# REAL AND COMPLEX INDEPENDENT SUBSPACE ANALYSIS BY GENERALIZED VARIANCE

**Zoltán Szabó** **András Lőrincz**
Department of Information Systems, Eötvös Loránd University,
Pázmány P. sétány 1/C, Budapest H-1117, Hungary
Research Group on Intelligent Information Systems
Hungarian Academy of Sciences
WWW home page: `http://nipg.inf.elte.hu`
`szzoli@cs.elte.hu, andras.lorincz@elte.hu`

## ABSTRACT

Here, we address the problem of Independent Subspace Analysis (ISA). We develop a technique that (i) builds upon joint decorrelation for a set of functions, (ii) can be related to kernel based techniques, (iii) can be interpreted as a self-adjusting, self-grouping neural network solution, (iv) can be used both for real and for complex problems, and (v) can be a first step towards large scale problems. Our numerical examples extend to a few 100 dimensional ISA tasks.

**Keywords:** Independent Subspace Analysis, joint f-decorrelation

## 1 INTRODUCTION

Uncovering independent processes is of high importance, because it breaks combinatorial explosion [10]. In cases, like Smart Dust, the problem is vital, because (i) elements have limited computational capacity and (ii) communication to remote distances is prohibitively expensive. Self-adjusting, self-grouping neural network solutions may come to our rescue here. Here, we present such an approach for Independent Subspace Analysis (ISA). The extension of ISA to Independent Process Analysis is straightforward under certain conditions [10].

Our paper is built as follows. The $\mathbb{K}$-ISA model is introduced in Section 2. Section 3 is about our method. Illustrations are provided in Section 4.

## 2 THE $\mathbb{K}$-ISA MODEL

Section 2.1 defines the $\mathbb{K}$-ISA task to be studied, Section 2.2 treats the ambiguities of the model.

 ICA Research Network International Workshop (ICARN-2006), pp. 85-88, Liverpool, U.K., 2006.

### 2.1 The $\mathbb{K}$-ISA Equations

We treat real and complex ISA tasks: Let $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$. Assume that we observe the mixture of multidimensional independent i.i.d. sampled sources (*components*):

$$\mathbf{z}(t) = \mathbf{A}\mathbf{s}(t), \qquad \mathbf{s}(t) = [\mathbf{s}^1(t); \ldots; \mathbf{s}^M(t)], \tag{1}$$

where $D = \sum_{m=1}^{M} d_m$ is the total dimension of the components, $\mathbf{A} \in \mathbb{K}^{D\times D}$ is the invertible *mixing matrix*. The task is to recover the hidden components $\mathbf{s}^m(t) \in \mathbb{K}^{d_m}$ by means of observations $\mathbf{z}(t) \in \mathbb{K}^D$. If $\mathbb{K} = \mathbb{R}$ ($\mathbb{K} = \mathbb{C}$) then we shall talk about Real (Complex) ISA [i.e., $\mathbb{R}$-ISA ($\mathbb{C}$-ISA)] task. The special $d_m = 1 (\forall m)$ case is the Real (Complex) Independent Component Analysis [i.e., $\mathbb{R}$-ICA ($\mathbb{C}$-ICA)].

### 2.2 Ambiguities of the $\mathbb{K}$-ISA Model

Identification of the $\mathbb{K}$-ISA model is ambiguous. However, ambiguities are simple: hidden components $\mathbf{s}^m$ can be determined up to permutation among subspaces and up to invertible transformation within subspaces. Details about $\mathbb{R}$-ISA and $\mathbb{C}$-ISA can be found in [13] and [12], respectively.

Ambiguities within subspaces can be lessened: given our assumption on the invertibility of matrix $\mathbf{A}$, we can assume without any loss of generality that both the sources and the observation are *white*, that is,

$$E[\mathbf{s}] = \mathbf{0}, cov[\mathbf{s}] = \mathbf{I}_D, \tag{2}$$
$$E[\mathbf{z}] = \mathbf{0}, cov[\mathbf{z}] = \mathbf{I}_D, \tag{3}$$

where $E[\cdot]$ denotes the expectation value, $\mathbf{I}_D$ is the $D$-dimensional identity matrix. Now, the $\mathbf{s}^m$ sources are determined up to (i) permutation *and* orthogonal transformation in the real case and (ii) permutation *and* unitary transformation in the complex case.

## 3 $\mathbb{K}$-ISA BY JOINT DECORRELATION

Components $\mathbf{s}^m$ are estimated by a neural network, which aims to 'decorrelate' (see below) the $\mathbf{y}^m \in \mathbb{K}^{d_m}$ parts of the $\mathbb{K}^D \ni \mathbf{y}(t) = [\mathbf{y}^1(t); \ldots; \mathbf{y}^M(t)]$ output of the network. The network executes mapping $\mathbf{z} \mapsto L(\mathbf{z}, \Theta)$ with network parameter $\Theta$.

### 3.1 Neural Network Candidates ($L$)

Choosing an RNN with feedforward ($\mathbf{F}$) and recurrent ($\mathbf{R}$) connections then the network assumes the form

$$\dot{\mathbf{y}}(\tau) = -\mathbf{y}(\tau) + \mathbf{F}\mathbf{z}(t) - \mathbf{R}\mathbf{y}(\tau) \tag{4}$$

and thus, upon relaxation it solves the

$$\mathbf{y}(t) = (\mathbf{I}_D + \mathbf{R})^{-1}\mathbf{F}\mathbf{z}(t) = L(\mathbf{z}(t); \mathbf{F}, \mathbf{R}) \tag{5}$$

input-output mapping [1, 9]. Another natural choice is a network with feedforward connections $\mathbf{W}$ that executes mapping

$$\mathbf{y}(t) = \mathbf{W}\mathbf{z}(t) = L(\mathbf{z}(t); \mathbf{W}). \tag{6}$$

### 3.2 Cost Function of $\mathbb{K}$-ISA

The neural network estimates hidden sources $\mathbf{s}^m$ by non-linear ($\mathbf{f}$) decorrelation of $\mathbf{y}^m$s, components of network output $\mathbf{y}$. Formally:

Let us denote the empirical $\mathbf{f}$-covariance matrix of $\mathbf{y}(t)$ and $\mathbf{y}^m(t)$ for function $\mathbf{f} = [\mathbf{f}^1; \ldots; \mathbf{f}^M]$ over $[1, T]$ by

$$\mathbf{\Sigma}_{\mathbb{K}}(\mathbf{f}, T) = \widehat{cov}\left(\mathbf{f}\left[\varphi_{\mathbb{K}}(\mathbf{y})\right], \mathbf{f}\left[\varphi_{\mathbb{K}}(\mathbf{y})\right]\right), \tag{7}$$

$$\mathbf{\Sigma}_{\mathbb{K}}^{i,j}(\mathbf{f}, T) = \widehat{cov}\left(\mathbf{f}^i\left[\varphi_{\mathbb{K}}\left(\mathbf{y}^i\right)\right], \mathbf{f}^j\left[\varphi_{\mathbb{K}}\left(\mathbf{y}^j\right)\right]\right), \tag{8}$$

respectively, where $i, j = 1, \ldots, M$, $\varphi_{\mathbb{R}}(\mathbf{v}) = \mathbf{v}$, $\varphi_{\mathbb{C}}$ is the mapping

$$\varphi_{\mathbb{C}} : \mathbb{C}^L \ni \mathbf{v} \mapsto \mathbf{v} \otimes \begin{bmatrix} \Re(\cdot) \\ \Im(\cdot) \end{bmatrix} \in \mathbb{R}^{2L}. \tag{9}$$

Here, $\Re(\cdot)$ [$\Im(\cdot)$] denotes the real (imaginary) part, $\otimes$ is the Kronecker-product. Then minimization of the following non-negative cost function (in $\Theta$)

$$Q_{\Theta}(\mathbf{f}, T) := -\frac{1}{2}\log\left\{\frac{\det[\mathbf{\Sigma}_{\mathbb{K}}(\mathbf{f}, T)]}{\prod_{m=1}^{M}\det[\mathbf{\Sigma}_{\mathbb{K}}^{m,m}(\mathbf{f}, T)]}\right\} \tag{10}$$

gives rise to *pairwise*[1] $\mathbf{f}$-uncorrelatedness:

**Theorem 1.** *For the separation carried out by the network minimizing cost function* (10)*, the following statements are equivalent:*

*i)* $\mathbf{f}$*-uncorrelatedness:* $\mathbf{\Sigma}_{\mathbb{K}}^{i,j}(\mathbf{f}, T) = 0 \quad (\forall i \neq j)$.

*ii)* $Q_{\Theta}$ *is minimal:* $Q_{\Theta}(\mathbf{f}, T) = 0$.

**Proof** (sketch)**.** *The statement follows from the inequality related to the multi-dimensional Shannon differential entropy $H$: Let $\mathbf{u} = [\mathbf{u}^1; \ldots; \mathbf{u}^M] \in \mathbb{R}^D$ ($\mathbf{u}^m \in \mathbb{R}^d$) denote a random variable. Then*

$$H(\mathbf{u}^1, \ldots, \mathbf{u}^M) \leq \sum_{m=1}^{M} H(\mathbf{u}^m), \tag{11}$$

*and equality holds iff $\mathbf{u}^m$s are independent. Hint: one can choose $\mathbf{u}$ as a normal random variable with covariance $\mathbf{\Sigma}_{\mathbb{K}}(\mathbf{f}, T)$ and insert the expression of the entropy of normal variables.*

[1]We note that – unlike in the the 1-dimensional case, i.e., unlike for $d = 1$ – pairwise independence is *not* equivalent to mutual independence. Nonetheless, according to our numerical experiences it is an efficient approximation.

**Note 1.** *For the special case $\mathbb{K} = \mathbb{R}$, $\Theta = (\mathbf{F}, \mathbf{R})$, $\mathbf{f}(\mathbf{z}) = \mathbf{z}$ and $d = 1$, see [9].*

**Note 2.** *Cost function $Q_{\Theta}$ of* (10) *is attractive from the point of view of computing its gradient. This gradient for the case of an RNN architecture [see Eq.* (5)*] may give rise to self-organization [9].*

**Note 3.** *For real random variables, the separation, which is aimed by cost function* (10)*, can be related to the more general principle, the Kernel Generalized Variance (KGV) technique [3]. This technique aims to separate the $\mathbf{y}^m$ components of $\mathbf{y}$, the transformed form of input $\mathbf{z}$. To this end, KGV estimates mutual information $I(\mathbf{y}^1, \ldots, \mathbf{y}^M)$ in Gaussian approximation*[2] *by means of the covariance matrix of variable $\mathbf{y}$. Here, the transformation of the KGV technique is realized by the neural network parameterized with variable $\Theta$ and by the function $\mathbf{f}$.*

**Note 4.** *We note that KGV is related to the kernel covariance (KC) method [7], which makes use of the supremum of 1-dimensional covariances as a measure of independence. Our approximation may also be improved by minimizing $Q_{\Theta}(\mathbf{f}, T)$ on $\mathcal{F}(\ni \mathbf{f})$, i.e., on a set of functions.*

### 3.3 The $\mathbb{K}$-ISA Algorithm

Below, our proposed $\mathbb{K}$-ISA method is introduced. A decomposition principle called $\mathbb{K}$-ISA Separation Theorem has been formulated in [12]. It says that (under certain conditions) the $\mathbb{K}$-ISA task can be solved in 2 steps: In the first step, 1-dimensional $\mathbb{K}$-ICA estimation is executed that provides separation matrix $\mathbf{W}_{\mathbb{K}\text{-ICA}}$ and estimated sources $\hat{\mathbf{s}}_{\mathbb{K}\text{-ICA}}$. In the second step, optimal permutation of the $\mathbb{K}$-ICA elements ($\hat{\mathbf{s}}_{\mathbb{K}\text{-ICA}}$) is searched for, the $\mathbb{K}$-ICA elements are grouped.

This principle is adapted to linear feedforward neural networks [see, Eq. (6)] here.[3] Separation matrix $\mathbf{W} = \mathbf{W}_{\mathbb{K}\text{-ISA}}$ is searched in the form

$$\mathbf{W}_{\mathbb{K}\text{-ISA}} = \mathbf{P}\mathbf{W}_{\mathbb{K}\text{-ICA}}, \tag{12}$$

where matrix $\mathbf{P} \in \mathbb{R}^{D \times D}$ denotes the desired permutation matrix. We search for the hidden sources $\mathbf{s}^m$ by pairwise decorrelation of the components $\mathbf{y}^m$ of the output of the network using function manifold $\mathcal{F}$ ($\mathcal{F}$: see, Note 4). Thus, given Theorem 1, our cost function is:

$$Q(\mathcal{F}, T, \mathbf{P}) := \sum_{\mathbf{f} \in \mathcal{F}} \|\mathbf{M} \circ \mathbf{\Sigma}_{\mathbb{K}}(\mathbf{f}, T, \mathbf{P})\|^2 \to \min_{\mathbf{P}}. \tag{13}$$

Here: (i) $\mathcal{F}$ denotes a set of functions, each function $\mathbb{R}^D \mapsto \mathbb{R}^D$ (if $\mathbb{K} = \mathbb{C}$ then $\mathbb{R}^{2D} \mapsto \mathbb{R}^{2D}$), and each function acts on each coordinate separately, (ii) $\circ$ denotes pointwise multiplication (Hadamard product), (iii) $\mathbf{M}$ masks according to the subspaces [$\mathbf{M} = \mathbf{I}_M \otimes \mathbf{E}_d$, where all elements of matrix $\mathbf{E}_d \in \mathbb{R}^{d \times d}$ are equal to 1 (if $\mathbb{K} = \mathbb{C}$ then $\mathbf{E}_d$ is replaced by $\mathbf{E}_{2d}$)], (iv) $\|\cdot\|^2$ denotes

[2]A complex variable is normal if its image using map $\varphi_{\mathbb{C}}$ is real multivariate normal [4]. Thus, relation $I(\mathbf{y}^1, \ldots, \mathbf{y}^M) = I[\varphi_{\mathbb{C}}(\mathbf{y}^1), \ldots, \varphi_{\mathbb{C}}(\mathbf{y}^M)]$ $\mathbf{y}^m \in \mathbb{C}^d$ extends the KGV based interpretation to the complex case, too [see Eqs. (7)-(8)].

[3]For the sake of simplicity we assume that all components have the same dimension, i.e., $d = d_m (\forall m)$.

Table 1: $\mathbb{K}$-ISA Algorithm - pseudocode

**Input of the algorithm**
  observation: $\{\mathbf{z}(t)\}_{t=1,\ldots,T}$
**Optimization**[a]
  **$\mathbb{K}$-ICA** : on whitened observation $\mathbf{z}$,
    $\Rightarrow \hat{\mathbf{s}}_{\mathbb{K}\text{-ICA}}$ estimation
  **Permutation search**
    $\mathbf{P} := \mathbf{I}_D$
    repeat
      sequentially for $\forall p \in \mathcal{G}^{m_1}, q \in \mathcal{G}^{m_2}$
        $(m_1 \neq m_2)$ :
        if $Q(\mathcal{F}, T, \mathbf{P}_{pq}\mathbf{P}) < Q(\mathcal{F}, T, \mathbf{P})$
          $\mathbf{P} := \mathbf{P}_{pq}\mathbf{P}$
        end
    until $Q(\mathcal{F}, T, \cdot)$ decreases in the *sweep* above
**Estimation**
  $\hat{\mathbf{s}}_{\mathbb{K}\text{-ISA}} = \mathbf{P}\hat{\mathbf{s}}_{\mathbb{K}\text{-ICA}}$

[a] Let $\mathcal{G}^1, \ldots, \mathcal{G}^M$ denote the indices of the $1^{st}, \ldots, M^{th}$ subspaces, i.e., $\mathcal{G}^m := \{(m-1)d+1, \ldots, md\}$, and permutation matrix $\mathbf{P}_{pq}$ exchanges coordinates $p$ and $q$.

the square of the Frobenius norm (sum of squares of the elements), (v) in $\mathbf{\Sigma}_K(\mathbf{f}, T, \mathbf{P})$, $\mathbf{y} = \mathbf{P}\hat{\mathbf{s}}_{\mathbb{K}\text{-ICA}}$, and (vi) $\mathbf{P}$ is the $D \times D$ permutation matrix to be determined.

Greedy permutation search is applied: 2 coordinates of different subspace are exchanged if this change lowers cost function $Q(\mathcal{F}, T, \cdot)$. Note: Greedy search could be replaced by a *global* one for a higher computational burden [11]. Table 1 contains the pseudocode of our technique.

# 4 Illustrations

The $\mathbb{K}$-ISA identification algorithm of Section 3.3 is illustrated below (due to the lack of space, illustrations are provided for $\mathbb{K} = \mathbb{R}$ only). Test cases are introduced in Section 4.1. The quality of the solutions will be measured by the normalized Amari-distance (Section 4.2). Numerical results are provided in Section 4.3.

## 4.1 Databases

Two databases were defined to study our algorithm. The databases are illustrated in Fig. 1 and Fig. 2.

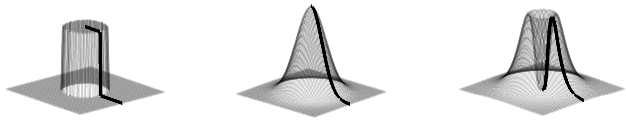
### *4.1.1 The $A\omega$ Database*

Here, hidden sources $\mathbf{s}^m$ are uniform distributions defined by the 2-dimensional images ($d = 2$) of letters on A-Z and $\alpha - \omega$. This is called database $A\omega$, which has 50 components ($M = 50$), see Fig. 1 for an illustration. This test falls outside of the (known) validity domain of the $\mathbb{R}$-ISA Separation Theorem.

### *4.1.2 The $d$-Spherical Database*

Here, hidden sources $\mathbf{s}^m$ are spherically symmetric random variables that have representation of the form $\mathbf{v} \stackrel{\text{distr}}{=} \rho\mathbf{u}^{(d)}$, where $\mathbf{u}^{(d)}$ is uniformly distributed on the $d$-dimensional unit sphere, and $\rho$ is a non-negative scalar random variable independent of $\mathbf{u}^{(d)}$ ($\stackrel{\text{distr}}{=}$ denotes equality in distribution). This *d-spherical* database: (i) can be scaled in dimension $d$, (ii) satisfies conditions of the $\mathbb{R}$-ISA Separation Theorem, and (iii) can be defined by $\rho$. (See [5, 6] for spherical variables.) Our choices for $\rho$ are shown in Fig. 2.

Figure 1: Database $A\omega$. 100-dimensional task of 50 pieces of 2-dimensional components ($D = 100$, $M = 50$, $d = 2$). Hidden sources are uniformly distributed variables on the letters of the English and the Greek alphabets.



(a) (b) (c)

Figure 2: Database $d$-spherical. Stochastic representation of the 3 ($M = 3$) hidden sources. (a): $\rho$ is uniform on $[0, 1]$, (b): $\rho$ is exponential with parameter $\mu = 1$, and (c): $\rho$ is lognormal with parameters $\mu = 0, \sigma = 1$, respectively.

## 4.2 Normalized Amari-distance

The precision of our algorithm was measured by the normalized Amari-distance as follows. The optimal estimation of the $\mathbb{R}$-ISA model provides matrix $\mathbf{B} := \mathbf{W}\mathbf{A} \in \mathbb{R}^{D\times D}$, a block-permutation matrix made of $d \times d$ sized blocks. Let us decompose matrix $\mathbf{B} \in \mathbb{R}^{D\times D}$ into $d \times d$ blocks: $\mathbf{B} = \left[\mathbf{B}^{i,j}\right]_{i,j=1,\ldots,M}$. Let $b^{i,j}$ denote the sum of the absolute values of the elements of matrix $\mathbf{B}^{i,j} \in \mathbb{R}^{d\times d}$. Then the *normalized* version of the Amari-distance [2] (as it was introduced in [11] for $\mathbb{R}$-ISA) is defined as:

$$r(\mathbf{B}) := \frac{1}{2M(M-1)}\left[\sum_{i=1}^{M}\left(\frac{\sum_{j=1}^{M} b^{ij}}{\max_j b^{ij}} - 1\right) + \sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{M} b^{ij}}{\max_i b^{ij}} - 1\right)\right]. \quad (14)$$

For matrix $\mathbf{B}$ we have that $0 \leq r(\mathbf{B}) \leq 1$, and $r(\mathbf{B}) = 0$ if, and only if $\mathbf{B}$ is a block-permutation matrix with $d \times d$ sized blocks. Thus, $r = 0$ corresponds to perfect estimation (0% error), $r = 1$ is the worst estimation (100% error). This performance measure can be used for $\mathbb{K} = \mathbb{C}$, too.

## 4.3 Simulations

Results on databases $A\omega$ and *d-spherical* are provided here. In our simulations, sample number of observations $\mathbf{z}(t)$ changed: $1000 \leq T \leq 30000$. Mixing matrix $\mathbf{A}$ was chosen randomly from the orthogonal group. Manifold $\mathcal{F}$ was $\mathcal{F} := \{\mathbf{z} \mapsto \cos(\mathbf{z}), \mathbf{z} \mapsto \cos(2\mathbf{z})\}$ (functions

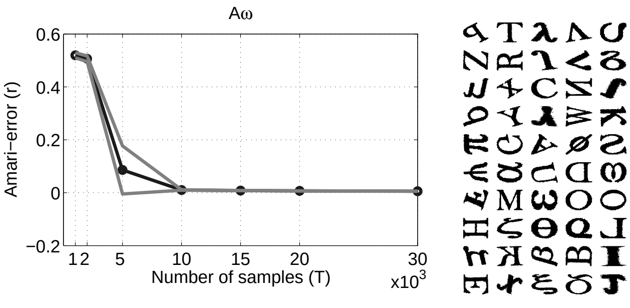



Figure 3: Estimations on database $A\omega$. (a) Amari-error as a function of the number of samples. Average±deviation for 30000 samples: $0.58\% \pm 0.04$, (b) estimation with average error for 30000 samples: the hidden components are recovered up to permutation and orthogonal transformation ($\mathbb{R}$-ISA ambiguity).

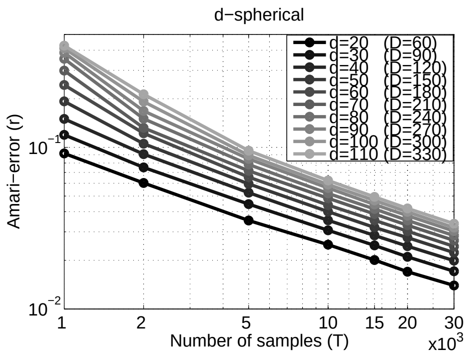



Figure 4: Estimations of database $d$-spherical: Amari-error as a function of the number of samples on loglog scale for different dimensional ($d$) subspaces. Task dimension: $D$. Errors are approximately linear, so they scale according to power law, like $r(T) \propto T^{-c}$ ($c > 0$). For numerical values, see Table 2.

operated on coordinates separately). Scaling properties of the approximation were studied for database $d$-spherical by changing the value of $d$ between 20 and 110 [i.e., the number of subspaces ($M$) was fixed, but the dimension of the subspaces was increased.] For each parameters [$T$ for database $A\omega$, $(T, d)$ for database $d$-spherical] ten experiments were averaged. Qualities of the solutions were measured by the Amari-error (see Section 4.2). We have chosen FastICA [8] for the $\mathbb{R}$-ICA module (see Table 1).

Precision of our method is shown: (i) for database $A\omega$ in Fig. 3 as a function of sample number, (ii) for database $d$-spherical in Fig. 4 as a function of sample number and source dimension ($d$) (for details, see Table 4). The figures demonstrate that the algorithm was able to uncover the hidden components with high precision. In the case of database $d$-spherical the Amari error decreases according to power law $r(T) \propto T^{-c}$ ($c > 0$).

In our numerical simulations, the number of sweeps before the iteration of the permutation optimization stopped (see Table 1) varied between 2 and 6.

Table 2: Amari-error for database $d$-spherical, for different $d$ values: average ± deviation. Number of samples: $T = 30000$.

| $d = 20$ | $d = 30$ | $d = 40$ |
|---|---|---|
| 1.40% (±0.03) | 1.71% (±0.03) | 1.99% (±0.03) |
| $d = 50$ | $d = 60$ | $d = 70$ |
| 2.23% (±0.03) | 2.44% (±0.03) | 2.65% (±0.03) |
| $d = 80$ | $d = 90$ | $d = 100$ |
| 2.85% (±0.03) | 3.03% (±0.04) | 3.19% (±0.02) |
| $d = 110$ | | |
| 3.37% (±0.03) | | |

# References


[1] S. Amari, A. Cichocki, and H.H. Yang. Recurrent neural networks for blind separation of sources. NOLTA'95, pp. 37-42, 1995.

[2] S. Amari, A. Cichocki, and H. Yang. A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems*, 1996.

[3] Francis R. Bach and Michael I. Jordan. Kernel ICA. *JMLR*, 3:1–48, 2002.

[4] Jan Eriksson and Visa Koivunen. Complex random vectors and ICA models: Identifiability, uniqueness and separability. *IEEE TIT*, 52(3), 2006.

[5] Kai-Tai Fang, Samuel Kotz, and Kai Wang Ng. *Symmetric multivariate and related distributions*. Chapman and Hall, 1990.

[6] Gabriel Frahm. *Generalized elliptical distributions: Theory and applications*. PhD thesis, University of Köln, 2004.

[7] A. Gretton, R. Herbrich, and A. Smola. The kernel mutual information. In *IEEE ICASSP*, volume 4, pages 880–883, 2003.

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

[9] Anke Meyer-Bäse, Peter Gruber, Fabian Theis, and Simon Foo. Blind source separation based on self-organizing neural network. *Engng. Appl. of AI*, 19: 305–311, 2006.

[10] B. Póczos and A. Lőrincz. Non-combinatorial estimation of independent AR sources. *Neurocomp.*, 2006. accepted.

[11] Z. Szabó, B. Póczos, and A. Lőrincz. Cross-entropy optimization for independent process analysis. In *Proc. of ICA*, LNCS 3889, pages 909–916. Springer-Verlag, 2006.

[12] Z. Szabó, B. Póczos, and A. Lőrincz. Separation theorem for $\mathbb{K}$-independent subspace analysis with sufficient conditions. Technical report, Eötvös Loránd University, Budapest, 2006. http://arxiv.org/abs/math.ST/0608100.

[13] F.J. Theis. Uniqueness of complex and multidimensional independent component analysis. *Signal Processing*, 84(5):951–956, 2004.