REVIEW

ICA Methods for Blind Source Separation of Instantaneous Mixtures: A Case Study

Niva Das

Department of Electronics and Telecom. Engineering, I.T.E.R. Bhubaneswar India

Aurobinda Routray

Department of Electrical Engineering, I.I.T. Kharagpur India

Pradipta Kishore Dash

Director C.O.E. Bhubaneswar India

(Submitted on January 24, 2007)

Abstract – The paper presents comparative assessment of Blind Source Separation methods for instantaneous mixtures. The study highlights the underlying principles of Principal Component Analysis (PCA) and Independent Component Analysis (ICA) in this context. These methods have been tested on instantaneous mixtures of synthetic periodic signals, monotonous noise from electromechanical systems and speech signals. In particular, methods based on Nonlinear PCA, Maximum Entropy, Mutual Information Minimization and Fast ICA, have been compared for their separation ability, processing time and accuracy. The quality of the output, the complexity of the algorithms and the simplicity (implementation) of the methods are some of the performance measures which are highlighted with respect to the above signals.

Keywords – ICA, blind source separation, nonlinear PCA

1. Introduction

The problem of Blind source separation (BSS) has received considerable attention in recent years because of its significant potential applications spanning over a wide range of diverse disciplines like sonar and radar signal processing, telecommunications, array signal processing [1], wireless communication, geophysical exploration, biomedical signal processing [49-51][2-4][55], speech and image processing[52][65].

The objective in a basic BSS problem is to extract unknown source signals that are mutually independent from the observed signals obtained from the sensors. The separation task becomes indispensable as the sources are mixed by an unknown medium and finally the mixed signals are delivered by the sensors. The term 'blind' stresses the fact that the mixing structure is unknown.

Multidimensional data analysis plays a major role in solving BSS problems. Proper representation of multivariate data commonly encountered in signal processing, pattern recognition, neural networks and statistical analysis is essential for visualization of the underlying geometry of the data structure. Linear transformations are used to exploit possible dependencies and reduce dimensionality of the multivariate data sets. Principal Component Analysis (PCA) and Independent Component Analysis (ICA) are the two prominent methods for linear transformation. Particularly ICA has been very effective in separating independent sources from the mixed signals. Hence, the principles of ICA are essential to deal with the issues of BSS.

The problem of Blind source separation (BSS) arises in many application areas where ICA methods can be used as a solution [69-70]. Several instances are highlighted: ICA methods have been successfully applied in real time applications like extraction of fetal electrocardiogram from abdominal recordings [2][53], removal of artifacts in EEG signal and analysis of seizures [54][13][56]. In radar applications ICA methods can be employed to enhance the quality of remote sensing data in a synthetic aperture radar imagery unit [57]. In radio communication, ICA method is applied to the observations corresponding to the outputs of several antenna elements which also include the effects of mutual couplings of the antenna elements to effectively separate out the individual transmitted signals. In wireless communication, ICA is applied to the received signal to extract the actual transmitted signal which has been transmitted through an unknown channel [58]. Recent developments in ICA utilities have been in text document clustering [59], facial feature representation [60], financial data analysis, neurobiological signal processing [61] and speech processing. In speech processing ICA finds applications in separation of multiple speakers, cancellation of reverberation, analysis of speaker variations [62], speaker recognition [64] and extraction of speech features [63].

This paper focuses on the most popular methods such as standard PCA [5-6], neural implementation of nonlinear PCA [14-16], ICA based on information theoretic approaches [7-8][10-12] and fastICA [17][19] for solving the BSS problem. The paper is organized as follows: Section 2 discusses the blind source separation technique. Section 3 presents PCA. The necessary background on ICA [27][28] and four major techniques of ICA are described in section 4. Section 5 focuses on the results for both self generated data and real world data and Section 6 on the discussions pertaining to the performance related issues. Section 7 is the conclusion of the work.

2. The Blind Source Separation Problem

The problem of source separation concerns extracting source signals from their mixtures. The observations are obtained at the output of a set of sensors, each receiving a different combination of the source signals. Separation may be achieved in different ways according to the amount of prior information available. BSS [37-38] seeks to recover original source signals from their mixtures without any prior information on the sources or the parameters of the mixtures. In other words the BSS problem can be stated as the estimation of *n* sources from *m* measurements that are unknown function of the sources. The basic BSS model is shown in Figure 1. In the figure, the source data i.e., s(n) are mixed by a mixing matrix A to produce x(n) that serve as sensor data. Optimization algorithms like the ICA act on x(n) to produce a separating matrix W that has the capability to extract the original sources y(n) i.e., replica of s(n) from the mixed sources.

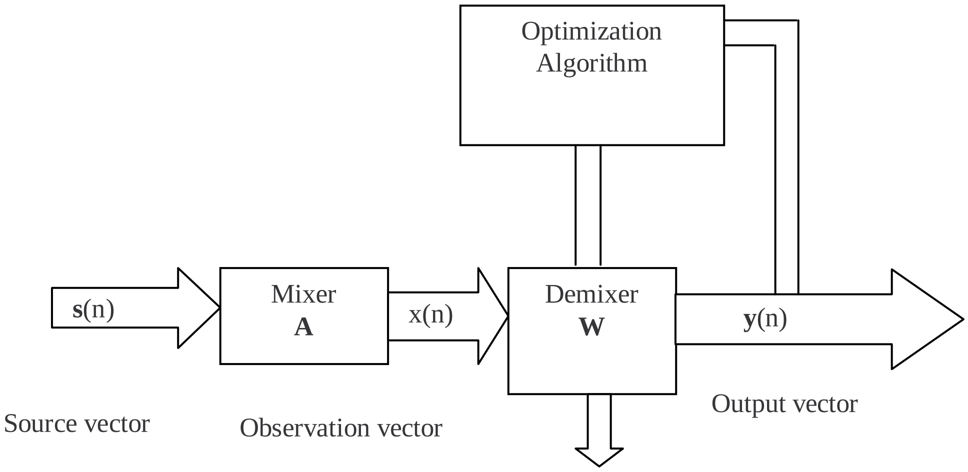


Figure 1. The basic block diagram of the BSS process

In the noise free linear instantaneous underdetermined case, the number of observations is smaller than the number of sources. The observation vector can be expressed as a linear transformation on the source vector as given by:

$$\mathbf{As}=\mathbf{x}$$

where

$\mathbf{A} \rightarrow$ is the mixing matrix $(m\times n)$

$\mathbf{s} \rightarrow$ is the source vector $(n\times 1)$ components

$\mathbf{x} \rightarrow$ is the observation vector $(m \times 1)$ components

$m \geq n$ → over determined

$m < n$ → under determined.

If the number of measurements is greater than or equal to the number of sources (*m*>*n*), it is possible to separate statistically independent sources under the condition that one of them is Gaussian. This represents an over-determined case. Once the mixing matrix is known, the sources can be obtained by the matrix-inversion when *m*=*n*. Similarly the sources can be estimated by using the pseudo inverse, when *m*>*n*. But there is no unique inverse when *m*<*n*, which means that there exist an infinite number of source vectors that are solutions of the above linear model. The solution to the Blind Source Separation problem [47] depends on issues like:

- mixture is linear or non-linear,
- mixing process is time varying or time invariant,
- mixing operation is convolutive or non-convolutive (instantaneous),
- sensors are noisy or noise-less and
- relation between number of sources (*n*) and number of measurements (*m*)

2.1 Basic BSS Model

Most of the linear BSS models in the simplest form can be expressed algebraically as:

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{n}(k) \; for \; k{=}1,2,3,\ldots\ldots,N$$

where $\mathbf{x}(\mathrm{k}) = [x_1(k) \quad x_2(k) \quad \cdots \quad x_m(k)]^T$, is a vector of the observed signals at the discrete time instant k, $\mathbf{s}(\mathrm{k}) = [s_1(k) \quad s_2(k) \quad \cdots \quad s_n(k)]^T$, is a vector of the source components at the same time instant, **n**(k) is the additive noise independent of sources. The source signals are assumed to be statistically independent. **A** is the non-singular mixing matrix having dimension ($m \times n$). The problem can be formulated as the computation of an unmixing or separating matrix **W,** whose output **y** is given by $\mathbf{y} = \mathbf{W}\mathbf{x}$, **y** being an estimate of the vector **s** of the source signals.

The basic BSS model considers as many sensors as sources (*m*=*n*), nonconvolutive mixtures, and noise free observations. The mixing can be instantaneous or convolutive. Instantaneous mixing can be seen in studio recordings, where audio signals are mixed using a mixing desk, without any delay or reverberations. In biomedical applications such as fMRI [18] and EEG, signals and images are almost all instantaneous mixture problems.

For instantaneous noise-free mixing, we have:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad \text{and} \quad \mathbf{y} = \mathbf{W}\mathbf{x}$$

where the task is to recover the original sources by finding **W** ,which is theoretically equal to the inverse of the unknown matrix **A**, i.e., $\mathbf{W} = \mathbf{A}^{-1}$, so that **y** is as close as possible to **s**.

Different approaches are available to BSS [20-21][48]:

- When stationary sources are assumed to be independent and identically distributed (iid) with no temporal structure, Higher Order Statistics (HOS) [23-26] can been used to solve the BSS problem. This method does not allow more than one Gaussian source.
- If sources are temporally correlated but spatially uncorrelated, then Second Order Statistics (SOS) is sufficient to estimate the mixing matrix. But SOS [22] methods can not separate sources having identical spectral shapes.

3. Principal Component Analysis

A common task in data analysis is to find an appropriate representation of multivariate data describing their essential structure. PCA is a way of identifying the underlying geometry in data and expressing them in such a way as to highlight their similarities and differences. It is a simple, nonparametric method of extracting relevant information from confusing data sets found in many fields- from biomedicine to computer graphics. The main advantage of PCA is that it provides a roadmap:

- To reduce a high dimensional complex data set to a lower dimension without much loss of information

- To reveal the hidden dynamics underlying the data. This technique is used in image compression.

PCA can be termed as a linear transformation method which can express the data as a linear combination of its basis vectors.

Let **X** and **Y** be $m \times n$ matrices related by a linear transformation **P**. **X** is the original observed data set and **Y** is the representation of that data set.

$$\mathbf{Y} = \mathbf{P}\mathbf{X} \tag{1}$$

Eq.(1) represents a change of basis. **P** is a matrix that transforms **X** into **Y** and the rows of **P**, are a set of new basis vectors for expressing the columns of **X**.

The covariance matrix of the input data space **X** is given by:

$$\mathbf{S}_{\mathrm{x}} = \left(\frac{1}{n-1}\right)[\mathbf{X}\mathbf{X}^T],$$

where n is the number of samples.

$\mathbf{S_x}$ is square and symmetric matrix. The diagonal terms of $\mathbf{S_x}$ are the variance of the particular measurement types. As multiple sensors record the same dynamic information, the covariance matrix describes all relationships between pair of measurements. So, redundancy is removed when covariance between separate measurements become negligible. In an optimized covariance matrix, all off diagonal terms are small.

To diagonalize the covariance matrix, $\mathbf{S}_y = [\mathbf{Y}\mathbf{Y}^T]/(n-1)$, the PCA selects a normalized direction in the m-dimensional space along which the variance is maximized. It then finds another direction perpendicular to the previous one along which variance is maximized. The search continues until m directions are selected. The resulting ordered set of **p** are the Principal Components, i.e., $\mathbf{P} = [\mathbf{p_1}\ \mathbf{p_2} \qquad \mathbf{p_m}]^T$.

3.1 Assumptions in PCA

- The basic assumption about PCA is that it is a linear transformation. Extension to the basic algorithm has been possible by applying non-linearity prior to performing PCA, which is popularly known as *kernel* PCA.
- Additional assumption in PCA is that the mean and variance entirely describe a probability distribution. The only zero-mean probability distribution that is fully described by the variance is the Gaussian distribution. Hence the probability distribution of the data vector must be Gaussian. If data vectors are not Gaussian distributed, then diagonalizing a covariance matrix might not give satisfactory results. This assumption formally guarantees that the Signal to Noise Ratio (SNR) and the covariance matrix fully characterize the noise and the redundancies.
- One important assumption that makes PCA soluble with linear decomposition techniques is that the principal components are orthogonal.
- PCA assumes that large variances have important dynamics like high SNR and hence principal components associated with larger variances represent interesting dynamics while those associated with lower variances represent noise.

3.2 Limitations of PCA

- PCA is a way of encoding second order dependencies by finding the directions of maximal variance.
- It decorrelates the input data but doesn’t address the higher order dependencies. It uses second order methods to reconstruct the signal in the mean square sense.
- In PCA, the data are represented in an orthonormal basis determined by the second order statistics (covariances) of the input data. Such a representation is adequate for Gaussian data but non-Gaussian data contain a lot of additional information in its higher order statistics. So, PCA fails to detect sources with non-Gaussian distribution.

Though the assumptions in PCA are too stringent, there might be situations where principal components need not be orthogonal, distributions along each dimension ($\mathbf{x_i}$) need not be Gaussian and the largest variances don’t correspond to meaningful axes. These problems can be solved by Independent Component Analysis (ICA), a recently developed technique which can be thought of as an extension to PCA. The ICA abandons all assumptions considered in PCA except linearity and finds direction of maximal independence in non-Gaussian data.

4. Independent Component Analysis

Independent Component Analysis (ICA) is a data model with different applications and Blind source separation is an application that can be solved using various theoretical approaches including but not limited to ICA. Potential applications of ICA [9] are seen in data analysis, array processing, blind deconvolution and feature extraction.

ICA of a random vector **x** consists of finding a linear transformation $\mathbf{s} = \mathbf{W}\mathbf{x}$, so that the components s_i are as independent as possible. This can be achieved by maximizing some function $F(s_1,s_2,......,s_m)$ that measures independence.

4.1 Assumptions in ICA

The observed signals are given by:

$$x_j(k) = \sum_{i=1}^{n} s_i(k) a(i,j) + n_j(k) \tag{8}$$

where j = 1,2 ,…,m indicates number of observations, i = 1,2,…,n indicates number of independent components, $n(k)$ indicates the noise sample.

Let $\mathbf{s}(k) = [s_1(k) \quad s_2(k) \quad \cdots \quad s_n(k)]^T$, $A = [a_1 \quad a_2 \quad \cdots \quad a_n]$, and $\mathbf{x}(k) = [x_1(k) \quad x_2(k) \quad \cdots \quad x_m(k)]^T$.

1. The number of observed linear mixtures m must be at least as large as the number of independent components n, i.e., $m \geq n$.
2. The independence condition can be defined by stating that the joint probability density of the source signals is equal to the product of the marginal probability densities of the individual signals, i.e.,

$$p(s) = \prod_{i=1}^{n} p[s_i(k)] \tag{9}$$

3. The source signals must be statistically independent of each other or in practice as independent as possible (it includes uncorrelatedness). This is difficult to verify in advance because distribution of data is not available in real world problems.
4. Each source signal must be stationary with zero mean. Only one source is allowed to have Gaussian distribution because a linear combination of Gaussian signals is Gaussian again making it impossible to separate them.
5. The matrix **A** must be of full rank. Signal **x** and **s** should be centered.

4.2 Ambiguities with ICA

1. The indeterminacy [45] associated with the ICA model is that the independent components and the columns of the mixing matrix **A** can be estimated up to a multiplicative constant, because any constant multiplying one independent component in the basic ICA model could be cancelled by dividing the corresponding column of the mixing matrix **A** by the same constant. So, to make the independent components unique up to a multiplicative sign, the sources should have unity variance.
2. There is a sign ambiguity (phase reversal or multiplication by –1) associated with the separated sources without affecting the model.
3. The order of the independent components can not be determined, i.e., the estimated source signals may be recovered in a different order. This is again due to the fact that the **s** and **A** matrices are unknown in the basic ICA model.

4.3 ICA Methods

Two categories of ICA algorithms exist. In the first type, source separation can be obtained by optimizing an objective function [30-31] which can be a scalar measure of some distributional property of the output **y**. More general measures are entropy, mutual independence, divergence between joint distribution of **y** and some given mode and higher order decorrelation.

The ICA method can be formulated as optimization of a suitable objective function which is also termed as the contrast function. The problem in optimization of contrast function is that, they rely on batch computation using the estimated higher order statistics of data or lead to complicated adaptive separation. It is often sufficient to use simple higher order statistics such as kurtosis, which is a fourth order cummulant with zero time lags. The kurtosis for the i^{th} source signal *s(i)* is given by:

$$Cum\left[s(i)^4\right] = E\left\{s(i)^4\right\} - 3\left[E\left\{s(i)^2\right\}\right]^2 \tag{10}$$

If *s(i)* is Gaussian, then its kurtosis is zero. Source signals that have negative kurtosis are called sub-Gaussian and have a probability distribution flatter than usual Gaussian distribution. Source signals having a positive kurtosis are called super-Gaussian and have a probability distribution with sharp peaks and longer tails than the standard Gaussian ones. A contrast function based on kurtosis is given by:

$$J_1(y) = \sum_{i=1}^{n}\left|Cum\left[y(i)^4\right]\right| = \sum_{i=1}^{n}\left|E\left\{y(i)^4\right\} - 3\left[E\left\{y(i)^2\right\}\right]^2\right| \tag{11}$$

It is maximized by a separating matrix **W**, if the sign of kurtosis is same as all the source signals. For pre-whitened input vector **x** and orthogonal separating matrices, $E[y(i)^2] = 1$ and (11) reduces to $E[y(i)^2] - 3$. Therefore, $J_1(y)$ is maximized, when $\sum_{i=1}^{n} E[y(i)^4]$ is minimized for sources having negative kurtosis and maximized for sources with positive kurtosis.

The second category uses neural implementation of ICA algorithms like non-linear PCA subspace learning [32-33] for achieving source separation. In this category, there are adaptive algorithms [49] like minimum mutual information method [34] and maximum entropy method [35] derived from information theoretic approach based on stochastic gradient optimization.

4.4 ICA using Neural Network

The basic neural network architecture to perform the separation of source signals is shown in Figure 2. The figure shows that the entire process of source separation requires three stages, i.e., whitening, separation and estimation.

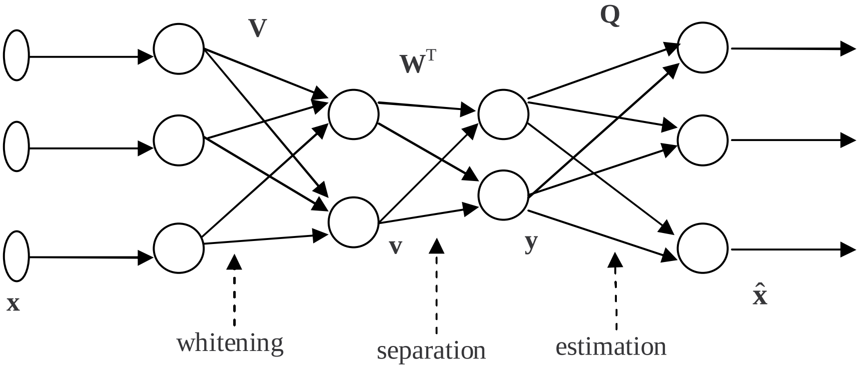


Figure 2. The ICA Network

In our discussion, we have taken: $m = n$, i.e., number of sources same as number of sensors. Feedback connections (not shown) are needed in the learning phase, but after learning these networks become feed forward, if data is stationary. Though the input-output mappings of the ICA network are linear after learning, nonlinearities must be used in the learning phase, so as to introduce higher order statistics into computations for achieving independence. This network is used in robust non-linear PCA learning algorithms. Each of the three layers shown in Figure 2 performs one of the processing tasks required to complete the process of ICA, i.e., preprocessing, separation and estimation of basis vectors. Any of these three tasks can be done either neurally or by conventional means. If the task is mere source separation, then the last layer is not needed.

The learning algorithms for all three phases are described under the heading of whitening, separation and estimation of ICA basis vectors.

Whitening is a critical procedure that makes the data suitable for separation algorithms to converge faster and to have better stability properties. Before whitening, preprocessing is done to ensure that the observed signals have zero mean and unity variance. On the other hand whitening can make the separation problem more difficult, if the mixing matrix is ill-conditioned or if some of the source signals are relatively weak compared to the other signals. Process of whitening can be thought of as a transformation which converts the observed data **x**(k) to a new vector, **v**(k) called as whitened vector with the help of a whitening matrix **V** so as to ensure, $E\{\mathbf{v}(k)\mathbf{v}(k)^T\} = \mathbf{I}\ (\text{identity matrix})$. The process can be mathematically expressed as:

$$\mathbf{v}(k) = \mathbf{V}\mathbf{x}(k) \tag{12}$$

The components of the whitened vector must be mutually uncorrelated and normalized to have unity variance. The whitening matrix can be determined by using a batch approach or by neural learning. The batch approach uses standard PCA technique to determine the whitening matrix as given by:

$$\mathbf{V} = \mathbf{D}^{-1/2}\mathbf{E}^{\mathrm{T}} \tag{13}$$

where **D** is a diagonal matrix with diagonal elements as eigen values of the covariance matrix $\mathbf{C} = E\{\mathbf{x}(k)\mathbf{x}(k)^T\}$ and **E** is the matrix containing eigen vectors of **C**. PCA is used in most of the over determined cases as it helps in reducing noise. The whitening matrix can also be learned using a simple neural algorithm as follows:

$$\mathbf{V}(k+1) = \mathbf{V}(k) - \mu(k)[\mathbf{v}(k)\mathbf{v}^T(k) - \mathbf{I}]\mathbf{V}(k) \tag{14}$$

where $\mu(k)$ is the learning rate parameter and can be adjusted as given by:

$$\mu(k) = \frac{1}{\gamma / \mu(k-1) + \|\mathbf{V}(k)\|_2^2} \quad . \tag{15}$$

In (15), γ is the forgetting factor and lies in the range $0<\gamma\leq 1.0$.

4.5 Non-linear PCA subspace learning

Neural implementation of the separating algorithm under unsupervised category for learning the separating matrix **W** is presented here [29][46]. The second stage of the ICA network is responsible for the separation of the whitened signals, $v(k)$. The linear separation transformation is given by:

$$\mathbf{y}(k) = \mathbf{W}^{\mathrm{T}}\mathbf{v}(k) \tag{16}$$

The separated signals are the outputs of the second stage, i.e., $\hat{\mathbf{s}}(k) = \mathbf{y}(k)$ where $\hat{\mathbf{s}}(k)$ is the estimated source signals.

The non-linear PCA subspace learning rule [16] is given by:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mu(\mathrm{k})\{\mathbf{v}(k) - \mathbf{W}(k)\mathbf{g}[\mathbf{y}(k)]\}\mathbf{g}[\mathbf{y}^{\mathrm{T}}(k)] \tag{17}$$

where $\mathbf{v}(k)$ is the whitened input vector, $\mu(k)$ is known as the learning rate parameter and is usually positive, ***g***(.) is a suitably chosen odd non-linearity, providing stability in the process of separation. This is a stochastic gradient descent algorithm which tries to minimize or maximize the performance criterion, i.e., sum of fourth moments (kurtosis). The performance criterion is expressed as:

$$J(\mathbf{W}) = \sum_{i=1}^{n} E\{f[\mathbf{y}(i)]\} \tag{18}$$

The objective function chosen for this is $f(t) = \beta^2 1\mathrm{n}[\cosh(t/\beta)]$. The derivative of $f(t)$ is $g(t)$ and is given by: $g(t) = f'(t) = \beta \tanh(t/\beta)$. If β=1 then, $g(t) = f'(t) = \tanh(t)$.

The Taylor's series expansion of *f*(*t*) for $\beta =1$ is given by *f*(*t*) = ln[cosh(*t*)] = $t^2/2 - t^4/12 + t^6/45 - \ldots$, and that of $g(t)$ can be written as $g(t) = t - t^3/3 + 2t^5/15 - \cdots\cdots\cdots$.

The second order term $t^2/2$, in the Taylor's series expansion of *f*(*t*) remains on an average constant due to whitening and the cubic term in the non-linearity *g*(*t*) will dominate and the learning rule will converge to a separating matrix **W**. But the columns of **W** are not exactly orthonormal. In this learning rule, the term $\mathbf{v}(k)\mathbf{g}[\mathbf{y}^T(k)]$ is responsible for learning the separating matrix **W**, while the other terms play the role of stabilizing and normalizing **W**. The problems associated with this algorithm are the choice of parameters i.e., β, μ, γ and the knowledge about the nature of the mixing matrix **A**.

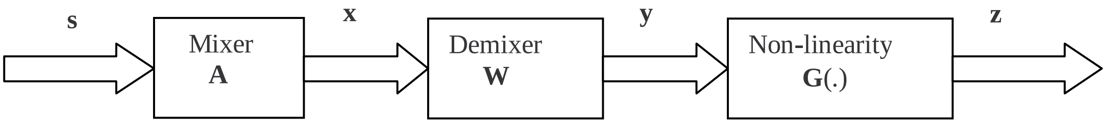


Figure 3. Maximum Entropy Method

4.6 Maximum Entropy Method

This is an adaptive algorithm based on information theoretic approach and was suggested by Bell & Sejnowski [7]. The block diagram in Figure 3 explains the maximum entropy method for blind source separation.

The demixer operates on the observed data **X** to produce an output **Y = WX**, which is an estimate of source **S**. The output **Y** is transformed into **Z** by passing it through a non-linearity $\mathbf{G}(\cdot)$, which is invertible and monotonic. For a given non-linearity $\mathbf{G}(\cdot)$, the maximum entropy method produces an estimate of source **S** by maximizing the entropy $h(\mathbf{Z})$ with respect to **W**. The mathematical representation of the whole process may be given as follows:

$$\mathbf{Z} = \mathbf{G}(\mathbf{Y}) = \mathbf{G}(\mathbf{WAS}) \quad \Rightarrow \mathbf{S} = \mathbf{A}^{-1}\mathbf{W}^{-1}\mathbf{G}^{-1}(\mathbf{Z}) = \psi(\mathbf{Z}) \tag{19}$$

where $\mathbf{G}^{-1}$ is the inverse non-linearity.

The probability density function of the output **Z** is defined in terms of that of the source **S** by

$$f[\mathbf{Z}(\mathbf{z})] = \left.\frac{f[\mathbf{S}(\mathbf{s})]}{|\det(\mathbf{J}(\mathbf{s}))|}\right|_{\mathrm{s}=\psi(\mathrm{z})}$$

where $\det(\mathbf{J}(\mathbf{s}))$ is the determinant of the Jacobian matrix **J(s)**. The *ij*-th element of the matrix **J(s)** is defined by $J_{ij} = \partial z_i / \partial s_j$. Hence, the entropy of the output **Z** at the output of the non-linearity $\mathbf{G}(\cdot)$ is

$$h(\mathbf{Z}) = -E[\log f_{\mathbf{Z}}(\mathbf{z})] = -E\left[\log\left(\frac{f_{\mathbf{S}}(\mathbf{s})}{|\det(\mathbf{J}(\mathbf{s}))|}\right)\right]_{\mathrm{s}=\psi(\mathrm{z})} = -D_{f_{\mathrm{S}}}\| |\det(\mathbf{J})| \quad \text{evaluated at } \mathbf{s} = \psi(\mathbf{z}) \ .$$

Hence, maximizing the entropy $h(\mathbf{Z})$ is equivalent to minimizing the Kullback-Leibler divergence between $f_{\mathbf{S}}(\mathbf{s})$ and a probability density function of **S**, defined by $|\det(\mathbf{J}(\mathbf{s}))|$.

If the random variable z_{i} (*i*th element of **z**) is uniformly distributed inside the interval [0,1] for all *i*, then the entropy $h(\mathbf{z})$ is equal to zero. Accordingly,

$$h(\mathbf{Z}) = -E[\log f_{\mathbf{Z}}(\mathbf{z})] = -E\left[\log\left(\frac{f_{\mathbf{S}}(\mathbf{s})}{|\det(\mathbf{J}(\mathbf{s}))|}\right)\right]_{\mathrm{s}=\psi(\mathrm{z})} \quad \Rightarrow f_{\mathbf{S}}(\mathbf{s}) = |\det(\mathbf{J}(\mathbf{S}))|$$

Under the ideal condition, $\mathbf{W} = \mathbf{A}^{-1}$, the above relationship reduces to

$$f_{\mathbf{S}_i}(s_i) = \left.\frac{\partial \mathrm{z}_{\mathrm{i}}}{\partial y_{\mathrm{i}}}\right|_{z_{\mathrm{i}}=g(s_{\mathrm{i}})} \quad \text{for all } i.$$

Conversely, the results from Maximum Entropy Method may be stated as follows:

Let the non-linearity at the demixer output be defined in terms of the original source distribution as

$$z_i = g_i(y_i) = \int_{-\infty}^{z_i} f_{S_i}(s_i)\,ds_i\,, \quad for \ \ i = 1,2,\cdots,n$$

Then, maximizing the entropy of the random vector **z** at the output of the non-linearity **G** is equivalent to $\mathbf{W} = \mathbf{A}^{-1}$, which yields perfect blind source separation. The maximum entropy and maximum likelihood methods for blind source separation are equivalent under the condition that the random variable z_{i} is uniformly distributed inside the interval [0,1] for all *i*. This relationship may be proven with the help of chain rule of calculus as

$$J_{ij} = \sum_{k=1}^{n} \frac{\partial z_i}{\partial y_i}\frac{\partial y_i}{\partial x_i}\frac{\partial x_i}{\partial s_i} = \sum_{k=1}^{n} \frac{\partial z_i}{\partial y_i} w_{\mathrm{ik}} a_{\mathrm{kj}}$$

The Jacobian matrix **J** is expressed as $\mathbf{J} = \mathbf{DWA}$, where **D** is a diagonal matrix given by

$$\mathbf{D} = diag\left(\frac{\partial z_1}{\partial y_1}, \frac{\partial z_2}{\partial y_2}, ..., \frac{\partial z_n}{\partial y_n}\right).$$

Hence, $\left|\det(\mathbf{J})\right| = \left|det(\mathbf{WA})\right| \prod_{i=1}^{n} \frac{\partial z_i}{\partial y_i}$.

In the light of the above equation, an estimate of the probability density function $f_{\mathbf{S}}(\mathbf{s})$ parameterized by the weight matrix **W** and the non-linearity **G** may be written formally as

$$f_{\mathbf{S}}(\mathbf{s}|\mathbf{W},\mathbf{G}) = \left| det(\mathbf{WA})\right| \prod_{i=1}^{n} \frac{\partial g_i(y_i)}{\partial y_i}$$

Therefore, under the above condition, maximizing the log-likelihood function $\{\log f_{\mathbf{S}}(\mathbf{s}|\mathbf{W},\mathbf{G})\}$ is equivalent to maximizing the entropy $h(\mathbf{Z})$ for blind source separation.

Referring to the expression $h(\mathbf{Z}) = -E[\log f_{\mathbf{Z}}(\mathbf{z})] = -E[\log(f_{\mathbf{S}}(\mathbf{s})/|\det(\mathbf{J}(\mathbf{s}))|)]_{\mathbf{s}=\psi(\mathbf{z})}$, it is seen that since the source distribution is fixed, maximizing the entropy $\mathrm{h}(\mathbf{Z})$ requires maximizing the expectation of the denominator term $\{\log|\det(\mathbf{J}(\mathbf{s}))|\}$ with respect to the separating matrix **W**.

To do the computation using an adaptive algorithm that will maximize the objective function, the instantaneous objective function ϕ may be considered as:

$$\phi = \log\left|\det(\mathbf{J})\right| \tag{20}$$

On expanding (20), we get:

$$\phi = \log\left|\det(\mathbf{A})\right| + \log\left|\det(\mathbf{W})\right| + \sum_{i=1}^{n} \log\left(\frac{\partial z_i}{\partial y_i}\right) \quad \text{and} \quad \frac{\partial \phi}{\partial \mathbf{W}} = \mathbf{W}^{-T} + \sum_{i=1}^{n} \frac{\partial}{\partial \mathbf{W}} \, log\left(\frac{\partial z_i}{\partial y_i}\right). \tag{21}$$

The non-linear function should be judiciously selected to deal with the super-Gaussian, sub-Gaussian, stationary and non-stationary signals. The popular non-linearities used are logistic function and hyperbolic tangent function:

$$z_i = \mathrm{g}(y_i) = \frac{1}{1+\mathrm{e}^{-y_i}} \qquad z_i = \mathrm{g}(y_i) = tanh(y_i) \quad i=1,2,\ldots,n$$

The non-linear functions should be monotonic and invertible.

Finding out $\partial\phi/\partial\mathbf{W}$ using the above non-linearity, we obtain $\partial\phi/\partial\mathbf{W} = \mathbf{W}^{-T} + (\mathbf{1}-2\mathbf{z})\mathbf{x}^T$, where **x** is the observed source vector, **z** is the non-linearly transformed output vector and **1** is a corresponding vector of ones. Using the steepest ascent method to maximize the entropy h(**Z**), the change in weight matrix **W** is given by $\Delta\mathbf{W} = \eta\frac{\partial\phi}{\partial\mathbf{W}} = \eta\left(\mathbf{W}^{-T} + (\mathbf{1}-2\mathbf{z})\mathbf{x}^T\right)$, where η is the learning rate parameter. The generalized final version for the update on **W** or the learning rule is obtained by using the natural gradient, which is equivalent to multiplying the expression for $\Delta\mathbf{W}$ by $\mathbf{W}^T\mathbf{W}$ instead of evaluating $\mathbf{W}^{-T}$ as given below:

$$\Delta\mathbf{W} = \eta(\mathbf{W}^{-T} + (\mathbf{1}-2\mathbf{z})\mathbf{x}^T)\mathbf{W}^T\mathbf{W} = \eta\left(\mathbf{I} + (\mathbf{1}-2\mathbf{z})(\mathbf{Wx})^T\right)\mathbf{W} = \eta\left(\mathbf{I} + (\mathbf{1}-2\mathbf{z})\mathbf{y}^T\right)\mathbf{W}$$

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \eta\left(\mathbf{I} + (\mathbf{1}-2\mathbf{z}(k))\mathbf{y}^T(k)\right)\mathbf{W}(k) \tag{22}$$

where **y** is the output of the demixer before passing through the non-linearity, **I** is the unity matrix and η is a fixed learning rate parameter with value less than 1.

The algorithm gives better result when applied on pre-whitened data. It is sensitive to the learning rate parameter and works better for super-Gaussian signals.

4.7 Minimization of Mutual Information

This is another adaptive algorithm that uses information theoretic approach [11][35]. To retain statistical independence between the estimated source signals, a possible measure is to minimize the mutual information between them, i.e., $I(y_i: y_j) = 0$, where y_i and y_j constitute two independent components of the output vector **y** and $I(\cdot)$ is the mutual information. Minimizing the mutual information is same as minimizing the Kullback-Leibler divergence between (1) the parameterized probability density function $f_{\mathbf{y}}(\mathbf{y},\mathbf{W})$ and (2) the corresponding factorial distribution defined by $\tilde{f}_{\mathbf{y}}(\mathbf{y},\mathbf{W})=\prod_{i=1}^{n}\tilde{f}_{y_i}(y_i,\mathbf{W})$, where $\tilde{f}_{\mathbf{y}_i}(y_i,\mathbf{W})$ is the marginal probability density function of y_i. The Kullback–Leibler Divergence between the probability density functions $f_{\mathbf{y}}(\mathbf{y},\mathbf{W})$ and $\tilde{f}_{\mathbf{y}}(\mathbf{y},\mathbf{W})$ is defined as $D_{f\,||\,\tilde{f}}(\mathbf{W})=-h(\mathbf{y})+\sum_{\mathrm{i}=1}^{\mathrm{n}}\tilde{h}(y_i)$, where $h(\mathbf{y})$ is the entropy of the random vector $\mathbf{y}$ at the output of the demixer and $\tilde{h}(y_i)$ is the marginal entropy of the i^{th} element of **y**. The BSS can be stated as : Given any random matrix **x** (observed) representing a linear combination of *n* independent source signals, the transformation of **x** by a neural algorithm into output **y** should be done in such a way that the Kullback-Leibler Divergence between $f_{\mathbf{y}}(\mathbf{y},\mathbf{W})$ and $\tilde{f}_{\mathbf{y}}(\mathbf{y},\mathbf{W})$ is minimized with respect to the unknown parameter matrix **W** (demixer). As we know,

$$\mathbf{y}=\mathbf{W}\mathbf{x} \text{ and } h(\mathbf{y})=h(\mathbf{W}\mathbf{x})=h(\mathbf{x})+log\left|\,det(\mathbf{W})\right|.$$

The marginal entropy is difficult to calculate when **y** has a higher dimension and can be approximated in terms of higher order moments of the random y_i by following the expansion of Gram-Charlier series [39]. The Gram-Charlier expansion of the parameterized marginal probability density function $\tilde{f}_{y_i}(y_i,\mathbf{W})$ is described by $\tilde{f}_{y_i}(y_i,\mathbf{W})=\alpha(y_i)[1+\sum_{k=3}^{\infty}c_k\,H_k(y_i)$, where the various terms are defined as follows:

1. $\alpha(y_i)$ is a normalized Gaussian random variable: $\alpha(y_i)=(1/2\pi)e^{-y_i^2/2}$.
2. $H_k(y_i)$ is Hermite polynomial.
3. The coefficients of the expansion, $\{c_k : k=3,4,....,\}$, are defined in terms of the cummulants of the random variable y_i. The series should be truncated after the 6th order term, because it is on the same order of magnitude as the 4th order [39]. Following the grouping order *k* ={ (0), (3), (4,6), (5,7,9), …} and truncating the Gram-Charlier series at *k* = (4,6), the marginal probability density function $\tilde{f}_{y_i}(y_i)$ is given by

$$\tilde{f}_{y_i}(y_i)\approx\alpha(y_i)\left[1+\frac{k_{i,3}}{3!}H_3(y_i)+\frac{k_{i,2}^2}{4!}H_4(y_i)+\frac{\left(k_{i,6}+10k_{i,3}^2\right)}{6!}H_6(y_i)\right],$$

where $k_{i,k}$ is the k^{th} order cummulant of y_i. Let $m_{i,k}$ denote the k^{th} order moment of y_i, i.e., $m_{i,k}=\mathrm{E}[y_i^k]=\mathrm{E}[(\sum_{j=1}^{n}w_{ij}x_j)^k]$, then, we can define $k_{i,k}$ in terms of the moments as

$$k_{i,3}=m_{i,3}\,,\quad k_{i,4}=m_{i,4}-3m_{i,2}^2\,,\quad k_{i,6}=m_{i,6}-10m_{i,3}^2-15m_{i,2}m_{i,4}+30m_{i,2}^2\;.$$

Taking logarithm on both sides of the above expression for $\tilde{f}_{Y_i}(y_i)$, we get

$$log\;\tilde{f}_{y_i}(y_i)\approx log\,\alpha(y_i)+log\left[1+\frac{k_{i,3}}{3!}H_3(y_i)+\frac{k_{i,2}^2}{4!}H_4(y_i)+\frac{\left(k_{i,6}+10k_{i,3}^2\right)}{6!}H_6(y_i)\right].$$

Using (i) the expansion of the logarithm : $\log(1+y)\approx y-y^2/2$ (where all the terms of order three and higher are neglected), (ii) the formula for the marginal entropy of y_i : $\tilde{h}(y_i)=-\int_{-\infty}^{\infty}\tilde{f}_{y_i}(y_i)log\,\tilde{f}_{y_i}(y_i)dy_i$, i =1,2,...,n, (iii) certain integrals that involve the normalized Gaussian density $\alpha(y_i)$ and various Hermite polynomials $H_k(y_i)$, the approximate formula for marginal entropy is given as

$$\begin{aligned}\tilde{h}(y_i)\;\approx\;&\frac{1}{2}\log(2\pi e)-\frac{k_{i,3}^2}{12}-\frac{k_{i,4}^2}{48}-\frac{\left(k_{i,6}+10k_{i,3}^2\right)^2}{1440}+\frac{3}{8}k_{i,3}^2k_{i,4}+\frac{k_{i,3}^2\left(k_{i,6}+10k_{i,3}^2\right)}{24}\\&+\frac{k_{i,4}^2\left(k_{i,6}+10k_{i,3}^2\right)}{24}+\frac{k_{i,4}\left(k_{i,6}+10k_{i,3}^2\right)^2}{64}+\frac{k_{i,4}^3}{16}+\frac{\left(k_{i,6}+10k_{i,3}^2\right)^3}{432}\end{aligned}$$

The objective is to minimize the objective function, which is the Kullback-Leibler Divergence $D_{f\,||\,\tilde{f}}$ for the source separation problem using gradient descent technique.

$$\begin{aligned} D_{f\,||\,\tilde{f}}(\mathbf{W}) &= -h(\mathbf{x}) - \log\left|\det(\mathbf{W})\right| + \frac{n}{2}\log(2\pi e) \\ &\quad - \sum_{i=1}^{n}\left(\begin{array}{l} \dfrac{k_{i,3}^2}{12} + \dfrac{k_{i,4}^2}{48} + \dfrac{\left(k_{i,6}+10k_{i,3}^2\right)^2}{1440} - \dfrac{3}{8}k_{i,3}^2 k_{i,4} - \dfrac{k_{i,3}^2\left(k_{i,6}+10k_{i,3}^2\right)}{24} - \\ \dfrac{k_{i,4}^2\left(k_{i,6}+10k_{i,3}^2\right)}{24} - \dfrac{k_{i,4}^2\left(k_{i,6}+10k_{i,3}^2\right)^2}{64} - \dfrac{k_{i,4}^3}{16} - \dfrac{\left(k_{i,6}+10k_{i,3}^2\right)^3}{432} \end{array} \right) \end{aligned} \tag{23}$$

where the cummulants $k_{i,k}$ are functions of weight matrix **W**.

To evaluate the Kullback-Leibler divergence as in (23), a procedure is needed to compute all higher order cummulants of a random vector. To develop a learning algorithm by minimizing the function in (23), we differentiate both sides of the equation with respect to **W** and thereby formulate the activation function $\varphi(\mathrm{y_i})$

$$\frac{\partial}{\partial w_{i,k}} D_{f\,||\,\tilde{f}}(\mathbf{W}) \approx -\left(\mathbf{W}^{-T}\right)_{i,k} + \varphi(y_i)x_k \tag{24}$$

where $\varphi(y_i)$ is given by:

$$\varphi(y_i) = \frac{1}{2}y_i^5 + \frac{2}{3}y_i^7 + \frac{15}{2}y_i^9 + \frac{2}{15}y_i^{11} - \frac{112}{3}y_i^{13} + 128y_i^{15} - \frac{512}{3}y_i^{17}\ . \tag{25}$$

It is non-monotonic in nature and y_i is confined to [-1,1]. The learning rule based on gradient ascent may be formulated by defining the adjustment applied to weight w_{ik} as $\Delta w_{i,k} = -\eta\dfrac{\partial}{\partial w_{ik}} D_{f\,||\,\tilde{f}} = \eta\left(\left(\mathbf{W}^{-T}\right)_{ik} - \varphi(y_i)x_k\right)$.

The entire adjustment matrix may be expressed as

$$\Delta\mathbf{W} = \eta\left(\left(\mathbf{W}^{-T}\right) - \varphi(\mathrm{y})\mathrm{x}^T\right), \quad \varphi(\mathrm{y}) = \left[\varphi(y_1) \quad \varphi(y_2) \quad \cdots \quad \varphi(y_n)\right]^T$$

$$\mathrm{y}^T = \mathrm{x}^T\mathbf{W}^T, \quad \Delta\mathbf{W} = \eta\left[\mathrm{I} - \varphi(\mathrm{y})\mathrm{x}^T\mathbf{W}^T\right]\mathbf{W}^{-T} = \eta\left[\mathrm{I} - \varphi(\mathrm{y})\mathrm{y}^T\right]\mathbf{W}^{-T}$$

The final version for the update on **W** or the learning rule is obtained by using the natural gradient (which is equivalent to multiplying the expression for **ΔW** by $\mathbf{W}^T\mathbf{W}$ instead of evaluating $\mathbf{W}^{\text{-T}}$) which may be produced as

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \eta(k)\left[\mathbf{I} - \varphi(\mathbf{y}(k))\mathbf{y}^T(k)\right]\left(\mathbf{W}(k)\mathbf{W}^T(k)\right)\mathbf{W}^{-T}(k) = \mathbf{W}(k) + \eta(k)\left[\mathbf{I} - \varphi(\mathbf{y}(k))\mathbf{y}^T(k)\right]\mathbf{W}(k) \tag{26}$$

In (26) **W** should be invertible and η should be small enough to maintain the estimates of cummulants reliable. Stability is achieved when the slope of the activation function is positive.

4.8 Fast Fixed Point ICA

Adaptive algorithms based on stochastic gradient descent may be problematic when used in an environment where no adaptation is needed. This is the case in many practical situations. The convergence is often slow, and depends crucially on the choice of the learning rate sequence. As a remedy for this problem, one can use batch (block) algorithms based on fixed-point iteration [17][40]. In [17], a fixed-point algorithm, namely fastICA, was introduced using kurtosis and was later on generalized for general contrast functions. The fastICA algorithm uses a simple, highly efficient fixed point iteration scheme to find the local extrema of the contrast function biased output vector. The objective function based on kurtosis may be expressed as:

$$J(\mathbf{w}) = E\left\{\left(\mathbf{w}^T\mathbf{x}\right)^4\right\} - 3\|\mathbf{w}\|^4 + F\left(\|\mathbf{w}\|^2\right) \tag{27}$$

where $kurt\left(\mathbf{w}^T\mathbf{x}\right) = E\left\{\left(\mathbf{w}^T\mathbf{x}\right)^4\right\} - 3\left[E\left\{\left(\mathbf{w}^T\mathbf{x}\right)^2\right\}\right]^2 = E\left\{\left(\mathbf{w}^T\mathbf{x}\right)^4\right\} - 3\|\mathbf{w}\|^4$, under the constraint ||**w**|| =1 and F is a penalty factor due to the constraint. The objective function is simplified because the data is whitened. The learning rule has the form:

$$\mathbf{w}(k+1)=\mathbf{w}(k)\pm\mu(k)\left[\mathbf{x}(k)\left(\mathbf{w}(k)^T\mathbf{x}(k)\right)^3-3\|\mathbf{w}(k)\|^2\mathbf{w}(k)+f\left(\|\mathbf{w}(k)\|^2\right)\mathbf{w}(k)\right] \tag{28}$$

where **x**(k) is the observation sequence, $\mu(k)$ is the learning rate sequence and f is the derivative of $F/2$. The expectations are removed by instantaneous values. The terms inside the square bracket are the gradients of kurtosis (first two terms) and the gradient of $F(\|\mathbf{w}\|^2)$(third term). The gradient of $F(\|\mathbf{w}\|^2)$ has the form (scalar × **w**) as long as this is a function of $\|\mathbf{w}\|^2$ only. positive sign before the bracket means finding the local maxima and –ve sign before the bracket means finding the local minima. The fixed points **w** of the learning rule in (28) are obtained by taking the expectations and equating the change in weight (**w**) to zero, which may be expressed as

$$E\left\{\mathbf{x}\left(\mathbf{w}^T\mathbf{x}\right)^3\right\}-3\|\mathbf{w}\|^2\mathbf{w}+f\left(\|\mathbf{w}\|^2\right)\mathbf{w}=0\ . \tag{29}$$

As the third term in (29) can be written in the form (scalar × **w**), the final form of (29) is

$$\mathbf{w}=scalar\times\left(E\left\{\mathbf{x}\left(\mathbf{w}^T\mathbf{x}\right)^3\right\}-3\|\mathbf{w}\|^2\mathbf{w}\right) \tag{30}$$

The scalar term in the above equation is insignificant and the effect can be eliminated with normalization.

FastICA is neural in the sense that it is parallel and distributed, but not adaptive in the sense that instead of using every data point immediately for learning, it uses sample averages computed over larger samples of the data. It is a general algorithm that can be used to optimize both one unit and multi-unit contrast function. The basic mathematical relationships are as $\mathbf{x}=\mathbf{A}\mathbf{s}$, $\mathbf{v}=\mathbf{V}\mathbf{x}$ so that $E[\mathbf{v}\mathbf{v}^T]=\mathbf{I}$ i.e., the elements of **v** are mutually uncorrelated, where **x** is the observed signal, **s** is the source signal, **V** is the whitening matrix, **A** is the mixing matrix and **v** is the whitened signal. Here, $\mathbf{v}=\mathbf{VAs}=\mathbf{Bs}$, where $\mathbf{B}=\mathbf{VA}$ is an orthogonal matrix, i.e., $E[\mathbf{B}\mathbf{B}^T]=\mathbf{I}$. The objective is to determine an orthogonal matrix **B** that does the separation of independent signals. The pseudo-inverse of the mixing matrix **A** is given by $\mathbf{A}^+=\mathbf{B}^T\mathbf{V}$.

Fixed Point Algorithm for ICA: The steps of fastICA (Deflation approach, i.e., the algorithm estimates the independent components one by one) are shown.

1. Prewhiten the observed data **x** to obtain **v**.

2. Take a random initial vector **w**(0) and normalize it to unity, i.e. $\mathbf{w}(0)=\mathbf{w}(0)/\|\mathbf{w}(0)\|_2$, and set $k=1$.

3. Set $\mathbf{w}(k)=E\left\{\mathbf{x}\left(\mathbf{w}(k-1)^T\mathbf{x}\right)^3\right\}-3\mathbf{w}(k\text{-}1)$. The expectation operator can be estimated using a large number of samples. Normalize **w**(*k*) to unity length. To ensure that different independent components are estimated each time, an orthogonalizing projection is included. The columns of matrix **B** are orthonormal. Thus, the basic idea is that estimation of independent components can be carried out one by one by projecting the current solution **w**(*k*) on the space orthogonal to the columns of matrix **B** previously found.

4. Set $\mathbf{w}(k)=\mathbf{w}(k)-\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T\mathbf{w}(k)$, where $\tilde{\mathbf{B}}$ is a matrix whose columns are previously found columns of $\mathbf{B}$. Set $\mathbf{w}(k)=\mathbf{w}(k)/\|\mathbf{w}(k)\|_2$, i.e., normalize **w**(*k*).

5. If $\left|\mathbf{w}^T(k)\mathbf{w}(k-1)\right|$ is not close to 1, then set $k=k+1$and repeat step 3. Otherwise output vector **w**(*k*) .

6. Using **w**(*k*), one of the separated signals is given by $\mathbf{s}(k)=\mathbf{w}^T(k)\mathbf{v}(k)$. If there are *n* independent components to be estimated, then the above algorithm is run for *n* times.

4.8.1 Estimation of Basis Vectors of ICA

ICA basis vectors can be estimated in two ways. Using the batch approach (PCA method), the matrix $\hat{\mathbf{A}}$ is the estimated basis vectors and can be expressed as

$$\hat{\mathbf{A}}=\mathbf{E}\mathbf{D}^{1/2}\mathbf{W} \tag{31}$$

where **D** and **E** are already defined by (13) and **W** is the separating matrix. A neural implementation of the algorithm for estimating the basis vectors is also available by minimizing the mean square error and is given by

$$J(\mathrm{Q})=\frac{1}{2}\left\|\mathbf{X}-\hat{\mathbf{X}}\right\|_2^2=\frac{1}{2}\left\|\mathbf{X}-\mathbf{Q}\mathbf{y}\right\|_2^2 \tag{32}$$

where $\mathrm{Q} \rightarrow$ the estimated mixing matrix $(\hat{\mathrm{A}})$ and $\hat{\mathrm{x}} \rightarrow$ the estimated observation matrix. Using the steepest descent approach, the learning rule may be expressed as

$$\mathbf{Q}(k+1) = \mathbf{Q}(k) + \mu(k)[\mathbf{x}(k) - \mathbf{Q}(k)\mathbf{y}(k)]\mathbf{y}^T(k) \tag{33}$$

5. Results

The performance of all the techniques presented in the previous section are tested on a variety of data and the result of each is presented in this section. For each type, the source signal (if available), observed signal and the separated signal are shown. The CPU time of each of the algorithms has been specified along with the number samples for training and the number of iterations taken. For each test case the estimation of the basis vectors of ICA was done using standard PCA. The Performance Index (PI) of the separating algorithm used for each data type is found to be very good. All the simulations were done using Matlab 6.5 on an AMD machine with speed of 1.8GHz.

5.1 Testing

Test 1: Three sinusoids (all deterministic) with frequencies 1000Hz, 300hz and 600Hz have been generated. The sampling frequency chosen was 10,000Hz.The source distribution is known as the signals were artificially generated. In the first case no noise was added to the observed data. ii)In the second case additive white Gaussian noise was added to the observed signals to test the performance of the various algorithms. Three signals, i.e., s1: amplitude modulated signal, s2: frequency signal and s3: uniformly random noise like signal were generated to serve as original signals. The sampling frequency chosen was 10,000 Hz.

Test 2: The sound of induction motors running at different speeds, i.e., 1000rpm, 600rpm and 300rpm were recorded at the chosen sampling frequency 22,050 Hz to serve as original source signals.

Test 3: Voice of two speakers were recorded at the chosen sampling frequency of 22,050 Hz.

5.2 Preprocessing

The source signals(s) in each test case were preprocessed to ensure that each has zero mean and unity variance. The observed signals in each test case were produced by a linear combination of the source signals with a randomly generated mixing matrix. The mixing matrix in all cases must be invertible. The observed signals in each test case were processed to have zero mean and unity variance.

For whitening the observed data in each test case, the whitening matrix was created using the standard PCA. The separating algorithms that work on whitened data are Non-linear PCA, fastICA, Maximum entropy ICA. Natural gradient based mutual information minimization algorithm doesn't require the observed data to be whitened.

5.3 Non-linear PCA

The non-linear function chosen was: $g(t) = \beta \tanh(t/\beta)$.

Test 1:

i) Sinusoids without noise:

The values of $\mu=0.01$, $\beta=1$, and $\gamma=0.9$. Number of samples considered for training = 100.
Number of iterations for proper learning and separation = 190.
The separated signals, source signals and observed signals are shown in Figure 4(a).
CPU time spent: 0.05 sec.
Correlation between source signal and estimated signal: 0.9996,-0.9997,-0.9996.

ii) Gaussian noise was added to the observed data in Test 1 as in the model as $\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{n}(k)$.
The values of μ, β and γ are chosen as: $\mu = 0.0005$, $\beta = 1$, $\gamma = 0.9$.
Number of samples considered for training = 10,000.
Number of iterations(epochs) for proper learning and separation = 500.
Noise in dB (additive Gaussian white) = 30.
The separated signals, source signals and observed signals are shown in Figure 4(b).

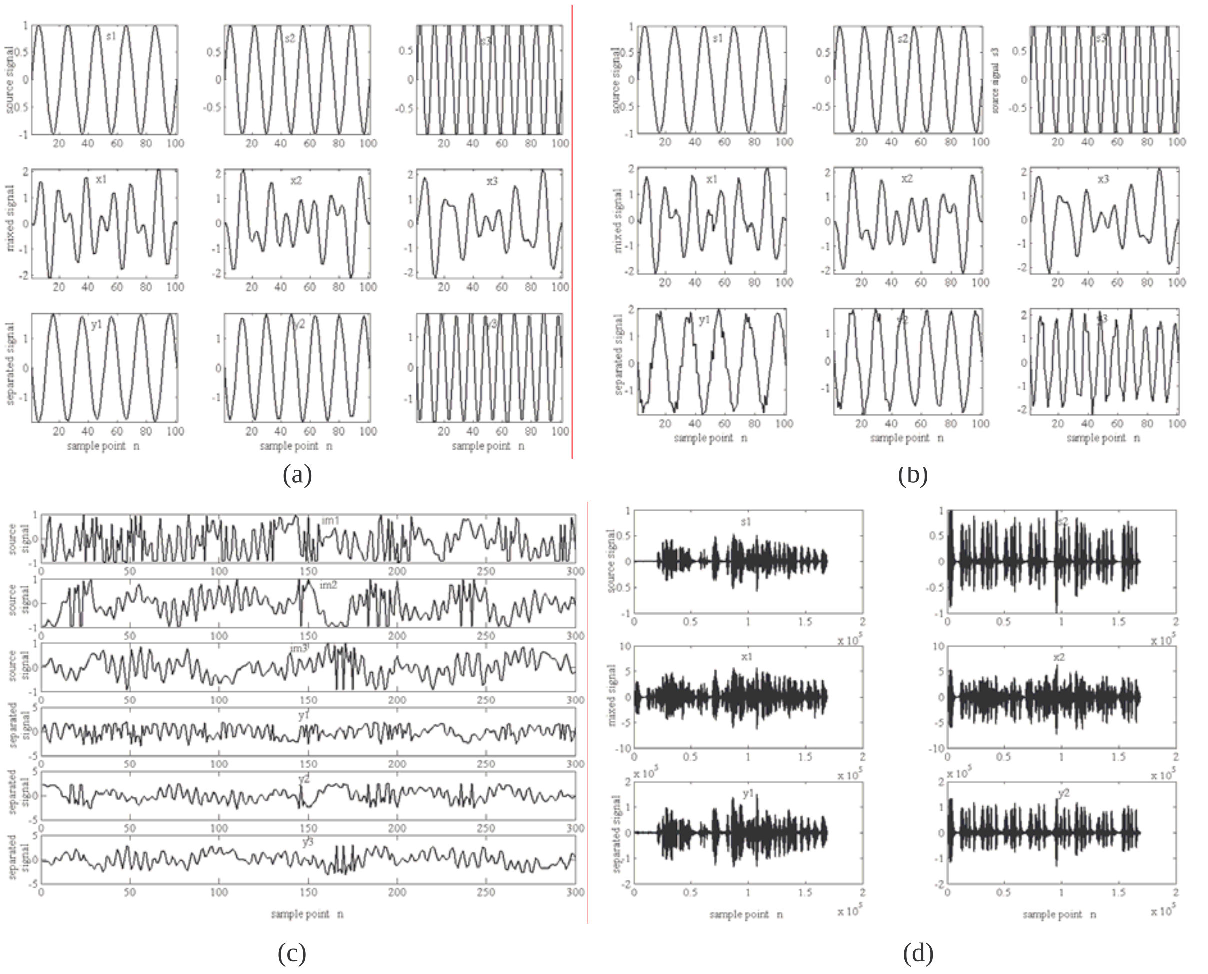


Figure 4. (a) Nonlinear PCA (sinusoids without noise), (b) Nonlinear PCA (sinusoids with 30dB noise), (c) Nonlinear PCA (Three induction machine signals), (d) Nonlinear PCA (Two speech signals).

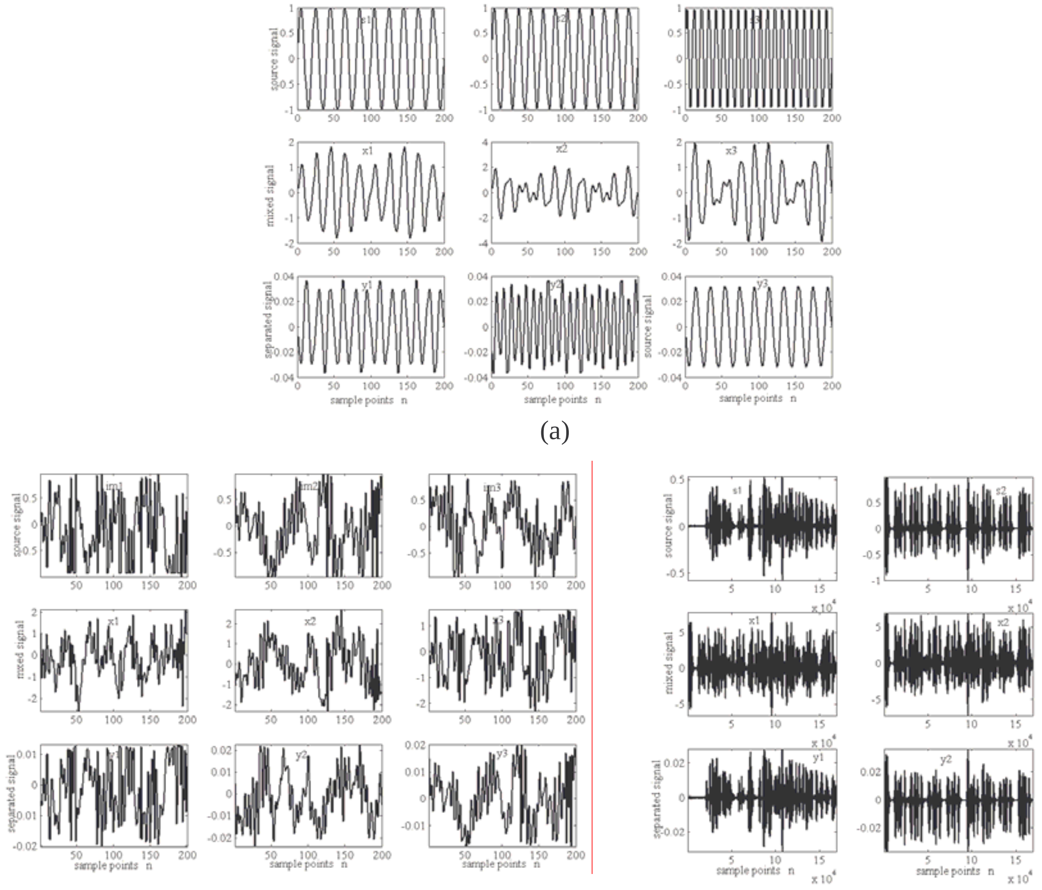
CPU time spent: 15 sec.
Correlation between source signal and estimated signal: -0.9789,-0.9862,-0.9846.

Test 2 (Induction motor data)
The source data is of stationary type. Best separation results were observed for $\mu = 0.001$, $\beta = 1$, $\gamma = 0.9$.
Number of samples considered for training = 3000.
Number of iterations for proper learning and separation = 350.
The separated signals, source signals and observed signals are shown in Figure 4(c).
CPU time spent: 3.15 sec.
Correlation between source signal and estimated signal: -0.9799, -0.9930, -0.9846.

Test 3: (Speech data) Best separation of speech signals were observed for $\mu = 0.1$, $\beta = 2$, $\gamma = 0.9$.
Number of samples considered for training = 169248.
Number of iterations for proper learning and separation =100.
The separated signals, source signals and observed signals are shown in Figure 4(d).
CPU time spent: 19 sec.
Correlation between source signal and estimated signal: -0.9999, 0.9916.

5.4 Maximum Entropy Method

Test 1 (self generated data without noise)
Best separation results were observed when $\eta = 0.09$.



(a)

(b)

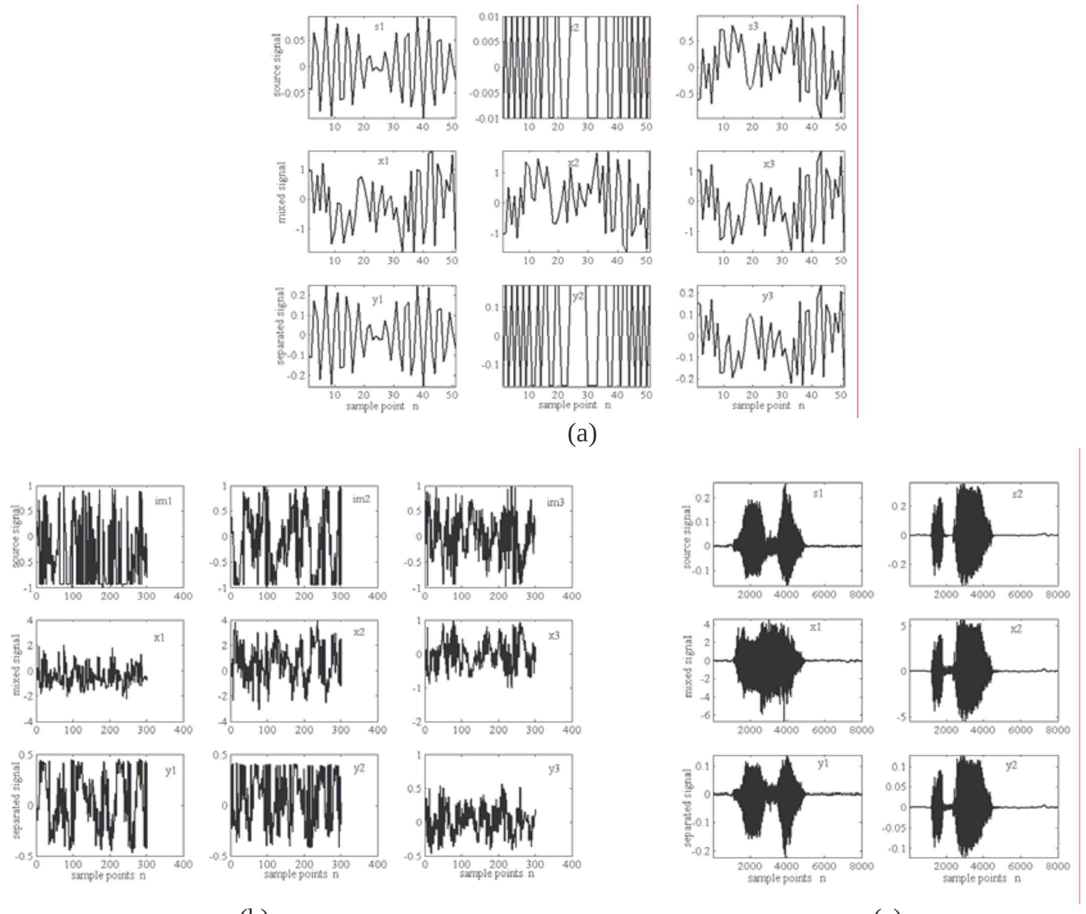
(c)

Figure 5. (a) Maximum Entropy Method (Sinusoids without noise), (b) Maximum Entropy Method (Three induction machine signals), (c) Maximum Entropy Method (Two speech signals)

Number of samples considered for training =1000.
Number of iterations (epochs) for proper learning and separation =100.
The non-linear function chosen was: $z = \tanh(y)$.
The separated signals, source signals and observed signals are shown in Figure 5(a).
CPU time spent: 0.094 sec.
Correlation between source signal and estimated signal (output not in order): -0.9492, -0.9730, -0.9299.

Test 2 (Induction motor data)
Best separation results were observed when η =0.002.
Number of samples considered for training =20000.
Number of iterations for proper learning and separation =1000.
The non-linear function chosen was: $z_i = 1/(1+e^{-y_i})$.
The separated signals, source signals and observed signals are shown in Figure 5(b)
CPU time spent: 19 sec.
Correlation between source signal and estimated signal: (out of order): -0.9981, 0.9972, -0.9967.

Test 3 (Speech data)
Best separation results were observed when η =0.001.
Number of samples considered for training =169248.
Number of iterations for proper learning and separation =400.
The non-linear function chosen was: $z_i 1/(1+e^{-y_i})$.



(a)

(b)

(c)

Figure 6. (a) Kullback-Leibler Divergence Minimization (Self generated signals & noise), (b) Kullback-Leibler Divergence Minimization (Three induction machine signals), (c). Kullaback-Leibler Divergence Minimization (Two speech signals).

The separated signals, source signals and observed signals are shown in Figure 5(c)
CPU time spent: 55 sec.
Correlation between source signal and estimated signal: 0.9999, -1.

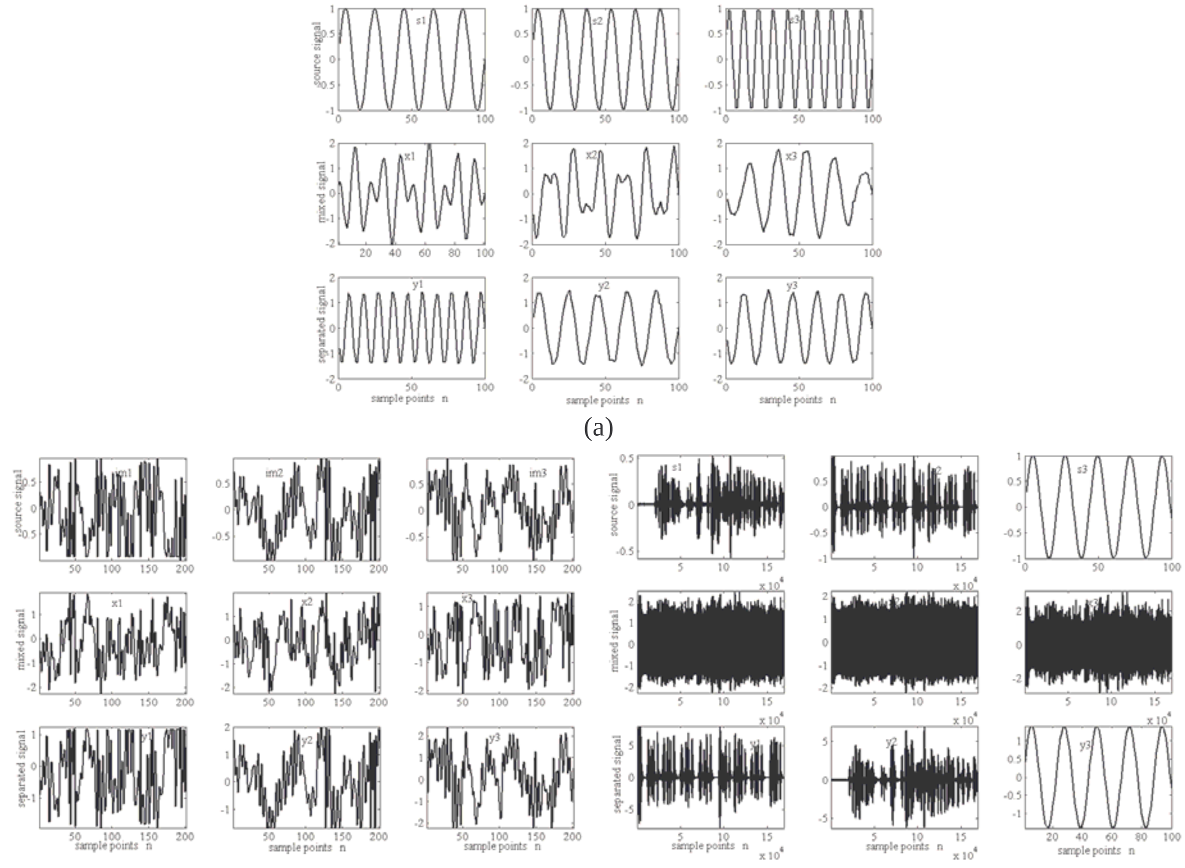
5.5 Minimization of Mutual Information

Test 1

The artificially generated signals were neither whitened nor noise-added.
Best separation results are obtained for the parameter values as $\eta = 0.1$.
Number of samples considered for training =1000.
Number of iterations for proper learning and separation =300.
The separated signals, source signals and observed signals are shown in Figure 6(a).
CPU time spent: 4.6875 sec
Correlation between source signal and estimated signal (out of order): 1, 1,-1.

Test 2 (Induction motor data)

Best separation results are obtained for the parameter values as given by: η =0.1
Number of samples considered for training = 1000.
Number of iterations for proper learning and separation =300.
The separated signals, source signals and observed signals are shown in Figure 6(b)
CPU time: 4.65 sec.
Correlation between source signal and estimated signal (not in order): -0.9976, -0.9946, -0.9998.



(a)

(b)

(c)

Figure 7(a). FastICA Method (Sinusoids with 30dB noise), (b) FastICA Method (Three induction machine signals), (c) FastICA Method (Two speech signal & one sinusoidal signal)

Test 3 (Speech data)

Best separation results are obtained for the parameter values: $\eta = 0.002$
Number of samples considered for training = 8000.
Number of iterations for proper learning and separation = 100
The separated signals, source signals and observed signals are shown in Figure 6(c)
CPU time: 10 sec.
Correlation between source signal and estimated signal: **-**0.9978, 0.9978.

5.6 Fast Fixed Point ICA

Test 1

Gaussian noise was added to the observed data in Test 1 as in the model: $\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{n}(k)$**.**
Best separation results are obtained when 30dB noise.
Number of samples considered for training =20000.
Number of iterations for proper learning and separation =11.
The separated signals, source signals and observed signals are shown in Figure 7(a)
CPU time: 0.062 sec.
Correlation between source signal and estimated signal (out of order): 0.9979, -0.9992, -0.9993.

Test 2 (Induction motor data)

Best separation results are obtained when:
Number of samples considered for training = 45431.
Number of iterations (epochs) for proper learning and separation =14.
The separated signals, source signals and observed signals are shown in Fig.7(b).

CPU time: 0.125 sec.
Correlation between source signal and estimated signal: -1, 1, 0.9997.

Test 3 (Speech data and a sinusoid)
Best separation results are obtained when:
Number of samples considered for training = 169248 .
Number of iterations for proper learning and separation =11.
The separated signals, source signals and observed signals are shown in Fig.7(c).
CPU time: 0.282 sec.
Correlation between source signal and estimated signal: (out of order): -0.9999, 1, 1.

6. Discussions

From the results it is seen that each of the algorithms perform reasonably well when applied on different types of data. The performance of a separation algorithm depends on a number of factors like distribution of source signals, learning rate parameters, nonlinearities etc. In our experiments, fastICA algorithm showed best results in terms of convergence time and separation ability.

Nonlinear PCA produces very good results at the cost of higher convergence time. It is sensitive to the values of μ, β and γ. Whitening the data is essential for nonlinear PCA-type algorithm to achieve good separation results except when the mixing matrix is ill-conditioned. The reason being without whitening, the algorithm still largely responds to the 2nd order statistics despite the nonlinear transformations. Typical nonlinear function chosen is $g(t) = \beta\,\tanh(t/\beta)$, where $g(t) = \partial f(t)/\partial t$ and $f(t) = \beta^2\, 1n(t/\beta)$ .The value of β plays an important role in the process of separation. $g(t)$ is linear and $f(t)$ is quadratic, while (t/β) lies in the range [-1, 1]. Hence all training data lying outside the range will be rejected by the algorithm. The optimum range of β has been found to vary between 0.7 and 10. The advantage of nonlinear PCA is that it can be realized fairly easily using hardware by slightly modifying the standard PCA rule.

Natural gradient based mutual information minimization technique is particularly useful for solving iterative optimization problems. The purpose of the nonlinearity used in the simplest form of the algorithm discussed in this paper is to incorporate higher order statistics necessary to solve the BSS problem. The performance of the algorithm strongly depends on the activation function which in turn depends upon the pdf's of the source signals. These algorithms are very successful when prior information about the sources i.e., super-Gaussian or sub-Gaussian, are available. For instance: if all the sources are super-Gaussian, then nonlinear function of type $f_i(y_i) = \alpha_i y_i + tanh(\beta_i y_i)$ and if all sources are sub-Gaussian in nature, then nonlinear function of type $f_i(y_i) = \alpha_i y_i + y_i^3$ are suitable for separation. α_i and β_i are small positive constants with ($0 \le \alpha_i \le 1$). However, if sensor signals are mixtures of both sub-Gaussian and super-Gaussian source signals the algorithm may fail to separate these signals reliably in which case approaches like the one in [66] may be followed. Though these algorithms demonstrate excellent performance for noiseless signal mixtures, their performance deteriorates with noisy measurements. The algorithms based on natural gradient based mutual information minimization technique are globally stable and exhibit good convergence behaviour for small learning rates. Stability is achieved for a range of output variable where the activation function is positive. One disadvantage with the technique is that mutual information is difficult to estimate requiring the estimates of the pdfs of the source signals. Approximation of the Edgeworth or Gram-Charlier expansion of the output variable solves the problem in some cases with some a priori information about the sources.

Maximum Entropy or Infomax algorithm is based on maximization of entropy of the ICA network which is under some conditions equivalent to maximum likelihood approach. The Infomax algorithm can separate only signals with positive kurtosis where as Extended Infomax [35] is capable of separating a mixture of sub-Gaussian and super-Gaussian signals. Infomax fails when more than one source is Gaussian and the mixing matrix is ill-conditioned. The convergence of these algorithms greatly depends upon the selection of nonlinearity. In our experiment the speech signals and the induction machine data were best separated when the nonlinear function was $z_i = 1/(1+e^{-yi})$, whereas sinusoidal signals were separated with long convergence time when nonlinear function was $z_i = \tanh(y_i)$.

The convergence of fastICA is cubic under the assumption of ICA data model which is very fast compared to all the other adaptive algorithms discussed in this paper. Contrary to gradient based algorithms, fastICA does not require any learning rate parameter and hence is easy to use. The algorithm finds directly the independent components of any non-Gaussian distribution (one source can be Gaussian) using any contrast function without requiring any prior knowledge on the distribution of the sources. It requires the mixing/demixing matrix to be orthogonal which is quite impractical for noisy measurements.

The commonality in all the algorithms, i.e., Infomax or Maximum Entopy, Natural gradient based Mutual Information Minimization, Nonlinear PCA, fastICA is the use of a nonlinear function to achieve separation. All algorithms are adaptive in nature except fastICA which is suitable for batch processing. Except Natural gradient based mutual information minimization method all other methods employ prewhitening. FastICA and Nonlinear PCA techniques have the ability to separate sources from a broad class of distributions with no *a priori* information about distribution of sources. The convergence properties of the adaptive algorithms i.e., Infomax or Maximum Entropy, Natural gradient based Mutual Information Minimization, Nonlinear PCA depend largely on the sources distribution and non-linearities used. The computational complexity for the adaptive algorithms and the fastICA algorithm are shown in Table 1.

Table 1. Computational Complexity of Algorithms

Sl. No	***Algorithm***	***Computational Complexity*** (*n = number of sources = number of sensors*)
1.	Natural gradient based Mutual Information Minimization	$(2n^3 + 3n^2)/\,sample$
2.	Maximum Entropy (Infomax)	$(2n^3 + 8n^2 + 2n)/\,sample$
3.	Nonlinear PCA	$(n^3 + 7n^2 + n + 2)/\,sample$
4.	FastICA (Batch)	$(n^3 + 6n^2 + 2n) \times (number\ of\ samples\ per\ block)$

7. Conclusion

In this paper we have evaluated the performance of Fast Fixed Point scheme and a class of neural algorithms for performing ICA [41]. The fast fixed point ICA type algorithms have been applied to large scale practical problems [18][42].

In the complete ICA network as shown in Figure 2, it may be advisable to use neural learning only for the vital separation process. For whitening and estimation of basis vectors standard numerical methods will suffice. The poor performance in terms of computation time and separation results may be attributed to the presence of noise and statistical dependency amongst the original source signals. The separation results could probably be improved by adding to the basic network structure, another separating stage where a different nonlinearity may be employed to introduce higher order statistics.

BSS techniques, at the present state of art, do not exhibit satisfactory performance in noisy environment. Whether to go for noise modeling or to use higher order cummulants to combat the problem is an open question and provides scope for further research in the direction of noisy-ICA [43]. Further, issues like propagation delays, time varying mixture [67], convolutive mixture [44] and non-stationary sources [68] make the BSS problem more complicated in real time. Extension of the present ICA algorithms to address these issues provides scope for future directions in the ICA-BSS research.

References

[1] J. Karhunen, A. Hyvärinen, R. Vigario, J. Hurri, and E. Oja. Applications of neural blind separation to signal and image processing. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'97)*, pages 131-134, Munich, Germany, 1997.

[2] L.De Lathauwer,B. de Moor, J. Vandewalle. Fetal electrocardiogram extraction by source subspace separation. In Proc. HOS’95,pages 134-8, Aiguablava,Spain, June 1995.

[3] S. Makeig, A.J. Bell, T.-P. Jung, and T.-J. Sejnowski. Independent component analysis of electroencephalographic data. In *Advances in Neural Information PRocessing Systems 8*, pages 145-151. MIT Press, 1996.

[4] R. Vigário, V. Jousmäki, M. Hämäläinen, R. Hari, and E. Oja. Independent component analysis for identification of artifacts in magnetoencephalographic recordings. In *Advances in Neural Information Processing 10 (Proc. NIPS'97)*, pages 229-235, Cambridge, MA, 1998. MIT Press.

[5] I.T. Jolliffe. *Principal Component Analysis.* Springernegativerlag, 1986.

[6] J. Karhunen and J. Joutsensalo. Generalizations of principal component analysis, optimization problems, and neural networks. *Neural Networks*, 8(4):549-562, 1995.

[7] Bell and T. Sejnowski, “An information-maximization approach to blind separation and blind deconvolution,” *NeuralComputation*, vol.7, pp.1129–1159, 1995.

[8] J.J. Atick. Entropy minimization: A design principle for sensory perception? *International Journal of Neural Systems*, 3:81-90, 1992. Supp. 1992.

[9] A.J. Bell and T.J. Sejnowski. The 'independent components' of natural scenes are edge filters. *Vision Research*, 37:3327-3338, 1997.

[10] Belouchrani and J.-F. Cardoso. Maximum likelihood source separation by the expectation-maximization technique: deterministic and stochastic implementation. In *Proc. NOLTA*, pages 49-53, 1995.

[11] J.-F. Cardoso. Infomax and maximum likelihood for source separation. *IEEE Letters on Signal Processing*, 4:112-114, 1997.

[12] M. Wax and T. Kailath. Detection of signals by information-theoretic criteria. *IEEE Trans. on ASSP*, 33:387-392, 1985.

[13] R. Vigário. Extraction of ocular artifacts from EEG using independent component analysis. *Electroenceph. clin. Neurophysiol.*, 103(3):395-404, 1997.

[14] Oja. A simplified neuron model as a principal component analyzer. *J. of Mathematical Biology*, 15:267-273, 1982.

[15] E. Oja. Neural networks, principal components, and subspaces. *Int. J. on Neural Systems*, 1:61-68, 1989.

[16] E. Oja. The nonlinear PCA learning rule in independent component analysis. *Neurocomputing*, 17(1):25-46, 1997.

[17] Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483-1492, 1997.

[18] M. McKeown, S. Makeig, S. Brown, T.-P. Jung, S. Kindermann, A.J. Bell, V. Iragui, and T. Sejnowski. Blind separation of functional magnetic resonance imaging (fMRI) data. *Human Brain Mapping*, 6(5-6):368-372, 1998.

[19] Hyvärinen. A family of fixed-point algorithms for independent component analysis. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'97)*, pages 3917-3920, Munich, Germany, 1997.

[20] N. Murata and S. Ikeda, “An on-line algorithm for blind source separation on speech signals,” *Proceedings of 1998 International Symposium on Nonlinear Theory and Its Application (NOLTA ’98)*, vol.3, pp.923–926, Sep. 1998.

[21] H. Saruwatari, S. Kurita, K. Takeda, F. Itakura, and K.Shikano, “Blind source separation based on subband ICA and beamforming,” *Proc. ICSLP2000*, vol.3, pp.94–97, Oct.2000.

[22] Belouchrani, K. Abed Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique based on second order statistics. *IEEE Trans. on S.P.*, 45(2):434-44, 1997.

[23] L. De Lathauwer, B. De Moor, and J. Vandewalle. A technique for higher-order-only blind source separation. In *Proc. ICONIP*, Hong Kong, 1996.

[24] Nikias and J. Mendel. Signal processing with higher-order spectra. *IEEE Signal Processing Magazine*, pages 10-37, July 1993.

[25] J.-F. Cardoso. Iterative techniques for blind source separation using only fourth-order cumulants. In *Proc. EUSIPCO*, pages 739-742, Brussels, Belgium, 1992.

[26] J.-F. Cardoso. Source separation using higher order moments. In *Proc. ICASSP'89*, pages 2109-2112, 1989.

[27] P.Common, “Independent component analysis, a new concept?,” *Signal Processing*, vol.36, pp.287–314, 1994.

[28] J.-F. Cardoso and P. Comon. Independent component analysis, a survey of some algebraic methods. in *Proc. ISCAS'96*, volume 2, pages 93-96, 1996.

[29] J. Karhunen, P. Pajunen, and E. Oja. The nonlinear PCA criterion in blind source separation: Relations with other approaches. *Neurocomputing*, 22:5-20, 1998.

[30] J. F. Cardoso. Entropic contrasts for source separation. In S. Haykin, editor, *Adaptive Unsupervised Learning*. 1999.

[31] Moreau and O. Macchi. New self-adaptive algorithms for source separation based on contrast functions. In *Proc. IEEE Signal Processing Workshop on Higher Order Statistics*, pages 215-219, Lake Tahoe, USA, June 1993.

[32] S.-I. Amari. Neural learning in structured parameter spaces -- natural riemannian gradient. In *Advances in Neural Information Processing 9*, pages 127-133. MIT Press, Cambridge, MA, 1997.

[33] S. Amari, A. Cichocki, and H.H. Yang. A new learning algorithm for blind source separation. in *Advances in Neural Information Processing 8*, pages 757-763. MIT Press, Cambridge, MA, 1996.

[34] Hyvärinen. Independent component analysis by minimization of mutual information. Technical Report A46, Helsinki University of Technology, Laboratory of Computer and Information Science, 1997.

[35] T.-W. Lee, M. Girolami, and T. J. Sejnowski. Independent component analysis using an extended infomax algorithm for mixed sub-Gaussian and super-Gaussian sources. *Neural Computation*, pages 609-633, 1998.

[36] Hyvärinen. Purely local neural principal component and independent component learning. In *Proc. Int. Conf. on Artificial Neural Networks*, pages 139-144, Bochum, Germany, 1996.

[37] D.-T.Pham. Blind separation of instantaneous mixture of sources via an independent component analysis. Research report RT 119, LMC IMAG, Grenoble, France, 1994.

[38] D.-T.Pham. Blind separation of instantaneous mixture of sources via an independent component analysis. IEEE Trans. On Sig. Proc., 44(11):2768-2779, Nov.1996.

[39] S. Haykin. *Neural Networks*. Pearson Education Asia, Second Edition, 1999.

[40] N. Delfosse and P. Loubaton. Adaptive blind separation of independent sources: a deflation approach. *Signal Processing*, 45:59-83, 1995.

[41] J. Karhunen, E. Oja, L. Wang, R. Vigario, and J. Joutsensalo. A class of neural networks for independent component analysis. *IEEE Trans. on Neural Networks*, 8(3):486-504, 1997.

[42] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on PAMI*, 11:674-693, 1989.

[43] Cichocki, S. C. Douglas, and S.-I. Amari. Robust techniques for independent component analysis with noisy data. *Neurocomputing*, 22:113-129, 1998.

[44] L. Parra and C. Spence, "Convolutive blind separation of non-stationary sources," *IEEE Trans. Speech & Audio Process.*, vol.8, pp.320–327, 2000.

[45] L. Tong, R.-W. Liu, V.C. Soon, and Y.-F. Huang. Indeterminacy and identifiability of blind identification. *IEEE Trans. on Circuits and Systems*, 38, 1991.

[46] H.B. Barlow. Unsupervised learning. *Neural Computation*, 1:295-311, 1989.

[47] J.-F. Cardoso. Blind signal separation: statistical principles. *Proceedings of the IEEE*, 9(10):2009-2025, 1998.

[48] Cichocki and R. Unbehauen, "Robust neural networks with on-line learning for blind identification and blind separation of sources," *IEEE Trans. Circuits and Systems I*, vol.43, no.11, pp.894–906, 1996.

[49] S.-I. Amari and A. Cichocki, "Adaptive blind signal processing - neural network approaches," *Proceedings of the IEEE*, vol.86, (9), pp.2026-2048, 1998.

[50] T. W. Lee, *Independent Component Analysis*, Kluwer Academic Publishers, 1998.

[51] A. Mansour, A. K. Barros and N. Ohnishi, "Blind separation of sources: Methods, assumptions and applications,"*IEICE Trans. On Fundamental of Electronics, Communications and Computer Sciences*, vol.E83-A, pp. 1498-1512, 2000.

[52] S. Van Gerven, "Adaptive noise cancellation and signal separation with applications to speech enhancement," *Ph.D dissertation*, Catholic University Leuven, Leuven, Belgium, Mar. 1996.

[53] Martens SM, Rabotti C, Mischi M, Sluijter RJ, "A robust fetal ECG detection method for abdominal recordings," *Physiol Meas*. 2007 Apr; 28(4): 373-88. Epub 2007 Mar 7.

[54] Iriarte J, Urrestarazu E, Valenica M, Alegre M, Malanda A, Viteri C, Artieda J, "Independent Component Analysis as a tool to eliminate artifacts in EEG: A quantitative study," *J Clin Neurophysiol*. 2003 Jul-Aug; 20(4): 249-57.

[55] James CJ, Hesse CW, “Independent Component Analysis for biomedical signals,” *Physiol Meas*. 2005 feb; 26(1): R15-39.

[56] Urrestarazu E, Iriarte J, “Independent Component Analysis 9ICA) in the study of electroencephalographic signals,” *Neurologia*. 2005 Jul-Aug; 20(6): 299-310.

[57] Fiori S, “Overview of independent component analysis technique with an application to synthetic aperture radar (SAR) imagery processing,” *Neural Netw*. 2003 Apr-May; 16(3-4): 453-67.

[58] A.-J. van der Veen, S.Talvar, and A. Paulraj, “A subspace approach to blind space-time signal processing for wireless communication systems, “*IEEE Trans. Signal Processing*, vol.45, pp. 173-190, Jan. 1997.

[59] A. Kaban and M. Girolami, “Clustering of text documents by skewness maximization,”in *Proc. Int. workshop on Independent Component Analysis and blind Signal Separation*, 2000, pp. 435-440.

[60] P. C. Yuen and J. h. Lai, “Face representation using independent component analysis,” Pattern Recognit., vol.35, pp. 1247-1257, 2002.

[61] A. Hyvarinen and E. Oja, “Independent Component Analysis: algorithms and applications,” *Neural Netw*. Vol.13, pp. 411-430, 2000.

[62] C. Huang, T.Chen, S. Li, E. Chang and J. Zhou, “Analysis of speaker variability,” in *Proc. Eur. Conf. Speech Communication Technology (EUROSPEECH)*, 2001, pp.1377-1380.

[63] J. H. Lee, H.-Y. Jung, T.-W. Lee and S. Y. Lee, “Speech feature extraction using independent component analysis,” in *Proc. Int. Conf. Acoustic Speech Signal Processing (ICASSP)*, vol.3, 2000, pp. 1631-1634.

[64] G.-J. Jang, T.-W. Lee and Y.-H. Oh, “Learning statistically efficient features for speaker recognition,” *Neurocomputers*, no.49, pp.329-348, 2002.

[65] T.-W. Lee and M. S. Lewicki, “Unsupervised classification, segmentation and denoising of images using ICA mixture models,” *IEEE Trans. On Image Processing*, January-2000.

[66] S. C. Douglas, A. Cichocki and S. Amari, “Multichannel blind separation and deconvolution of sources with arbitrary distributions,” in *Neural Networks for signal processing , Proc. 1997 IEEE Workshop (NNSP-97)*, Sept. 1997, pp.436-445.

[67] Mihai Enescu and Visa Koivunen, “Tracking time varying mixing system in blind separation,”*Proc. Of the IEEE Workshop on Sensor Array and Multichannel Signal Processing*, 16-17 March, 200, Page(s): 291-295.

[68] Robust Aichner, Herbert Buchner, Shoko Araki, Shoji Makino, “Online time-domain blind source separation of non-stationary convolved sources,” *Intl. Symposium on Independent Component Analysis and Blind Signal Separation*, (ICA),Nara, Japan, April-2003.

[69] S. Amari, A. hyvarinen, S.-Y. Lee, and V. D. Sanchez, “Blind signal separation and independent component analysis,” *Neurocomputing*, 49(12): 1-5, 2002.

[70] S. Choi, A. Cichocki, H.-M. Park, and S.-Y. Lee, “Blind source separation and Independent component analysis: A Review,” Neural Information Processing- Letters and Reviews, vol.6, No.1, January 2005.