International Journal of Neural Systems, Vol. 8, Nos. 5 & 6 (October/December 1997) 601–612

LEAST-SQUARES METHODS FOR BLIND SOURCE SEPARATION BASED ON NONLINEAR PCA

PETTERI PAJUNEN* and JUHA KARHUNEN†
Helsinki University of Technology, Laboratory of Computer and Information Science,
P.O. Box 5400, FIN-02015 HUT, Espoo, Finland

Received November 10, 1997
Accepted May 18, 1998

In standard blind source separation, one tries to extract unknown source signals from their instantaneous linear mixtures by using a minimum of *a priori* information. We have recently shown that certain nonlinear extensions of principal component type neural algorithms can be successfully applied to this problem. In this paper, we show that a nonlinear PCA criterion can be minimized using least-squares approaches, leading to computationally efficient and fast converging algorithms. Several versions of this approach are developed and studied, some of which can be regarded as neural learning algorithms. A connection to the nonlinear PCA subspace rule is also shown. Experimental results are given, showing that the least-squares methods usually converge clearly faster than stochastic gradient algorithms in blind separation problems.

1. Introduction

In recent years, blind signal processing has become an important application and research domain of both unsupervised neural learning and statistical signal processing. In the basic blind source separation (BSS) problem, the goal is to separate mutually statistically independent but otherwise unknown source signals from their instantaneous linear mixtures without knowing the mixing coefficients. BSS techniques have applications in a wide variety of problems for example in communications, speech processing, array processing, and medical signal processing. References and a brief description of some applications of neural BSS techniques to signal and image processing can be found in Refs. 1, 2.

Blind source separation is based on the strong but often plausible requirement that the separated sources must be statistically independent (or as independent as possible). Because direct verification of the independence condition is very difficult, some suitable higher-order statistics are in practice used for achieving separation. In neural BSS methods, higher-order statistics are incorporated into processing implicitly by using suitable nonlinearities in the learning algorithms. Different neural approaches to BSS and to the closely related Independent Component Analysis (ICA)[3,4] are reviewed in Refs. 2, 5. It has turned out that fairly simple neural algorithms[4,6–12] are able to learn a satisfactory separating solution in many instances.

In particular, we have recently shown that several nonlinear PCA (Principal Component Analysis) type neural algorithms can successfully separate a number of sources on certain conditions. Blind separation using stochastic gradient type neural learning algorithms based on nonlinear PCA approaches is discussed in detail in Ref. 10. This work is based on our earlier extensions of standard neural principal component analysis into various forms containing some simple nonlinearities.[13,14] The idea of extending

*E-mail: Petteri.Pajunen@hut.fi
†E-mail: Juha.Karhunen@hut.fi

neural PCA learning rules so that some nonlinear processing is involved was more or less independently proposed by Sanger,[15] Oja,[16] and Xu.[17]

However, neural and adaptive algorithms proposed thus far for blind source separation are typically stochastic gradient algorithms which apply a coarse instantaneous estimate of the gradient. Such algorithms are fairly simple, but they require careful choice of the learning parameters for providing acceptable performance. If the learning parameter is too small, convergence can be intolerably slow; on the other hand the algorithm may become unstable if the learning parameter is chosen too large.

In this paper, we introduce efficient recursive least-squares (RLS) type algorithms for the blind source separation problem. These algorithms minimize in a different way the same nonlinear PCA criterion which we have used previously as a basis of separation.[10] The proposed basic algorithms use relatively simple operations, and they can still be realized using nonlinear PCA networks. The main advantage of these algorithms is that the learning parameter is determined automatically from the input data so that it becomes roughly optimal. This usually leads to a significantly faster convergence compared with the corresponding stochastic gradient algorithms where the learning parameter must be chosen using some ad hoc rule.

Recursive least-squares methods have a long history in statistics, adaptive signal processing, and control; see Refs. 18, 19. For example in adaptive signal processing, it is well-known that RLS methods converge much faster than the standard stochastic gradient based least-mean square (LMS) algorithm at the expense of somewhat greater computational cost.[18] Similar properties hold for the RLS algorithms presented in this paper.

Our basic RLS algorithms are obtained by modifying approximate RLS algorithms proposed by Yang[20] for the standard linear PCA problem. A fundamental difference between Yang's algorithms and ours is that the former ones do not contain any nonlinearities, and hence utilize second-order statistics only. Therefore, they cannot be directly applied to blind source separation. Apart from Yang's work, some other authors (for example Refs. 21, 22) have applied different RLS approaches to the standard linear PCA problem.

The contents of the rest of the paper is as follows. In the next section the necessary background on the blind source separation problem and associated neural network models is briefly presented. Then the nonlinear PCA criterion is shown to be a contrast function with a suitable choice of nonlinearity. Connections between the nonlinear PCA subspace criterion and the Bussgang criterion as well as the EASI algorithm are discussed. After this, we introduce the basic recursive least-squares algorithms and some variants of them. After presentation of selected experimental results, the paper ends with conclusions and some remarks.

2. Neural Blind Source Separation

2.1. *The blind separation problem*

The blind source separation problem has the following basic form. Assume that there exist m zero-mean source signals $s_1(t), \ldots, s_m(t)$ that are scalar-valued and mutually statistically independent at each time instant or index value t. The original sources $s_i(t)$ are unknown, and we observe n possibly noisy but different linear mixtures $x_1(t), \ldots, x_n(t)$ of the sources. The constant mixing coefficients are also unknown. In blind source separation, the task is to find the waveforms $\{s_i(t)\}$ of the sources, using only the mixtures $x_j(t)$. Some examples of source signals are speech signals (cocktail party problem), EEG signals, and digital images.

Denote by $\mathbf{x}(t) = [x_1(t), \ldots, x_n(t)]^T$ the n-dimensional data vector made up of the mixtures at discrete time (or point) t. The BSS signal model can then be written in the matrix form

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t)\,. \tag{1}$$

Here $\mathbf{s}(t) = [s_1(t), \ldots, s_m(t)]^T$ is the source vector, and $\mathbf{A}$ is a constant full-rank $n \times m$ mixing matrix whose elements are the unknown coefficients of the mixtures. The additive noise term $\mathbf{n}(t)$ is often omitted from (1), because it is usually impossible to separate noise from the sources without some prior knowledge on noise.

The number of available different mixtures n must be at least as large as the number of sources m. Usually m is assumed to be known, and the number of sources is the same as the number of mixtures ($m = n$). Furthermore, each source signal $s_i(t)$ is

assumed to be a stationary zero-mean stochastic process. Only one of the sources is allowed to have a Gaussian distribution. In practice, it is often possible to separate the sources approximately even though they are not strictly mutually independent.[24]

Essentially the same data model (1) is used in Independent Component Analysis. Assumptions on the model are described in more detail in Refs. 3, 10, 23. It is possible to extend the basic data model (1) into various directions for example to include time delays etc. Some possibilities and references are listed in Ref. 5. In particular, various methods for handling cases where the number of mixtures is different (usually greater) than that of sources are discussed in Ref. 24, and suppression of noise in Ref. 25.

2.2. *Neural network model*

In neural and adaptive BSS, an $m \times n$ separating matrix $\mathbf{B}(t)$ is updated so that the m-vector

$$\mathbf{y}(t) = \mathbf{B}(t)\mathbf{x}(t) \tag{2}$$

becomes an estimate $\mathbf{y}(t) = \hat{\mathbf{s}}(t)$ of the original independent source signals. In neural realizations, $\mathbf{y}(t)$ is the output vector of the network, and the matrix $\mathbf{B}(t)$ is the total weight matrix between the input and output layers. Only the waveforms of the source signals can be recovered, since the estimate $\hat{s}_i(t)$ of the ith source signal may appear in any component $y_j(t)$ of $\mathbf{y}(t)$. The amplitudes and signs of the estimates $y_j(t)$ may also be arbitrary due to the inherent inderminacies of the BSS problem.[26] The estimated sources are typically scaled to have a unit variance.

In several BSS algorithms, the data vectors $\mathbf{x}(t)$ are preprocessed by whitening them through a linear transform $\mathbf{V}$ so that the covariance matrix $\mathrm{E}\{\mathbf{x}(t)\mathbf{x}(t)^T\}$ becomes the unit matrix $\mathbf{I}_m$. Whitening can be done in many ways, for example using standard PCA or simple adaptive neural algorithms; see Ref. 10. After prewhitening, the separation task becomes somewhat easier, because the components of the whitened vectors $\mathbf{x}(t)$ are already uncorrelated which is a necessary prerequisite of independence. Also the subsequent $m \times m$ separating matrix, denoted here by $\mathbf{W}^T(t)$, can be taken orthogonal: $\mathbf{W}^T(t)\mathbf{W}(t) = \mathbf{I}_m$. The total separating matrix between input and output layers is $\mathbf{B}(t) = \mathbf{W}^T(t)\mathbf{V}(t)$.

These considerations lead to a two-layer network structure with weight matrices $\mathbf{V}$ and $\mathbf{W}^T$. Feedback connections are needed in the learning phase between the neurons in each layer. In the standard stationary case, the whitening and separating matrices converge to some constant values during learning, and the network becomes purely feedforward after learning. However, the same model can be used in the general nonstationary situation by keeping these matrices time-varying. The ensuing network structure is discussed in more detail in Ref. 10.

Roughly speaking, the currently existing neural blind separation algorithms can be divided into two main groups: the methods in the first group try to find the total separating matrix $\mathbf{B}(t)$ directly, while the methods of the second group use prewhitening. Whitening has some advantages mentioned before but also has disadvantages; especially if some of the source signals are very weak or the mixture matrix is ill-conditioned, prewhitening may greatly lower the accuracy of separation. In the following, we will use the notation $\mathbf{x}(t)$ for both whitened and non-whitened mixture vectors, explicitly mentioning when whitening is required.

A simple criterion for separating prewhitened sources having the same known sign of kurtosis is the sum of kurtoses of the outputs of the network or the separating system.[27] Our approaches are related to this criterion, because it provides simple but yet sufficiently efficient neural algorithms. The kurtosis of the ith output $y_i(t)$ is defined as

$$\kappa_4[y_i(t)] = \mathrm{E}\{y_i(t)^4\} - 3[\mathrm{E}\{y_i(t)^2\}]^2 . \tag{3}$$

Due to the prewhitening, $\mathrm{E}\{y_i(t)^2\} = 1$, and it suffices to consider the sum of the fourth moments of the outputs. This criterion is minimized for sub-Gaussian sources (for which the kurtosis is negative), and maximized for super-Gaussian sources (having a positive kurtosis value). For Gaussian sources, the kurtosis is zero. The theory of separation is presented in more detail in Refs. 5, 10, 27.

3. The Nonlinear PCA Criterion

3.1. *The basic criterion*

Standard principal component analysis (PCA) is a well-defined and fairly unique technique, but it utilizes second-order statistics of the data only. There exist many neural learning algorithms for performing standard PCA.[28,29] However, the PCA problem can be solved efficiently using numerical eigenvector

algorithms, so that neural gradient based algorithms are often not competitive in practical applications.

If the standard PCA problem is extended so that some nonlinearities are involved, the situation changes considerably. The nonlinearities introduce at least implicitly some higher-order statistics into computations. This is often desirable for non-Gaussian data, which may contain a lot of useful information in their higher-order statistics. Furthermore, neural approaches become more competitive from the computational point of view because there usually does not exist any simple algebraic solution to the nonlinear problem. These issues are discussed in more detail in Refs. 13 and 14, where several simple approaches to nonlinear PCA are introduced by generalizing optimization problems leading to standard PCA. Generally, nonlinear PCA is a non-unique concept. There exist many possible nonlinear extensions of PCA which often lead to somewhat different solutions. In addition to Refs. 13 and 14, various neural approaches to nonlinear PCA are discussed and introduced in, for example, Refs. 16, 17, 30–33.

In this paper, we concentrate on a specific form of nonlinear PCA which has been turned out to be especially useful in blind source separation. This is obtained by minimizing the nonlinear PCA subspace criterion[17,13]

$$J_1(\mathbf{W}) = \mathrm{E}\{\|\mathbf{x} - \mathbf{W}\mathbf{g}(\mathbf{W}^T\mathbf{x})\|^2\} \qquad (4)$$

with respect to the $m \times m$ weight matrix $\mathbf{W}$. Here $\mathbf{g}(\mathbf{y})$ denotes the vector which is obtained by applying a nonlinear function $g(t)$ componentwise to the vector $\mathbf{y}$. In the nonlinear PCA criterion (4), $g(t)$ is usually some odd function such as $g(t) = \tanh(t)$ or $g(t) = t^3$. The criterion (4) was first proposed in a more general context by Xu in Ref. 17.

Denoting the ith column of the matrix $\mathbf{W}$ by $\mathbf{w}_i$, the criterion (4) can be written in the form

$$J_1(\mathbf{W}) = \mathrm{E}\{\|\mathbf{x} - \sum_{i=1}^{m} g(\mathbf{w}_i^T\mathbf{x})\mathbf{w}_i\|^2\}. \qquad (5)$$

This shows that the sum which tries to approximate the data vector $\mathbf{x}$ is linear with respect to the weight vectors $\mathbf{w}_i$, the nonlinearity $g(t)$ appearing only in the coefficients $g(\mathbf{w}_i^T\mathbf{x})$ of the expansion. This simple form of nonlinear PCA is not the best possible if the goal is to approximate the data vector $\mathbf{x}$ for example in data compression or representation, but the coefficients introduce higher-order statistics which are needed in blind separation.

The criterion (4) can be approximately minimized using the stochastic gradient descent algorithm

$$\Delta\mathbf{W} = \mu[\mathbf{x} - \mathbf{W}\mathbf{g}(\mathbf{W}^T\mathbf{x})]\mathbf{g}(\mathbf{x}^T\mathbf{W}) \qquad (6)$$

This nonlinear PCA subspace rule has been independently derived in Refs. 17 and 13. In (6), μ is a positive learning parameter, and we have omitted the time index t from all the quantities for simplicity.

The algorithm (6) was first proposed by Oja *et al.* in Ref. 16 on heuristic grounds. For more details and related algorithms, see Refs. 16, 17, 13, 14.

3.2. *Application to blind source separation*

In applying the criterion (4) and the algorithm (6) to the blind separation problem, it is essential that the data vectors $\mathbf{x}(t)$ are first preprocessed by whitening them. Thus the nonlinear PCA learning rule is applied to BSS problems in the form

$$\Delta\mathbf{W} = \mu[\mathbf{x} - \mathbf{W}\mathbf{g}(\mathbf{y})]\mathbf{g}(\mathbf{y}^T) \qquad (7)$$

where $\mathbf{y} = \mathbf{W}^T\mathbf{x}$. Later on we have justified in several papers summarized in Refs. 10 that for the prewhitened mixture vectors $\mathbf{x}(t)$, $\mathbf{W}^T(t)$ becomes an orthogonal $m \times m$ separating matrix provided that all the source signals are of the same type, namely either sub-Gaussian or super-Gaussian. In practice, this condition can be mildened somewhat so that one of the sources can be of different type if its kurtosis has the smallest absolute value.[23,10]

In order to achieve separation, it suffices that the nonlinearity $g(t)$ is of right type.[36,10] More precisely, for sub-Gaussian sources $g(t)$ should grow less than linearly. Later, it is seen that this condition is closely connected with a general nonlinearity determined by the source signal density functions. We have used in our earlier experiments the sigmoidal nonlinearity $g(t) = \tanh(t)$ with good results. The robustness of the blind separation problem against choosing a non-optimal nonlinearity is discussed in Refs. 36 and 26.

The nonlinear PCA rule (7) can be applied also for super-Gaussian sources using Fahlman type activation functions.[38] Alternatively, one could use the cubic nonlinearity $g(t) = t^3$. However, this kind of

fast growing nonlinearity often requires extra measures (some kind of normalization) for keeping the algorithm stable.

The separation properties of the algorithm (7) have been analyzed rigorously in simple cases in Ref. 37. Recently it has been shown[38] that the criterion function (4) is approximately related to a separating contrast function derived in Ref. 3 Below, we show an exact correspondence using a specific nonlinearity. The orthogonality of the separating matrix $\mathbf{W}$ ($\mathbf{W}^T\mathbf{W} = \mathbf{W}\mathbf{W}^T = \mathbf{I}$, where $\mathbf{I}$ is the unit matrix) allows us to analyze the criterion (4) in more detail for prewhitened data vectors $\mathbf{x}$. In this case, one can express the criterion (4) in the form

$$\begin{aligned} J_1(\mathbf{W}) &= \mathrm{E}\{\|\mathbf{x} - \mathbf{W}\mathbf{g}(\mathbf{W}^T\mathbf{x})\|^2\} \\ &= \mathrm{E}\{\|\mathbf{W}^T\mathbf{x} - \mathbf{W}^T\mathbf{W}\mathbf{g}(\mathbf{W}^T\mathbf{x})\|^2\} \\ &= \mathrm{E}\{\|\mathbf{y} - \mathbf{g}(\mathbf{y})\|^2\} \\ &= \sum_{i=1}^{m} \mathrm{E}\{[y_i - g(y_i)]^2\}\,. \end{aligned} \tag{8}$$

If we now define an odd quadratic function illustrated in Fig. 1 as

$$g(y) = \begin{cases} y^2 + y, & y \geq 0; \\ -y^2 + y, & y < 0, \end{cases} \tag{9}$$

the criterion (8) becomes

$$J_1(\mathbf{W}) = \sum_{i=1}^{m} \mathrm{E}\{[y_i - y_i \pm y_i^2]^2\} = \sum_{i=1}^{m} \mathrm{E}\{y_i^4\} \tag{10}$$

The statistic $J_1(\mathbf{W}) = \sum_i \mathrm{E}\{y_i^4\}$ has been rigorously shown to be a contrast in Ref. 27 under the following assumptions: all the sources have the same known sign of kurtosis, and the data have been prewhitened. Therefore, each of the global minima (there are several) of J_1 corresponds exactly to the separating solutions. Experimentally it has been found that local optimization of J_1 usually provides the separated sources.

The simple form (8) makes it easier to understand the effect of any nonlinearity to separation, and allows a comparison to known contrast functions.[3] Consider for example the sigmoidal nonlinearity $g(t) = \tanh(t)$ which has been used widely in neural separation algorithms. Inserting the Taylor series expansion $\tanh(t) = t - t^3/3 + 2t^5/15 - \ldots$ into (8), one can easily see that in this case the criterion $J_1(\mathbf{W})$ actually depends on sixth order (and higher) statistics.

3.3. *Relationships to other approaches*

3.3.1. *Blind equalization using Bussgang approaches*

The form $\mathrm{E}\{\|\mathbf{y} - \mathbf{g}(\mathbf{y})\|^2\}$ is similar to the Bussgang blind equalization cost,[18,39] in which the nonlinearity is chosen to be

$$g(x) = \frac{-\mathrm{E}\{|x|^2\} p_x'(x)}{p_x(x)}\,, \tag{11}$$

where p_x is the probability density function of x. Since the mixtures are whitened, the expectation $\mathrm{E}\{|x|^2\} = 1$, and for $p_x(x) = 1/\cosh(x)$ we get

$$g(x) = \frac{\sinh(x)\,\cosh(x)}{\cosh^2(x)} = \tanh(x)\,. \tag{12}$$

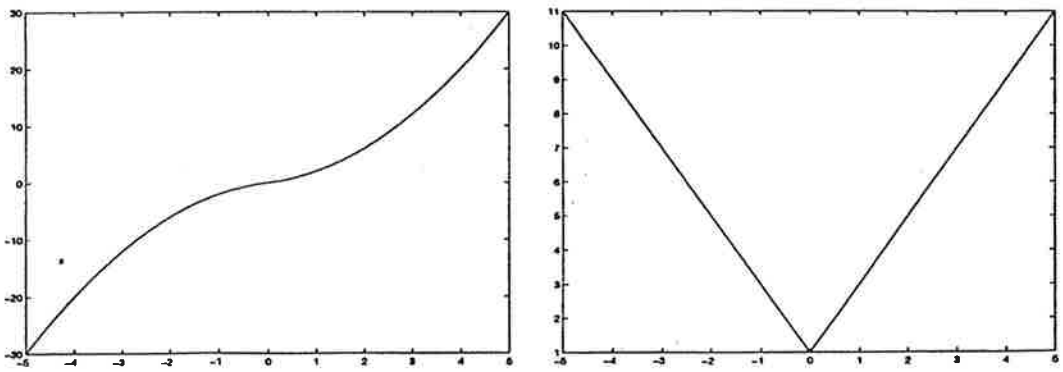

Fig. 1. Left: The odd quadratic function g. Right: The derivative of g.

This means that (4) with $g(x) = \tanh(x)$ is a Bussgang blind equalization cost for sources with a density proportional to $p_x(x) = 1/\cosh(x)$. This corresponds to a signal with a positive kurtosis. However, in the nonlinear PCA learning rule the nonlinearity $g(x) = \tanh(x)$ makes it possible to separate signals with *negative* kurtosis. Conversely, nonlinearities corresponding to signals with negative kurtosis according to (11) can be used to separate signals with positive kurtosis using nonlinear PCA learning rule. The reasons for this are discussed in Ref. 40. Although in the Bussgang blind equalization cost it is assumed that the density p_x is known, it has been shown that it is not necessary to exactly match the nonlinearity with the source density.[36]

Based on Lambert's work,[39] we have also shown in Ref. 40 that the minimization of the cost $\mathrm{E}\{\|\mathbf{y} - \mathbf{g}(\mathbf{y})\|^2\}$ can be interpreted as finding an extremal point of the sum of negentropies

$$J = \sum_i E\{\log p_{y_i}(y_i)/\log p_G(y_i)\}$$

Here p_{y_i} is the probability density function of the ith output signal y_i and p_G is the density of the Gaussian distribution with the same variance as y_i. Thus minimization of the nonlinear PCA criterion for prewhitened data is also closely related to a meaningful information-theoretic criterion, since the sum of negentropies measures the non-Gaussianity of the outputs.

3.3.2. *The EASI algorithm*

A well-known stochastic gradient algorithm for blind separation without prewhitening is the EASI algorithm introduced by Cardoso and Laheld in Ref. 23. The general update formula for the separating matrix $\mathbf{B}$ is in EASI

$$\Delta\mathbf{B} = \mu_k[\mathbf{I} - \mathbf{y}\mathbf{y}^T - \mathbf{g}(\mathbf{y})\mathbf{y}^T + \mathbf{y}\mathbf{g}(\mathbf{y}^T)]\mathbf{B}\,. \quad (13)$$

In Ref. 40 we have derived a closely related algorithm from the nonlinear PCA rule (7) as follows. For the total separating matrix $\mathbf{B} = \mathbf{W}^T\mathbf{V}$, the update rule is in general

$$\Delta\mathbf{B} = \mu_k[\Delta\mathbf{W}^T\mathbf{V} + \mathbf{W}^T\Delta\mathbf{V}]\,. \quad (14)$$

Using the nonlinear PCA learning rule for $\Delta\mathbf{W}$ and the simple whitening algorithm

$$\Delta\mathbf{V} = [\mathbf{I} - \mathbf{x}\mathbf{x}^T]\mathbf{V} \quad (15)$$

and constraining $\mathbf{W}$ to be orthogonal, the following algorithm is obtained for the total separating matrix $\mathbf{B}$:

$$\Delta\mathbf{B} = \mu_k[\mathbf{I} - \mathbf{y}\mathbf{y}^T + \mathbf{g}(\mathbf{y})\mathbf{y}^T - \mathbf{g}(\mathbf{y})\mathbf{g}(\mathbf{y}^T)]\mathbf{B} \quad (16)$$

A comparison with the EASI algorithm (13) shows that the derived algorithm (16) differs only slightly from it (the sign of the nonlinear part $\mathbf{g}(\mathbf{y})\mathbf{y}^T - \mathbf{g}(\mathbf{y})\mathbf{g}(\mathbf{y}^T)$ is not important). In Ref. 23, the EASI algorithm is derived by making rather heavy but sensible approximations.

4. Least-Squares Algorithms for the Nonlinear PCA Criterion

4.1. *Symmetric recursive algorithm*

Our basic recursive least-squares algorithms are closely related to Yang's recent work,[20] in which he has derived a RLS algorithm called PAST for adaptive tracking of signal subspaces. A signal subspace models the "signal" part in the data, and it is essentially the same as a PCA subspace of suitable dimension. Signal subspaces are used especially in linear eigenvector methods of array processing and sinusoidal frequency estimation.

Yang derives his PAST algorithm from the cost function

$$J_2(\mathbf{W}) = \mathrm{E}\{\|\mathbf{x} - \mathbf{W}\mathbf{W}^T\mathbf{x}\|^2\}\,. \quad (17)$$

This criterion differs from the nonlinear PCA criterion (4) in that the nonlinear function $g(t)$ is lacking from it. Therefore, the criterion (17) does not take into account any higher-order statistics in the data even implicitly.

The cost function (17) has been considered in several earlier papers dealing with PCA neural networks.[34,13,14,28,17,35] It is well-known that the minimum of (17) is provided by any orthogonal matrix $\mathbf{W}$ whose columns span the PCA subspace defined by the principal eigenvectors of the data covariance matrix $\mathrm{E}\{\mathbf{x}\mathbf{x}^T\}$ (for zero mean data). Some of these results have been rederived in Ref. 41 together with a convergence analysis of the PAST algorithms.

In this paper we extend Yang's PAST algorithm so that it can be used for minimizing the nonlinear cost function (4), and apply the resulting modified algorithm to blind source separation. Our modified symmetric (or subspace type) nonlinear PAST

algorithm adapted for the BSS problem reads as follows:

$$\begin{aligned}
\mathbf{z}(t) &= \mathbf{g}(\mathbf{W}^T(t-1)\mathbf{x}(t)) = \mathbf{g}(\mathbf{y}(t))\,,\\
\mathbf{h}(t) &= \mathbf{P}(t-1)\mathbf{z}(t)\,,\\
\mathbf{m}(t) &= \mathbf{h}(t)/(\beta + \mathbf{z}^T(t)\mathbf{h}(t))\,,\\
\mathbf{P}(t) &= \frac{1}{\beta}\mathrm{Tri}[\mathbf{P}(t-1) - \mathbf{m}(t)\mathbf{h}^T(t)]\,,\\
\mathbf{e}(t) &= \mathbf{x}(t) - \mathbf{W}(t-1)\mathbf{z}(t)\,,\\
\mathbf{W}(t) &= \mathbf{W}(t-1) + \mathbf{e}(t)\mathbf{m}^T(t)\,.
\end{aligned} \tag{18}$$

The constant $0 < \beta \leq 1$ is a forgetting term which should be close to unity. The notation Tri means that only the upper triangular part of the argument is computed and its transpose is copied to the lower triangular part, making thus the matrix $\mathbf{P}(t)$ symmetric. The simplest way to choose the initial values is to set both $\mathbf{W}(0)$ and $\mathbf{P}(0)$ to $m \times m$ unit matrices. According to the theory, the data vectors $\mathbf{x}(t)$ must be prewhitened prior to using them in the algorithm (18).

The symmetric algorithm (18) can be regarded either as a neural network learning algorithm or an adaptive signal processing algorithm. It does not require any matrix inversions, because the most complicated operation is division by a scalar. In particular, the sequential version discussed in the next subsection becomes relatively simple when written out for each weight vector.

The algorithm (18) and its modified versions can be derived using the following principles.[20,19] Because the expectation in (4) is unknown, it is first replaced by the sum over the last samples, leading to the respective least-squares criterion. Then we approximate the unknown vector $\mathbf{g}(\mathbf{W}^T(t)\mathbf{x}(t))$ by the vector $\mathbf{z}(t) = \mathbf{g}(\mathbf{W}^T(t-1)\mathbf{x}(t))$. This vector can be easily computed because the estimated weight matrix $\mathbf{W}^T(t-1)$ from the previous iteration step $t-1$ is already known. The approximation error is usually rather small after initial convergence, because the update term $\Delta\mathbf{W}(t)$ then becomes small compared to weight matrix $\mathbf{W}(t)$. These considerations yield the modified least-squares type criterion

$$J_3(\mathbf{W}(t)) = \sum_{i=1}^{t} \beta^{t-i} \|\mathbf{x}(i) - \mathbf{W}(t)\mathbf{z}(i)\|^2\,. \tag{19}$$

If the forgetting factor $\beta = 1$, all the samples are given the same weight, and no forgetting of old data takes place. Choosing $\beta < 1$ is useful especially in tracking nonstationary changes in the sources. The cost function (19) is now of the standard form used in recursive least-squares methods. Any of the available algorithms[19] can be used for solving the weight matrix $\mathbf{W}(t)$ iteratively. We have in this paper used the efficient algorithm (18).

4.2. *Sequential recursive algorithm*

The algorithm (18) updates the whole weight matrix $\mathbf{W}(t)$ simultaneously, treating all the weight vectors or the columns $\mathbf{w}_1(t), \ldots, \mathbf{w}_m(t)$ of the matrix $\mathbf{W}(t)$ in a symmetric way. Alternatively, we can compute the weight vectors $\mathbf{w}_i(t)$ in a sequential manner using a deflation technique. The resulting algorithm has the following form:

$$\begin{aligned}
&\mathbf{x}_1(t) = \mathbf{x}(t)\,;\\
&\text{For each } i = 1, \ldots, m \text{ compute}\\
&\quad z_i(t) = g(\mathbf{w}_i^T(t-1)\mathbf{x}_i(t))\,,\\
&\quad d_i(t) = \beta d_i(t-1) + [z_i(t)]^2\,,\\
&\quad \mathbf{e}_i(t) = \mathbf{x}_i(t) - \mathbf{w}_i(t-1)z_i(t)\,,\\
&\quad \mathbf{w}_i(t) = \mathbf{w}_i(t-1) + \mathbf{e}_i(t)[z_i(t)/d_i(t)]\,,\\
&\quad \mathbf{x}_{i+1}(t) = \mathbf{x}_i(t) - \mathbf{w}_i(t)z_i(t)\,.
\end{aligned} \tag{20}$$

Here again the data vectors $\mathbf{x}(t)$ must be prewhitened, and $d_i(t)$ provides an individual learning parameter $1/d_i(t)$ for each weight vector $\mathbf{w}_i(t)$ from a simple recursion formula.

Let us now compare the algorithms (18) and (20) to the nonlinear PCA subspace learning rule (7). Consider for ease of comparison the single neuron case $m = 1$. Both the symmetric and sequential versions then reduce to the algorithm

$$\mathbf{w}(t) = \mathbf{w}(t-1) + \frac{1}{d(t)}[\mathbf{x}(t) - \mathbf{w}(t-1)z(t)]z(t) \tag{21}$$

where $z(t) = g(\mathbf{w}^T(t-1)\mathbf{x}(t)) = g(y(t))$, and

$$d(t) = \beta d(t-1) + [z(t)]^2\,. \tag{22}$$

The algorithm (21) is exactly the same as the nonlinear PCA learning rule (7) except for the scalar learning parameter. In (21), the learning parameter $1/d(t)$ is determined automatically from the properties of the data using the recursion (22) so that it becomes roughly optimal due to the minimization of

the least-squares criterion (19). On the other hand, in (7) the learning parameter $\mu(t)$ is usually a constant which is chosen in a somewhat ad hoc manner or by tuning it to the average properties of the data. It is just this nearly optimal choice of the learning parameter that yields the algorithms (18) and (20) their superior convergence properties compared to standard stochastic gradient type learning algorithms such as (7).

The original PAST algorithms[20] which estimate either the PCA subspace or the PCA eigenvectors themselves have a similar relationship to the well-known Oja's one-unit rule,[42] which is the seminal neural algorithm for learning the first PCA eigenvector. However, they cannot be applied to the BSS problem because no higher-order statistics or nonlinearities are used.

4.3. *Batch Versions*

4.3.1. *Symmetric batch algorithm*

The previous considerations and the form of the cost function (4) suggest straightforward batch algorithms for optimizing the matrix $\mathbf{W}$. These algorithms are no longer neural because they are nonadaptive and use all the data vectors during each iteration cycle. However, they are derived by iteratively minimizing the same nonlinear PCA criterion as before.

If it is assumed for the moment that the matrix $\mathbf{W}$ in the term $\mathbf{g}(\mathbf{W}^T\mathbf{x})$ is a constant, (4) can be regarded as the least-squares error for the linear model

$$\mathbf{X} = \mathbf{W}\mathbf{g}(\mathbf{W}^T\mathbf{X}) + \mathbf{e} = \mathbf{W}\mathbf{G} + \mathbf{e}\,, \tag{23}$$

where $\mathbf{e}$ is the modeling error or noise term, the matrix

$$\mathbf{G} = [\mathbf{g}(\mathbf{W}^T\mathbf{x}(1)), \ldots, \mathbf{g}(\mathbf{W}^T\mathbf{x}(N))] \tag{24}$$

is a constant, and the data matrix $\mathbf{X}$ is defined by

$$\mathbf{X} = [\mathbf{x}(1), \ldots, \mathbf{x}(N)]\,. \tag{25}$$

Now a new value for the weight matrix $\mathbf{W}$ can be computed by minimizing the least-squares error $\|\mathbf{e}\|^2$ for the linearized model (23). This is equivalent to finding the best approximate solution to the linear matrix equation

$$\mathbf{X} = \mathbf{W}\mathbf{G} \tag{26}$$

in the least-squares error sense. It is well-know (see for example Ref. 43, p. 54) that the optimal solution to this problem is $\hat{\mathbf{W}} = \mathbf{X}\mathbf{G}^+$, where $\mathbf{G}^+$ is the pseudoinverse of $\mathbf{G}$. Now $\mathbf{G}$ actually depends on $\mathbf{W}$, but we can anyway determine $\hat{\mathbf{W}}$ using the following iterative symmetric algorithm:

1. Choose an initial value for $\mathbf{W}$.
2. Compute $\mathbf{G}$ using the current value of $\mathbf{W}$.
3. Update the weight matrix $\mathbf{W}$:

$$\mathbf{W} = \mathbf{X}\mathbf{G}^T(\mathbf{G}\mathbf{G}^T)^{-1} = \mathbf{X}\mathbf{G}^+ \tag{27}$$

4. Continue iteration from step 2 until convergence

The basic idea is to treat the weight matrix in the argument of $\mathbf{g}$ as a constant, and then optimize with respect to the weight matrix $\mathbf{W}$ outside $\mathbf{g}$ using the standard linear least-squares method. After sufficient number of iterations, $\mathbf{W}$ should converge close to a local minimum of (4). Recall that these local minima yield a separating matrix $\mathbf{W}$ for prewhitened mixture vectors $\mathbf{v}$.

4.3.2. *Sequential batch algorithm*

It is possible to find the weight matrix $\mathbf{W}$ one column at a time by minimizing the cost function

$$J_s = \mathrm{E}[\|\mathbf{x} - \mathbf{w}g(\mathbf{w}^T\mathbf{x})\|^2]\,, \tag{28}$$

where $\mathbf{w}$ is a weight vector. A minimizing solution $\hat{\mathbf{w}}$ gives one of the separated sources as the inner product $\hat{\mathbf{w}}^T\mathbf{x}$. We can directly use the derivation of the symmetric algorithm by replacing the matrix $\mathbf{W}$ by the weight vector $\mathbf{w}$. The linear updating step (27) becomes

$$\mathbf{w} = \frac{\sum g(y(j))\mathbf{x}(j)}{\sum g(y(j))^2}\,. \tag{29}$$

Once $\mathbf{w}$ is found, the algorithm is repeated with a different initial value for $\mathbf{w}$ until all the sources have been found. After each iteration, $\mathbf{w}$ must be orthogonalized against the previously found weight vectors using for example the well-known Gram-Schmidt procedure.[28] This ensures that the algorithm does not converge to a previously found vector (see Ref. 44). It seems that in practice it is better to compute only the numerator in (29), and then normalize $\mathbf{w}$ explicitly by $\mathbf{w} = \mathbf{w}/\|\mathbf{w}\|$.

5. Experimental Results

In the experiments below, we used four sub-Gaussian source signals: a ramp, a sinusoid, a binary signal, and uniformly distributed white noise. Three of the sources are deterministic for easy visual comparison and inspection of the results. These sources were mixed linearly using a 4×4 mixing matrix, whose elements were Gaussian random numbers.

In the first experiment the convergence speed of different algorithms was compared. The nonlinear PCA subspace rule was compared with the proposed recursive least-squares algorithms. Convergence was measured by the following cost C, which is similar to the one proposed in Ref. 6:

$$D = \sum_i \left(\sum_j \frac{|p_{ij}|^2}{\max_k |p_{ik}|^2} - 1 \right) + \sum_j \left(\sum_i \frac{|p_{ij}|^2}{\max_k |p_{kj}|^2} - 1 \right)$$

$$C = \sqrt{D} \tag{30}$$

The elements p_{ij} are from the total system matrix $\mathbf{P} = \mathbf{W}^T\mathbf{V}\mathbf{A}$, where $\mathbf{V}$ is the whitening matrix. For a separating solution, the value of C is zero and positive otherwise. A typical learning rate $\mu = 0.01$ was chosen and for the RLS algorithms, a forgetting factor $\beta = 1 - \mu = 0.99$ was chosen. All algorithms used the same data, 1000 samples of four linear mixtures of four sub-Gaussian sources.

In Fig. 2, the value of the performance index C is depicted as a function of iterations. The convergence curves clearly show that the recursive least-squares algorithms perform better than the nonlinear PCA subspace rule. Furthermore, the symmetric algorithm converges faster than the sequential version. In a similar experiment with the parameter values $\mu = 0.02$ and $\beta = 0.98$ the results were similar (see Fig. 3).

In another experiment, the sequential batch algorithm (29) using explicit normalization was compared to the nonlinear PCA learning rule (7) by finding one basis vector $\mathbf{w}$ of four mixtures of four sub-Gaussian sources. The number of data vectors $\mathbf{x}(i)$ was 100, and each algorithm was run 50 cycles. In (7) one cycle means using each sample once (100 iterations). We used the sigmoidal nonlinearity

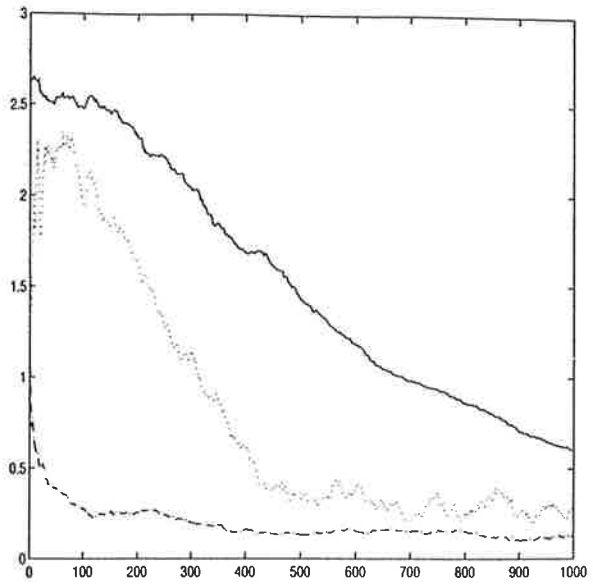

Fig. 2. The convergence speed of the recursive least-squares algorithms is compared with the nonlinear PCA subspace rule using $\mu = 0.01$ and $\beta = 0.99$. Solid: Nonlinear PCA subspace rule. Dashed: Symmetric RLS algorithm. Dotted: Sequential RLS algorithm. The curves show the performace index (30) as a function of iterations.

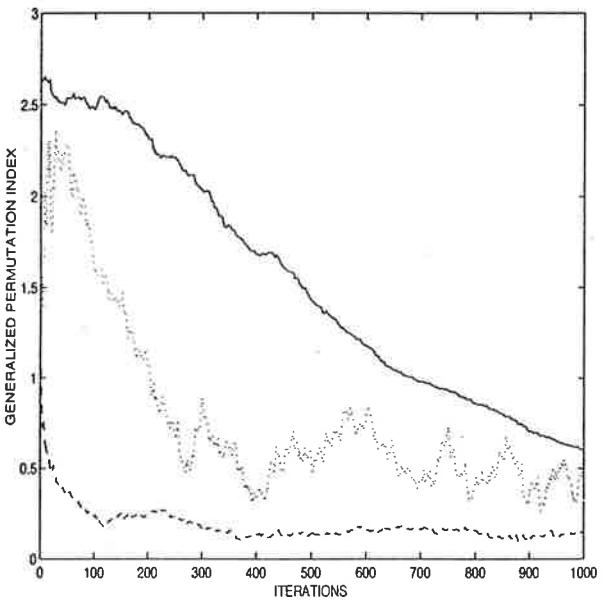

Fig. 3. The convergence of the recursive least-squares algorithms is compared with the nonlinear PCA subspace rule using $\mu = 0.02$ and $\beta = 0.98$. Solid: Nonlinear PCA subspace rule. Dashed: Symmetric RLS algorithm. Dotted: Sequential RLS algorithm. The curves show the performace index (30) as a function of iterations.

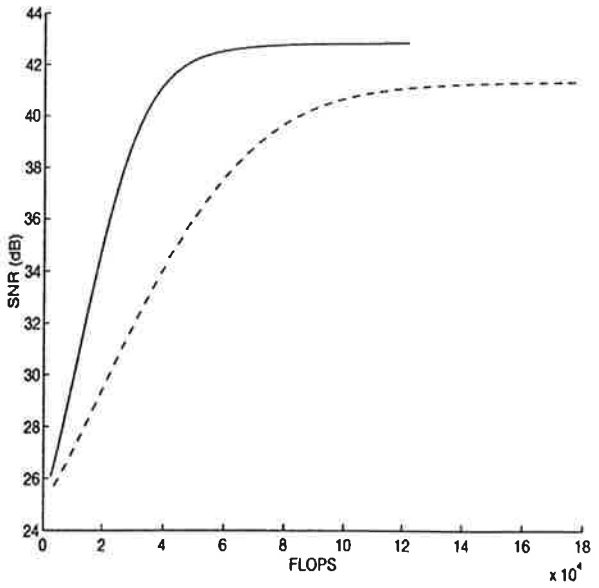

Fig. 4. Solid line: Sequential batch algorithm. Dashed line: Nonlinear PCA algorithm. The x-axis represents the number of floating point operations and the y-axis signal-to-noise ratio of the separated source vs. the corresponding original source.

$g(y) = \tanh(y)$. Figure 4 shows that the proposed batch algorithm converges faster and achieves a better final accuracy.

5.1. *Discussion*

The experimental results presented above are typical ones achieved using these algorithms. In all the computer simulations that we have made thus far the proposed least-squares algorithms (18) and (20) converged faster than the existing adaptive neural BSS algorithms. The difference in the convergence speed is often of the order of magnitude or even higher compared with the nonlinear PCA subspace rule (and other stochastic gradient type algorithms that have a roughly similar performance.[5])

In general, the final accuracy achieved by RLS algorithms can be improved in stationary situations by increasing the forgetting parameter β from its initial value (say $\beta = 0.95$) closer to unity after initial convergence has taken place. Similarly, in gradient type algorithms the value of the learning parameter μ can be decreased with time for achieving a better accuracy. The convergence speed of the nonlinear PCA subspace rule (6) may depend greatly on the chosen initial values of the weight vectors.[38] The least-squares algorithms introduced in this paper are more robust in this respect.

Although the proposed adaptive RLS algorithms utilize a fixed nonlinearity $\mathbf{g}$, it is straightforward to extend the algorithms so that the nonlinearity is not fixed. This can be done e.g. by estimating the sign of the kurtosis of the outputs online (see Ref. 45, 46), which makes it possible to separate sources with different sign of kurtosis.

All the adaptive learning algorithms can be used also for simultaneous tracking and separation of sources in nonstationary situations. This problem is very difficult but important in practice. We have presented some tracking experiments with the proposed RLS type algorithms in Refs. 47, 48.

6. Conclusions

In this paper we have introduced several new algorithms for blind source separation and possible other applications based on a nonlinear PCA criterion. In particular, we have discussed minimization of the nonlinear PCA cost function (4) using approximative least-squares approaches. The proposed nonlinear recursive least-squares type algorithms (18) and (20) provide faster convergence in blind source separation compared with the corresponding stochastic gradient algorithms, in the same sense as recursive least-squares (RLS) algorithms are fast compared with stochastic gradient LMS algorithms in adaptive filtering.[18] According to the experiments made thus far, they provide a very good performance with a fairly low computational load. In some instances, it may be computationally more efficient to use the batch versions of the least-squares algorithms.

We have also mentioned connections of the nonlinear PCA criterion to some well-known existing approaches such as Bussgang methods in blind equalization and the adaptive EASI blind source separation algorithm.

Together with earlier works, the results in this paper demonstrate that nonlinear PCA is a versatile and useful starting point for blind signal processing with close connections to some other well-known approaches. There exist several possibilities for further research such as taking into account robustness, time delays etc.

References

1. J. Karhunen, A. Hyvärinen, R. Vigário, J. Hurri and E. Oja 1997, "Applications of neural blind separation to signal and image processing," in *Proc. 1997 IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP'97)*, Munich, Germany, April 1997, pp. 131–134.
2. E. Oja, J. Karhunen, A. Hyvärinen, R. Vigário and J. Hurri 1997, "Neural independent component analysis — approaches and applications," in *Brain-like Computing and Intelligent Information Systems*, eds. S.-I. Amari and N. Kasabov, (Springer-Verlag, Singapore, 1998), pp. 167–188.
3. P. Comon 1994, "Independent component analysis — A new concept?" *Signal Processing* **36**, 287–314.
4. C. Jutten and J. Herault 1991, "Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture," *Signal Processing* **24**(1), 1–10.
5. J. Karhunen 1996, "Neural approaches to independent component analysis and source separation," in *Proc. 4th European Symp. on Artificial Neural Networks (ESANN'96)*, Bruges, Belgium, April 1996, pp. 249–266.
6. S. Amari, A. Cichocki and H. Yang 1996, "A new learning algorithm for blind signal separation," in *Advances in Neural Information Processing Systems 8*, eds. D. S. Touretzky *et al.* (MIT Press, Cambridge, MA), pp. 757–763.
7. A. Bell and T. Sejnowski 1995, "An information-maximisation approach to blind separation and blind deconvolution," *Neural Computation* **7**, 1129–1159.
8. A. Cichocki and R. Unbehauen 1996, "Robust neural networks with on-line learning for blind identification and blind separation of sources," *IEEE Trans. Circuits Syst. I* **43**, 894–906.
9. L. Wang, J. Karhunen and E. Oja 1995, "A bigradient optimization approach for robust PCA, MCA, and source separation," in *Proc. 1995 IEEE Int. Conf. on Neural Networks*, Perth, Australia, November 1995, pp. 1684–1689.
10. J. Karhunen, E. Oja, L. Wang, R. Vigario and J. Joutsensalo 1997, "A class of neural networks for independent component analysis," *IEEE Trans. Neural Networks* **8**, 486–504.
11. G. Deco and D. Obradovic 1996, *An Information-Theoretic Approach to Neural Computing* (Springer-Verlag, New York).
12. M. Girolami and C. Fyfe 1996, "A temporal model of linear anti-hebbian learning," *Neural Processing Lett.* **4**, 139–148.
13. J. Karhunen and J. Joutsensalo 1994, "Representation and separation of signals using nonlinear PCA type learning," *Neural Networks* **7**(1), 113–127.
14. J. Karhunen and J. Joutsensalo 1995, "Generalizations of principal component analysis, optimization problems, and neural networks," *Neural Networks* **8**(4), 549–562.
15. T. Sanger 1989, "An optimality principle for unsupervised learning," in *Advances in Neural Information Processing Systems 1*, ed. D. Touretzky, (Morgan Kaufmann, Palo Alto, CA), pp. 11–19.
16. E. Oja, H. Ogawa and J. Wangviwattana 1991, "Learning in nonlinear constrained Hebbian networks," in *Artificial Neural Networks* (Proc. ICANN'91, Espoo, Finland), eds. T. Kohonen *et al.*, (North-Holland, Amsterdam), pp. 385–390.
17. L. Xu 1993, "Least mean square error reconstruction principle for self-organizing neural-nets," *Neural Networks* **6**, 627–648.
18. S. Haykin 1996, *Adaptive Filter Theory*, 3rd ed. (Prentice-Hall).
19. J. Mendel 1995, *Lessons in Estimation Theory for Signal Processing, Communications, and Control* (Prentice-Hall: Englewood Cliffs).
20. B. Yang 1995, "Projection approximation subspace tracking," *IEEE Trans. Signal Processing* **43**, 95–107.
21. S. Bannour and M. Azimi-Sadjadi 1995, "Principal component extraction using recursive least squares learning," *IEEE Trans. Neural Networks* **6**, 457–469.
22. W. Kasprzak and A. Cichocki 1996, "Recurrent least squares learning for quasi-parallel principal component analysis," in *Proc. 4th European Symp. Artificial Neural Networks (ESANN'96)*, Bruges, Belgium, April 1996, pp. 223–228.
23. J.-F. Cardoso and B. Laheld 1996, "Equivariant adaptive source separation," *IEEE Trans. Signal Processing* **44**, 3017–3030.
24. A. Cichocki, J. Karhunen, W. Kasprzak and R. Vigario 1996, "On neural blind separation with unequal numbers of sources, sensors, and outputs," submitted manuscript.
25. J. Karhunen, A. Cichocki, W. Kasprzak and P. Pajunen 1997, "On neural blind separation with noise suppression and redundancy reduction," *Int. J. Neural Systems* **8**(2), 219–237.
26. J.-F. Cardoso 1998, "Entropic contrasts for source separation," in *Adaptive Unsupervised Learning*, ed. S. Haykin, Chapter 2.
27. E. Moreau and O. Macchi 1996, "High-order contrasts for self-adaptive source separation," *Int. J. Adaptive Control and Signal Processing* **10**, 19–46.
28. K. Diamantaras and S. Kung 1996, *Principal Component Networks — Theory and Applications* (John Wiley, New York).
29. F.-L. Luo and R. Unbehauen 1997, *Applied Neural Networks for Signal Processing* (Cambridge Univ. Press, Cambridge).
30. L. Xu 1994, "Theories for unsupervised learning: PCA and its nonlinear extensions," in *Proc. IEEE Int. Conf. on Neural Networks (ICNN'94)*, Orlando, Florida, June–July 1994, pp. 1252a–1257.

31. F. Palmieri 1994, "Hebbian learning and self-association in nonlinear neural networks," in *Proc. IEEE Int. Conf. on Neural Networks (ICNN'94)*, Orlando, Florida, June–July 1994, pp. 1258–1263.
32. A. Sudjianto, M. Hassoun and G. Wasserman 1996, "Extensions of principal component analysis for nonlinear feature extraction," in *Proc. 1996 IEEE Int. Conf. on Neural Networks (ICNN'96)*, Washington D.C., USA, June 1996, pp. 1433–1438.
33. R. Hecht-Nielsen 1996, "Data manifolds, natural coordinates, replicator neural networks, and optimal source coding," in *Proc. 1996 Int. Conf. on Neural Information Processing (ICONIP'96)*, Hong Kong, September 1996, pp. 1207–1210.
34. P. Baldi and K. Hornik 1989, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks* **2**, 53–58.
35. F. Palmieri and J. Zhu 1995, "Self-association and Hebbian learning in linear neural networks," *IEEE Trans. Neural Networks* **6**, 1165–1184.
36. J.-F. Cardoso 1997, "Infomax and maximum likelihood for blind source separation," *IEEE Signal Processing Lett.* 4, 112–114.
37. E. Oja 1997, "The Nonlinear PCA learning rule in independent component analysis," *Neurocomputing* **17**(1), 25–46.
38. M. Girolami and C. Fyfe 1998, "Stochastic ICA contrast maximisation using Oja's nonlinear PCA algorithm," *Int. J. Neural Systems*, this issue.
39. R. Lambert 1996, "Multichannel blind deconvolution: FIR matrix algebra and separation of multipath mixtures," Ph.D. dissertation, University of Southern California, Dept. of Electrical Eng., May 1996.
40. J. Karhunen, P. Pajunen and E. Oja 1998, "The nonlinear PCA criterion in blind source separation: relations with other approaches," to appear in *Neurocomputing*.
41. B. Yang 1996, "Asymptotic convergence analysis of the projection approximation subspace tracking algorithms," *Signal Processing* **50**, 123–146.
42. E. Oja 1982, "A simplified neuron model as a principal component analyzer," *J. Math. Biol.* 15, 267–273.
43. T. Kohonen 1989, *Self-Organization and Associative Memory*, 3rd ed. (Springer-Verlag, New York).
44. A. Hyvärinen and E. Oja 1997, "A fast fixed-point algorithm for independent component analysis," *Neural Computation* **9**(7), 1483–1492.
45. J. Karhunen and P. Pajunen 1996, "Hierarchic nonlinear PCA algorithms for neural blind source separation," in *NORSIG 96 Proceedings — 1996 IEEE Nordic Signal Processing Symposium*, pp. 71-74.
46. A. Hyvärinen and E. Oja 1998, "Independent component analysis by general nonlinear Hebbian-like learning rules," in *Signal Processing* **64**(3), 301–313.
47. J. Karhunen and P. Pajunen 1997, "Blind source separation using least-squares type adaptive algorithms," in *Proc. 1997 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'97)*, Munich, Germany, April 1997, pp. 3361–3364.
48. J. Karhunen and P. Pajunen 1997, "Blind source separation and tracking using nonlinear PCA criterion: a least-squares approach," in *Proc. 1997 Int. Conf. on Neural Networks (ICNN'97)*, Houston, Texas, June 1997, pp. 2147–2152.