Fast Kernel Density Independent Component Analysis

Aiyou Chen

Bell Labs, Lucent Technologies
aychen@research.bell-labs.com

Abstract. We develop a super-fast kernel density estimation algorithm (FastKDE) and based on this a fast kernel independent component analysis algorithm (KDICA). FastKDE calculates the kernel density estimator exactly and its computation only requires sorting n numbers plus roughly $2n$ evaluations of the exponential function, where n is the sample size. KDICA converges as quickly as parametric ICA algorithms such as FastICA. By comparing with state-of-the-art ICA algorithms, simulation studies show that KDICA is promising for practical usages due to its computational efficiency as well as statistical efficiency. Some statistical properties of KDICA are analyzed.

Keywords: independent component analysis, kernel density estimation, nonparametric methods.

1 Introduction

Independent component analysis (ICA) has been a powerful tool for blind source separation in many applications such as image and acoustic signal processing, brain imaging analysis (Hyvarinen, Karhunen and Oja 2001). Suppose that an observable signal, say $\mathbf{X}$, can be modeled as an unknown linear mixture of m mutually independent hidden sources $(S_1, \cdots, S_m)$. Denote $\mathbf{S} \equiv (S_1, \cdots, S_m)^T$, so

$$\mathbf{X} = \mathbf{A}\mathbf{S} \tag{1}$$

for some matrix $\mathbf{A}$. Assume that $\{\mathbf{X}(t) : 1 \leq t \leq n\}$ are n i.i.d. observations of $\mathbf{X}$, where t is the time index. That is, at time t the hidden sources produce signals $\mathbf{S}(t) \equiv (S_1(t), \cdots, S_m(t))^T$ that are observed as $\mathbf{X}(t) = \mathbf{A}\mathbf{S}(t)$. The problem is to recover $\{\mathbf{S}(t) : 1 \leq t \leq T\}$ without knowing either $\mathbf{A}$ or the distributions of $\mathbf{S}$. In order to solve this problem, it is necessary that $\dim(\mathbf{X}) \geq m$. Without loss of generality, we may assume that the dimension of $\mathbf{X}$ is the same as $\mathbf{S}$ and that A is an $m \times m$ nonsingular matrix. It is well-known that $\mathbf{W} = \mathbf{A}^{-1}$ (called the unmixing matrix) is identifiable up to permutation and scale transformations of the rows of $\mathbf{A}$ if $\mathbf{S}$ has at most one Gaussian component (Comon, 1994). The order and scale can be controlled such that $\mathbf{W}$ is unique. The ICA problem becomes to estimate $\mathbf{W}$.

Classical ICA algorithms such as FastICA fit parametric models for the hidden sources and thus are limited to particular families of hidden sources

J. Rosca et al. (Eds.): ICA 2006, LNCS 3889, pp. 24–31, 2006.

(Cardoso 1998). It has been realized that the unknown distributions of hidden sources can be estimated by nonparametric methods, which can be applied to a wide range of distribution families. For example, Hastie and Tibshirani (2002) proposed penalized maximal likelihood based on log-spline density estimation. Miller and Fisher (2003) proposed the RADICAL algorithm based on the neighborhood density estimator. Vlassis & Motomura (2001), Boscolo et al. (2004) and recently Shwartz et al. (2005) used kernel density estimation to deal with the unknown source distributions. These nonparametric algorithms are in general more accurate and more robust but on the other side are computationally much heavier than classical parametric ICA algorithms such as FastICA. The computational bottleneck is the nonparametric density estimators[1]. There exists other nonparametric ICA algorithms such as KCCA, KGV (Bach & Jordan 2002), CFICA (Eriksson & Koivunen 2003), PCFICA (Chen & Bickel 2005) and *kernel mutual information* (Gretten et al 2005), which do not deal with the source density functions directly. Among different nonparametric density estimators, the kernel density estimator (KDE) is most popular. But naive implementation requires $O(n^2)$ complexity, where n is the sample size. In the statistical literature, the binning and clustering techniques have been used to reduce the complexity, see Silverman (1986). For example, Pham (2004) applied the binning technique in the ICA literature. Fast Gauss transform (Greengard & Strain 1991) and the dual-tree algorithm by Gray & Moore (2003) are alternative fast algorithms for KDE. All these KDE algorithms are based on different approximation techniques and are faster than $O(n^2)$. But these techniques require careful choices of certain tuning parameters in order to balance computational speed-up and approximation errors, and occasionally are as slow as $O(n^2)$ in order to achieve good performance.

In this paper, we develop a super-fast kernel density estimation algorithm (FastICA) and based on this a fast kernel ICA algorithm (KDICA). The remaining of the paper is structured as follows. In Section 2, the FastKDE algorithm is developed. In Section 3, the KDICA algorithm is described. In Section 4, some simulation studies are used to show both computational and statistical efficiency of KDICA. In Section 4, some statistical properties of KDICA are analyzed. Section 5 concludes the paper. From now on, vectors and matrices are in bold and capital. W_k denotes the kth row vector of $\mathbf{W}$.

2 The FastKDE Algorithm

Let $\{x_i : 1 \leq i \leq n\} \subset \mathcal{R}$ be from a density function $p(\cdot)$. The kernel density estimator of $p(\cdot)$ is defined by

$$\hat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x_i - x}{h}), \tag{2}$$

[1] The neighborhood density estimator used by RADICAL only requires $n \log n$ complexity, but it does not produce a continuous objective function w.r.t. W.

where $K(\cdot)$ is a kernel density function and h is the bin width, usually $h = O(n^{-1/5})$. Popular choices of $K(\cdot)$ are symmetric density functions such as Gaussian kernel, Laplacian kernel, Uniform, Epanechnikov, etc. We need to evaluate $\hat{p}(x)$ for $x \in \{x_i : i = 1, \cdots, n\}$. Direct evaluation requires $O(n^2)$ complexity, and alternative algorithms based on approximation are available with complexity less than $O(n^2)$, but are *not* fast enough for ICA.

It is known that the choice of K is not crucial for KDE. Here we use the Laplacian kernel and develop a simple fast algorithm. The Laplacian kernel is $K(x) = \frac{1}{2}e^{-|x|}$, $x \in \mathcal{R}$. Although $K(x)$ is not differentiable at $x = 0$, $\hat{p}(x) \approx \int p(x + th)K(t)dt$ is differentiable wherever $p(x)$ is.

First the sample points $\{x_i\}$ are sorted. Sorting n numbers can be performed very quickly, for example the *quick sort* algorithm has complexity in the worst case $O(n \log n)$ and the *bucket sort* algorithm requires linear time only. Without loss of generality, let $x_1 \leq \cdots \leq x_n$. It is not hard to show that for $k = 1, \cdots, n$,

$$\hat{p}(x_k) = \frac{1}{2nh}\{\exp(\frac{x_k}{h}) \sum_{i=k+1}^{n} \exp(-\frac{x_i}{h}) + \exp(-\frac{x_k}{h}) \sum_{i=1}^{k} \exp(\frac{x_i}{h})\}.$$

Then FastKDE can now be described as follows.

Algorithm. FastKDE (given h and $x_1 \leq \cdots \leq x_n$)

1. Initialize $\underline{\mathbf{s}}_1 = e^{x_1/h}$ and $\overline{\mathbf{s}}_n = 0$, then calculate for $i = 2, \cdots, n$,

$$\underline{\mathbf{s}}_i = \underline{\mathbf{s}}_{i-1} + \exp(\frac{x_i}{h}) \text{ and } \overline{\mathbf{s}}_{n-i+1} = \overline{\mathbf{s}}_{n-i+2} + \exp(-\frac{x_{n-i+2}}{h}).$$

2. For $i = 1, \cdots, n$, compute

$$\hat{p}(x_i) = \frac{1}{2nh}\{\underline{\mathbf{s}}_i \exp(-\frac{x_i}{h}) + \overline{\mathbf{s}}_i \exp(\frac{x_i}{h})\}.$$

The exponential values $\{(\exp(x_i/h), \exp(-x_i/h)) : 1 \leq i \leq n\}$ only need to be computed once and saved for both Step 1 and Step 2. Then Step 1 and Step 2 require about $3n$ summations in total. Thus the total complexity of FastKDE is about $2n$ exponential evaluations. The bin width h is chosen for simplicity by the reference method which minimizes $\int(\hat{p}(x) - p(x))^2dx$ and gives $h = O(n^{-1/5})$ (Silverman 1986). We recommend to use

$$\hat{h} = 0.6\hat{\sigma}n^{-1/5} \tag{3}$$

where $\hat{\sigma}$ is the sample standard deviation of $\{x_i\}$.

3 The KDICA Algorithms

In this section we develop the KDICA algorithm, for which the FastKDE algorithm as the key technology is implemented. We use the maximum profile likelihood and later establish its relationship with criteria derived from information theory in Section 5.

3.1 Maximum Profile Likelihood

Suppose each S_k has a density function $r_k(\cdot)$, for $k = 1, \cdots, m$. Then the density function of $\mathbf{X}$ can be expressed as $p_{\mathbf{X}}(\mathbf{x}) = |\det(\mathbf{W})| \prod_{k=1}^{m} r_k(W_k \mathbf{x})$, where W_k is the kth row of $\mathbf{W}$. The classical maximum likelihood estimator (MLE) maximizes the likelihood of observations of X with respect to all the parameters $(\mathbf{W}, r_1, \cdots, r_m)$. However, since $(r_1, \cdots, r_m)$ are unknown functions, model (1) is called *semiparametric* (Bickel et al. 1993) and direct implementation of MLE does not work by using finite samples. In this scenario, maximum profile likelihood (MPLE) can serve as an alternative of MLE (see Murphy and van der Vaart 2000). If $\mathbf{W}$ is known, then r_k is identical to the density function of $W_k\mathbf{X}$. Thus r_k can be estimated by the kernel density estimator $\hat{r}_{W_k}(x) = (nh)^{-1} \sum_{t=1}^{n} K((W_k\mathbf{X}(t) - x)/h)$, where for the KDICA algorithm the Laplacian kernel is used for K. The profile likelihood, say l_p, is to modify the likelihood function by replacing r_k by $\hat{r}_{W_k}$, that is,

$$l_p(\mathbf{W}) = \frac{1}{n} \sum_{t=1}^{n} \sum_{k=1}^{m} \log \hat{r}_{W_k}(W_k\mathbf{X}(t)) + \log|\det(\mathbf{W})|. \tag{4}$$

Since $l_p(\mathbf{W})$ is just a function of $\mathbf{W}$, the maximum profile likelihood estimator (MPLE) is defined by

$$\hat{\mathbf{W}} = \arg\max l_p(\mathbf{W}). \tag{5}$$

Obviously the computational bottleneck of MPLE is to evaluate $\{\hat{r}_{W_k}$ $(W_k\mathbf{X}(t)) : t = 1, \cdots, n\}_{k=1}^{m}$. By using the FastKDE algorithm developed above, the complexity of MPLE is reduced to $O(mn)$.

3.2 Algorithm

This subsection describes the KDICA algorithm which implements the estimator (5). Since prewhitening can reduce computational complexity while keeps statistical consistency (Chen & Bickel 2005), we use this technique to preprocess the data. That is, let $\tilde{\mathbf{X}}(t) = \hat{\Sigma}_{\mathbf{X}}^{-1/2}\mathbf{X}(t)$ for $t = 1, \cdots, n$, where $\hat{\Sigma}_{\mathbf{X}}$ is the sample variance-covariance matrix of $\mathbf{X}(t)$. By assuming unitary variances for $\mathbf{S}$, $\tilde{\mathbf{X}}$ can be separated by a rotation matrix. Then we seek for a rotation matrix $\hat{\mathbf{O}}$, such that

$$\hat{\mathbf{O}} = \arg \min_{O \in \mathcal{O}(m)} F(\mathbf{O}), \tag{6}$$

where $F(\mathbf{O}) = -\sum_{k=1}^{m} \frac{1}{n} \sum_{t=1}^{n} \log \tilde{r}_{O_k}(O_k\tilde{\mathbf{X}}(t))$, and $\tilde{r}_{O_k}(s) = \frac{1}{nh} \sum_{t=1}^{n}$ $K((O_k\tilde{\mathbf{X}}(t) - s)/h)$ is the Laplacian kernel density estimator for $O_k\tilde{\mathbf{X}}$. $\mathcal{O}(m)$ is the set of $m \times m$ rotation matrices. Since $O_k\tilde{\mathbf{X}}$ has unitary variance, by (3), $h = 0.6n^{-1/5}$.

The optimization of (6) can be done efficiently by using the gradient algorithm on the Stiefel manifold (Edelman, Arias & Smith 1999). We refer to Bach & Jordan (2002) for how to implement it. The KDICA algorithm then has three steps as follows. Note that the KDICA does not need any tuning parameters.

Algorithm. KDICA ($h_n = 0.6n^{-1/5}$)

1. Prewhiten : $\tilde{\mathbf{X}}(t) = \hat{\Sigma}_{\mathbf{X}}^{-1/2}\mathbf{X}(t)$ for $t = 1, \cdots, n$, where $\hat{\Sigma}_{\mathbf{X}}$ is the sample variance-covariance matrix of $\{\mathbf{X}(t) : 1 \leq t \leq n\}$.
2. Optimize $\hat{\mathbf{O}} = \arg\max_{\mathbf{O}\in\mathcal{O}(m)} F(\mathbf{O})$ using the gradient algorithm.
3. Output $\hat{\mathbf{W}} = \hat{\mathbf{O}}\hat{\Sigma}_{\mathbf{X}}^{-1/2}$.

4 Simulation Studies

We compare KDICA with several well-known ICA algorithms such as the generalized FastICA (Hyvarinen 1999), JADE (Cardoso 1999) and KGV (Bach and Jordan 2002). Some recent algorithms such as NPICA (Boscolo et al 2004) and EFFICA (Chen 2004) are also included for comparison. FastICA is used to initialize KGV, NPICA, EFFICA and KDICA. We used $m = 4$ and $m = 8$ sources with different sample sizes 1000 and 4000. The 8 sources were generated from: $\mathcal{N}(0, 1)$, exp(1), t(3), lognorm(1,1), t(5), logistic(0,1), Weibull(3,1), and exp(10)+$\mathcal{N}(0, 1)$. When $m = 4$, the first four distributions were used for hidden sources. Each experiment was replicated 100 times and the boxplots of Amari errors were reported. Figure 1 shows that KDICA is comparable to EFFICA which has been proven to be asymptotically efficient under mild conditions, and like other nonparametric algorithms, KDICA performs much better than FastICA and JADE. The right panel of Figure 1 reports the average running time of all algorithms. The plot shows that KDICA is more than 20 times faster than NPICA which uses the FFT based KDE algorithm and 50 times faster than KGV. KDICA is about 10 times slower than but comparable to FastICA and JADE. The KDICA algorithm exhibits very good simulation performance. But due to space limitation, we are not allowed to report further simulation studies.

We next apply the KDICA algorithm for blind separation of mixtures of images. Two natural images and a Gaussian noise image are given in the first row of Figure 2, each of size 80 × 70 pixels (black/white). First, each pixel matrix is reshaped into a column and each column is normalized by its sample standard deviation. Second, a random 3 × 3 matrix W $\in \Omega$ is inverted to obtain three columns $\{\mathbf{X}(t) \in \mathcal{R}^3 : 1 \leq t \leq 5600\}$ and each column is reshaped into a matrix of size 80 × 70. This gives three contaminated images, as shown in the second row of Figure 2. Third, $\{\mathbf{X}(t)\}$ is separated into three vectors by using KDICA, and each vector is reshaped into an image with 80 × 70 pixels. Three random restarting points were used in KDICA. It is surprising that human eyes can hardly tell the difference between natural images and separated images. This type of experiments have also been done by several different researchers in the ICA literature (e.g. Yang and Amari 1997).

We ran this experiment 10 times with random $\mathbf{W}$ by using KDICA and several other ICA algorithms. The average running times for the generalized FastICA, JADE, and KDICA are 0.05, 0.03 and 1.82 seconds separately. Other nonparametric algorithms such as NPICA and KGV take more than one minute.

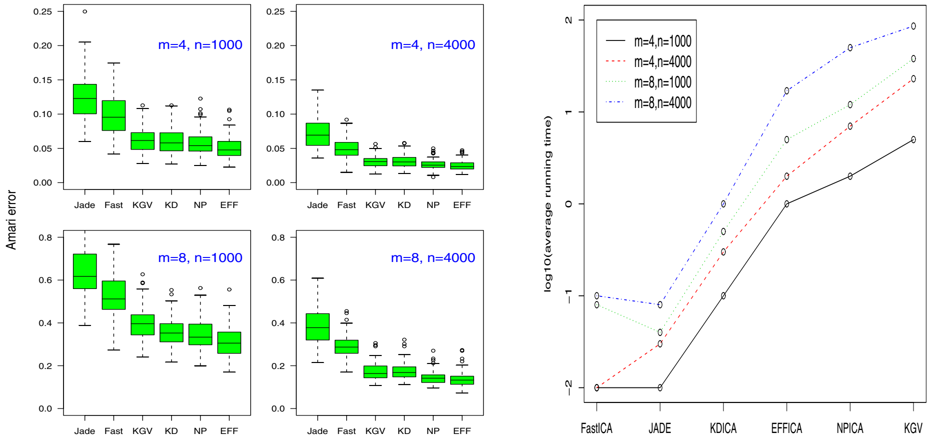

Fig. 1. Left panel: Comparison of KDICA and other ICA algorithms in terms of the Amari errors, where the numbers below the x-labels are the average running time (seconds/per experiment) of the corresponding algorithms. Right panel: Comparison of running time of different ICA algorithms.

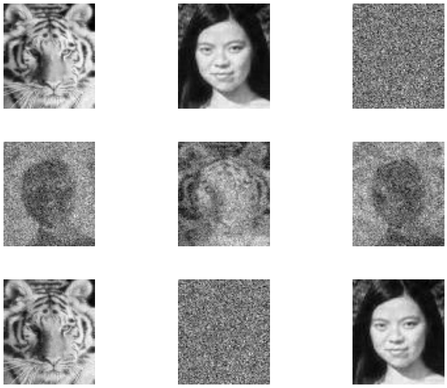

Fig. 2. Face identification by KDICA, where the three original, mixed and separated images are given in the three rows separately

5 Statistical Consistency and Efficiency of KDICA

In this Section, we study the statistical properties of the estimator (5). Obviously as $n \uparrow \infty$, $\hat{r}_{W_k} \to r_{W_k}$, the density function of $W_k X$. Thus for $n = \infty$, the profile likelihood is equal to $l_p(\mathbf{W}) = E\sum_{k=1}^{m} \log r_{W_k}(W_k X) + \log|\det(\mathbf{W})|$. Let $p_{\mathbf{W}}(\cdot)$ be the joint density function of $(W_1\mathbf{X}, \cdots, W_m\mathbf{X})$, then $p_{\mathbf{W}}(\mathbf{W}\mathbf{x}) = p_{\mathbf{X}}(\mathbf{x})/|\det(\mathbf{W})|$. Thus the mutual information of $(W_1\mathbf{X}, \cdots, W_m\mathbf{X})$ is equal to

$$\mathbf{I}(\mathbf{W}) = E \log \frac{p_{\mathbf{W}}(\mathbf{W}\mathbf{X})}{\prod_{k=1}^{m} r_{W_k}(W_k\mathbf{X})} = E \log p_{\mathbf{X}}(\mathbf{X}) - l_p(\mathbf{W}).$$

Notice that $E \log p_{\mathbf{X}}(\mathbf{X})$ does not depend on the parameter $\mathbf{W}$. The above equation implies that the profile likelihood criteria is equivalent to the mutual information criteria which has been popularly used in the ICA literature. Thus we would expect the statistical performance of KDICA to be similar to or better than other nonparametric ICA algorithms. General connection between likelihood inference and information theory criteria has been studied by Lee, Girolami, Bell & Sejnowski (2000). We obtain statistical consistency of the KDICA algorithm as summarized in Theorem 1, whose technical conditions and proof are omitted here due to space limitation but refer to Chen (2004).

Theorem 1. *Suppose that $\mathbf{W}$ is identifiable and the density functions of the hidden sources are continuous and satisfy mild smoothness conditions. If $h_n = O(n^{-1/5})$. Then the estimator $\hat{\mathbf{W}}$ given by (5) is consistent, that is, $||\hat{\mathbf{W}} - \mathbf{W}_P|| = o_P(1)$, where $\mathbf{W}_P$ is the true unmixing matrix.*

6 Concluding Remarks

In this paper, we have presented the FastKDE and KDICA algorithms. Due to its computational and statistical efficiency, KDICA makes nonparametric ICA applicable for large size problems of blind source separation. We conjecture that FastKDE will make it convenient to deal with nonlinear independent component analysis (Jutten et. al, 2004) in a truly nonparametric manner.

Acknowledgment

The paper is extracted from Chapter 4 of the author's Ph.D. thesis. The author would like to thank his advisor Peter J. Bickel for support and many helpful discussions, and thank Michael I. Jordan, Anant Sahai and Diane Lambert for helpful comments.

References

1. Bach, F. and Jordan, M. (2002). Kernel independent component analysis. *Journal of Machine Learning Research* **3** 1-48.
2. Bickel, P., Klaassen, C. , Ritov, Y. and Wellner, J. (1993). *Efficient and Adaptive Estimation for Semiparametric Models.* Springer Verlag, New York, NY.
3. Boscolo, R., Pan, H. and Roychowdhury V. P. (2004). Independent component analysis based on nonparametric density estimation. *IEEE Trans. Neural Networks*, **15** (1): 55-65.
4. Cardoso, J. F. (1998). Blind signal separation: statistical principles. *Proceedings of the IEEE,* **9**(10) 2009-2025.
5. Cardoso, J. F. (1999). High-order contrasts for independent component analysis. *Neural Computation* **11**(1) 157-192.

6. Chen, A. (2004). Semiparametric inference for independent component analysis. *Ph.D Thesis, Advisor: Peter J. Bickel, Department of Statistics, University of California, Berkeley, 2004.*
7. Chen, A. and Bickel, P. J. (2005). Consistent independent component analysis and prewhitening. *IEEE Trans. on Signal Processing*, Vol. 53, 10, page 3625-3632.
8. Comon, P. (1994). Independent component analysis, a new concept? *Signal Processing* **36**(3):287-314.
9. Edelman, A., Arias, T. and Smith, S. (1999). The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, **20**(2): 303-353.
10. Eriksson, J. and Visa Koivunen (2003). Characteristic-function based independent component analysis. *Signal Processing*, Vol. 83, pp2195-2208.
11. Greengard, L. and Strain, J. (1991). The fast Gauss transform. *SIAM J. Sci. Stat. Comput.*, 12, page 79-94.
12. Gray, A. and Moore, A. (2003). Very fast multivariate kernel density estimation via computational geometry. *Proceedings of the Joint Statistical Meeting*, San Francisco, CA, 2003.
13. Gretton, A., Herbrich, R., Smola, A., Bousquet, O. and Scholkopf, B. (2005). Kernel methods for testing independence. Submitted.
14. Hastie, T. and Tibshirani, R. (2002). Independent component analysis through product density estimation, *Technical report*, Department of Statistics, Stanford University.
15. Hyvarinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. on Neural Networks* **10**(3) 626-634.
16. Hyvarinen, A., Karhunen, J. and Oja, E. (2001). *Independent Component Analysis.* John Wiley & Sons, New York, NY.
17. Jutten, C., Babaie-Zadeh, M, and Hosseini, S. (2004). Three easy ways for separating nonlinear mixtures? *Signal Processing*, 84, pp.217-229.
18. Lee, T. W., Girolami, M., Bell, A. and Sejnowski, T. (2000). A unifying information-theoretic framework for independent component analysis. *Computers and Mathematics with Applications* **39** 1-21.
19. Miller, E. and Fisher, J. (2003). ICA using spacings estimates of entropy. *Journal of Machine Learning Research*, **4**, pp. 1271-1295.
20. Murphy, S. and van der Vaart, A. (2000). On profile likelihood. *Journal of the American Statistical Association* **95** 449-485.
21. Pham, D. T. (2004). Fast algorithms for mutual information based independent component analysis. *IEEE Trans. on Signal Processing*, Vol. 52, 10, page 2690-2700.
22. Shwartz, S., Zibulevsky, M. and Schechner, Y. (2005). Fast kernel entropy estimation and optimization. *Signal Processing*, **85**, 1045-1058.
23. Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis.* Chapman Hall: London.
24. Vlassis, N. and Motomura, Y. (2001). Efficient source adaptivity in independent component analysis. *IEEE Trans. Neural Networks* **12**(3) 559-565.
25. Yang, H. H. and Amari, S. (1997). Adaptive on-line learning algorithms for blind separation - maximum entropy and minimum mutual information. *Neural Computation*, **9**(7): 1457-1482.