Neural Approaches to Independent Component Analysis and Source Separation

Juha Karhunen

Helsinki University of Technology
Laboratory of Computer and Information Science
Rakentajanaukio 2 C, FIN-02150 Espoo, Finland
Email: `Juha.Karhunen@hut.fi`

Abstract

Independent Component Analysis (ICA) is a recently developed technique that in many cases characterizes the data in a natural way. The main application area of the linear ICA model is blind source separation. Here, unknown source signals are estimated from their unknown linear mixtures using the strong assumption that the sources are mutually independent. In practice, separation can be achieved by using suitable higher-order statistics or nonlinearities. Various neural approaches have recently been proposed for blind source separation and ICA. In this paper, these approaches and the respective learning algorithms are briefly reviewed, and some extensions of the basic ICA model are discussed.

1. Introduction

A recent trend in neural network research is to study various forms of unsupervised learning beyond standard Principal Component Analysis (PCA). Such techniques are often called nonlinear PCA methods. They can be developed from various starting points, usually leading to different solutions [20]. Independent Component Analysis (ICA) [12, 18] is a useful extension of PCA that has been developed some years ago in context with blind source separation (BSS) problems. In BSS, the goal is to extract independent sources signals from their linear mixtures using a minimum of a priori information. Such blind techniques are needed in several areas. The application of neural BSS approaches have already been considered in communications [6, 14], speech processing [29, 38], and medical signal processing [25] for example.

Roughly speaking, in ICA the data vectors are represented in a linear basis which is determined by requiring that the coefficients of expansion must be mutually independent (or as independent as possible). Therefore, the basis vectors of ICA are generally nonorthogonal, and higher-order statistics are needed in determining the ICA expansion. However, this kind of representation often characterizes the fundamental properties of the data better than standard PCA. For example in blind source separation the ICA expansion leads to separation of the original source signals.

Lately, there has been considerable interest in various neural realizations of ICA and BSS. In these approaches, the higher-order statistics are typically taken into account by using suitable nonlinearities in the learning phase, even though

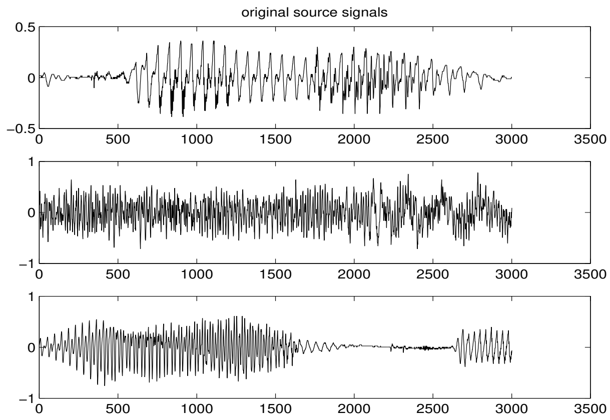

Figure 1: The three original voice sources.

the final input-output mapping is still linear. In this paper, we attempt to give a short tutorial review of some recent neural approaches to ICA and BSS.

2. The data model

The basic data model used in defining both ICA [12] and the BSS problem [18, 24] for linear memoryless channels has the following form [23].

Assume that there exist M zero mean source signals $s_k(1), \ldots, s_k(M)$, $k = 1, 2, \ldots$, that are scalar-valued and mutually statistically independent for each sample value k. An example is given in Figure 1, which shows 3 sampled voice waveforms. They are usually at least approximately independent for different voice sources. We assume that the original sources are unobservable, and all that we have are L possibly noisy but different linear mixtures $x_k(1), \ldots, x_k(L)$ of the sources. Three such mixtures of the voice sources in Fig. 1 are shown in Fig. 2.

Denote by $\mathbf{x}_k = [x_k(1), \ldots, x_k(L)]^T$ the L-dimensional kth data vector made up of the mixtures at discrete time (or point) k. The ICA signal model can then be written in the vector form

$$\mathbf{x}_k = \mathbf{A}\mathbf{s}_k + \mathbf{n}_k = \sum_{i=1}^{M} s_k(i)\mathbf{a}(i) + \mathbf{n}_k. \tag{1}$$

Here $\mathbf{s}_k = [s_k(1), \ldots, s_k(M)]^T$ is the source vector consisting of the M source signals (independent components) $s_k(i)$ $(i = 1, \ldots, M)$ at the index value k. $\mathbf{A}$ $= [\mathbf{a}(1), \ldots, \mathbf{a}(M)]$ is a constant $L \times M$ mixing matrix whose elements are the unknown coefficients of the mixtures. The columns $\mathbf{a}(i)$ are the basis vectors of ICA. The additive noise term $\mathbf{n}_k$ is often omitted from (1), because it is usually impossible to separate it from the source signals.

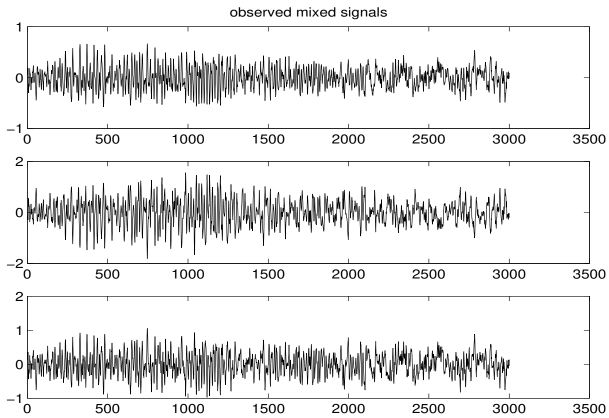

Figure 2: Linear mixtures of the voice sources (input data).

In addition to the independence assumption, we assume that the number of available different mixtures L is at least as large as the number of sources M. Usually M is assumed to be known in advance, and often $M = L$. Furthermore, each source signal $s_k(i)$ is a stationary zero-mean stochastic process. Only one of the source signals $s_k(i)$ is allowed to have a Gaussian distribution. This follows from the fact that it is impossible to separate several Gaussian sources from each other [12, 42].

Note that very little prior information is assumed on the matrix $\mathbf{A}$. Therefore, the strong independence assumptions are required in determining the ICA expansion (1). Even then, only the directions of the ICA basis vectors $\mathbf{a}(i)$, $i = 1, \ldots, M$, are defined, because their magnitudes and the amplitudes of the source signals $s_k(i)$ can be interchanged in the model (1). Also the order of the terms in the sum in (1) can be arbitrary. To get a more unique expansion (1), one can either require that each source $s_k(i)$ has unit variance or normalize the basis vectors $\mathbf{a}(i)$ to unit length (and then arrange them according to the powers of the sources, see [12]).

Linear models of the form (1) are used in several known techniques, but the assumptions are different. In the standard least-squares method, it is assumed that the matrix $\mathbf{A}$ is completely known. Then it is easy to estimate the vector $\mathbf{s}_k$: $\hat{\mathbf{s}}_k = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{x}_k$. If the matrix $\mathbf{A}$ is known except for a few parameters, subspace type methods or the maximum likelihood method can be used for estimating the unknown parameters [37]. In standard PCA, the expansion (1) is determined by requiring that the basis vectors $\mathbf{a}(i)$ are mutually orthonormal and the coefficients $s_k(i)$ have maximal variances.

3. Blind source separation

In blind source separation, the task is to find the waveforms $\{s_k(i)\}$ of the

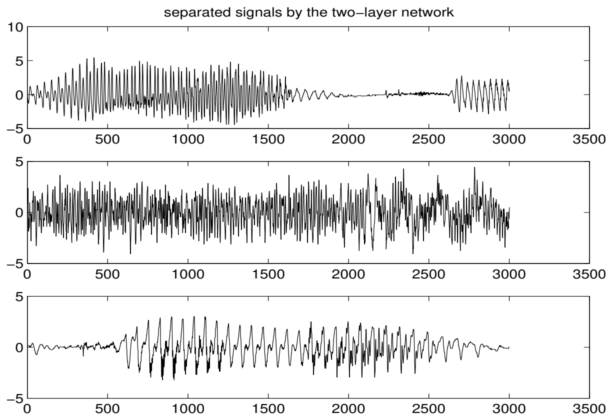

Figure 3: The separated outputs given by the bigradient algorithm.

sources, knowing only the data vectors $\mathbf{x}_k$ and the number M of sources. Both batch type and data-adaptive BSS algorithms have been suggested. In neural realizations, adaptive learning algorithms that are as simple as possible but yet provide sufficient performance are desirable.

In adaptive source separation [6, 18, 24], an $M \times L$ separating matrix $\mathbf{B}_k$ is updated so that the M-vector

$$\mathbf{y}_k = \mathbf{B}_k \mathbf{x}_k \tag{2}$$

is an estimate $\mathbf{y}_k = \hat{\mathbf{s}}_k$ of the original independent source signals. In neural realizations, $\mathbf{y}_k$ is the output vector of the network, and the matrix $\mathbf{B}_k$ is the total weight matrix between the inputs and outputs. The estimate $\hat{s}_k(i)$ of the ith source signal may appear in any component $y_k(j)$ of $\mathbf{y}_k$. The amplitudes of the sources $s_k(i)$ and their estimates $y_k(j)$ are typically scaled so that they have unit variance. Figure 3 shows the separated sources given by the bigradient algorithm (see Section 6) for the voice data of Fig. 2. These can be compared with the original sources in Fig. 1. This experiment is described in more detail in [41].

In several BSS algorithms, the data vectors $\mathbf{x}_k$ are preprocessed by whitening (sphering) them, so that their covariance matrix becomes the unit matrix. After prewhitening, the separating matrix can be assumed orthogonal. Whitening has certain advantages and drawbacks; see Section 5.

A crucial issue in BSS and ICA is reliable verification of the independence condition. It is impossible to do this directly because the involved probability densities are usually unknown. The first proposed separation algorithms are based either on direct minimization of a sum of (typically) fourth order cumulants, or on somewhat heuristic approaches where the learning algorithms have such a form that they satisfy some kind of independence condition after convergence; see [6, 8, 12] for references.

A mathematically more exact procedure [12], is to measure the degree of dependence using mutual information. This is minimized when the output components (estimated sources) are mutually independent. Approximating the mutual information using Edgeworth expansion leads to contrast functions which are maximized by separating matrices [12]. Even these contrast functions require fairly intensive batch type computations using the estimated higher-order statistics of the data, or lead to pretty complicated adaptive separation algorithms.

Fortunately, it is often sufficient to use the simple higher-order statistics called kurtosis. For the ith source signal $s(i)$, the (unnormalized) kurtosis is defined by

$$\mathrm{cum}[s(i)^4] = \mathrm{E}\{s(i)^4\} - 3[\mathrm{E}\{s(i)^2\}]^2. \tag{3}$$

If $s(i)$ is Gaussian, its kurtosis $\mathrm{cum}[s(i)^4] = 0$. Source signals that have a negative kurtosis are often called sub-Gaussian ones. Typically, their probability distribution is "flatter" than Gaussian, for example bimodal [16]. Sources with a positive kurtosis (super-Gaussian sources) have usually a distribution which has a longer tail and sharper peak than standard Gaussian distribution [3, 16].

The division of sources into sub-Gaussian and super-Gaussian ones is important, because the separation capability of many simple algorithms depends on this. Consider the situation where the sign of the kurtosis (3) is the same for all the sources $s_k(i)$, $i = 1, \ldots, M$, and the input vectors have been prewhitened. In [28] it is proved that one can then use a particularly simple contrast function, the sum of the fourth moments

$$J(\mathbf{y}) = \sum_{i=1}^{M} \mathrm{E}\{y(i)^4\}. \tag{4}$$

A separating matrix $\mathbf{B}$ minimizes (4) for sub-Gaussian sources, and maximizes it for super-Gaussian sources. We have used the criterion (4) because it is simple enough, and can be applied in a straightforward way to our nonlinear PCA type neural learning algorithms [22, 23, 40]. In many practical situations the sources are either sub-Gaussian or super-Gaussian. For example speech signals are typically super-Gaussian [3].

Recently, Bell and Sejnowski [3] have suggested minimization of the mutual information in another way. They maximize the joint entropy of the outputs of a neural network, and derive an explicit learning rule to this end. A similar approach is followed in [2], where the Gram-Charlier expansion is used instead of the Edgeworth expansion in approximating the mutual information. These neural learning rules will be discussed in more detail later on.

4. Neural network models

Consider now neural estimation of the complete ICA expansion (1). Let us denote the estimated expansion by

$$\mathbf{x}_k = \mathbf{Q}\mathbf{y}_k + \mathbf{n}'_k = \hat{\mathbf{x}}_k + \mathbf{n}'_k. \tag{5}$$

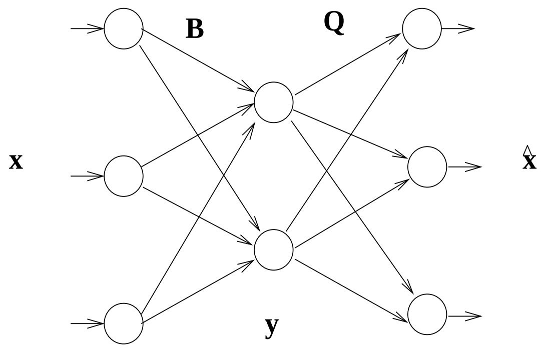

Figure 4: The basic ICA network structure.

Here, the $L \times M$ matrix $\mathbf{Q}$ denotes the estimate of the ICA basis matrix $\mathbf{A}$, $\mathbf{y}_k$ is the estimate of the source (or independent component) vector $\mathbf{s}_k$, and $\mathbf{n}'_k$ is the noise term.

For estimation, we use the 2-layer feedforward network shown in Figure 4. The L inputs of the network are the components of the vector $\mathbf{x}$ (at discrete sample index k). In the hidden layer there are M neurons, and the output layer consists again of L neurons. $\mathbf{B}$ denotes the $M \times L$ separating weight matrix between the inputs and the hidden layer, and $\mathbf{Q}$ respectively the $L \times M$ weight matrix between the hidden and output layers. The ICA expansion (1) can be estimated using the network of Fig. 4 in two subsequent stages as follows:

1. Learn an $M \times L$ separating weight matrix $\mathbf{B}$ for which the components of $\mathbf{y} = \mathbf{Bx}$ are as independent as possible;

2. Learn an $L \times M$ weight matrix $\mathbf{Q}$ which minimizes the mean-square error $\mathrm{E}\{\| \mathbf{n}'_k \|^2\} = \mathrm{E}\{\| \mathbf{x}_k - \mathbf{Q}\mathbf{y}_k \|^2\}$ with respect to $\mathbf{Q}$.

This network structure is justified in more detail in [21, 23]. In BSS the basis vectors of ICA are not of much interest, and the last layer is usually omitted from the network of Fig. 4.

If prewhitening is used, the first stage is further divided into two subsequent parts. First, the data (input) vectors $\mathbf{x}_k$ are whitened by applying the transformation

$$\mathbf{v}_k = \mathbf{V}\mathbf{x}_k, \tag{6}$$

where $\mathbf{v}_k$ denotes the kth whitened vector, and $\mathbf{V}$ is an $M \times L$ whitening matrix. If $L > M$, $\mathbf{V}$ simultaneously reduces the dimension of the data vectors from L to M. After this, the sources (independent components) are separated:

$$\mathbf{y}_k = \mathbf{W}^T\mathbf{v}_k. \tag{7}$$

Here $\mathbf{W}^T$ denotes for clarity the orthonormal ($\mathbf{W}^T\mathbf{W} = \mathbf{I}_M$) $M \times M$ separating matrix that the network should learn. Figure 5 shows the ensuing 2-layer source

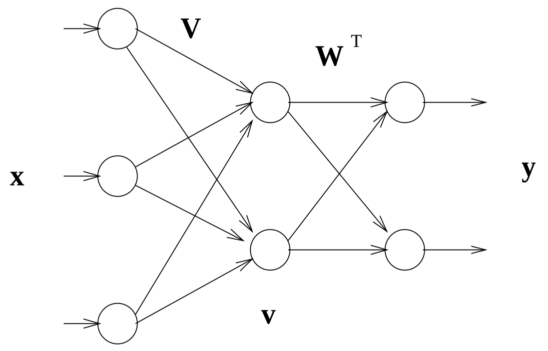

Figure 5: The two-layer network structure used in source separation.

separation network structure, where now $\mathbf{B} = \mathbf{W}^T\mathbf{V}$. If the basis vectors of ICA are needed, an extra layer with the weight matrix $\mathbf{Q}$ can be appended to this network quite similarly as in the network of Fig. 4.

In the ICA/BSS networks of Figs. 4 and 5, the number of sources M is often equal to L, the dimension of the input vectors. In this case, no data compression takes place in the hidden layer, but the independence constraint anyway provides an ICA solution. As usual, feedback connections (not shown) are needed in the learning phase, but after learning these networks become purely feedforward if the data is stationary. Even though the input-output mappings of the proposed ICA networks are linear after learning due to the linear data model (1), nonlinearities must be used in learning the separating matrix $\mathbf{B}$ or $\mathbf{W}^T$. They introduce the necessary higher-order statistics into computations.

Feedforward network structures are currently popular in source separation, but they are not the only possibility. In [1, 18, 26] recurrent neural network structures have been studied in the BSS problem. They may have some advantages over feedforward networks in hardware implementation [1]. Recently, feedforward network structures containing several subsequent separation layers have been proposed in [10]. Such networks seem to allow separation of sources in difficult cases (weak sources or ill-conditioned problems) provided that the data vectors $\mathbf{x}_k$ do not contain noise and obey the ICA model (1) exactly.

5. Whitening

If the data vectors $\mathbf{x}_k$ have a nonzero mean, it is usually first subtracted from them. Furthermore, the effects of second-order statistics can be removed by using the whitening transformation (6). The matrix $\mathbf{V}$ is chosen so that the covariance matrix $\{\mathbf{v}_k\mathbf{v}_k^T\}$ becomes the unit matrix $\mathbf{I}_M$. Thus the components of the whitened vectors $\mathbf{v}_k$ are mutually uncorrelated and they have unit variance. Uncorrelatedness is a necessary prerequisite for the stronger independence condition; so after prewhitening the separation task usually becomes somewhat easier.

There exist infinitely many solutions for whitening the input data (provided that $L \geq M$).

Standard PCA is often used for whitening, because one can then simultaneously compress information optimally in the mean-square error sense and filter possible noise [23, 19]. The PCA whitening matrix is given by

$$\mathbf{V} = \mathbf{D}^{-1/2}\mathbf{E}^T. \tag{8}$$

In (8), $\mathbf{D} = \mathrm{diag}[\lambda(1), \ldots, \lambda(M)]$ is $M \times M$ diagonal matrix , and the $L \times M$ matrix $\mathbf{E} = [\mathbf{c}(1), \ldots, \mathbf{c}(M)]$, where $\lambda(i)$ is the ith largest eigenvalue of the data covariance matrix $\mathrm{E}\{\mathbf{x}_k\mathbf{x}_k^T\}$, and $\mathbf{c}(i)$ denotes the respective ith principal eigenvector. PCA whitening can be done either using standard software or neurally [23]. Furthermore, it can be used for estimating the number of the sources or independent components (see Section 8).

Probably the simplest neural algorithm for learning the whitening matrix $\mathbf{V}_k$ is

$$\mathbf{V}_{k+1} = \quad \mathbf{V}_k - \mu_k[\mathbf{v}_k\mathbf{v}_k^T - \mathbf{I}]\mathbf{V}_k. \tag{9}$$

This has been independently proposed in [24, 35], and is utilized as a part of the EASI (PFS) separation algorithm [6, 24]. The algorithm (9) can be applied also to simultaneous data compression with $M < L$, but it does not have any optimality properties in this respect.

Whitening and related procedures have been sometimes criticized [7], because they do not provide so-called uniform performance in subsequent separation. In uniform performance methods, the separation capability does not depend on the condition number of the mixing matrix $\mathbf{A}$. In theory, it is then possible to separate even very weak sources or use almost similar mixtures as inputs [7, 10]. However, this property presumes that the input data obeys the ICA model (1) exactly with no noise. In our experiments, separation algorithms that require prewhitening usually performed quite well in normal conditions, for example when the mixing matrix $\mathbf{A}$ was chosen randomly [22, 23].

6. Neural separation algorithms

During the last years, various neural algorithms have been proposed for learning either the total separating matrix $\mathbf{B}$ directly or the orthogonal separating matrix $\mathbf{W}^T$ after prewhitening. In the direct approach, the output vector is defined by $\mathbf{y}_k = \mathbf{B}_k\mathbf{x}_k$; in the prewhitening approach the relationships (6) and (7) are used. In the following, we list some relevant possibilities. The learning parameter μ_k in the algorithms below is usually positive.

1. The Herault-Jutten (HJ) algorithm [18, 8]. This seminal neural separation algorithm has inspired a lot of later work. In its basic form, the separating matrix $\mathbf{B}$ is sought in the form $\mathbf{B} = (\mathbf{I}+\mathbf{S})^{-1}$, and the off-diagonal elements of $\mathbf{S}$ are updated using the rule

$$\mathbf{S}_{k+1} = \quad \mathbf{S}_k + \mu_k\mathbf{g}(\mathbf{y}_k)\mathbf{h}(\mathbf{y}_k^T). \tag{10}$$

The diagonal elements of $\mathbf{S}$ are zero, and $\mathbf{g}(\mathbf{y}_k)$ denotes the column vector whose components are $g(y_k(1)), \ldots, g(y_k(M))$. Similarly, $\mathbf{h}(\mathbf{y}_k^T)$ is a row vector with components $h(y_k(i))$. Here, $g(t)$ and $h(t)$ are two different odd functions, and the learning parameter $\mu_k > 0$.

The basic HJ algorithm is simple and local, but may fail in separating more than two sources. Various odd functions $g(t)$ and $h(t)$ such as t, t^3, $\mathrm{sgn}(t)$, and $\tanh(t)$ have been used in (10). In [14], the choices $g(t) = t^3$, $h(t) = t$ are recommended for sub-Gaussian sources, and respectively $g(t) = t$, $h(t) = t^3$ for super-Gaussian sources. The HJ algorithm is derived and discussed in the papers [18, 11, 36]. Its convergence properties have been studied in several papers, and it can be realized using either feedback or feedforward type architectures [27]. The algorithm has been extended for convolutive mixtures (time delays) in [34, 29].

Especially in discrete realization, the separating matrix is often computed from the approximation

$$\mathbf{B}_{k+1} = \mathbf{I} - \mathbf{S}_{k+1} \tag{11}$$

which avoids the matrix inversion. In practice, the approximative algorithm performs similarly or sometimes even better than the original HJ algorithm. Cichocki [9, 10] has recently proposed various improved versions of the basic HJ algorithm, where also the diagonal elements are updated.

2. The EASI (or PFS) algorithm. This has been introduced and justified as an adaptive signal processing algorithm in [6, 24], but it can as well be used as a learning algorithm of a nonlinear PCA type network. The general update formula for the separating matrix $\mathbf{B}$ is

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \mu_k[\mathbf{y}_k\mathbf{y}_k^T - \mathbf{I} + \mathbf{g}(\mathbf{y}_k)\mathbf{h}(\mathbf{y}_k^T) - \mathbf{h}(\mathbf{y}_k)\mathbf{g}(\mathbf{y}_k^T)]\mathbf{B}_k \tag{12}$$

In the original EASI algorithm $h(t) = t$; the learning rule (12) is a generalized form introduced in [22]. If the functions $g(t)$ or $h(t)$ grow faster than linearly, the algorithm (12) should be stabilized in practice; see [6, 22, 24]. The generalized form has the advantage that by using for example the functions $g(t) = t$, $h(t) = \tanh(t)$ for sub-Gaussian sources and $g(t) = \tanh(t)$, $h(t) = t$ for super-Gaussian sources, separation can usually be achieved without the extra stabilization. An advantage of (12) is that it provides uniform performance [6, 24].

3. Bell's and Sejnowski's algorithm. This is derived in the insightful paper [3] using an information theoretic approach. The learning algorithm for the separating matrix $\mathbf{B}$ is using our notation

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \mu_k[\mathbf{B}_k^{-T} + \mathbf{z}_k\mathbf{x}_k^T] \tag{13}$$

Here $\mathbf{B}_k$ must be a square matrix ($M = L$), and the i:th element of the column vector $\mathbf{z}$ is obtained from the formula [3, 4]

$$z(i) = \frac{\partial}{\partial u(i)}\frac{\partial u(i)}{\partial y(i)} \tag{14}$$

where $u(i)$ is the i:th element of the vector $\mathbf{u} = \mathbf{f}(\mathbf{y}) = \mathbf{f}(\mathbf{Bx})$, and $f(t)$ is usually some sigmoidal function. If for example $\mathbf{u} = \tanh(\mathbf{y})$, $\mathbf{z} = -2\tanh(\mathbf{y})$. The algorithm (13) can easily be generalized for mixtures having a nonzero mean, blind deconvolution, etc. [3]. In [4] it is justified that the optimal functions $f(t)$ are the cumulative distribution functions of the sources (if known). In practice, many other choices provide separation.

It is easy to get rid of the annoying inverse matrix $\mathbf{B}_k^{-T}$ in the original algorithm (13). The first possibility is to prewhiten the data [41, 4]; then $\mathbf{B}_k^{-T} = \mathbf{B}_k$ (or actually $\mathbf{W}_k^{-1} = \mathbf{W}_k^T$ using our notation). Even better[1], one can use the so-called natural gradient approach [2] where essentially the update term in (13) is multiplied by $\mathbf{B}_k^T\mathbf{B}_k$. This yields the modified algorithm

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \mu_k[\mathbf{I} - 2\mathbf{g}(\mathbf{y}_k)\mathbf{y}_k^T]\mathbf{B}_k \tag{15}$$

where usually the function $g(t) = \tanh(t)$. Both whitening and the modification (15) often increase the convergence speed of Bell's and Sejnowski's algorithm (13) by orders of magnitude [41, 4] and allow a truly neural realization. Furthermore, (15) provides a uniform performance and needs not prewhitening.

The form of the algorithm (15) (without the constant 2) is derived in [2] by extending and combining the ideas in [3, 12]. Furthermore, approximation of the mutual information using a Gram-Charlier expansion yields a specific "universal" expression for the nonlinear function $g(t)$ [2].

4. Cichocki and Amari with their co-workers have recently proposed a variety of new separation algorithms in [9, 10, 1]. For achieving separation, different matrix functions $\mathbf{G}(\mathbf{y})$ depending on the output vector $\mathbf{y}$ can be defined, such as

$$\mathbf{G}(\mathbf{y}) = \mathbf{I} - \mathbf{g}(\mathbf{y})\mathbf{h}(\mathbf{y}^T), \tag{16}$$

$$\mathbf{G}(\mathbf{y}) = \mathbf{I} - \mathbf{y}\mathbf{y}^T - \mathbf{g}(\mathbf{y})\mathbf{h}(\mathbf{y}^T) + \mathbf{h}(\mathbf{y})\mathbf{g}(\mathbf{y}^T), \tag{17}$$

$$\mathbf{G}(\mathbf{y}) = \mathbf{I} - \sum_{i=0}^{p} \mathbf{y}_k\mathbf{y}_{k-iT}^T. \tag{18}$$

In the last expression, T is a suitably chosen time delay. Each of these choices can be used for learning the separating matrix $\mathbf{B}$ in the general algorithm

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \mu_k\mathbf{G}(\mathbf{y}_k) \tag{19}$$

for feedforward networks [10]. The first choice (16) yields a modified version of the HJ algorithm, and the second matrix (17) resembles closely the update term in the generalized EASI algorithm (12).

The same matrices $\mathbf{G}(\mathbf{y})$ can be used also in learning the separating matrix in recurrent networks [9, 1]. The learning algorithm is

$$\mathbf{B}'_{k+1} = \mathbf{B}'_k - \mu_k(\mathbf{B}'_k + \mathbf{I})\mathbf{G}(\mathbf{y}_k), \tag{20}$$

[1] This part is largely based on comments provided by Dr. Kari Torkkola

where the output vector $\mathbf{y}_k$ is now computed from the formula

$$\mathbf{y}_k = [\mathbf{I} + \mathbf{B}'_k]^{-1}\mathbf{x}_k. \tag{21}$$

The computation of the inverse matrix can be avoided; see [1]. The properties of the algorithm (20) have not yet been thoroughly investigated.

5. The bigradient algorithm [39, 40, 41]. This learning algorithm for the orthogonal separating matrix $\mathbf{W}$ after prewhitening reads

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k \mathbf{v}_k \mathbf{g}(\mathbf{y}_k^T) + \gamma_k \mathbf{W}_k(\mathbf{I} - \mathbf{W}_k^T \mathbf{W}_k). \tag{22}$$

In (22), the learning parameter μ_k can be either positive or negative, and γ_k is another positive learning parameter, usually about 0.5 or 1 in practice. The first update term $\mu_k \mathbf{v}_k \mathbf{g}(\mathbf{y}_k^T)$ is essentially a nonlinear Hebbian term, and the second term $\gamma_k \mathbf{W}_k(\mathbf{I} - \mathbf{W}_k^T \mathbf{W}_k)$ keeps the weight matrix $\mathbf{W}_k$ roughly orthonormal. The bigradient algorithm is derived and analyzed in [39, 40]. One of its best features is flexibility. The algorithm (22) can be applied with slightly different forms and choices to separating either sub-Gaussian or super-Gaussian sources, but also to standard PCA and MCA (Minor Component Analysis). It is also easy to modify the algorithm (22) so that the weight vectors of the neurons (columns of the matrix $\mathbf{W}_k$) are computed sequentially in a hierarchic order; see [39].

6. Nonlinear PCA subspace rule [31, 22]. This learning algorithm was originally introduced by Oja some years ago [30] as an extension of his well-known PCA subspace rule; see [20] for early references. In BSS, it is used quite similarly as (22), but the update formula for $\mathbf{W}$ is different:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k[\mathbf{v}_k - \mathbf{W}_k \mathbf{g}(\mathbf{y}_k)]\mathbf{g}(\mathbf{y}_k^T). \tag{23}$$

Here the learning parameter μ_k must be positive for stability reasons, restricting the applicability of (23) mainly to sub-Gaussian sources. Without prewhitening, the nonlinear PCA subspace rule is able to somehow separate sinusoidal type sources only, because the algorithm responds in this case still largely to the second-order statistics.

A major advantage of the learning rule (23) is that it can be realized using a simple modification of one-layer standard symmetric PCA network [30, 20], allowing a simple and local neural implementation. An interesting feature is that the underlying data model is actually slightly nonlinear [20]: the coefficients (here sources) $s_k(i)$ in (1) are replaced by nonlinear coefficients $g(s_k(i))$. Nonlinear PCA subspace rule (23) can then be derived by approximately minimizing the mean-square error $\mathrm{E}\{\| \mathbf{n}_k \|^2\}$. In spite of this, the algorithm performs well for sub-Gaussian sources even in large problems [22, 23]. The separation properties of (23) have been analyzed mathematically in detail in [31, 23], where local convergence of the algorithm to a separating solution is shown in certain cases.

The list above is by no means exhaustive. For example, we have made some preliminary experiments showing that the hierarchic versions of the nonlinear and robust PCA subspace rule (called nonlinear or robust GHA algorithms) [20] as well as the hierarchic bigradient rule [39] work well with prewhitening in separation with suitable choices. Many of the proposed algorithms have not been analyzed mathematically, and their properties are still largely unexplored. No extensive comparisons have been made. Thus it is difficult to say much about the superiority of these algorithms. In several cases, rather similar stochastic nonlinear Hebbian type terms are effectively used in learning, suggesting that the final performance given by various algorithms is often almost the same. This was observed in our simulations [22, 23].

7. Estimation of the basis vectors of ICA

The basis vectors $\mathbf{a}(1), \ldots, \mathbf{a}(M)$ of the linear ICA model (1) are the counterparts of the principal eigenvectors in PCA. Therefore, they should be useful in much the same applications [12], providing in many cases a more meaningful characterization of the data. There exist several methods for estimating them; see [23, 21] for a more detailed discussion.

Assuming that the $M \times L$ matrix $\mathbf{B}_k$ has converged to a separating solution $\mathbf{B}$, the basis vectors $\mathbf{a}(i)$ can be estimated from the pseudoinverse $\hat{\mathbf{A}} = \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}$. The columns of $\hat{\mathbf{A}}$ are the desired estimated basis vectors of ICA. If $\mathbf{B}$ is a square matrix, $\hat{\mathbf{A}} = \mathbf{B}^{-1}$. Even though this method is simple, it is not feasible for a truly neural realization because of the required matrix inversion.

Alternatively, the basis vectors can be estimated neurally using the extra $\mathbf{Q}$ layer shown in Fig. 4. In [21], we have derived the stochastic gradient algorithm

$$\mathbf{Q}_{k+1} = \mathbf{Q}_k + \mu_k(\mathbf{x}_k - \mathbf{Q}_k\mathbf{y}_k)\mathbf{y}_k^T \tag{24}$$

($\mu_k > 0$) for learning the $L \times M$ weight matrix $\mathbf{Q}_k$ of this additional layer. The algorithm (24) is based on the minimization of the mean-square representation error $\mathrm{E}\{\| \mathbf{x}_k - \mathbf{Q}\mathbf{y}_k \|^2\}$ (see Section 4). Again, the columns of the $\mathbf{Q}_k$ constitute the estimates of the basis vectors of ICA after convergence (in any order). The algorithm (24) can be used in context with any adaptive separation algorithm, and it seems to perform well in practice. It would be rather straightforward to design more accurate and faster converging but more complicated algorithms for learning the matrix $\mathbf{Q}$ using the same MSE error minimization approach.

8. Extensions of the basic models

The standard problem considered in most neural network papers dealing with the ICA model (1) is blind source separation, assuming that there is no noise and the number M of sources is known. In this section, we discuss various extensions of the basic model and assumptions made on it in Section 2. Estimation of the basis vectors of ICA discussed before is already one such extension.

1. Nonstationary data. The basis vectors $\mathbf{a}(i)$ of ICA (mixture coefficients)

in (1) are usually taken as constants, and the sources $s_k(i)$ are assumed stationary. In practice, it is important that a separation algorithm could adapt to (slow) changes in the statistics of the sources and mixture coefficients. Only a little has been published on this problem, perhaps because it is generally difficult. A neural network for the nonstationary case is proposed in [26]; however, only the sources $s_k(i)$ are assumed nonstationary, and the simulations have been made using cyclostationary rather than truly nonstationary sources. In [29], some experimental results are given for nonstationary speech data.

2. The effect of noise. In most papers, it is assumed that the noise term $\mathbf{n}_k$ in (1) is zero, and sometimes noise is regarded as an extra source. Generally, this is not the case, which can be seen by expressing the noise vector $\mathbf{n}_k$ in the form

$$\mathbf{n}_k = \sum_{i=1}^{M} n_k(i)\mathbf{a}(i) + \mathbf{e}_k. \tag{25}$$

Here $n_k(i)$ is the projection of $\mathbf{n}_k$ onto $\mathbf{a}(i)$, and $\mathbf{e}_k$ is the part of the noise vector $\mathbf{n}_k$ lying in the subspace orthogonal to the basis vectors $\mathbf{a}(1), \ldots, \mathbf{a}(M)$. Inserting (25) into the ICA model (1) shows that noise adds a component $n_k(i)$ to each source. Experimental results confirm this: noise smears the separated sources. If the number L of mixtures is greater than the number M of sources, it is possible to filter some of the noise out. This can be done by projecting the input data $\mathbf{x}_k$ onto its M-dimensional signal subspace using for example PCA whitening; see [23, 37].

3. Estimation of the number M of the sources (independent components). Usually M is assumed known, which is often not the case in practice, especially if one wants to use ICA as a data analysis tool. If the signal-to-noise ratio is good enough, it is relatively easy to estimate the number of sources M using standard PCA [23, 19]. In practice, this is done by first estimating the data covariance matrix $\mathrm{E}\{\mathbf{x}_k\mathbf{x}_k^T\}$ from the data vectors $\mathbf{x}_k$ and then computing its eigenvalues. The M largest "signal" eigenvalues should be clearly larger than the rest "noise" eigenvalues [37]. From this, one can deduce the number M of the sources. If some of the sources are weak or the power of the noise is not small, it is difficult to estimate the correct value of M using this simple method. Similar model order estimation problems are encountered elsewhere, for example in sinusoidal frequency estimation. It is still an open question whether the more advanced methods developed there could be applied also to the source separation problem on some conditions.

4. More/less mixtures than sources. Often it is assumed that the number L of mixtures (the dimension of the data vectors $\mathbf{x}_k$) is equal to the number M of sources. As discussed before, several separation algorithms can directly handle the case $L > M$ (more mixtures than sources). If not, it is always

possible to linearly compress the dimensionality of the data vectors from L to M using for example standard PCA (provided that M is known).

The interesting case where there are less mixtures than sources ($L < M$) is studied theoretically in [42]. The result is that it is still often possible to separate the sources into L distinct groups. This phenomenon has been experimentally confirmed in our experiments with the nonlinear PCA algorithm (23). In the case $L < M$, the outputs were either some almost pure sources or linear combinations of some of them. In many potential applications, there are not available several different linear mixtures of the source signals. This fact greatly limits the number of practical problems to which ICA and BSS are applicable.

5. Nonlinear ICA and source separation. A natural extension of linear ICA and BSS is to assume that the components of the data vectors $\mathbf{x}_k$ depend nonlinearly on some statistically independent components (source signals). This important but difficult problem has been first addressed in [5], and later on in a series of papers [13, 32, 33] by Deco and Parra with their co-workers. The approaches proposed thus far seem to be computationally fairly complicated or require explicit estimation of higher-order statistics. Simulations have been made in small-dimensional cases only (typically $M = 2$).

6. Delays and convolutive mixtures. A problem related to blind source separation is blind deconvolution [17], where the task is to recover an unknown source signal mixed with unknown time-delayed versions of itself. A combination of both these problems is called blind identification [3], or separation of convolutive mixtures [29]. There we have generally L unknown linear mixtures of M independent source signals having different time delays in each mixture. This situation can be modeled by replacing the elements of the mixing matrix $\mathbf{A}$ in (1) by filters (typically FIR filters). The blind identification problem naturally arises for example in practical speech separation: the delays of each speaker are different at different microphones. This extension has been studied at least for the HJ algorithm in [34, 29], and for Bell's and Sejnowski's algorithm in [3, 38]. If the source signals are generated by technical systems (for example in communications), it may be possible to handle time delays using synchronisation sequences [14].

7. Use of prior information [19]. If there are available some additional information on the sources, this can be often utilized in improving the separation results or in devising simpler algorithms. In particular, when the sources are temporally correlated, separation may be based on second-order statistics only, and is possible also for Gaussian sources. If the source signals are discrete valued, typically binary, simpler separation approaches exist. If the distributions of the sources are known, one can devise optimal nonlinearities [4] or contrast functions. A good recent review of various approaches, mostly nonneural, that utilize prior information is [19].

8. Application of ICA to other problems than BSS. The ICA model should be useful in much the same applications [12] as standard PCA, which is widely applied to various signal and information processing tasks. However, almost all the papers on ICA are related to the BSS problem. In particular, the basis vectors of ICA provide a good description of the data. They are useful in finding interesting directions in the data for example in projection pursuit [15]. Fyfe and Baddeley [16] have already considered this application without showing an explicit connection to ICA. In [23] we have pointed out that their projection pursuit directions are in fact basis vectors of ICA.

9. Concluding remarks

Blind signal processing is a rapidly emerging, promising new application area of unsupervised neural learning. In this paper, we have attempted to give a tutorial review of some neural approaches to blind source separation and to the closely related Independent Component Analysis model. Many new learning algorithms have been recently proposed. Their theoretical properties, range of applicability, and mutual comparisons are still largely unexplored. Another important task is to design methods which extend the simple basic linear model to cope with realistic practical situations.

Acknowledgements. The author is grateful to Prof. Erkki Oja and Dr. Kari Torkkola for useful discussions and comments, and to Liuyue Wang for providing the figures in the paper.

References

[1] S. Amari, A. Cichocki, and H. Yang, "Recurrent neural networks for blind separation of sources," in *Proc. 1995 Int. Symp. on Nonlinear Theory and Applications (NOLTA-95)*, Las Vegas, USA, December 1995, vol. 1, pp. 37-42.

[2] S. Amari, A. Cichocki, and H. Yang, "A new learning algorithm for blind signal separation," to appear in *Advances in Neural Information Processing Systems 8 (Proc. NIPS'95).* Cambridge, MA: MIT Press, 1996.

[3] A. Bell and T. Sejnowski, "An information-maximisation approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129-1159, 1995.

[4] A. Bell and T. Sejnowski, "Fast blind separation based on information theory," in *Proc. 1995 Int. Symp. on Nonlinear Theory and Applications (NOLTA-95)*, Las Vegas, USA, December 1995, vol. 1, pp. 43-47.

[5] G. Burel, "Blind separation of sources: a nonlinear neural algorithm," *Neural Networks*, vol. 5, pp. 937-947, 1992.

[6] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," to appear in *IEEE Trans. on Signal Processing.*

[7] J.-F. Cardoso, "The invariant approach to source separation," in *Proc. 1995 Int. Symp. on Nonlinear Theory and Applications (NOLTA-95)*, Las Vegas, USA, December 1995, vol. 1, pp. 55-60.

[8] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing, 2nd ed.* New York: John Wiley, 1994.

[9] A. Cichocki and R. Unbehauen, "Robust neural networks with on-line learning for blind identification and blind separation of sources," submitted to *IEEE Trans. on Circuits and Systems*, June 1994.

[10] A. Cichocki, W. Kasprzak, and S. Amari, "Multi-layer neural networks with a local adaptive learning rule for blind separation of source signals," in *Proc. 1995 Int. Symp. on Nonlinear Theory and Applications (NOLTA-95)*, Las Vegas, USA, December 1995, vol. 1, pp. 61-66.

[11] P. Comon, C. Jutten, and J. Herault, "Blind separation of sources, part II: problems statement," *Signal Processing*, vol. 24, no. 1, pp. 11-20, July 1991.

[12] P. Comon, "Independent component analysis - a new concept?," *Signal Processing*, vol. 36, pp. 287-314, 1994.

[13] G. Deco and W. Brauer, "Nonlinear higher-order statistical decorrelation by volume-conserving neural architectures," *Neural Networks*, vol. 8, no. 4, pp. 525-535, 1995.

[14] Y. Deville and L. Andry, "Application of blind source separation techniques to multi-tag contactless identification systems," in *Proc. 1995 Int. Symp. on Nonlinear Theory and Applications (NOLTA-95)*, Las Vegas, USA, December 1995, vol. 1, pp. 73-78.

[15] J. Friedman, "Exploratory projection pursuit," *J. American Statistical Association*, vol. 82, no. 397, pp. 249-266, March 1987.

[16] C. Fyfe and R. Baddeley, "Non-linear data structure extraction using simple hebbian networks," *Biological Cybernetics*, vol. 72, pp. 533-541, 1995.

[17] S. Haykin, *Adaptive Filter Theory, 3rd ed.*. New Jersey: Prentice-Hall, 1996.

[18] C. Jutten and J. Herault, "Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, no. 1, pp. 1-10, July 1991.

[19] C. Jutten and J.-F. Cardoso, "Separation of sources: really blind," in *Proc. 1995 Int. Symp. on Nonlinear Theory and Applications (NOLTA-95)*, Las Vegas, USA, December 1995, vol. 1, pp. 79-84.

[20] J. Karhunen and J. Joutsensalo, "Generalizations of principal component analysis, optimization problems, and neural networks," *Neural Networks*, vol. 8, no. 4, pp. 549-562, 1995.

[21] J. Karhunen, L. Wang, and J. Joutsensalo, "Neural estimation of basis vectors in independent component analysis," in *Proc. Int. Conf. on Artificial Neural Networks (ICANN'95)*, Paris, France, October 1995, pp. 317-322.

[22] J. Karhunen, L. Wang, and R. Vigario, "Nonlinear PCA type approaches for source separation and independent component analysis," in *Proc. 1995 IEEE Int. Conf. on Neural Networks*, Perth, Australia, November 1995, pp. 995-1000.

[23] J. Karhunen, E. Oja, L. Wang, R. Vigario, and J. Joutsensalo, "A class of neural networks for independent component analysis," Lab. of Computer and Information Science, Helsinki Univ. of Technology (Espoo, Finland), Technical Report A28, October 1995. Submitted to a journal.

[24] B. Laheld and J.-F. Cardoso, "Adaptive source separation with uniform performance," in *Signal Processing VII: Theories and Applications (Proc. EUSIPCO-94)*, M. Holt et al. (Eds.). Lausanne: EURASIP, 1994, vol. 2, pp. 183-186.

[25] S. Makeig, A. Bell, T.-P. Jung, and T. Sejnowski, "Independent component analysis of electroencephalographic data," to appear in *Advances in Neural Information Processing Systems 8 (Proc. NIPS'95).* Cambridge, MA: MIT Press, 1996.

[26] K. Matsuoka, M. Ohya, and M. Kawamoto, "A neural net for blind separation of nonstationary signals," *Neural Networks*, vol. 8, no. 3, pp. 411-419, 1995.

[27] E. Moreau and O. Macchi, "Two novel architectures for the self adaptive separation of signals," in *Proc. IEEE Int. Conf. on Communications*, Geneva, Switzerland, May 1993, pp. 1154-1159.

[28] E. Moreau and O. Macchi, "New self-adaptive algorithms for source separation based on contrast functions," in *Proc. IEEE Signal Proc. Workshop on Higher Order Statistics*, Lake Tahoe, USA, June 1993, pp. 215-219.

[29] H.-L. Nguyen Thi and C. Jutten, "Blind source separation for convolutive mixtures," *Signal Processing*, vol. 45, pp. 209-229, 1995.

[30] E. Oja, H. Ogawa, and J. Wangviwattana, "Learning in nonlinear constrained Hebbian networks," in *Artificial Neural Networks (Proc. ICANN-91)*, T. Kohonen et al. (Eds.). Amsterdam: North-Holland, 1991, pp. 385-390.

[31] E. Oja, "The Nonlinear PCA learning rule and signal separation – mathematical analysis," Lab. of Computer and Information Science, Helsinki Univ. of Technology (Espoo, Finland), Technical Report A26, August 1995. Submitted to a journal.

[32] L. Parra, G. Deco, and S. Miesbach, "Redundancy reduction with information-preserving nonlinear maps," *Network*, vol. 6, pp. 61-72, 1995.

[33] L. Parra, "Symplectic nonlinear component analysis," to appear in *Advances in Neural Information Processing Systems 8 (Proc. NIPS'95).* Cambridge, MA: MIT Press, 1996.

[34] J. Platt and F. Faggin, "Networks for the separation of sources that are superimposed and delayed," in *Advances in Neural Information Processing Systems 4*, J. Moody et al. (Eds.). San Mateo, California: Morgan Kaufmann, 1991, pp. 730-737.

[35] M. Plumbley, "A Hebbian/anti-Hebbian network which optimizes information capacity by orthonormalizing the principal subspace," in Proc. IEE Conf. on Artificial Neural Networks, Brighton, UK, May 1993, pp. 86-90.

[36] E. Sorouchyari, "Blind separation of sources, part III: stability analysis," *Signal Processing*, vol. 24, no. 1, pp. 21-29, July 1991.

[37] C. Therrien, *Discrete Random Signals and Statistical Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1992.

[38] K. Torkkola, "Blind separation of delayed sources based on information maximization," to appear in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'96)*, Atlanta, USA, May 1996.

[39] L. Wang and J. Karhunen, "A unified neural bigradient algorithm for robust PCA and MCA," to appear in *Int. J. of Neural Systems.*

[40] L. Wang, J. Karhunen, and E. Oja, "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.

[41] L. Wang, J. Karhunen, E. Oja, and R. Vigario, "Blind separation of sources using nonlinear PCA type learning algorithms," in *Proc. Int. Conf. on Neural Networks and Signal Processing*, Nanjing, China, December 1995, pp. 847-850.

[42] J. Zhu, X.-R. Cao, and R.-W. Liu, "Blind source separation based on output independence - theory and implementation," in *Proc. 1995 Int. Symp. on Nonlinear Theory and Applications (NOLTA-95)*, Las Vegas, USA, December 1995, vol. 1, pp. 97-102.