EIS 2004
Island of Madeira, Portugal

CONNECTING GEOMETRIC INDEPENDENT COMPONENT ANALYSIS TO UNSUPERVISED LEARNING ALGORITHMS

Fabian J. Theis[1] *fabian@theis.name*
Peter Gruber [1] *petergruber@gmx.net*
Carlos G. Puntonet[2] *carlos@atc.ugr.es*
Elmar W. Lang[1] *elmar.lang@biologie.uni-r.de*

[1]Institute of Biophysics, AG Neuro- and Bioinformatics,
University of Regensburg, D-93040 Regensburg, Germany
[2] Dept. Arquitectura y Tecnologia de Computadores, Escuela Técnica Superior de Ingenieria Informatica,
Universidad de Granada, E-18071 Granada

ABSTRACT

The goal of independent component analysis (ICA) lies in transforming a mixed random vector in order to render it as independent as possible. This paper shows how to use adaptive learning and clustering algorithms to approximate mixture space densities thus learning the mixing model. Here, a linear square-model is assumed, and as learning algorithm either a self-organizing map (SOM) or a neural gas (NG) is used. These result in a considerable improvement in separation quality in comparison to other mixture-space analysis ('geometric') algorithms, although the computational cost is rather high. By establishing this connection between neural networks and ICA, applications like for example transferring convergence proofs for SOMs to geometric ICA algorithms now seem possible.

INTRODUCTION

In independent component analysis (ICA) one tries to find statistically independent data within a given random vector. An application of ICA lies in blind source separation (BSS), where it is furthermore assumed that the given vector has been mixed using a fixed set of independent sources. The advantage of applying ICA algorithms to BSS problems in contrast to correlation-based algorithms is the fact that ICA tries to make the output signals as independent as possible by also including higher-order statistics.

Since the first introduction of the ICA method by Herault and Jutten [6] various algorithms have been proposed to solve the blind source separation problem [4] [2] [3].

Most of them are based on information theory, but there also exist geometric ICA algorithms which are based on mixture space analysis. They were first proposed in [12]; the theoretical background for geometric ICA has been studied in detail [15], and a convergence condition has been formulated, which then resulted in a new, faster geometric algorithm called FastGeo. Here, we will propose two new 'geometric' algorithms, which analyze the mixture space using a self-organizing map respectively a neural gas.

ICA AND BSS

For $m, n \in \mathbb{N}$ let $\mathrm{Mat}(m \times n)$ be the $\mathbb{R}-$vectorspace of real $m \times n$ matrices, and $\mathrm{Gl}(n) := \{W \in \mathrm{Mat}(n \times n) \mid \det(W) \neq 0\}$ be the general linear group of $\mathbb{R}^n$. Let $[1:n] := [1,n] \cap \mathbb{N} = \{1, \ldots, n\}$ for $n \in \mathbb{N}$. $\mathrm{Cov}(X) := E(XX^T)$ denotes the covariance matrix of a random vector X.

Given an independent random vector $S : \Omega \longrightarrow \mathbb{R}^n$, which will be called **source vector** with zero mean and symmetric distribution, where Ω is a fixed probability

space, and $A \in \mathrm{Gl}(n)$ is a quadratic invertible matrix, we call the random variable $X := A \cdot S$ the **mixed vector**. The goal of linear ICA is to recover the sources and the **mixing matrix** A from the given mixture X.

In the following we denote two matrices $B, C \in \mathrm{Mat}(m \times n)$ to be **equivalent**, $B \sim C$, if C can be written as $C = BPL$ with an invertible diagonal matrix (scaling matrix) $L \in \mathrm{Gl}(n)$ and an invertible matrix with unit vectors in each row (permutation matrix) $P \in \mathrm{Gl}(n)$. Uniqueness of linear ICA states that if at most one of the source variables S_i is Gaussian then for any **solution to the symmetric ($m = n$) BSS problem**, i.e. any $D \in \mathrm{Gl}(n)$ such that $D \circ X$ is independent, D^{-1} is equivalent to A [4]. Vice versa, any matrix $D \in \mathrm{Gl}(n)$ such that D^{-1} is equivalent to A solves the BSS problem, since we calculate for the transformed mutual information

$$I(D \circ X) = I(LPA^{-1} \circ X) = I(A^{-1} \circ X) = I(S) = 0,$$

taking into account that the information is invariant under scaling and permutation of coordinates.

Without loss of generality let $E(X) = 0$; this can be accomplished using a translation of the data vectors. Then also $E(S) = 0$, so both the mixtures and the sources are **centered**. Furthermore, by applying a whitening transformation to the mixtures (full rank principal component analysis), we can already decorrelate the data. Then

$$I = \mathrm{Cov}(X) = E(ASS^\top A^\top) = A\,\mathrm{Cov}(S)A^\top = AA^\top$$

if we assume that also the sources are whitened. This means that solving the orthogonal BSS problem will also solve the general BSS problem, so we can restrict ourselves to the case $A \in O(n)$.

ICA USING A SELF-ORGANIZING MAP

The **selforganizing map algorithm (SOM)** is a clustering algorithm often used for the visualization of high-dimensional data. SOMs have been developed by Kohonen in 1981 [9] and have since then become a widely used and studied visualization and clustering technique.

In this section we want to hybridize the two concepts of ICA and SOMs. There have already been some other approaches to this like Local ICA [8], where the mixture data is first clustered using a SOM, and the ICA is applied to each cluster, or nonlinear BSS using a SOM as approximation to the demixing mapping [11]. Our approach is somewhat similar to Pajunen et al.'s idea [11] in the linear case but it does not require the sources to be subgaussian.

The idea of what we call **SOMICA** is very simple, based on the ideas of geometric ICA. Figure 1 shows

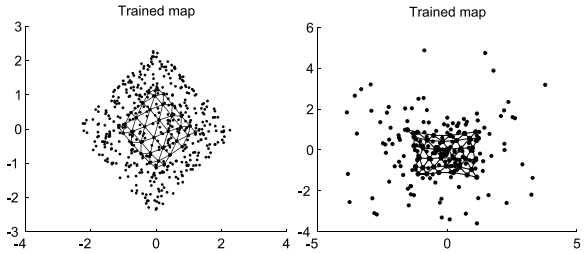

Fig. 1. SOMICA algorithm, sub(left)- and super(right)-gaussian case, Separation of a mixture of two uniform respectively Laplacian signals. The 2-dimensional SOM is used to approximate the whitened mixtures. The extremal units (those at the corners of the grid) are then images of the unit vectors or their sum, depending on super- or subgaussianity of the sources, so here $m_{11} = \lambda A(e_1 + e_2)$ (left image) and $m_{11} = \lambda A(e_1 - e_2)$ (right image) for some $\lambda \neq 0$. Crosstalking error of the separation in the subgaussian case was 0.108 and in the supergaussian case 0.0846.

the basic idea of SOMICA: Given observations X first whiten them such that $\mathrm{Cov}(X) = I$. Then use a SOM to approximate X. The corner unit locations then contain similar to geometric ICA the information of the mixing matrix A.

So assume S is an independent non-gaussian 2-dimensional symmetric non-deterministic random vector, and let $X = AS$ with $A \in O(n)$ already whitened. Let $r \in \mathbb{N}$ and define a 2-dimensional SOM on the input grid

$$R = [1 : r] \times [1 : r].$$

Note that we index the processing units by 2-tuples $(i, j) \in R$. Use the SOM-learning-algorithm to approximate the whitened mixtures X. Let m_{ij} be the processing unit location of unit (i, j) after the learning process has converged. Define

$$B = (m_{11} - m_{rr} | m_{1r} - m_{r,1})$$

and

$$\hat{B} = (m_{11} + m_{1r} - m_{r1} - m_{rr} | m_{11} - m_{1r} + m_{r1} - m_{rr}).$$

We claim that if S is supergaussian, then B is equivalent to A and that if S is subgaussian, then $\hat{B}$ is equivalent to A.

Note that we only have to show that $m_{11} + m_{1r} - m_{r1} - m_{rr}$ and $m_{11} - m_{1r} + m_{r1} - m_{rr}$ are proportional to Ae_1 and Ae_2 or Ae_2 and Ae_1 respectively, because then the B and A have the same columns except for scaling and permutation, so they are equivalent. We will not give a full explanation of this claim here. A

proof should follow the lines of the geometric case [14] and may use the convergence results of SOMs [5].

The intuitive idea of why this conjecture should be true is that for example in the uniform case (more general subgaussian case) the corners $m_{11}, m_{1r}, m_{r1}, m_{rr}$ of the SOM correspond to 'corners' of the mixture distribution, which are identified as $A(\pm e_1 \pm e_2)$. So the matrix $(m_{11} + m_{1r}|m_{11} - m_{1r})$ will have to be equivalent to A. Using symmetry we in fact use matrix $\hat{B}$, which takes a mean over both opposite corners in order to stabilize the algorithm a bit. In the supergaussian case, left picture of figure 1, however, the corners of the SOM should correspond directly to $A(e_i)$, so $(m_{11}|m_{1r})$ will be equivalent to A. Again we use B as above for stability reasons.

Figure 2 shows that SOMICA does not work without whitening. The reason is that the SOM algorithm converges to output neuron positions which correspond to the mean of the input distribution within their receptive fields, whereas in geometric ICA we know that the fixed points fulfill the GCC i.e. they correspond to the medians of the input data distributions within their receptive fields. However after whitening, we have orthogonal structures, so median and mean are the same.

Note that if it is not known in the beginning whether the sources are super- or subgaussian then one can determine the correct solution by comparing the covariance C of both recoveries $B^{-1}X$ and $\hat{B}^{-1}X$ and taking the better solution in terms of minimal $\|C - I\|$. A similar idea has been applied in the LatticeICA algorithm [13], where the geometric structure of the mixture space is approximated using a histogram.

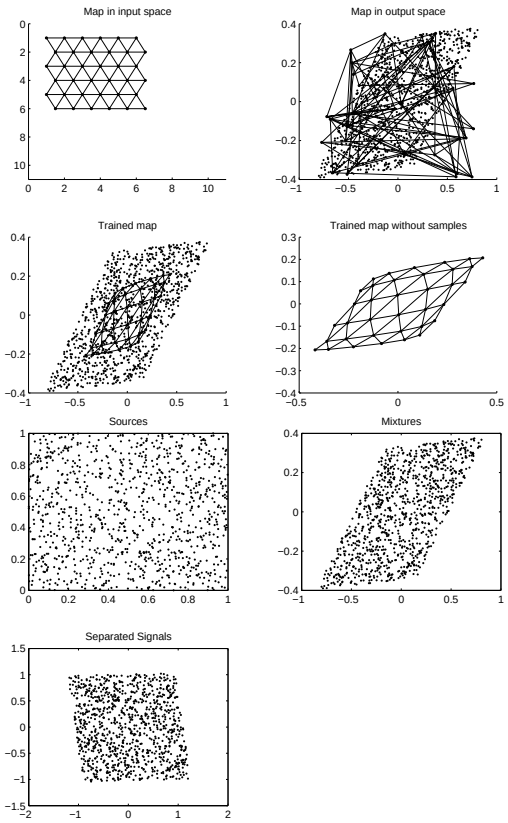

Fig. 2. SOMICA algorithm without whitening: The separation does not work properly without whitening as shown in the figure with two uniform sources (crosstalking error 0.352930). The top left figure shows the SOM itself, the top right one the randomly initialized SOM before learning. The next two figures show the learnt map with and without samples. Clearly the corner points of the SOM have not reached the corners of the transformed square. The next three figures show the source, mixture and recovered sample density.

ICA USING A NEURAL GAS

Now, we want to use a similar algorithm as in section 3, but instead of using a SOM we will use a neural gas.

The term **neural gas (NG)**, first introduced by Martinetz et al [10] describes an adaptive neural system with a growing architecture; they were introduced to improve vector quantization techniques by converging better to lower approximation errors than other methods. Similar to a SOM, a NG consists of a set of neurons located in an input space (**centroid** of the neuron) together with corresponding output vectors in order to realize a correspondence between the input space and the output space. In this paper, we use the neural gas algorithm implemented by the 'SOM Toolbox' from the Helsinki group[1]; in contrast to SOMs it does not track a neighborhood relationship. However, we will see that this is not necessary in order to separate the data.

The algorithm which we denote by **NGICA** now works very similar to the SOMICA algorithm. The idea is shown in figure 3 for the sub- and supergaussian case: Given whitened observations X, we use a neural gas to approximate X. Units with maximal modulus then contain the mixture matrix information as seen above.

In practice for the two-dimensional case we use a neural gas with only 4 units. Let the unit positions in $\mathbb{R}^2$ be p_1, p_2, p_3 and p_4; we assume that the indices 1 to 4 have been chosen in such a way that p_i and p_{i+2} are opposite each other in the sense that the modulus of their sum is minimal under all indices permutations.

[1] http://www.cis.hut.fi/projects/somtoolbox/

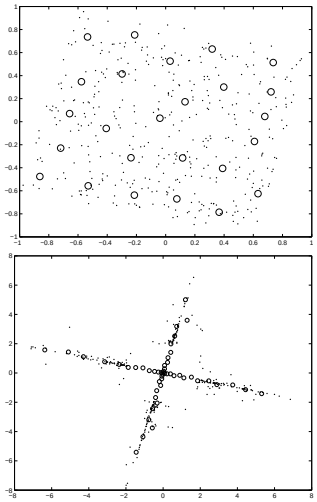

Fig. 3. NGICA algorithm, sub(left)- and super(right)-gaussian case: Separation of a mixture of two uniform respectively gamma-distributed signals; here a scatter plot of the mixtures is displayed. A 2-dimensional NG with 25 respectively 40 units is used to approximate the whitened mixtures. Again, extremal units are the images of the unit vectors or their sum, depending on super- or subgaussianity of the sources, so here $p_1 = \lambda A(e_1 + e_2)$ in the left image or $p_1 = \lambda A(e_1 - e_2)$ in the right image for some $\lambda \neq 0$.

After this index ordering, we proceed exactly as in section 3, where the vectors $m_{11}, m_{1,r}, m_{rr}$ and $m_{r,1}$ are to be replaced by p_1, p_2, p_3 and p_4.

Again, we claim that

$$B = (p_1 - p_3 | p_2 - p_4)$$

respectively

$$\hat{B} = (p_1 + p_2 - p_3 - p_4 | p_1 - p_2 - p_3 + p_4)$$

are equivalent to A, depending on whether S is super- or subgaussian.

A proof of this claim is strongly connected to a proof of the SOMICA algorithm, because it is quite easy to see that in standard settings the corner points of a SOM are opposite each other in the sense stated above, so the SOMICA algorithm can be translated into the NGICA algorithm and vice versa. For proper proofs however, apart from convergence details more care has to be taken when analyzing the different update rules of a SOM and a NG.

An advantage of NGICA clearly lies in the fact that it is easy to generalize to higher dimensions - after all only the number of points ($2n$) and their ordering has to be adapted. Typical SOM algorithms however are restricted to 2 or 3 dimensions.

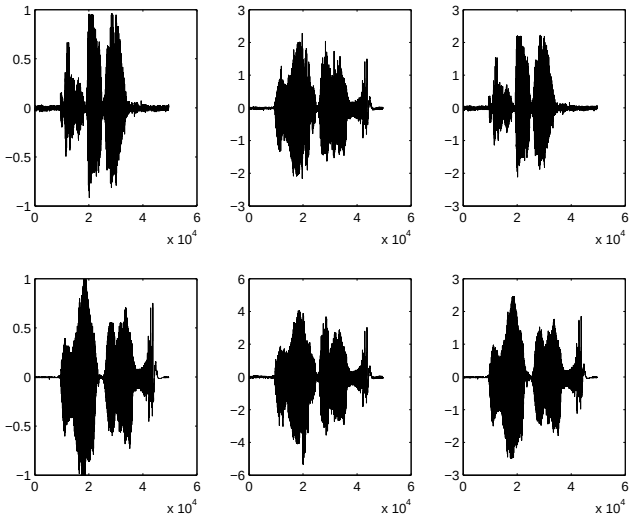

Fig. 4. Example 1: Mixture of two speech signals. The first column shows the two independent source signals, two speech signals ('peace and love', 'hello, how are you'; both spoken by the same person). The mixing matrix A was chosen to be . The middle column shows the mixture of those two signals, and the right column shows the recovered sources.

EXAMPLES

In this section, we first give two example applications of the two adaptive algorithms from above. Then we compare SOMICA and NGICA with other ICA algorithms Calculations have been performed on a P4-2000 PC with Windows and Matlab using the SOM Toolbox.

For comparison, we calculate the **performance index** E_1 or **crosstalking error** as proposed by Amari [1]

$$E_1 = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \frac{|p_{ij}|}{\max_k |p_{ik}|} - 1 \right) + \sum_{j=1}^{n} \left(\sum_{i=1}^{n} \frac{|p_{ij}|}{\max_k |p_{kj}|} - 1 \right)$$

where $P = (p_{ij}) = B^{-1}A$ with B the calculated estimate of A.

In our first explicit example, we consider a mixture of two real-world speech signals (supergaussian) using the mixture matrix

$$A = \begin{pmatrix} 1 & 2 \\ -2 & 4 \end{pmatrix}.$$

In figure 4, the source, mixture and recovered signals are plotted, and figure 5 presents a scatterplot of the mixture density. Already from the scatterplot, the

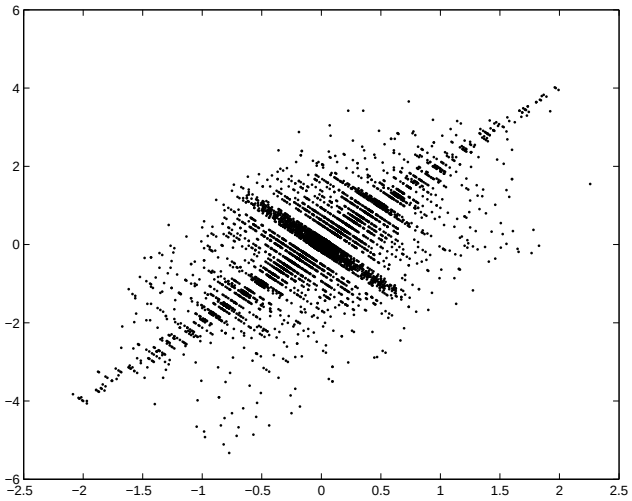

Fig. 5. Example 1: Mixture of two speech signals, scatterplot. Plotted is the mixture scatterplot of the two signals from figure 4.

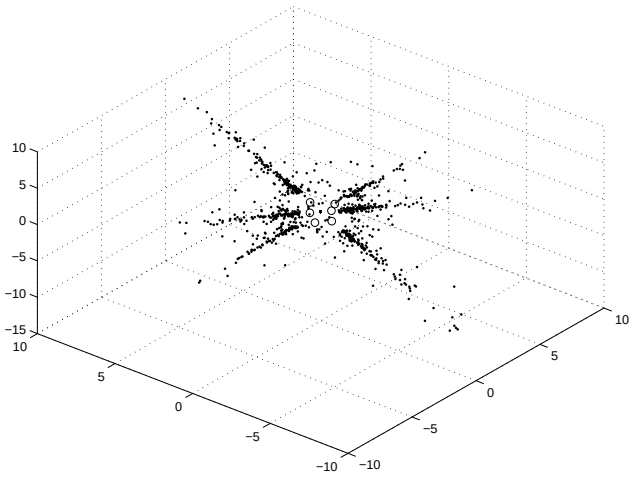

Fig. 6. Example 2: Mixture of three gamma signals, scatterplot. Plotted is the mixture scatterplot of the three mixed signals. For ease of presentation a ball of radius 2 has been cut out and the learnt neurons have been added, marked by the 6 larger circles.

mixing matrix columns can be guessed by trying to fit in a 'cross'. This is precisely what geometric algorithms do [15]. In this example, we apply SOMICA to recover the sources; we get a reconstructed mixing matrix B as follows:

$$B = \begin{pmatrix} 0.37 & 0.81 \\ -1.0 & 1.6 \end{pmatrix}.$$

The crosstalking error between A and B is 0.16 which is quite good.

The second example applies NGICA to three high-kurtotic gamma-distributed random signals. They were mixed using the easy to picture mixing matrix

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix}.$$

Figure 6 shows a mixture scatter plot together with the trained neural gas, which in this three-dimensional case consists of 6 different neurons. Using NGICA to approximate the three dimensional mixture, we get a recovered mixing matrix

$$B = \begin{pmatrix} -0.62 & 0.0080 & 0.011 \\ 0.036 & -0.66 & -0.65 \\ -0.0056 & -0.68 & 0.65 \end{pmatrix},$$

which results in a crosstalking error of 0.21, which is good for three dimensional data sets.

We now compare the two adaptive ICA algorithms with other algorithms, namely the **FastICA** algorithm [7] by Hyvärinen and Oja, the two geometric algorithms **FastGeo** [15] and an early implementation of **LatticeICA** [13], and an easy PCA-based algorithm, which we denote by **SimpleICA**: given two-dimensional signals X, then $\left(E\sqrt{D}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\right)^{-1} X$ is independent if E and D are calculated using the Matlab eigenvalue decomposition algorithm; this only works for problems with the same source component distributions and normalized mixing matrices. We give performance comparison with SimpleICA in order to show how much algorithm time is used up by prewhitening.

At first, we consider a mixture of two Laplacian signals. These results as well as those of the following three examples are shown in table 1: for each algorithm we measure the mean elapsed cpu-time per run and the mean crosstalking error E_1 with its standard deviation. FastICA, SimpleICA (PCA) and the two geometric algorithms are fast, SOMICA is very slow in comparison, but we have not yet done any optimization and SOM algorithms usually tend to be rather slow. In terms of accuracy however, SOMICA performs better than FastICA. NGICA is twice as fast as SOMICA, but somewhat less accurate. Note that this can be changed by varying the number of learning epochs of the neural gas algorithm.

In our second and third example, we compare these algorithms for uniform and delta-like data distributions. Again both SOMICA and NGICA are very accurate but slow; FastICA and LatticeICA seem to have problems with the delta case, which is not surprising

Table 1. Comparison of time per run and crosstalking error of ICA algorithms for a random mixture of various signals. Means and standard deviations were taken over 100 runs with 1000 samples and uniformly distributed mixing matrix elements.

Src type	Algorithm	time/run [ms]	index E_1
Laplacian	FastICA	6	0.27±0.22
	FastGeo	18	0.51 ±0.73
	LatticeICA	57	1.0±0.71
	SimpleICA	2	0.23±0.43
	SOMICA	852	0.15±0.11
	NGICA	501	0.22±0.14
uniform	FastICA	5	0.099±0.059
	FastGeo	17	0.52 ±0.65
	LatticeICA	64	0.27±0.39
	SimpleICA	1	0.17±0.44
	SOMICA	850	0.095±0.055
	NGICA	510	0.16±0.12
delta-like	FastICA	1487	0.39±1.2
(deterministic)	FastGeo	29	0.49 ±0.29
	LatticeICA	73	2.0±1.2
	SimpleICA	13	0.36±0.82
	SOMICA	868	0.012±0.001
	NGICA	511	0.014±0.012
sound	FastICA	41	0.49±0.45
(speech)	FastGeo	49	0.30 ±0.65
	LatticeICA	88	0.74±0.65
	SimpleICA	35	0.60±0.74
	SOMICA	883	0.20±0.12
	NGICA	536	0.40±0.29

considering the fact that the delta distribution is not independent; they are, nonetheless, separable by geometric algorithms and show that geometric algorithms can only be used in ICA problems where a BSS mixing model is indeed given.

The fourth example deals with real-world data: two audio signals (two speech signals). The results are similar to the above toy examples. FastICA outperforms SOMICA and NGICA in terms of speed, the accuracy of FastICA, SOMICA and FastGeo however are comparable, followed by NGICA. The SimpleICA algorithm is less accurate, mainly due to the different source distributions.

CONCLUSION

We presented a new approach for linear ICA similar to geometric ICA using a SOM and a neural gas for the mixture space approximation. Both SOMICA and NGICA are very stable, and give accuracy results comparable or slightly better than those of the FastICA algorithm. The adaptive algorithms are very accurate but in its current non-improved state very slow, so they are mostly interesting from a theoretical point of view, especially if one tries to generalize convergence and other theoretical results from neural networks to ICA algorithms; for example, we hope to prove convergence of the geometric algorithm in a manner similar to the SOM convergence proof in one dimension.

Simulations with non-symmetrical and non-unimodal distributions will have to be performed. This is the subject of ongoing research in our group. In the future, the algorithms could be extended to the non-linear case similar to [11]. Currently, we are working on a generalization to the postnonlinear case, which works quite well at least for subgaussian sources.

REFERENCES

[1] S. Amari, A. Cichocki, and H.H. Yang. A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems*, 8:757–763, 1996.

[2] A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind seperation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[3] J.-F. Cardoso. Blind signal seperation: Statistical principals. *Proc. IEEE*, 86:2009–2025, 1998.

[4] P. Comon. Independent component analysis - a new concept? *Signal Processing*, 36:287–314, 1994.

[5] M. Cottrell and J.-C. Fort. Étude d'un processus d'auto-organisation. *Annales de l'Institut Henri Poincaré*, 23(1):1–20, 1987.

[6] J. Hérault and C. Jutten. Space or time adaptive signal processing by neural network models. In J.S. Denker, editor, *Neural Networks for Computing. Proceedings of the AIP Conference*, pages 206–211, New York, 1986. American Institute of Physics.

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

[8] J. Karhunen, S. Malaroiu, and M. Ilmoniemi. Local linear independent component analysis based on clustering. *Int. J. of Neural Systems*, 10(6), 2000.

[9] Teuvo Kohonen. Self-organizing formation of topologically correct feature maps. *Biol. Cyb.*, 43(1):59–69, 1982.

[10] T.M. Martinetz, S.G. Berkovich, and K.J. Schulten. 'neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569, 1993.

[11] P. Pajunen, A. Hyvarinen, , and J. Karhunen. Nonlinear blind source separation by self-organizing maps. *Progress in Neural Information Processing, Proc. of the International Conference on Neural Information Processing (ICONIP'96), Hong Kong*, 2:1207–1210, 1996.

[12] C.G. Puntonet and A. Prieto. An adaptive geometrical procedure for blind separation of sources. *Neural Processing Letters*, 2, 1995.

[13] M. Rodriguez-Alvarez, F. Rojas, C.G. Puntonet, J. Ortega, F.J. Theis, and E.W. Lang. A geometric ICA procedure based on a lattice of the observation space. *ICA 2003 submitted*, 2003.

[14] F.J. Theis, A. Jung, E.W. Lang, and C.G. Puntonet. A theoretic model for geometric linear ICA. *Proc. of ICA 2001*, pages 349–354, 2001.

[15] F.J. Theis, A. Jung, C.G. Puntonet, and E.W. Lang. Linear geometric ICA: Fundamentals and algorithms. *Neural Computation*, 15:1–21, 2002.