KURTOSIS APPROACH FOR NONLINEAR BLIND SOURCE SEPARATION

Vu A. Duong[1] and Allen R. Stubberud[2]
[1]Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109
[2]Department of EECS
University of California, Irvine
Irvine, CA 92127
Vu.A.Duong@jpl.nasa.gov, arstubbe@uci.edu

Abstract: In this paper, we introduce a new algorithm for blind source signal separation for post-nonlinear mixtures. The mixtures are assumed to be linearly mixed from unknown sources first and then distorted by memoryless nonlinear functions. The nonlinear functions are assumed to be smooth and can be approximated by polynomials. Both the coefficients of the unknown mixing matrix and the coefficients of the approximated polynomials are estimated by the gradient descent method conditional on the higher order statistical requirements. The results of simulation experiments presented in this paper demonstrate the validity and usefulness of our approach for nonlinear blind source signal separation
Keywords: Independent Component Analysis, Kurtosis, Higher order statistics

1. Introduction

As it has been shown in the literature [2,4,7,12,18] that linear ICA has been successful in separating linear mixtures of the blind source signals. However, most of the mixing mechanisms in our real life follow the non-linear manners in which linear ICA algorithm is unable to separate the source signals. One example is source signals, which propagate in the ocean, are likely to be non-linearly mixed due to the inhomogeneous of the water layers in the ocean. Intrinsic nonlinearities in microphone that has been used in speech recording, is another example of nonlinear mixing of source signals. To cope with these practical problems, there is a need for nonlinear ICA. Nonlinear independent component analysis can be defined as a nonlinear mapping **g** which takes a random vector **x** into a random vector **y** = **g**(**x**) with a factorizable density, *i.e.* a product of its marginal densities, which implies that the components of **y** are independent.

Non-linear ICA, in general, is an intractable problem, since the indeterminacies in the separating solutions are much more severe than in the linear case. For example, if X and Y are two independent random variables, then any variable f(X) and g(Y), where f and g are one-to-one functions, are also independent. Thus if there is a linear algorithm, which is able to find some independent outputs, they may be different from the original source signals. Recently, researchers have started addressing the formulation to the nonlinear mixing models. Burel [3] proposed a nonlinear ICA algorithm for parametric nonlinearity. The nonlinearity in [20] is assumed to depend on unknown coefficients. Pajunen and Karhunen [17] introduced the idea of using self-organizing map (SOM) for nonlinear ICA. However, this approach can be successfully applied only to the separation of sources having uniform distributions. Taleb and Jutten [20] have applied a maximum likelihood approach for separating sources from post-nonlinear mixtures. Yang, Amari and Cichocki have extended their natural gradient to this case using an MLP network.

Due to the complexity of the general nonlinear ICA, this paper focuses on a subclass of non-linear ICA called post non-linear ICA. Section 1 reviews the current development of nonlinear ICA. Section 2 is the model for nonlinear ICA. Section 3 is the algorithm and section 4 is the simulation results.

2. Non-linear Independent Component Analysis Model

The simple post non-linear mixing and unmixing models are shown in figure 1. The mixing model consists of two parts: a linear mixing part and a memoryless invertible nonlinear transfer part. The unmixing system is the inverse sequence of the mixing system in which the non-linear transfer function is inverted in each channel at first and the results are unmixed by a linear transform **W**. The source signals are recovered if each operation in the unmixing sequence is the inverse of the corresponding operation in the mixing sequence.
Let $\mathbf{s}(n) = [s_1(n)\ s_2(n)\ \ldots s_N(n)]^T$ is an Nx1 column vector of N independent sources at time n, **A** is NxN matrix, $\mathbf{f} = [f_1\ f_2\ \ \ f_N]$ is Nx1 vector of nonlinear functions and **f** operates componentwise and $\mathbf{x}(n) = [x_1(n),\ x_2(n),\ldots,\ x_N(n)]^T$ is a Nx1 column vector of mixed signal observed at time n.

The mixing model for the post nonlinear ICA can be mathematically represented as

$$\mathbf{x}(n) = \mathbf{f}(\mathbf{A}\mathbf{s}(n)), \qquad (1)$$

and the umixing model is represented as

$$\mathbf{u}(n) = \mathbf{W}\mathbf{g}(\mathbf{x}(n)), \qquad (2)$$

where $\mathbf{u}(n) = [u_1(n), u_2(n), \ , u_N(n)]^T$ is an Nx1 vector of the estimated sources at time n, $\mathbf{W}$ ideally is an inverted matrix of $\mathbf{A}$ and $\mathbf{g}$ ideally is inverted mapping of f componentwise.

Model (1) can be written in term of componentwise as

$$x_i = f_i\left(\sum_{j=1}^{N} a_{ij} s_j\right), \quad i = 1, \ldots, N \qquad (3)$$

Thus the sources s_j, j=1,...,N are first mixed linearly according to the basic linear ICA and followed by applying a nonlinear functions f_i to get the final observations x_i. It has been shown [28] that for the post nonlinear mixtures, the indeterminacies are usually the same as for standard linear ICA that are the sources can be separated up to the scaling, permutation, and sign indeterminacies under weak conditions on the mixing matrix $\mathbf{A}$ and source distributions. The post-nonlinearity assumption is useful and reasonable in many signal-processing applications because the model can be thought as a nonlinear distortion of a sensor system.

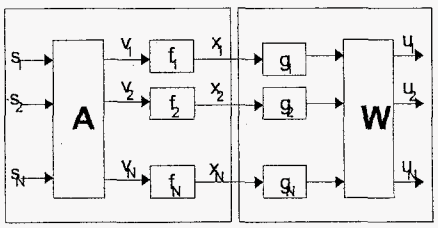


Figure 1: The mixing and unmixing models for Post Nonlinear ICA

3. Separation of post-nonlinear mixtures using *mutual information*

Taleb and Jutten have developed a method for post-nonlinear ICA. Their results have been reported in [20]. A summary of their method is in the following.

A separation procedure for the post-nonlinear mixtures (1) is in two subsequent stages:

1. A nonlinear stage, which cancels the nonlinear distortion f_i, i=1,...,N. The objective of this part is to estimate nonlinear functions $g_i(\theta_i, u)$. The parameters θ_i for each nonlinear g_i are adjusted so that the cancellation is achieved.
2. A linear stage that separate the approximately linear mixture v, obtained by the nonlinear stage. This can be achieved by learning an NxN separating matrix $\mathbf{W}$ for which components of the output vector $\mathbf{u} = \mathbf{W}\mathbf{v}$ of the separating system are statistically independent.

To achieve the goals for both stages, minimization of mutual information $I(\mathbf{u})$ between the components $u_1,\ldots,u_N$ of the output vector is used as the objective function for both stages. For the linear part, minimization of the mutual information leads to the Bell-Sejnowski algorithm [2].

$$\frac{\partial I(\mathbf{u})}{\partial \mathbf{W}} = (\mathbf{W}^T)^{-1} - \Psi(\mathbf{u})\mathbf{x}^T \qquad (4)$$

where component ψ_i of the vector ψ are score functions of components u_i of the output vector $\mathbf{u}$,

$$\psi_i(u_i) = -\frac{d}{du_i}\log p_i(u_i) = \frac{\frac{\partial p(u_i)}{\partial u_i}}{p(u_i)} \qquad (5)$$

here $p(u_i)$ and $\frac{\partial p(u_i)}{\partial u_i}$, i=1,...,N are the probability density functions and their derivatives, respectively. The probability $p(u_i)$, however, are not available, they can be estimated using the kernel method [wolfgang],

$$\hat{p}_y(u_i) = \frac{1}{hT}\sum_{k=1}^{T} K(u_i - y(k)) \qquad (6)$$

where $y(k)$ is the observation, and K is a kernel function.

For the nonlinear stage, the learning rule can be derived using the gradient method,

$$\frac{\partial I(\mathbf{u})}{\partial \theta_k} = -E\left\{\frac{\partial \log\left|g_k'(\theta_k, x_k)\right|}{\partial \theta_k}\right\} - E\left\{\sum_{i=1}^{N} \psi_i(u_i) w_{ik} \frac{\partial g_k(\theta_k, x_k)}{\theta_k}\right\} \qquad (7)$$

where x_k is the kth component of the input vector, w_{ik} is the element at row i and column k of the matrix $\mathbf{W}$, and g_k' is the derivative of the kth nonlinear function g_k. The detailed learning rule particularly depends on the specific parameter form of the choosing nonlinear mapping $g_k(\theta_k, x_k)$. In [20], a multilayer perceptron network is used for modeling the nonlinear functions $g_k(\theta_k, x_k)$, k=1,...,N.

In the linear ICA, it suffices that the score function in (5) is theoretically the right type for achieving separation. However, their estimation is critical for the performance of the separation of both linear and nonlinear models. By using the kernel estimator in (6), the score function in (5) can be adaptively

Table 1: A list of the kernel functions

Kernel	K(u)
Uniform	$\frac{1}{2}, \quad \lvert u\rvert \le 1$
Triangle	$1-\lvert u\rvert, \quad \lvert u\rvert \le 1$
Epanechnikov	$\frac{3}{4}(1-u^2), \quad \lvert u\rvert \le 1$
Quartic	$\frac{15}{16}(1-u^2)^2, \quad \lvert u\rvert \le 1$
Triweight	$\frac{35}{32}(1-u^2)^3, \quad \lvert u\rvert \le 1$
Gaussian	$\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}u^2)$
Conius	$\frac{\pi}{4}\cos(\frac{\pi}{2}u), \quad \lvert u\rvert \le 1$

estimated from the output vector **u** and hence it helps to improve the performance of the separation

4. Separation of post-nonlinear mixtures using higher order statistics

In the previous section, we introduced a method of separation a post-nonlinear ICA based on the minimization of the mutual information between components of the output vector. In this section, we are going to develop a new method for separating a post-nonlinear ICA but based on the higher order statistics, particularly kurtosis. While the mutual information approach achieves the separation by estimating the probability density functions of the components of the output vector, the higher order statistical approach achieves its separation by using the fourth order statistics of the components of the output vector.

As in the mutual information approach, the separation procedure for the post-nonlinear mixtures using higher order statistics also consists of two subsequent stages:

1. A nonlinear stage that is to estimate nonlinear functions g_i, the inverse of the nonlinear distortion f_i, $i=1,\ldots,N$. In addition, we make a further assumption that the nonlinear functions f_i are smooth enough so that they can be approximated by polynomials of Qth-order. This nonlinear stage may be described as

$$f_i(u_i) = \sum_{j=1}^{Q} f_{ij} u_i^{j-1} \qquad (8)$$

The inverse g_i of the f_i in equation (9) is in general complicated to determine. Therefore, it is assumed that the inverse may be approximately by the pth-order polynomials as follow:

$$g_i(u_i) = \sum_{j=1}^{P} g_{ij} u_i^{j-1} \qquad (9)$$

2. A linear stage that separates the approximately linear mixture **v** obtained by the nonlinear stage. This can be achieved by learning an NxN separating matrix **W** for which components of the output vector **u** = **Wv** of the separating system are statistically independent.

To achieve the goals for both stages, minimization of the fourth order cummulant (kurtosis) of the components $u_1,\ldots,u_N$ of the output vector is used as the objective function for both stages.

The essential idea of using kurtosis for the separation criterion is as follow: according to the central limit theorem, the distribution of a sum of independent random variables asymptotically converges to a Gaussian distribution, under certain conditions. Loosely speaking, a sum of two independent random variables usually has a distribution that is closer to Gaussian than any of the two original random variables [13]. Hence, the non-Gaussianity is used as a measure to determine the independency of a component of a random vector.

In the basic linear ICA problem, the data **x** is a linear combination of independent components s_i

$$\mathbf{x} = \mathbf{A}\mathbf{s}$$

Assume that all the independent components s_i have identical distributions. Consider a random variable $y = \mathbf{w}^T\mathbf{x}$, where y is a scalar, $\mathbf{w}$ is a vector to be determined, and $\mathbf{x}$ is the observation vector. Random variable y can be written in term of $\mathbf{s}$ as $y = \mathbf{w}^T\mathbf{A}\mathbf{s}$.. Thus y is a certain linear combination of the s_i, with coefficients given by $\mathbf{w}^T\mathbf{A}$. If $\mathbf{w}$ were one of the rows of the inverse of $\mathbf{A}$, the linear combination of $\mathbf{w}^T\mathbf{x}$ would equal to one of the component s_i.

Now varying the coefficients in $\mathbf{w}^T\mathbf{A}$ and observe how the distribution of $y = \mathbf{w}^T\mathbf{A}\mathbf{s}$ changes. Since y is a linear combination of independent random variable s_i, y is theoretically more Gaussian than any of the s_i and becomes least Gaussian when it in fact equals to one of the s_i.

However, the original vector $\mathbf{s}$ is unavailable, therefore instead of varying $\mathbf{w}^T\mathbf{A}$, we vary the coefficients of $\mathbf{w}$ since $y = (\mathbf{w}^T\mathbf{A})\mathbf{s} = \mathbf{w}^T\mathbf{x}$.
Therefore the vector $\mathbf{w}$ may be determined by maximizing the non-gaussianity of $\mathbf{w}^T\mathbf{x}$ which give us one of the independent components.

To measure the non-Gaussianity of a random variable y, we use the kurtosis, which is the fourth-order cumulant of a random variable. The kurtosis of a random variable y, denoted as kurt(y), is defined as,

$$kurt(y) = E\left\{y^4\right\} - 3\left(E\left\{y^2\right\}\right)^2 \quad (10)$$

here we assume that y is a zero mean random variable. The kurtosis of random variable y is simpler if y is further assumed to have unit variance, that is $E\{y^2\}=1$. Then the kurtosis of y becomes

$$kurt(y) = E\left\{y^4\right\} - 3 \quad (11)$$

If y is a Gaussian random variable then its fourth-order moment equals $3(E\left\{y^2\right\})^2$. Thus, kurtosis is zero for a Gaussian random variable, and for most of non-Gaussian random variables, kurtosis is nonzero, which can be positive or negative. Typically, non-Gaussianity is measured by the absolute value of kurtosis. The square of kurtosis can also be used.

Learning Rule for Post Nonlinear ICA

Let y now represent for a component of the output vector $\mathbf{u}$, $y = u_i$, $i=1,\ldots,N$.
The objective function for one point post-nonlinear ICA is defined as,

$$J(y) = |\mathrm{kurt}(y)| = |\mathrm{kurt}(\mathbf{w}^T\mathbf{g}(\mathbf{x}))| \quad (12)$$

Where $\mathbf{w}^T = [w_1, w_2,..,w_N]$ is a row vector of the separating matrix $\mathbf{W}$ and $\mathbf{g}(\mathbf{x}) = [g_1(x_1), g_2(x_2),\ldots,g_N(x_N)]^T$, and $g_i(x_i)$ was defined in equation (10).
Substitution of equation (10) into (13) yields,

$$J(y) = |kurt(y)| = |kurt(\sum_{k=1}^{N} w_k \sum_{j=1}^{N} g_{kj} x_k^{j-1})| \quad (13)$$

For the linear part, the maximization of the objective function can be obtained using the gradient of $J(y)$ respects to $\mathbf{w}$, which leads to the fixed point algorithm [17].

$$\Delta\mathbf{w} \propto sign(kurt(\mathbf{w}^T\mathbf{g}(\mathbf{x})))E\left\{\mathbf{g}(\mathbf{x})(\mathbf{w}^T\mathbf{g}(\mathbf{x}))^3\right\} \quad (14)$$

$$\mathbf{w} \leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|} \quad (15)$$

An adaptive version of this algorithm can be obtained by dropping the expectation operation in the algorithm, which yields,

$$\Delta\mathbf{w} \propto sign(kurt(\mathbf{w}^T\mathbf{g}(\mathbf{x})))\left\{\mathbf{g}(\mathbf{x})(\mathbf{w}^T\mathbf{g}(\mathbf{x}))^3\right\} \quad (16)$$

$$\mathbf{w} \leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|} \quad (17)$$

For the nonlinear part, taking the derivative of (14) with respect to the g_{kj} yields,

$$\frac{\partial J(y)}{\partial g_{kj}} = sign(kurt(\mathbf{w}^T\mathbf{g}(\mathbf{x}))) * \{\frac{\partial E\left\{(\sum_{k=1}^{N} w_k \sum_{j=1}^{N} g_{kj} x_k^{j-1})^4\right\}}{\partial g_{kj}} - 3\frac{\partial\{E\left\{(\sum_{k=1}^{N} w_k \sum_{j=1}^{N} g_{kj} x_k^{j-1})^2\right\}\}^2}{\partial g_{kj}}\} \quad (18)$$

Equation (18) is rewritten in vector notation form as

$$\frac{\partial J(y)}{\partial g_{kj}} = sign(kurt(\mathbf{w}^T\mathbf{g}(\mathbf{x}))) * \{4E\left\{(\mathbf{w}^T\mathbf{g}(\mathbf{x}))^3 w_k x_k^{j-1}\right\} - 12E\{(\mathbf{w}^T\mathbf{g}(\mathbf{x}))^2\}E\left\{(\mathbf{w}^T\mathbf{g}(\mathbf{x}))w_k x_k^{j-1}\right\}\} \quad (19)$$

Since we are working with the instantaneous values of x_k, the expectation operator E in (19) can be dropped, and the adaptive rules for g_{kj} now becomes

$$\Delta g_{kj} \propto -8sign(kurt(\mathbf{w}^T\mathbf{g}(\mathbf{x}))) \{(\mathbf{w}^T\mathbf{g}(\mathbf{x}))^3 w_k x_k^{j-1}\} \quad (20)$$

Equation (20) is the learning rule for estimating one independent component. For multiple independent components, we run the algorithm several times with different initial points and the independent constrain

between components. Since each nonlinear function f_i does not contribute into the mixing process of the original signals, the nonlinear functions g_i, in the inverse process, are adaptively learned without any constrains. The linear unmixing matrix, on the other hand, is constrained due to the reason that the independence of components requires that they are uncorrelated, and in the whitened space, the expectation of $E\{(\mathbf{w}_i^T\mathbf{z})(\mathbf{w}_j^T\mathbf{z})\} = \mathbf{w}_i^T\mathbf{w}_j^T$, and therefore uncorellatedness is equivalent to orthogonality. This property is a direct consequence of the fact that after whitening, the mixing matrix can be taken to be orthogonal. Hence, the constrains is that the row vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N$ of the unmixing matrix $\mathbf{W}$ are orthogonal to each other.

5. Simulation and results

Example 1

Prepare the simulation data

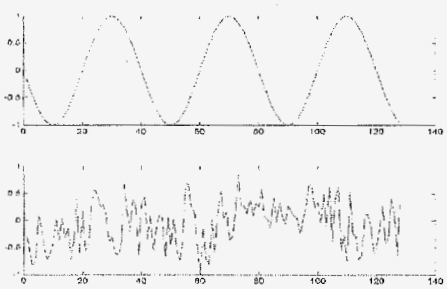


Figure 2a) Source signals: a sine wave of 100 Hz and Random noise

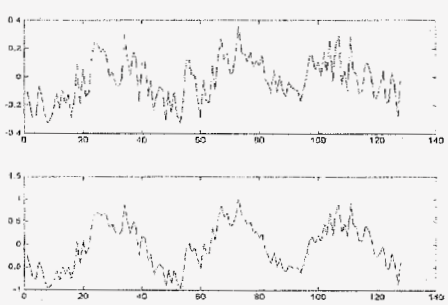


Figure 2b) Linearly mixed signals from the sources in Figure 2a

In this example, a sine wave of 100Hz and a random Gaussian noise were used as the two unknown independent source signals to create the mixed signals. The two independent sources are shown in Figure 2a). The procedure to generate the mixtures is as follow: The two independent source signals were linearly mixed together first by a randomly generated 2x2 matrix **A** and their plots are depicted in the Figure 2b). The linearly mixed signals were then nonlinearly distorted componentwise by the log sigmoid functions. The plots of the nonlinear mixtures are shown in the Figure 2c). The final mixtures are in the form.

$$\mathbf{x} = \text{logsig}(\mathbf{As}), \qquad (21)$$

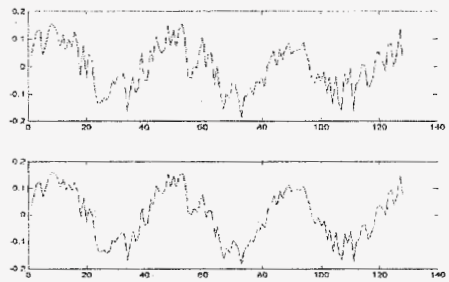


Figure 2c) Nonlinearly mixed signals form the linear mixtures in Figure 2b by log sigmoid function

Recover the original independent signal process

The independent sources were recovered from the mixture in the following procedure: First the DC biases of the mixtures were removed by subtracting the means from each signal. The resulting unbiases mixtures were then entered into a repeating loop which consist of following steps.

Step 1: the signals were passed through nonlinear functions that are approximated by 9th order polynomials with their coefficients adaptively adjusted after each iteration, in order to remove the nonlinear distorted which is the logsigmoid functions in this particular simulation.

Step 2: the signals which come out of the nonlinear stage are then whitened which produced uncorrelated mixture with unit variances.

Step 3: The whiten signal from step 3 were linearly recovered by the un-mixing matrix.

Step 4: coefficients of the polynomials from step 1 and coefficients of the un-mixing matrix were updated follow the learning rules 21 and 17, respectively.

The process goes back to step 1 until the criteria are met. It took 5 iterations for the simulation to converge. The plots of the recovered signals are shown in figure 2d). Comparing the original signals in the Figure 1a) and the recovered signals in the Figure 1d), we can see that the original signals have been successfully recovered except that the recovered signals phases changed 180 degree and their amplitudes have been scaled. However, these changes have been predicted theoretically in the previous section.

Example 2

In the example 1, we did the simulation with a sine wave of 100Hz and a random Gaussian noise as the unknown independent sources. In this example we are going to perform the experiment with two sine waves of 100 Hz and 400Hz, respectively as the unknown independent sources.

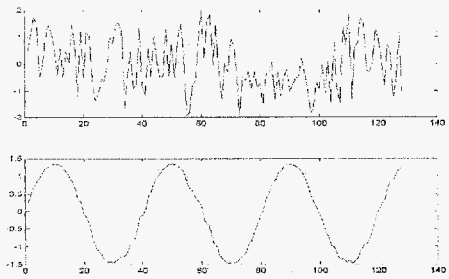


Figure 2d) Approximated signals from the nonlinear mixtures in Figure 2c

Prepare the data set

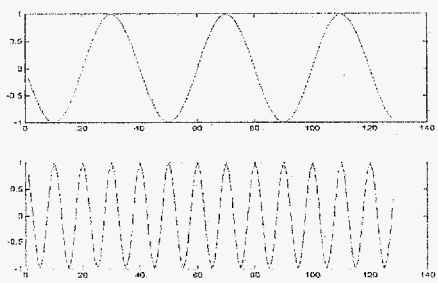


Figure 3a) Source signals: a sine wave of 100 Hz and a sine wave of 400 Hz

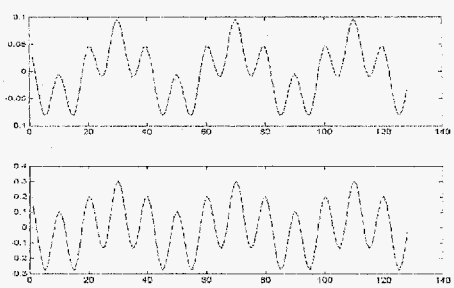


Figure 3b) Linearly mixed signals from the sources in Figure 3a

A sine wave of 100Hz and a sine wave of 400Hz a random Gaussian noise were used as the two unknown independent source signals to create the mixed signals. The two independent sources are shown in Figure 3a). The procedure to generate the mixtures is as the same as in the previous example: The two independent source signals were linearly mixed together first by a randomly generated 2x2 matrix **A** and their plots are depicted in the Figure 3b). The linearly mixed signals were then nonlinearly distorted componentwise by the log sigmoid functions. The plots of the nonlinear mixtures are shown in the Figure 3c).

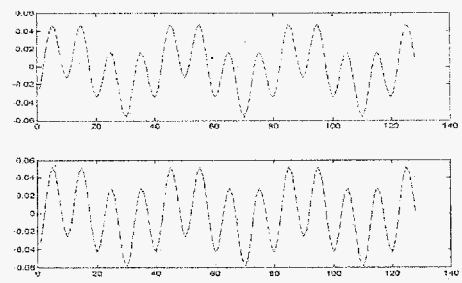


Figure 3c) Nonlinearly mixed signals from the linear mixtures in Figure 3b by log sigmoid function

Recover the original signals

As in the example 1, the independent sources were recovered from the mixtures first by removing the mean values from the mixtures. The zero-mean mixtures then entered into the training process where all the coefficients of the polynomials and the unmixing matrix adaptively updated in each iteration as described in the previous example. It took 5 iterations for the simulation to converge. As it is shown in the Figure 2.d) the recovered signals are very close to original signals except that the phases of the recovered signals have been reversed and the amplitudes have been scaled.

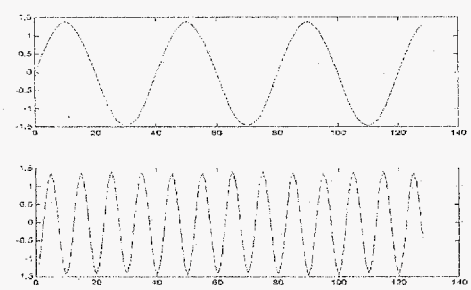


Figure 3 d) Approximated signals from the nonlinear mixture in Figure 3c

Signal to noise ratio experiment

In this section, we study the performance of the proposed post-nonlinear ICA algorithm with a variety of nonlinearities by the signal to noise ratios (SNR) of the recovered signals. It is noted that in the ICA problem, signal to noise ratio practically can not be determined due to following reasons: Firstly, the

original source signals are not available, secondly, the signs of the recovered signals can not be determined, i.e. the signs of the recovered signals may or may not changed compare to the original ones due to the 4th order cumulants and lastly, the amplitudes of the recovered signals can not be determined also. However, in simulation, the *unknown* independent sources are *unknown* only to the algorithm but they are available for the author and the readers since we have used them to create the mixtures. To overcome the problem of the signs of a particular recovered signal, we calculate the SNRs of both plus and minus signs of the recovered signal and chose the biggest one. The amplitude problem can be fixed in simulation by forcing the amplitudes of the *unknown* independent sources and the recovered signals are one by normalizing them. With these constrains, the signal to noise ratio is feasible to measure the performance of the proposed algorithm.

Table 2 shows the SNRs of the recovered signals from the mixtures of two original independent sources, a sine wave of 100 Hz and a uniformly random signal with a variety of nonlinearities. The nonlinearities are required to be monotonic and smooth so they can be approximated by polynomials. In this simulation, we used the 9th order polynomials to approximate the inverse of nonlinearities in the table. In the table, we used both. We used the same nonlinearity on both channels to create the mixtures and we also use used different nonlinearities on each channel to create the mixtures. The simulations converged in 5 to 10 iterations. As it can be seen from the table, the sine wave was recovered with SNR of around 30 db, which is very good and the noise was recovered with SNR of around 12 db. The results in the table show that the algorithm has performed well on the both increasing monotonic nonlinearity and decreasing monotonic nonlinearity, symmetric nonlinearity and asymmetric nonlinearity. The algorithm has also performed well when the same nonlinearity applied to both channels and as well as the different nonlinearity applied to each channel when creating the mixtures.

Table 3 shows the SNRs of the recovered signals from the mixtures of two independent sine waves, one is 100 Hz and the other is 400 Hz. The purpose of this simulation is to study the performance of the algorithm with different source signals while the other aspects such as nonlinearity, the order of polynomial, are not changed. As they are shown in the table 3, both sine waves were recovered with SNRs around 33 dB which is better than the performance when recovering the source signals from the mixtures of a sine wave and a noise.

The results from both tables 2 and 3 indicate that the proposed algorithm is successful to separate the post-nonlinear ICA mixtures with a variety of nonlinearities. The results also show that the algorithm is successful to separate post-nonlinear mixtures with different source signals. The performance, however, depends more on the source signals than on the nonlinearities.

6. Conclusions

In this paper, we introduced the nonlinear ICA problem and one approach for solving it. Through the counter examples, we have showed that the nonlinear ICA problem, in general, does not have a unique solution. Blindly reconstructing the nonlinear mapping has made the nonlinear ICA far more complicated than the linear ICA. To make the problem simpler and more practical, we just focused our study in the post-nonlinear subclass. This extra information has made it possible to express the nonlinear mapping in a parametric form and to inherit the results from the linear ICA. The main focus of this paper is the development of an algorithm for the nonlinear ICA using higher order statistics. The advantage of this method is that it does not need to approximate the probability density functions of the source signals, which are very crucial and sensitive to the performance of the algorithm using mutual information approach. Since this algorithm was developed from the fixed-point algorithm for linear ICA, it inherits the free-learning rate characteristic from the fixed-point method. The simulation results have shown that the proposed algorithm is successful to separate post non-linear mixtures from a variety of nonlinearities. The simulation results also indicate that the performance of the algorithm depends more on the type of the source signals than on the nonlinearity. For example, the sine waves have been recovered with the signal to noise ratio around 32 dB and the noises have been recovered with the signal to noise ratio around 12 dB.

Acknowledgements

The research described herein was performed jointly by the Dept. of EECS, University of California Irvine and the Jet Propulsion Laboratory, California Institute of Technology under contract with the National Aeronautics and Space Administration (NASA). The authors would like to thank Drs. Tuan A. Duong and T. Daud for their insightful discussion and suggestions.

Table 2: Signal to Noise Ratio of the Recovered Signal $s_1(t)$ and $s_2(t)$ Using Different Nonlinear Functions with Sine Wave and Random Noise as Source Signals

$$s_1(t) = \sin(2\pi 100t + \pi)$$
$$s_2(t) = \text{rand}(\text{size}(t)) - \text{rand}(\text{size}(t))$$

Nonlinear Functions	Recovered Signal $s_1(t)$ SNR	Recovered Signal s_2 SNR
y = acos(x)	26.5833	10.7848
y = asin(x)	22.4023	10.7452
y = 1/(1+exp(-x))	35.9658	12.0254
y = atan(x)	29.3673	12.2724
y = tanh(x)	29.2337	12.3314
y = logsig(x)	35.9658	12.0258
y = tansig	29.2337	12.3314
y = 1/(1+exp(-x)) for channel 1 y = logsig(x) for channel 2	35.0976	15.2312
y = erf(x)	28.2245	12.2451
y = erf(x) - 0.5	28.2245	12.2451
Y =erf(x) + 0.5	28.2245	12.2451

Table 3: Signal to Noise Ratio of the Recovered Signal $s_1(t)$ and $s_2(t)$ Using Different Nonlinear Functions with a Sine Wave of 100 Hz and a sine Wave 400 Hz as Source Signals

$$s_1(t) = \sin(2\pi\ 100t + \pi)$$
$$s_2(t) = \sin(2\pi\ 400t + \pi/2)$$

Nonlinear Functions	Recovered Signal s_1 SNR	Recovered Signal s_2 SNR
y = acos(x)	35.7854	29.3359
y = asin(x)	35.7854	29.3359
y = 1/(1+exp(-x))	35.7654	33.1584
y = atan(x)	31.9494	33.9473
y = tanh(x)	31.8424	33.9517
y = logsig(x)	35.7654	33.1584
y = tansig	31.8424	33.9517
y = 1/(1+exp(-x)) for channel 1 y = logsig(x) for channel 2	31.1534	28.1903
y = erf(x)	32.2882	33.9457
y = erf(x) - 0.5	32.2882	33.9457
Y =erf(x) + 0.5	32.2882	33.9457

REFERENCES:

[1]. L. Ameida. Linear and nonlinear ICA based on mutual information. In *Proc. IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (AS-SPCC)*, pages 117-122, Lake Louise, Canada, October 2000.

[2]. A. J. Bell and T. J. Sejnowski. An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7:1129-1159, 1995.

[3]. G. Burel. Blind Separation of sources: a nonlinear algorithm. *Neural Networks*, 5(6):937-947, 1992.

[4]. J. Cardoso. High-order contrast for independent component analysis. *Neural Computation*, 1998(a).

[5]. A. Cichocki, S. Amari,, and J. Cao. Blind signal separation of delayed and convolved sources with self-adaptive learning rate. In *Proc. Int. Sym. On Nonlinear Theory and Applications (NOLTA'96)*, page 229-232, Kochi, Japan, 1996

[6]. A. Cichocki, L. Zang, S. Choi, and S. Amari. Nonlinear dynamic independent component analysis using state space and neural network models. *Int. Workshop on Independent Component Analysis and Signal Separation (ICA' 99)*, page 99-104, Aussois, France, 1999.

[7]. P. Common. Independent component Analysis – a new concepts? *Signal Processing*, 36(3):287-314, 1994.

[8]. V. Duong and A. Stubberud. An application of independent component analysis to a sequence of images data. *Proceedings of the IASTED International Conference in Intelligent Systems and Control 2000*, Honolulu, Hawaii, USA, paper # 317-105.

[9]. V. Duong and A. Stubberud. "Independent Component Analysis for Space-Time Mixtures". *The IASTED International Conference on Signal Processing, Pattern Recognition and Applications 2003*, Rhodes, Greece, June 30th-July 2nd.

[10]. M. Girolami and C. Fyfe. Extraction of independent signal sources using a deflationary exploratory projection pursuit network with lateral inhibition. IEEE Proceedings on Vision, Image and Signal Processing Journal, 14(5):299-206, 1997.

[11]. J. Herault and C. Jutten. Space or time adaptive signal processing by neural network models. In Denker J. S. (ed), editor, *Neural network for computing: AIP Conference proceedings* 151, New York. American Institute for Physics, 1986.

[12]. A. Hyvaerinen and E. Oja. A fast fixed-point algorithms for independent component analysis. Neural Computation, 9:1483-1492, 1997.

[13]. . Hyvaerinen and P. Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3)429-439, 1999.

[14]. A. Hyvaerinen, J. Karhunen and E. Oja. *Independent Component Analysis*. John Wiley and Sons, 2001.

[15]. C Jutten and J. Herault. Blind separation of the sources, part I: an adaptive algorithm based on neuromimetic architecture. Signal Processing, 24:1-10, 1991.

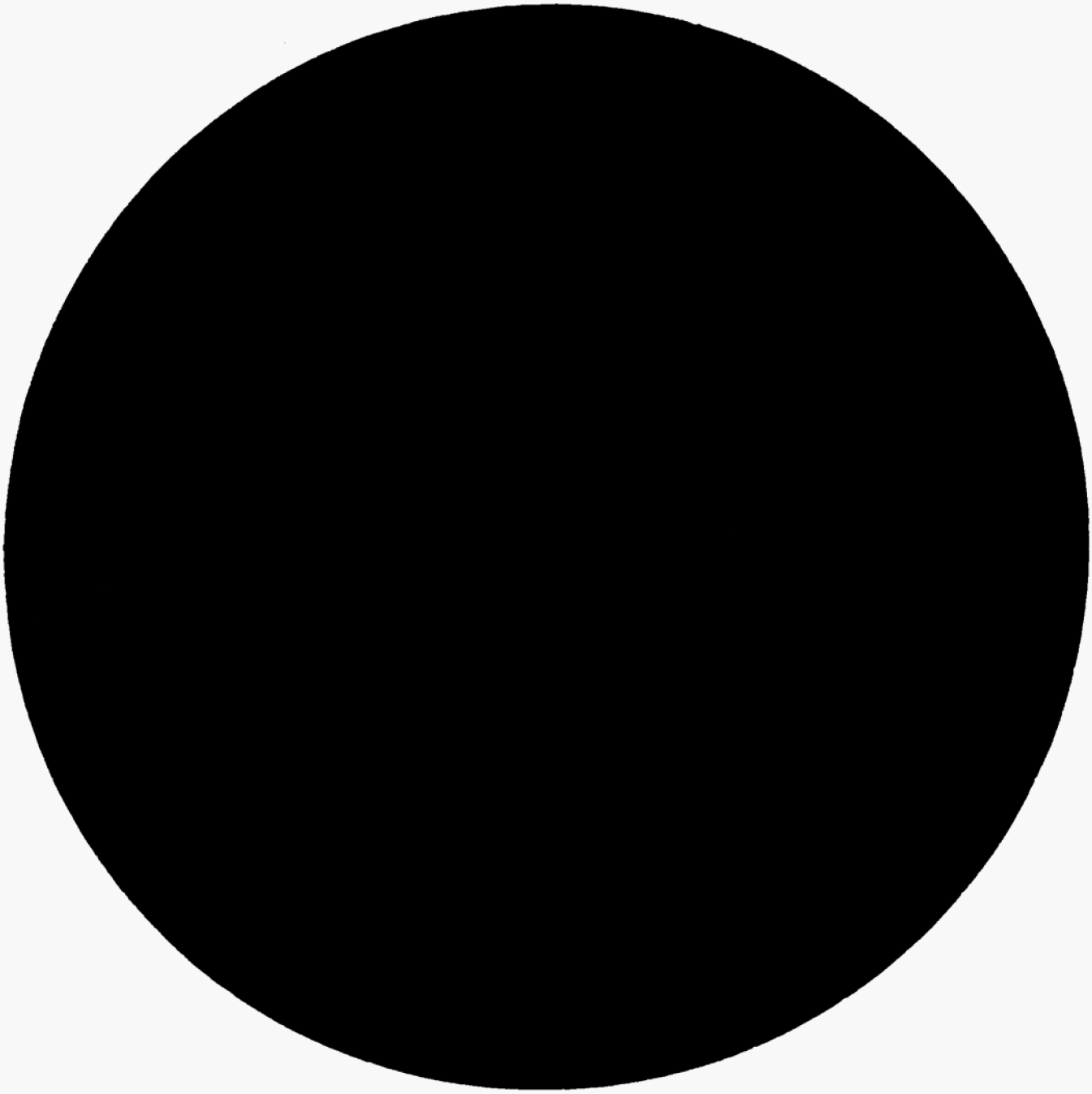
[16]. T.-W. Lee, A. J. Bell and R. Orglmeister. Blind source separation of real-world signals. In Proc. ICNN, pages 2129- 2135, Houston, USA, 1997.

[17]. P. Pajunen, A. Hyvarinen, and J. Karhunen. Nonlinear Blind source separation by self-organizing maps. In *Proc. Int. Conf. on Neural Information Processing*. Pages 1207-1210, Hong Kong, 1996.

[18]. D.-T. Pham, P. Garrat and C. Jutten. Separation of a mixtures of independent sources through a maximum likelihood approach. In *EUSIPCO* pages 771-774, 1992.

[19]. Z. Roth and Y. Baram. Multidimensional density shaping by sigmoids. *IEEE Trans. On Neural Networks*, 7(5):1291-1298, 1996.

[20]. A. Taleb and C. Jutten. Source separation in post- nonlinear mixtures. IEEE Trans. on Signal Processing, Vol. 47, No. 10, October, 1999.

[21]. D. Yellin and E. Weinstein. Criteria for multichannel signal separation. IEEE *Trans. on Signal Processing*, 42:2158-2167, 1994.

End of File