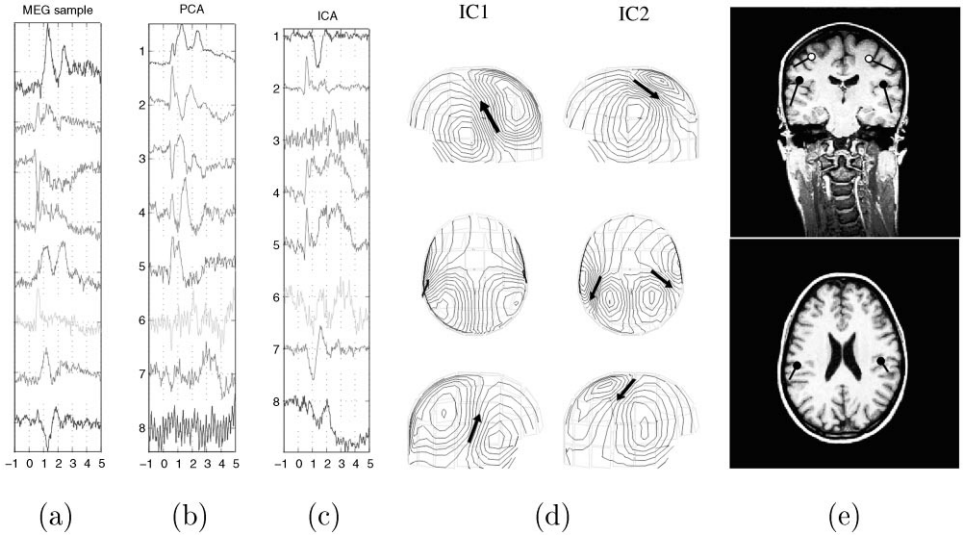


Fig. 11. (a) A subset of the 122-MEG channels. (b) Principal and (c) independent components of the data. (d) Field patterns corresponding to the first two independent components. In (e) the superposition of the localizations of the dipole originating IC1 (black circles, corresponding to the auditory cortex activation) and IC2 (white circles, corresponding to the SI cortex activation) onto magnetic resonance images (MRI) of the subject. The bars illustrate the orientation of the source net current.

sources represent stronger attractors that the others, so that the first independent components found usually correspond to the strongest sources.

Fig. 11 presents a subset of MEG channels with the strongest responses, together with the first eight principal and independent components. It depicts as well the field patterns associated to the first two ICs, superimposed onto helmet-shaped sensory array of the neuromagnetometer, viewed from the left, top and right (note that these patterns are columns of the estimated mixing matrix $\boldsymbol{A}$). The arrows inserted correspond to the equivalent current dipoles (ECDs) used to model the experimental data. Both the patterns and the ECDs agree with the physiological conventions for these type of evoked fields [29] (in IC1 the direction of the dipoles is inverted, but this corresponds to a negative scaling factor over the IC, which is still a valid ICA solution). The corresponding locations (black circles for AEFs, white circles for SEFs) are shown in e), superimposed onto MRI of the subject. These locations are respectively over the auditory, and primary somatosensory cortices, as expected from the experimental setting. The bars, indicating the orientation of the source net current, agree with the orientation expected for such cortical signals (perpendicular to the outer border of the cortex).

The results given by ICA are physiologically clearly more meaningful than those given by PCA in this experiment. In earlier papers, we have got very encouraging results on using ICA in removing artifacts from EEG and MEG data [49,50], indicating that ICA is a valuable and promising tool in biomedical signal processing.

5. Discussion and conclusions

The main topic of this paper is to study experimentally what happens in typical feed-forward neural networks proposed for blind source separation when the number of sources is different from the number of sensors and/or outputs of the networks. This is an important practical issue which is usually skipped by assuming that the number of sources is known and equals to the number of mixtures. We have presented both qualitative and quantitative results for the two dominant classes of such networks which differ with respect to the need of pre-whitening.

Two promising neural network approaches are presented for the problem of blind separation of unknown number of sources, where only the maximum possible number of active sources is known in advance. In such a case the number of sensors is usually larger than the number of source signals.

The main results of this paper can be summarized as follows:

1. A brief analysis and discussion of two pre-whitening rules, local and global (equivariant), is given. It is shown how they can be derived rigorously.
2. A generalization of the nonlinear PCA rule with two different nonlinear functions is proposed.
3. It is pointed out that some of the presented blind separation rules are nonlinear extensions and/or generalizations of the pre-whitening rules.

4. Recommendations of various pre-processing and post-processing methods are given for solving BSS problems with more sensors than sources.
5. The validity and performance of proposed solutions are illustrated by several computer simulations. It is demonstrated that all the discussed algorithms work properly for complex source signals like natural images, although their complexities and properties are different.
6. The main advantages of the proposed methods are their simplicity, adaptivity (the algorithms can be used on-line), and in some cases locality and/or robustness for badly scaled and ill-conditioned mixing matrices.
7. In large-scale real-world problems the number of source signals is generally unknown. Then it is possible to extract the most powerful sources one at a time using an efficient semi-neural fixed-point algorithm. We demonstrate the usefulness of this approach by extracting independent components from real-world auditory evoked fields where the original data vectors were 122-dimensional.

Some new issues arose from the results of our experiments. In the basic ICA/BSS model it is assumed that the source signals are mutually independent. For example, for the face images this does not generally hold even as an approximation, because they are usually clearly correlated. Due to this fact clear differences in the behavior of separation methods can be observed. In the methods applying pre-whitening, the orthogonality constraint set on the separating matrix forces the output signals to be mutually uncorrelated (except for the nonlinear PCA rule). The second class of learning algorithms tries to perform whitening and separation simultaneously in one or more layers. Thus, they respond to higher-order statistics of the source signals at the same time as the sources are de-correlated. As a result, the outputs of these networks are not necessarily uncorrelated for correlated source signals.

It is interesting to note that the algorithms discussed in this paper (with the exception of the fixed-point rule) can roughly retrieve the original source signals even though they are not completely independent or not even uncorrelated (which is a considerably milder condition than independence). Thus, they can achieve in the BSS problem more than the current theory promises. On the other hand, the output images given by these algorithms are in fact more independent than the original correlated source images. Hence, we expect that they are better estimates of the true independent components of the mixed data than the original sources. This is especially true for the natural/relative gradient rule which tries to minimize a measure of independence, namely the Kullback–Leibler divergence [1,10].

It seems that if the statistical independence assumptions made customarily in the BSS/ICA data model do not hold, the solutions of the ICA and BSS problems are in fact different. This is a very fundamental and interesting issue that needs to be studied in the future. Another topic is to study the behavior of the separating networks in the case where the amount of additive and/or multiplicative noise is not insignificant. Also a challenging and in practice important task is to investigate various possible generalizations of the proposed algorithms, like multichannel blind deconvolution/equalization, or separation of convolved and delayed sources with unknown delays.

Acknowledgements

The authors are grateful to Profs. Shun-ichi Amari and Erkki Oja for useful comments and discussions, to Markus Peltonen for making preliminary experiments for Section 4.2 and Jaakko Särelä for help in the experiments of Section 4.5. The authors thank as well Prof. Riitta Hari and Veikko Jousmäki for the original MEG data. Ricardo Vigário holds a grant from JNICT, under its Programa PRAXIS XXI.

The authors thank the anonymous reviewers for their professional and detailed comments which were very helpful in improving the revision.

Appendix A. Analysis of the whitening rule

A.1. Derivation of the whitening rule

The learning rule (12) can be easily derived by minimizing the following loss (cost) function:

$$J(\boldsymbol{V}) = \tfrac{1}{4}\|\boldsymbol{R}_{vv} - \boldsymbol{I}\|_{\mathrm{F}}, \tag{A.1}$$

where $\|.\|_{\mathrm{F}}$ means the Frobenius norm and $\boldsymbol{R}_{vv} = E[\boldsymbol{v}(t)\boldsymbol{v}^{\mathrm{T}}(t)]$ is the correlation matrix of output vector $\boldsymbol{v}(t)$.

It is interesting to notice that this correlation matrix can be expressed as

$$\boldsymbol{R}_{vv} = E[\boldsymbol{V}\boldsymbol{x}(t)\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{V}^{\mathrm{T}}] = \boldsymbol{V}\boldsymbol{R}_{xx}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{V}\boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{V}^{\mathrm{T}}. \tag{A.2}$$

Hence, without loss of generality, assuming that $\boldsymbol{R}_{ss} = E[\boldsymbol{s}(t)\boldsymbol{s}^{\mathrm{T}}(t)] = \boldsymbol{I}$ we have

$$\boldsymbol{R}_{vv} = \boldsymbol{P}(t)\boldsymbol{P}^{\mathrm{T}}(t), \tag{A.3}$$

where matrix $\boldsymbol{P}(t) = \boldsymbol{V}(t)\boldsymbol{A}$ describes the global system of mixing and whitening operations.

Multiplying Eq. (12) by the mixing matrix $\boldsymbol{A}$ from the right-hand side we get

$$\boldsymbol{V}(t+1)\boldsymbol{A} \overset{\mathrm{df}}{=} \boldsymbol{P}(t+1) = \boldsymbol{P}(t) + \eta(t)[\boldsymbol{I} - \boldsymbol{P}(t)\boldsymbol{s}(t)\boldsymbol{s}^{\mathrm{T}}(t)\boldsymbol{P}^{\mathrm{T}}(t)]\boldsymbol{P}(t). \tag{A.4}$$

Taking the expectation value of both sides of the above equation and assuming, without loss of generality, that the autocorrelation matrix is a unit matrix, i.e.

$$\boldsymbol{R}_{ss} \overset{\mathrm{df}}{=} E\{\boldsymbol{s}(t)\boldsymbol{s}^{\mathrm{T}}(t)\} = \boldsymbol{I}, \tag{A.5}$$

it is evident that a learning algorithm, employing the above rule (12), achieves equilibrium when the matrix $\boldsymbol{P}(t)$ becomes orthogonal, i.e. $\boldsymbol{P}^{-1} = \boldsymbol{P}^{\mathrm{T}}$. In this derivation we have made the simplifying assumption that $\boldsymbol{P}(t)$ is independent of the current source vector $\boldsymbol{s}(t)$. This holds at least as a good approximation, because $\boldsymbol{P}(t)$ depends through the whitening matrix $\boldsymbol{V}(t)$ only on the previous values $\boldsymbol{s}(i)$, $i < t$, of the source vector $\boldsymbol{s}(t)$.

A.2. Convergence analysis of the whitening rule

The algorithms which iteratively apply either the rule (11) or (12) achieve the equilibrium point when the output signal covariance matrix achieves

$$\boldsymbol{R}_{vv} = E[\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}] = E[\boldsymbol{V}\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}^{\mathrm{T}}] = \boldsymbol{V}E[\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}]\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{V}\boldsymbol{R}_{xx}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{I}. \tag{A.6}$$

In other words, for any non-singular matrix $\boldsymbol{V}$ we can write

$$\boldsymbol{V}\boldsymbol{R}_{xx}\boldsymbol{V}^{\mathrm{T}} - \boldsymbol{I} = (\boldsymbol{V}\boldsymbol{R}_{xx}\boldsymbol{V}^{\mathrm{T}} - \boldsymbol{I})\boldsymbol{V} = \boldsymbol{V}^{\mathrm{T}}(\boldsymbol{V}\boldsymbol{R}_{xx}\boldsymbol{V}^{\mathrm{T}} - \boldsymbol{I}) = \boldsymbol{0}. \tag{A.7}$$

Hence

$$\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V} = \boldsymbol{R}_{xx}^{-1}. \tag{A.8}$$

Taking into account that matrix $\boldsymbol{V}$ is symmetrical, i.e. $\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{V}$, we get the equilibrium point

$$\boldsymbol{V} = \boldsymbol{R}_{xx}^{-1/2}. \tag{A.9}$$

On the other hand, $\boldsymbol{R}_{vv}$ can be decomposed as

$$\boldsymbol{R}_{vv} = \tilde{\boldsymbol{U}}\boldsymbol{\Lambda}\tilde{\boldsymbol{U}}^{\mathrm{T}}, \tag{A.10}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues and $\tilde{\boldsymbol{U}}$ is an orthogonal matrix of corresponding eigenvectors of $\boldsymbol{R}_{vv}$. As $\boldsymbol{R}_{vv}$ tends to the unit matrix $\boldsymbol{I}$ and the matrix $\tilde{\boldsymbol{U}}$ is orthogonal, the matrix $\boldsymbol{\Lambda}$ must also tend to the unit matrix. This means that output signals $v_i(t)$ will be orthogonal signals with unit variances.

Appendix B. Derivation of adaptive learning algorithms for BSS

In this section we consider feed-forward neural networks of simple form which need not pre-whitening. Let us consider a single layer neural network described by

$$\boldsymbol{y}(t) = \boldsymbol{W}(t)\boldsymbol{x}(t), \tag{B.1}$$

where $\boldsymbol{W}$ is a square $n \times n$ non-singular matrix of synaptic weights.

It should be noted that in this model we assume that the number of outputs is equal to the number of sensors ($l = m$), although the number of sources can be less than the number of sensors ($n \leq m$). Such neural network model is justified by two facts. First, the number of sources can change over the time and it is generally unknown. Second, in practice, we expect that additive noise exists in each sensor. Such noise signals can itself be considered as auxiliary unknown sources. Thus, it is reasonable to use extra outputs in order to extract (if possible) also these noise signals.

In an ideal noiseless case our objective is to develop such a learning algorithm, which provides a decay to zero of all redundant ($m - n$) output signals y_i, while the remaining n output signals correspond to single recovered sources.

B.1. Cost functions

Stochastic independence of random variables is a more general concept than uncorrelatedness or whiteness. Independence can be expressed by the relationship $q(s_i, s_j) = q_i(s_i)q(s_j)$, where $q(s)$ denotes the probability density function (p.d.f.) of random variable s. More generally, a set of signals $\boldsymbol{s}$ consists of independent signals if their joint p.d.f. can be decomposed as

$$q(\boldsymbol{s}) = \prod_{i=1}^{m} q_i(s_i), \tag{B.2}$$

where $q_i(s_i)$ is the p.d.f. of the ith source signal.

In this paper we assume for simplicity that all variables are real-valued and the number of source signals is equal or less to the number of sensors and that the source signals are of zero mean ($E[s_i(t)] = 0$).

We also assume that additive noises are reduced in the pre-processing stage to a small level. Most learning algorithms are derived from heuristic considerations based on minimization or maximization of a loss or performance function [1,4,30,16,17,19]. It is remarkable that entropy maximization (infomax) [5], independent component analysis (ICA) [4,19], and maximization of likelihood (ML) [10] lead to a formulation based on the same type of loss functions [8].

The *Kullback–Leibler divergence* (relative entropy) between two probability density functions (p.d.f.s) $f_y(\boldsymbol{y})$ and $q(\boldsymbol{y})$ on $\boldsymbol{R}^n$ is defined as [1,2,8]:

$$J(\boldsymbol{y},\boldsymbol{W}) = D_{pq}(p_y(\boldsymbol{y})\|q(\boldsymbol{y})) = \int_{-\infty}^{\infty} p_y(\boldsymbol{y})\log\frac{p_y(\boldsymbol{y})}{q(\boldsymbol{y})}\,\mathrm{d}\boldsymbol{y}, \tag{B.3}$$

whenever the integral exists. The Kullback–Leibler divergence always take non-negative values, achieving zero if and only if $p_y(\boldsymbol{y})$ and $q(\boldsymbol{y})$ have the same distribution. It is invariant with respect to an invertible (monotonic) nonlinear transformation of variables, including amplitude rescaling and permutation, in which the variables y_i are rearranged. For the independent component analysis problem, we assume that $q(\boldsymbol{y})$ is the product of the distribution of independent variables y_i. It can be the product of the marginal p.d.f.s of $\boldsymbol{y}$, in particular,

$$q(\boldsymbol{y}) = f_M(\boldsymbol{y}) = \prod_{i=1}^{n} p_i(y_i), \tag{B.4}$$

where $p_i(y_i)$ are the marginal probability density functions of y_i ($i = 1,2,\ldots,n$). The marginal p.d.f. is defined as

$$p_i(y_i) = \int_{-\infty}^{\infty} p_y(\check{\boldsymbol{y}}^i)\,\mathrm{d}\check{\boldsymbol{y}}^i, \tag{B.5}$$

where the integration is taken over $\check{\boldsymbol{y}}^i = [y_1 \ldots y_{i-1}\ \ y_{i+1} \ldots y_n]^{\mathrm{T}}$, i.e. the vector remaining after removing the variable y_i. The natural measure of independence can be formulated as

$$J(\boldsymbol{y},\boldsymbol{W}) = \int_{-\infty}^{\infty} p_y(\boldsymbol{y})\log\frac{p_y(\boldsymbol{y})}{\prod_{i=1}^{n} p_i(y_i)}\,\mathrm{d}\boldsymbol{y}. \tag{B.6}$$

The above Kullback–Leibler divergence can be expressed in terms of the mutual information as

$$D_{pq} = -H(\boldsymbol{y}) - \sum_{i=1}^{n} \int_{-\infty}^{\infty} p_y(\boldsymbol{y}) \log p_i(y_i)\, \mathrm{d}\boldsymbol{y}, \tag{B.7}$$

where the differential entropy of output signals $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$ is defined as

$$H(\boldsymbol{y}) = -\int_{-\infty}^{\infty} p_y(\boldsymbol{y}) \log p_y(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}. \tag{B.8}$$

Taking into account that $\mathrm{d}\boldsymbol{y} = \mathrm{d}\check{\boldsymbol{y}}^i\, \mathrm{d}y_i$ the second part in Eq. (B.7) can be expressed by the marginal entropies as

$$\int_{-\infty}^{\infty} p_y(\boldsymbol{y}) \log p_i(y_i)\, \mathrm{d}\boldsymbol{y} = \int_{-\infty}^{\infty} \log p_i(y_i) \int_{-\infty}^{\infty} p_y(\boldsymbol{y}) \mathrm{d}\check{\boldsymbol{y}}^i\, \mathrm{d}y_i = \int_{-\infty}^{\infty} p_i(y_i) \log p_i(y_i)\, \mathrm{d}y_i$$
$$= E\{\log(p_i(y_i))\} = -H_i(y_i). \tag{B.9}$$

Hence, the Kullback–Leibler divergence can be expressed by the differential $H(\boldsymbol{y})$ and the marginal entropies $H_i(y_i)$ as

$$J(\boldsymbol{y},\boldsymbol{W}) = -H(\boldsymbol{y}) + \sum_{i=1}^{n} H_i(y_i). \tag{B.10}$$

Assuming $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$, the differential entropy can be expressed as

$$H(\boldsymbol{y}) = H(\boldsymbol{x}) + \log|\det(\boldsymbol{W})|, \tag{B.11}$$

where $H(\boldsymbol{x}) = -\int_{-\infty}^{\infty} f_x(\boldsymbol{x}) \log f_x(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$ is independent of matrix $\boldsymbol{W}$. Hence, we obtain a simple (cost) contrast function

$$J(\boldsymbol{y},\boldsymbol{W}) = -\log|\det(\boldsymbol{W})| - \sum_{i=1}^{n} E\{\log(p_i(y_i))\}. \tag{B.12}$$

B.2. Gradient rules

The standard gradient of the cost function can be expressed as

$$\Delta\boldsymbol{W} = \frac{\partial J}{\partial \boldsymbol{W}} = -\boldsymbol{W}^{-\mathrm{T}} + \langle \boldsymbol{f}(\boldsymbol{y})\boldsymbol{x}^{\mathrm{T}} \rangle, \tag{B.13}$$

where $\boldsymbol{f}(\boldsymbol{y}) = [f_1(y_1) \cdots f_n(y_n)]^{\mathrm{T}}$ contains the nonlinearities:

$$f_i(y_i) = -\frac{\mathrm{d}\log p_i(y_i)}{\mathrm{d}y_i} = -\frac{\mathrm{d}p_i(y_i)/\mathrm{d}y_i}{p_i(y_i)} = -\frac{p_i'(y_i)}{p_i(y_i)} \tag{B.14}$$

and $\langle . \rangle$ means time average (expectation) over the specified time window.

This leads to the well-known algorithm proposed by Bell and Sejnowski [5]:

$$\Delta\boldsymbol{W} = \eta(\boldsymbol{W}^{-\mathrm{T}} - \boldsymbol{f}(\boldsymbol{y})\boldsymbol{x}^{\mathrm{T}}) = \eta(\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y})\boldsymbol{y}^{\mathrm{T}})\boldsymbol{W}^{-\mathrm{T}}. \tag{B.15}$$

We can improve dramatically the performance of the above rule by applying the natural gradient developed by Amari [4]:

$$\Delta \boldsymbol{W} = -\eta \frac{\partial J}{\partial \boldsymbol{W}} \boldsymbol{W}^{\mathrm{T}}\boldsymbol{W} = \eta[\boldsymbol{I} - \langle \boldsymbol{f}(\boldsymbol{y})\boldsymbol{y}^{\mathrm{T}}\rangle]\boldsymbol{W}. \tag{B.16}$$

Alternatively, we can use the following filtering gradient [14]:

$$\Delta \boldsymbol{W} = -\eta \boldsymbol{W}\left[\frac{\partial J}{\partial \boldsymbol{W}}\right]^{\mathrm{T}} \boldsymbol{W} = \eta[\boldsymbol{I} - \langle \boldsymbol{y}\boldsymbol{g}(\boldsymbol{y}^{\mathrm{T}})\rangle]\boldsymbol{W}, \tag{B.17}$$

where $g(y)$ are inverse (dual) to $f(y) = -p_i(y_i)/p_i(y_i)$. The above learning rules could be merged (combined) together in order to build up a more general learning algorithm [17,18]:

$$\Delta \boldsymbol{W} = \eta[\boldsymbol{\Lambda} - \langle \boldsymbol{f}(\boldsymbol{y})\boldsymbol{g}(\boldsymbol{y}^{\mathrm{T}})\rangle]\boldsymbol{W}, \tag{B.18}$$

where $\boldsymbol{\Lambda}$ is an arbitrary diagonal positive-definite matrix.

It is interesting to note that in the special case for

$$\boldsymbol{g}(\boldsymbol{y}) = \boldsymbol{f}(\boldsymbol{y}) - \boldsymbol{y} \quad \text{and} \quad \boldsymbol{\Lambda} = 0 \tag{B.19}$$

we have

$$\Delta \boldsymbol{W} = \eta \boldsymbol{f}(\boldsymbol{y})[\boldsymbol{y}^{\mathrm{T}} - \boldsymbol{f}(\boldsymbol{y}^{\mathrm{T}})]\boldsymbol{W}. \tag{B.20}$$

Assuming further that the signals are pre-whitened signals, so that $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W} = \boldsymbol{I}$, we obtain the well-known nonlinear PCA rule (14) as

$$\Delta \boldsymbol{W} = \eta \boldsymbol{f}(\boldsymbol{y})[\boldsymbol{x}^{\mathrm{T}} - \boldsymbol{f}(\boldsymbol{y}^{\mathrm{T}})\boldsymbol{W}], \tag{B.21}$$

which need a pre-whitening process ($\boldsymbol{v} = \boldsymbol{V}\boldsymbol{x}$).

Assuming that the constraint $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W} = \boldsymbol{I}$ is satisfied during the learning process, we can easily prove that the above algorithm reduces approximately to the learning rule proposed by Cardoso and Laheld [10] as

$$\Delta \boldsymbol{W} = \eta[\boldsymbol{f}(\boldsymbol{y})\boldsymbol{y}^{\mathrm{T}} - \boldsymbol{y}\boldsymbol{f}(\boldsymbol{y}^{\mathrm{T}})]\boldsymbol{W}. \tag{B.22}$$

Connections of the nonlinear PCA rule (14) to other blind separation approaches are studied also in [38].

B.3. Generalized parameterized distribution models – Practical implementation of the algorithm for extended ICA

The performance of the BSS learning algorithms depends on the shape of the activation functions. Optimal selection of nonlinearities depends on the p.d.f. of source signals.

For finding quasi-optimal nonlinear activation functions we can use parameterized models of the probability density functions (p.d.f.s). For example for super-Gaussian

sources with positive kurtosis we can use unimodal model of p.d.f. as

$$p_i(y_i) = \frac{\exp(-2\alpha_i y_i)}{1+\exp(-2\alpha_i \boldsymbol{y}_i)^2}, \tag{B.23}$$

which leads to nonlinear activation functions

$$f_i(y_i) = -\frac{\partial \log p_i(y_i)}{\partial y_i} = \tanh(\alpha_i y_i), \tag{B.24}$$

where $\alpha_i > 2$ is a fixed or adaptively adjusted parameter. More general and flexible p.d.f. models are generalized Gaussian, Cauchy, or Rayleigh distributions.

Let us assume, for example, that the source signals have generalized Gaussian distributions of the form [14]

$$p_i(y_i) = \frac{r_i}{2\sigma_i \Gamma(1/r_i)} \exp\left(-\frac{1}{r_i}\left|\frac{y_i}{\sigma_i}\right|^{r_i}\right), \tag{B.25}$$

where $r_i > 0$ is a variable parameter, $\Gamma(r) = \int_0^\infty y^{r-1}\exp(-y)\,\mathrm{d}y$ is the gamma function and $\sigma_i^r = E\{|y_i|^r\}$ is a generalized measure of variance, known as the dispersion of distribution. The parameter r_i can vary from zero, through 1 (Laplace distribution), $r_i = 2$ (standard Gaussian distribution) to infinity (for uniform distribution). The locally optimal normalized nonlinear activation functions can be expressed in such cases as

$$f_i(y_i) = -\frac{\mathrm{d}\log(p_i(y_i))}{\mathrm{d}y_i} = |y_i|^{r_i-1}\,\mathrm{sign}(y_i), \quad r_i \geq 1. \tag{B.26}$$

Taking into account that $\mathrm{sign}\, y = y/|y|$ we obtain

$$f_i(y_i) = \frac{y_i}{|y_i|^{2-r_i}}. \tag{B.27}$$

For spiky or very impulsive signals the parameters r_i can take the value between zero and one. In such case we can use slightly modified activation functions:

$$f_i(y_i) = \frac{y_i}{[|y_i|^{2-r_i} + \varepsilon_i]}, \quad 0 < r_i < 1, \tag{B.28}$$

where ε_i is a very small positive parameter (typically 10^{-4}–10^{-5}), avoiding the singularity of the function for $y_i = 0$.

Alternatively, we can define the moving average of the instantaneous values of nonlinear function as

$$f_i(y_i) = \frac{y_i}{\langle |y_i|^{2-r_i}\rangle} = \frac{y_i(k)}{\hat{\sigma}_i^{(2-r_i)}(k)}, \quad 0 < r_i < \infty, \tag{B.29}$$

with estimation of $\hat{m}_{2-r_i} = \hat{\sigma}_i^{(2-r_i)}$ by the moving average as

$$\hat{\sigma}_i^{(2-r_i)}(k+1) = (1-\eta)\hat{\sigma}_i^{(2-r_i)}(k) + \eta|y_i(k)|^{2-r_i}. \tag{B.30}$$

Such activation function can be considered as "linear" time-variable function modulated in time by the fluctuating estimated moment $m_{2-r_i} = \hat{\sigma}_i^{(2-r_i)}$.

Moreover, when we do not have exact a priori knowledge about the source signal distributions, we can adapt the value of $r_i(t)$ for each output signal $y_i(t)$ according to its estimated distance from ideal Gaussian distribution. A simple gradient-based rule for adjusting each parameter $r_i(k)$ is

$$\Delta r_i(k) = -\eta_i(k)\frac{\partial J}{\partial r_i} = -\eta|y_i|^{r_i}\log|y_i|. \tag{B.31}$$

It is interesting to note that in the special case of very spiky signals corresponding to $r_i \simeq 0$, the optimal function is a "linear" time-variable function proposed by Matsuoka et al. [42] for non-stationary signals:

$$f_i(y_i) = \frac{y_i}{\langle |y_i|^2 \rangle} = \frac{y_i(k)}{\hat{\sigma}_i^2(k)}. \tag{B.32}$$

Summarizing, for blind separation of source signals, which have both positive and negative kurtosis (sub- and super-Gaussian sources) we can apply the learning rule

$$\Delta \boldsymbol{W}(t) = \eta[\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y})\boldsymbol{g}(\boldsymbol{y})^{\mathrm{T}}]\boldsymbol{W}(t) \tag{B.33}$$

with activation functions

$$g(y) = y \quad \text{and} \quad f(y) = \frac{y_i}{|y_i|^{2-r_i} + \varepsilon}, \tag{B.34}$$

where $r_i < 2$ for positive kurtosis and $r_i > 2$ for negative kurtosis.

Alternatively, we can use the following switching nonlinearities [14]:

$$\begin{aligned} f_i(y_i) &= \tanh(\alpha y_i) \qquad \text{for } \kappa_4(y_i) > 0, \\ f_i(y_i) &= |y_i|^{r_i}\mathrm{sign}(y_i) \quad \text{otherwise}, \end{aligned} \tag{B.35}$$

$$\begin{aligned} g_i(y_i) &= |y_i|^{r_i}\mathrm{sign}(y_i) \quad \text{for } \kappa_4(y_i) > 0, \\ g_i(y_i) &= \tanh(\alpha y_i) \qquad \text{otherwise}, \end{aligned} \tag{B.36}$$

with $r_i \geq 1$, $\alpha \geq 2$, where $\kappa_4(y_i) = E\{y_i^4\}/E^2\{y_i^2\} - 3$ is the normalized kurtosis value. The value of the kurtosis can be estimated on-line from the formula

$$E\{y_i^q(k+1)\} = (1-\eta)E\{y^q(k)\} + \eta|y_i(k)|^q \quad (q = 2,4). \tag{B.37}$$

The above learning algorithm (B.33), (B.35)–(B.36) monitors and estimates the statistics of each output signal and depending on the sign or value of its normalized kurtosis (which is the measure of distance from the Gaussianity) automatically selects (or switches) suitable nonlinear activation functions, such that successful (stable) separation of all non-Gaussian source signals is possible. The same on-line kurtosis estimation algorithm can be applied in context with other neural or adaptive blind separation algorithms as well.

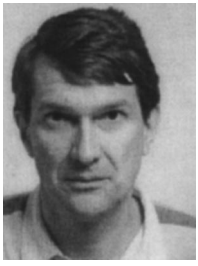
References

[1] S. Amari, Natural gradient works efficiently in learning, Neural Comput. 10 (1998) 251–276.
[2] S. Amari, T.-P. Chen, A. Cichocki, Stability analysis of adaptive blind source separation, Neural Networks 10 (1997) 1345–1351.
[3] S. Amari, A. Cichocki, H. Yang, Recurrent neural networks for blind separation of sources, Proc. Int. Symp. on Nonlinear Theory and its Applications, NOLTA-95, Las Vegas, December 1995, pp. 37–42.
[4] S. Amari, A. Cichocki, H. Yang, A new learning algorithm for blind signal separation, Advances in Neural Information Processing Systems 8, MIT Press, Cambridge, MA, 1996, pp. 757–763.
[5] A.J. Bell, T.J. Sejnowski, An information maximization approach to blind separation and blind deconvolution, Neural Comput. 7 (1995) 1129–1159.
[6] X.-R. Cao, R.-W. Liu, A general approach to blind source separation, IEEE Trans. Signal Process. 44 (1996) 562–571.
[7] J.-F. Cardoso, Infomax and maximum likelihood for blind source separation, IEEE Signal Process. Lett. 4 (1997) 112–114.
[8] J.-F. Cardoso, Entropic contrasts for source separation, in: S. Haykin (Ed.), Adaptive Unsupervised Learning, Ch. 4, John Wiley, 1999 (in press).
[9] J.F. Cardoso, A. Belouchrani, B. Laheld, A new composite criterion for adaptive and iterative blind source separation, Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP–94), vol. 4, 1994, 273–276.
[10] J.F. Cardoso, B. Laheld, Equivariant adaptive source separation, IEEE Trans. on Signal Process. 44 (1996) 3017–3030.
[11] A. Cichocki, W. Kasprzak, Local adaptive learning algorithms for blind separation of natural images, Neural Network World 6 (4) (1996) 515–523.
[12] A. Cichocki, W. Kasprzak, S. Amari, Multi-layer neural networks with a local adaptive learning rule for blind separation of source signals, Proc. Int. Symp. on Nonlinear Theory and its Applications, NOLTA-95, Las Vegas, December 1995, pp. 61–65.
[13] A. Cichocki, W. Kasprzak, S. Amari, Neural network approach to blind separation and enhancement of images, Signal Processing VIII, Proc. EUSIPCO-96, EURASIP/LINT Publ., Trieste, Italy, 1996, vol. I, pp. 579–582.
[14] A. Cichocki, I. Sabala, S. Choi, B. Orsier, R. Szupiluk, Self adaptive independent component analysis for sub-Gaussian and super-Gaussian mixtures with unknown number of sources and additive noise, in: Proc. 1997 Int. Symp. on Nonlinear Theory and its Applications, NOLTA-97, vol. 2, Hawaii, USA, December 1997, pp. 731–734.
[15] A. Cichocki, R. Thawonmas, S. Amari, Sequential blind signal extraction in order specified by stochastic properties, Electron. Lett. 33 (1997) 64–65.
[16] A. Cichocki, R. Unbehauen, Neural Networks for Optimization and Signal Processing, 2nd ed., New York, Wiley, 1994, pp. 461–471.
[17] A. Cichocki, R. Unbehauen, Robust neural networks with on-line learning for blind identification and blind separation of sources, IEEE Trans. Circuits and Systems I: Fundam. Theory Appl. 43 (11) (1996) 894–906.
[18] A. Cichocki, R. Unbehauen, E. Rummert, Robust learning algorithm for blind separation of signals, Electron. Lett. 30 (17) (1994) 1386–1387.
[19] P. Comon, Independent component analysis – a new concept? Signal Process. 36 (1994) 287–314.
[20] P. Comon, C. Jutten, J. Herault, Blind separation of sources, Part II: problem statement, Signal Process. 24 (1991) 11–20.
[21] N. Delfosse, P. Loubaton, Adaptive blind separation of independent sources: a deflation approach, Signal Process. 45 (1995) 59–83.
[22] S.C. Douglas, A. Cichocki, Neural networks for blind decorrelation of signals, IEEE Trans. Signal Process. 45 (11) (1997) 2829–2842.
[23] S.C. Douglas, A. Cichocki, S. Amari, Fast-convergence filtered regressor algorithms for blind equalization, Electron. Lett. 32 (23) (1996) 2114–2115.

[24] M. Girolami, C. Fyfe, Negentropy and kurtosis as projection pursuit indices provide generalized ICA algorithms, Advances in Neural Information Processing Systems, NIPS'96 Workshop, Denver, December 1996.
[25] M. Girolami, C. Fyfe, Generalized independent component analysis through unsupervised learning with emergent bussgang properties, Proc. IEEE Int. Conf. on Neural Networks, ICNN'97, Houston, TX June 1997, pp. 2147–2152.
[26] A. Hyvärinen, A family of fixed-point algorithms for independent component analysis, Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'97), Munich, Germany, April 1997, pp. 3917–3920.
[27] A. Hyvärinen, E. Oja, Simple neuron models for independent component analysis, Int. J. Neural Systems 7 (1996) 671–687.
[28] A. Hyvärinen, E. Oja, A fast fixed-point algorithm for independent component analysis, Neural Comput. 9 (1997) 1483–1492.
[29] V. Jousmäki, R. Hari, Somatosensory evoked fields to large-area vibrotactile stimuli, Electroenceph. Clin. Neurophysiol., under revision.
[30] C. Jutten, J. Herault, Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture, Signal Process. 24 (1991) 1–20.
[31] C. Jutten, H.L. Nguyen Thi, E. Dijkstra, E. Vittoz, J. Caelen, Blind separation of sources, an algorithm for separation of convolutive mixtures, Proc. Int. Workshop on High Order Statistics, Chamrousse, France, July 1991, pp. 273–276.
[32] J. Karhunen, Neural approaches to independent component analysis and source separation, Proc. 4th Eur. Symp. on Artificial Neural Networks, ESANN'96, Bruges, Belgium, April 1996, pp. 249–266.
[33] J. Karhunen, A. Cichocki, W. Kasprzak, P. Pajunen, On neural blind separation with noise suppression and redundancy reduction, Int. J. Neural Systems 8 (1997) 219–237.
[34] J. Karhunen, A. Hyvärinen, R. Vigario, J. Hurri, E. Oja, Applications of neural blind separation to signal and image processing, Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP'97, Munich, Germany, April 1997, pp. 131–134.
[35] J. Karhunen, J. Joutsensalo, Representation and separation of signals using nonlinear PCA type learning, Neural Networks 7 (1) (1994) 113–127.
[36] J. Karhunen, E. Oja, L. Wang, R. Vigario, J. Joutsensalo, A class of neural networks for independent component analysis, IEEE Trans. Neural Networks 8 (1997) 486–504.
[37] J. Karhunen, P. Pajunen, Blind source separation and tracking using nonlinear PCA criterion: a least-squares approach, Proc. Int. Conf. on Neural Networks, ICNN'97, Houston, TX, June 1997, pp. 2147–2152.
[38] J. Karhunen, P. Pajunen, E. Oja, The nonlinear PCA criterion in blind source separation: relations with other approaches, Neurocomputing (1998) Vol. 22, pp. 5–20.
[39] W. Kasprzak, A. Cichocki, Hidden image separation from incomplete image mixtures by independent component analysis, Proc. 13th Int. Conf. on Pattern Recognition, ICPR'96, Vienna, Austria, August 1996, IEEE Computer Society Press, Silver Spring, MD, pp. 394–398.
[40] J.L. Lacoume, P. Ruiz, Separation of independent sources from correlated inputs, IEEE Trans. Signal Process. 40 (1992) 3074–3078.
[41] X.-T. Ling, Y.-F. Huang, R. Liu, A neural network for blind signal separation, Proc. IEEE Int. Symp. on Circuits and Systems, ISCAS-94, London, UK, 1994, pp. 69–72.
[42] K. Matsuoka, M. Ohya, M. Kawamoto, A neural net for blind separation of non-stationary signals, Neural Networks 8 (1995) 411–419.
[43] E. Moreau, O. Macchi, New self-adaptive algorithms for source separation based on contrast functions, Proc. IEEE Signal Process. Workshop on Higher Order Statistics, Lake Tahoe, USA, June 1993, pp. 215–219.
[44] E. Moreau, O. Macchi, High order contrasts for self-adaptive source separation, Int. J. Adaptive Control Signal Processing 10 (1996) 19–46.
[45] E. Oja, The nonlinear PCA learning rule and signal separation – mathematical analysis, Neurocomputing 17 (1997) 25–45.

[46] E. Oja, J. Karhunen, A. Hyvärinen, R. Vigario, J. Hurri, Neural independent component analysis – approaches and applications, in: S. Amari, N. Kasabov (Eds.), Brain-Like Computing and Intelligent Information Systems, Springer, Singapore, 1997, pp. 167–188.
[47] F.M. Silva, L.B. Almeida, A distributed decorrelation algorithm, in: E. Gelenbe (Ed.), Neural Networks, Advances and Applications, North-Holland, Amsterdam, 1991, pp. 145–163.
[48] L. Tong, R. Liu, V.C. Soon, Y.-F. Huang, Indeterminacy and identifiability of blind identification, IEEE Trans. Circuits Systems 38 (1991) 499–509.
[49] R. Vigário, Extraction of ocular artifacts from EEG using independent component analysis, Electroenceph. Clin. Neurophysiol. 103, 395–404.
[50] R. Vigário, V. Jousmäki, M. Hämäläinen, R. Hari, E. Oja, Independent component analysis for identification of artifacts in magnetoencephalographic recordings, Neural Information Processing Systems, Eds. M. Jordan, MIT Press, Cambridge, MA, 1998, Vol. 10, pp. 229–235. NIPS'97, Denver, December 1997.
[51] R. Vigário, J. Särelä, E. Oja, Independent Component Analysis in Wave Decomposition of Auditory Evoked Fields, Proc. Int. Conf. on Artificial Neural Networks, ICANN'98, Skövde, Sweden, 1998, pp. 287–292.
[52] L. Wang, J. Karhunen, A unified neural bigradient algorithm for robust PCA and MCA, Int. J. Neural Systems 7 (1996) 53–67.
[53] E. Weinstein, M. Feder, A.V. Oppenheim, Multi-channel signal separation by de-correlation, IEEE Trans. Speech and Audio Process. 1 (1993) 405–413.
[54] H. Yang, S.-I. Amari, Adaptive online learning algorithms for blind separation: maximum entropy and minimum mutual information, Neural Comput. 9 (1997) 1457–1482.

Andrzej Cichocki received the M.Sc. (with honors), Ph.D. and 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 spent several years at University Erlangen-Nuremberg (Germany), at the Chair of Applied and Theoretical Electrical Engineering, as an Alexander-von-Humboldt Research Fellow and Guest Professor. He is currently the team leader of the laboratory for Open Information Systems, at Brain Science Institute of Riken (Japan), in the group conducted by Prof. Shun-ichi Amari. He is the co-author of two books: *MOS Switched-Capacitor and Continuous-Time Integrated Circuits and Systems* (Springer-Verlag, 1989) and *Neural Networks for Optimization and Signal Processing* (Teubner-Wiley, 1993). His current research interests include neural networks and nonlinear dynamic systems theory.



Juha Karhunen received the Dr.Tech. degree from the Department of Technical Physics of Helsinki University of Technology in 1984. In 1994, he became Docent of statistical signal analysis. Since 1976, he has been in the Laboratory of Computer and Information Science at Helsinki University of Technology, Finland, where he is currently Senior Research Fellow of Academy of Finland. His current research interests include neural networks, unsupervised learning, and their applications to signal processing. He has published many conference and journal papers on these topics, and served as a reviewer to the major journals in these fields. He is a senior member of IEEE and member of Int. Neural Network Society.

Wlodzimierz Kasprzak received his Ph.D. in computer science from Department of Electronics of the Warsaw University of Technology in 1987. In 1989 he was an AvH-research fellow at University of Erlangen, Germany; 1990–1995 he was with the research staff of the Bavarian Research Center FORWISS, Erlangen, Germany, and in 1996 he was a senior researcher in the Frontier Research Program Riken, Lab. for Artificial Brain Systems, Japan. From 1997 he is an Assistant Prof. at Warsaw University of Technology, Institute of Control and Computation Engineering. His research interests include image sequence analysis, object recognition, and neural networks in signal processing and computer vision.

Ricardo Vigário received the B.Sc. and M.Sc. degrees in applied and medical physics and biomedical engineering from the University of Lisbon, Portugal, in 1992 and 1994, respectively. Since 1993, he has been in the Laboratory of Computer and Information Science at Helsinki University of Technology, Finland, where he is currently working towards the Dr.Tech. degree. His current research interests include neural networks in signal processing, computer vision, and biomedical signal processing.