Generalizing Independent Component Analysis for Two Related Data Sets

Juha Karhunen, *Senior Member, IEEE*, and Tomas Ukkonen

***Abstract*— We introduce in this paper methods for finding mutually corresponding dependent components from two different but related data sets in an unsupervised (blind) manner. The basic idea is to generalize cross-correlation analysis for taking into account higher-order statistics. We propose independent component analysis (ICA) type extensions for the singular value decomposition of the cross-correlation matrix. They extend cross-correlation analysis in a similar manner as ICA extends standard principal component analysis for covariance matrices. We present experimental results demonstrating the usefulness of the proposed methods both for artificially generated data and for a cryptographic problem.**

I. Introduction

Principal component analysis (PCA) [5], [4], [17] and independent component analysis (ICA) [17], [4] are well-known techniques for unsupervised (blind) extraction of useful information from vector-valued data $\mathbf{x}$. While PCA is a well-established, old statistical technique, ICA has gained a lot of popularity during the last decade because it often provides more meaningful results.

Standard linear PCA and ICA are both based on the same type of simple linear latent variable model for the observed data vector $\mathbf{x}(t)$:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) = \sum_{i=1}^{n} s_i(t)\mathbf{a}_i \tag{1}$$

In this model, the data vector $\mathbf{x}(t)$ is expressed as a linear combination of scalar sources $s_i(t)$, $i = 1, 2, \ldots, n$, which multiply the respective constant basis vectors $\mathbf{a}_i$, $i = 1, 2, \ldots, n$. The sources are in different contexts called also latent variables, (hidden) factors, or (hidden) causes. The index t may denote time, position, or just number of the sample vector, again depending on the context. For simplicity, we assume here that both the data vector $\mathbf{x}(t)$ = $[x_1(t), x_2(t), \ldots, x_n(t)]^T$ and the source vector $\mathbf{s}(t)$ = $[s_1(t), s_2(t), \ldots, s_n(t)]^T$ are zero mean n-vectors, and that the mixing matrix $\mathbf{A}$ = $[\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ is a full-rank constant $n \times n$ matrix. The column vectors $\mathbf{a}_i$, $i = 1, 2, \ldots, n$ of the mixing matrix $\mathbf{A}$ comprise the basis vectors of PCA or ICA, and the components $s_i(t)$, $i = 1, 2, \ldots, n$, of the source vector $\mathbf{s}(t)$ are respectively principal or independent components corresponding to the data vector $\mathbf{x}(t)$. From now on, the index t is left out, assuming that the order of the data vectors $\mathbf{x}(t)$ is not important. This assumption is made in standard ICA.

In PCA, it is required that the mixing matrix is orthogonal: $\mathbf{A}^T\mathbf{A}$ = $\mathbf{I}$, leading to mutually orthonormal basis vectors $\mathbf{a}_i$. In ICA, there is no such requirement, and hence the mixing matrix $\mathbf{A}$ and the basis vectors $\mathbf{a}_i$ of ICA are generally non-orthogonal. In both the expansions, the components s_i must be mutually uncorrelated: $\mathrm{E}\{s_i s_j\} = 0$, $i \neq j$. To get the true principal components, the variances $\mathrm{E}\{s_i^2\}$ are in addition sequentially maximized for $i = 1, 2, \ldots, n$ [5], [18], [17], [4]. In ICA, the orthogonality condition of PCA is replaced by the strong but often realistic requirement that the components s_i of the source vector $\mathbf{s}$ should be statistically independent (or as independent as possible). This still leaves the sign, order, and scaling of the independent components s_i ambiguous [17]. Usually they are scaled so that their variances $\mathrm{E}\{s_i^2\} = 1$.

Assuming zero mean, $\mathrm{E}\{\mathbf{x}\} = \mathbf{0}$, the covariance matrix of the data $\mathbf{x}$ is both for PCA and ICA

$$\mathbf{C}_{xx} = \mathrm{E}\{\mathbf{x}\mathbf{x}^T\} = \mathbf{A}\mathrm{E}\{\mathbf{s}\mathbf{s}^T\}\mathbf{A}^T = \mathbf{A}\mathbf{C}_{ss}\mathbf{A}^T \tag{2}$$

where the covariance matrix $\mathbf{C}_{ss}$ = $\mathrm{E}\{\mathbf{s}\mathbf{s}^T\}$ of the source vector $\mathbf{s}$ is a diagonal matrix due to the uncorrelatedness of the components s_i.

Because PCA considers second-order statistics (covariances) only, it can be easily computed using the eigendecomposition of the covariance matrix $\mathbf{C}_{xx}$. An alternative though less accurate way is to apply linear PCA neural networks taught by Hebbian (and possibly anti-Hebbian) learning rules [5], [4]. Such stochastic gradient algorithms for estimating the PCA expansion were developed by the first author together with Prof. E. Oja in a somewhat different context already in early 1980's [20], [24]. The ICA expansion is somewhat more difficult to estimate, requiring higher-order statistics, but several good batch or adaptive neural type algorithms now exists for computing it, too [17], [4].

Both standard ICA and PCA have been generalized into many different directions [5], [17], [18], [4], [14]. In this paper, we consider a generalization in which one tries to find mutually dependent corresponding components from two different but related data sets $\mathbf{x}$ and $\mathbf{y}$. For simplicity, we assume in this paper that such dependences appear between transformed components of $\mathbf{x}$ and $\mathbf{y}$ pairwise, while their other component pairs are statistically fairly independent. We note here that it would be possible to consider more flexible and general subspace type models.

Juha Karhunen is with the Adaptive Informatics Research Centre, Laboratory of Computer and Information Science, Helsinki University of Technology, P.O.Box 5400, FIN-02015 HUT, Espoo, Finland (Fax: +358-9-451 3277; email: Juha.Karhunen@hut.fi; URL: http://www.cis.hut.fi/juha).

Tomas Ukkonen is currently with the Finnish Geodetic Institute, P.O.Box 15, FIN-02431 Masala, Finland (email: Tomas.Ukkonen@iki.fi; URL: http://www.fgi.fi).

A well-known related statistical technique is canonical correlation analysis [23]. There one tries to find linear combinations x^* and y^* of the components of the vectors $\mathbf{x}$ and $\mathbf{y}$, respectively, so that x^* and y^* have maximal correlations. Because canonical correlation analysis resorts to second-order statistics only, its solution can again be found using eigenanalysis and singular value decomposition of auto- and cross-covariance matrices of $\mathbf{x}$ and $\mathbf{y}$ [23]. Fyfe and Lai have considered a neural implementation of canonical correlation analysis in [22], and a nonlinear generalization of it using kernels in [8]. Furthermore, Koetsier et al. have presented in [21] an unsupervised neural algorithm called Exploratory Correlation Analysis for the extraction of common features in multiple data sources. This method, is also closely related with canonical correlation analysis.

In an interesting paper, Akaho and his co-authors [2] have considered an ICA style generalization of canonical correlation analysis which they call multimodal independent component analysis (MICA). In their method, standard linear ICA is first applied to both data sets $\mathbf{x}$ and $\mathbf{y}$ separately. Then the corresponding dependent components of the two ICA expansions are identified using a natural gradient type learning rule. Akaho's method seems to work in most cases in practice, but it has a potential theoretical weakness. If two scalar variables s_1 and s_2 are statistically independent and similarly t_1 and t_2, but s_1 and t_1 depend on each other and similarly s_2 and t_2, one cannot in general theoretically deduce anything on the dependence or independence of the variable pairs s_1 and t_2 or s_2 and t_1. For example, s_1 and t_2 may have a common part which does not appear in s_2 and t_1, which makes them statistically dependent.

II. Theoretical background

A. Removal of second-order dependencies

Consider two different but related data sets $\mathbf{x}$ and $\mathbf{y}$. The dimension m of $\mathbf{y}$ is in general different from the dimension n of $\mathbf{x}$. Assuming zero mean also for $\mathbf{y}$, the cross-covariance matrix of $\mathbf{x}$ and $\mathbf{y}$ is

$$\mathbf{C}_{xy} = \mathrm{E}\{\mathbf{x}\mathbf{y}^T\} \tag{3}$$

The elements $\mathrm{E}\{x_i y_j\}$ of this matrix are cross-covariances between the components x_i and y_j of the vectors $\mathbf{x}$ and $\mathbf{y}$, and they are in general nonzero.

The cross-covariance matrix $\mathbf{C}_{xy}$ can be diagonalized using its singular value decomposition (SVD) (see for example [13], [5]):

$$\mathbf{C}_{xy} = \mathbf{U}\mathbf{D}_{st}\mathbf{V}^T \tag{4}$$

Here $\mathbf{U}$ and $\mathbf{V}$ are $n \times n$ and $m \times m$ orthogonal matrices, respectively, and

$$\mathbf{D}_{st} = \mathrm{E}\{\mathbf{s}\mathbf{t}^T\} \tag{5}$$

is an $n \times m$ (pseudo)diagonal matrix (that is, a diagonal matrix appended with zeros if $m \neq n$ [13]). The matrices $\mathbf{U}$ and $\mathbf{V}$ and $\mathbf{D}_{st}$ are obtained from the eigendecompositions of the symmetric matrices $\mathbf{C}_{xy}\mathbf{C}_{xy}^T$ and $\mathbf{C}_{xy}^T\mathbf{C}_{xy}$, respectively [5], [13]. Standard PCA is a special case of SVD the expansion (4) in which $\mathbf{x} = \mathbf{y}$, $\mathbf{U} = \mathbf{V}$, and $\mathbf{s} = \mathbf{t}$. SVD can be estimated using neural PCA type algorithms [4], too, but we have in this work used more efficient and accurate standard numerical algorithms for computing it.

We can think that the diagonalization (4) of the cross-covariance matrix $\mathbf{C}_{xy}$ is realized via two orthogonal linear transformations $\mathbf{U}$ and $\mathbf{V}$:

$$\mathbf{x} = \mathbf{U}\mathbf{s}, \quad \mathbf{y} = \mathbf{V}\mathbf{t} \tag{6}$$

where the corresponding components s_i and t_i of the vectors $\mathbf{s}$ and $\mathbf{t}$ are correlated: $\mathrm{E}\{s_i t_i\} \neq 0$, but their different components are uncorrelated: $\mathrm{E}\{s_i t_j\} = 0$ for $i \neq j$. Later on in our experiments, to make the comparisons easier, the variances of the components of the vectors $\mathbf{x}$ and $\mathbf{y}$ are always normalized to unity.

The key idea in this work is to allow non-orthogonal square transformation matrices $\mathbf{A}$ and $\mathbf{B}$ instead of $\mathbf{U}$ and $\mathbf{V}$:

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \quad \mathbf{y} = \mathbf{B}\mathbf{t} \tag{7}$$

In a similar manner as in standard linear ICA for one data set $\mathbf{x}$, we require that the transformations $\mathbf{A}$ and $\mathbf{B}$ not only make the different components s_i and t_j, $i \neq j$, of the vectors $\mathbf{s}$ and $\mathbf{t}$ uncorrelated, but they should be as independent as possible. The goal is to concentrate the dependencies between the vectors $\mathbf{s}$ and $\mathbf{t}$ as far as possible to their corresponding components s_i and t_i, which are in turn required to be as dependent as possible.

Using the transformations (7), the cross-covariance matrix $\mathbf{C}_{xy}$ can be expressed as

$$\mathbf{C}_{xy} = \mathbf{A}\mathbf{D}_{st}\mathbf{B}^T \tag{8}$$

It should be noted that it is always possible to find orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ providing the singular value decomposition (4) and making the different components of the vectors $\mathbf{x}$ and $\mathbf{y}$ uncorrelated. By finding suitable transformations (7) among the considerably more flexible class of non-orthogonal matrices $\mathbf{A}$ and $\mathbf{B}$, one should therefore in general be able to achieve more than just decorrelation.

B. Removal of higher order dependencies

Our approach for computing the matrices $\mathbf{A}$ and $\mathbf{B}$ is based on nonlinear decorrelation and the FastICA algorithm [17]. The algorithm has converged to a good solution when

$$\mathrm{E}\{\mathbf{x}\mathbf{g}(\mathbf{x})^T\} \tag{9}$$

is a diagonal matrix, and the data vectors $\mathbf{x}$ have been preprocessed to have zero mean and unit variance. The vector $\mathbf{g}(\mathbf{x}) = [g(x_1), g(x_2), \ldots, g(x_n)]^T$ is a nonlinear transformation of the data vector $\mathbf{x}$. The nonlinearity $g(t)$ must be chosen carefully in order to get as independent signals as possible. Good nonlinearities for wide classes of signals are $g(t) = \tanh(t)$ or $g(t) = t^3$.

From this result we see that the signals x_i and x_j, $i \neq j$, are usually statistically independent when their nonlinear correlations

$$\mathrm{E}\{x_i g(x_j)\} = 0, \quad \mathrm{E}\{x_j g(x_i)\} = 0. \tag{10}$$

Therefore, to remove cross-dependencies between the zero mean vectors $\mathbf{x}$ and $\mathbf{y}$, we need to diagonalize the matrices

$$\mathrm{E}\{\mathbf{x}\mathbf{y}^T\}, \quad \mathrm{E}\{\mathbf{x}\mathbf{g}(\mathbf{y})^T\}, \quad \mathrm{E}\{\mathbf{g}(\mathbf{x})\mathbf{y}^T\}. \tag{11}$$

Furthermore, because the nonlinearities $g(x) = \tanh(x)$ and $g(x) = x^3$ above have the property that they preserve the sign of x, it is possible to diagonalize all these matrices by just diagonalizing a single matrix

$$\mathrm{E}\{\mathbf{x}\mathbf{y}^T + \mathbf{x}\mathbf{g}(\mathbf{y})^T + \mathbf{g}(\mathbf{x})\mathbf{y}^T\} \tag{12}$$

This matrix can be further generalized to

$$\mathrm{E}\{[\mathbf{x} + (\mathbf{g}(\mathbf{x}) - \mathrm{E}\{\mathbf{g}(\mathbf{x})\})][\mathbf{y} + (\mathbf{g}(\mathbf{y}) - \mathrm{E}\{\mathbf{g}(\mathbf{y})\})]^T\} \tag{13}$$

where the term $\mathrm{E}\{(\mathbf{g}(\mathbf{x}) - \mathrm{E}\{\mathbf{g}(\mathbf{x})\})(\mathbf{g}(\mathbf{y}) - \mathrm{E}\{\mathbf{g}(\mathbf{y})\})^T\}$ vanishes when the vectors $\mathbf{x}$ and $\mathbf{y}$ are independent.

We can diagonalize this form by simple use of SVD. In general we want to diagonalize the matrix

$$\mathrm{E}\{\mathbf{f}(\mathbf{x})\mathbf{g}(\mathbf{y})^T\} = \mathbf{U}\mathbf{S}\mathbf{V}^T. \tag{14}$$

We can do this nonlinearly with tranforms

$$\mathbf{x}' = \mathbf{f}^{-1}(\mathbf{U}^T\mathbf{f}(\mathbf{x})), \quad \mathbf{y}' = \mathbf{g}^{-1}(\mathbf{V}^T\mathbf{g}(\mathbf{y})) \tag{15}$$

provided that the inverse functions $\mathbf{f}^{-1}(\cdot)$ and $\mathbf{g}^{-1}(\cdot)$ exist.

Assume now that the data vectors $\mathbf{x}$ and $\mathbf{y}$ have been whitened and cross-decorrelated. For vector-valued functions $\mathbf{f}(\mathbf{x})$ which map their components independently, the optimal linear mapping in the mean-square error sense is then $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x} = a\mathbf{I}\mathbf{x}$, and similarly for $\mathbf{g}(\mathbf{y})$. This can be used to find linear approximations to the non-linear diagonalizing transforms (15):

$$\mathbf{x}' = a^{-1}\mathbf{I}\mathbf{U}^T a\mathbf{I}\mathbf{x} = \mathbf{U}^T\mathbf{x} \tag{16}$$

$$\mathbf{y}' = b^{-1}\mathbf{I}\mathbf{V}^T b\mathbf{I}\mathbf{y} = \mathbf{V}^T\mathbf{y} \tag{17}$$

These approximations, although suboptimal, turned out to give good results in our experiments.

III. Methods

We have developed and tested several somewhat heuristic methods based on the above ideas. We restricted our testing to matrices which have a similar form as in Eq. (12). Eqs. (16) and (17) could also be used iteratively to totally or significantly reduce non-diagonal values of this form of matrix containing nonlinear correlations. In the following, we present the two methods which performed on average best in our experiments.

A. Method 1

In the first method (Method 1), we first estimate the independent components[1] of the vectors $\mathbf{x}$ and $\mathbf{y}$. Let us denote the vectors containing these estimated independent components by $\hat{\mathbf{s}}_x$ and $\hat{\mathbf{s}}_y$:

$$\mathbf{x} = \mathbf{A}\hat{\mathbf{s}}_x, \quad \mathbf{y} = \mathbf{B}\hat{\mathbf{s}}_y \tag{18}$$

[1]Or the most independent components if strictly statistically independent components do not exist.

Here $\mathbf{A}$ is an $n \times n$ matrix and $\hat{\mathbf{s}}_x$ an n-dimensional vector, and $\mathbf{B}$ is an $m \times m$ matrix and $\hat{\mathbf{s}}_y$ an m-dimensional vector. Furthermore, the variances of vectors $\hat{\mathbf{s}}_x$ and $\hat{\mathbf{s}}_y$ were normalized to unity for getting suitable starting vectors.

After this, the singular value decomposition of the matrix

$$\mathbf{F}_{xy} = \mathrm{E}\{\mathbf{x}\mathbf{y}^T + \tanh(\mathbf{x})\mathbf{y}^T + \mathbf{x}\tanh(\mathbf{y})^T\} = \mathbf{U}_F\mathbf{D}_F\mathbf{V}_F^T \tag{19}$$

containing nonlinear correlations of the vectors $\mathbf{x}$ and $\mathbf{y}$ is computed quite similarly as for the standard cross-correlation matrix $\mathbf{C}_{xy}$ in (4). On the right hand side of Eq. (19), $\mathbf{U}_F$ and $\mathbf{V}_F$ denote the orthogonal left and right matrices of the singular value decomposition of the matrix $\mathbf{F}_{xy}$, and the diagonal matrix $\mathbf{D}_F$ contains the respective singular values. The nonlinearity, in (19) $\tanh(\cdot)$, is applied to each component of the vectors $\mathbf{x}$ and $\mathbf{y}$ separately.

Finally, the estimated source (independent component) vectors $\hat{\mathbf{s}}_x$ and $\hat{\mathbf{s}}_y$ in Eq. (18) are rotated using the singular vector matrices $\mathbf{U}_F$ and $\mathbf{V}_F$, yielding the final results

$$\mathbf{s}_x^* = \mathbf{U}_F^T\hat{\mathbf{s}}_x, \quad \mathbf{s}_y^* = \mathbf{V}_F^T\hat{\mathbf{s}}_y \tag{20}$$

The basic idea behind this and the following method is to include nonlinear correlations of the components of the vectors $\mathbf{x}$ and $\mathbf{y}$ into computation of the matrix $\mathbf{F}_{xy}$. In (19), the sigmoidal $\tanh(\cdot)$ nonlinearity is applied to $\mathbf{x}$ and $\mathbf{y}$ to achieve this goal.

This is a heuristic way to try to concentrate the dependencies between $\mathbf{x}$ and $\mathbf{y}$ into their corresponding components. But due to the averaged nature of the expectation defining the matrix $\mathbf{F}_{xy}$, one cannot in general claim that other than targeted component pairs of $\mathbf{s}_x^*$ and $\mathbf{s}_y^*$ are statistically independent or even uncorrelated.

B. Method 2

In the second method, the zero mean data vectors $\mathbf{x}$ and $\mathbf{y}$ are first whitened using PCA. This takes place by first computing the PCA expansions of $\mathbf{x}$ and $\mathbf{y}$:

$$\mathbf{C}_{xx} = \mathbf{U}_x\mathbf{D}_{xx}\mathbf{U}_x^T, \quad \mathbf{C}_{yy} = \mathbf{U}_y\mathbf{D}_{yy}\mathbf{U}_y^T \tag{21}$$

where the column vectors of the orthogonal matrices $\mathbf{U}_x$ and $\mathbf{U}_y$ contain the eigenvectors of the covariance matrices $\mathbf{C}_{xx}$ and $\mathbf{C}_{yy}$ of $\mathbf{x}$ and $\mathbf{y}$, respectively. The diagonal matrices $\mathbf{D}_{xx}$ and $\mathbf{D}_{yy}$ consist of their eigenvalues in the same order. The whitened data vectors are given by [17]

$$\mathbf{x}' = \mathbf{D}_{xx}^{-1/2}\mathbf{U}_x^T\mathbf{x}, \quad \mathbf{y}' = \mathbf{D}_{yy}^{-1/2}\mathbf{U}_y^T\mathbf{y} \tag{22}$$

The components of the preprocessed vectors $\mathbf{x}'$ are mutually uncorrelated and have unit variance, and similarly for $\mathbf{y}'$.

The whitened vectors $\mathbf{x}'$ and $\mathbf{y}'$ are then rotated further using alternatively the singular value decompositions of the nonlinear correlation matrices $\mathbf{F}_{xy}$ in (19) and $\mathbf{G}_{xy}$:

$$\mathbf{G}_{xy} = \mathrm{E}\{\mathbf{x}\mathbf{y}^T + \mathbf{x}^3\mathbf{y}^T + \mathbf{x}(\mathbf{y}^T)^3\} = \mathbf{U}_G\mathbf{D}_G\mathbf{V}_G^T \tag{23}$$

This takes place quite similarly as in our first method,

$$\mathbf{s}_x^* = \mathbf{U}^T\mathbf{x}', \quad \mathbf{s}_y^* = \mathbf{V}^T\mathbf{y}' \tag{24}$$

where the matrices $\mathbf{U}$ and $\mathbf{V}$ are alternatively taken from the SVDs (19) and (23). The iteration is continued until convergence. Thus Method 2 uses in addition to the sigmoidal nonlinearity $\tanh(\cdot)$ the cubic nonlinearities $\mathbf{x}^3$ and $\mathbf{y}^3$ for including nonlinear correlations of the vectors $\mathbf{x}$ and $\mathbf{y}$ into computations.

We tried several slightly different methods of similar type in our experiments. The two methods described above were selected to this paper because they provided on average the best results and are computationally sufficiently efficient.

C. Method 3

The just described Methods 1 and 2 try to find one-dimensional signal pairs s_i, t_i where all the relevant information about the ith component t_i of the vector $\mathbf{t}$ has been concentrated onto the corresponding component s_i of the vector $\mathbf{s}$ and visa versa. These ideas can be also used to find a linear mapping between two sets of signals.

Method 3 extends the linear mean-squared error minimization to a more generic linear method. The method relaxes assumptions about distributions of signals and errors. The idea is to solve signal pairs with one of the methods described above, and then find one-dimensional mappings minimizing the mean-square error between s_i and t_i pairs. These one-dimensional mappings are sufficient for cross-independent signal pairs, where the signals $s_j, j \neq i$ do not contain any information about the correct value of t_i. An optimum linear mapping minimizing the mean-square error changes only the sign and scaling of zero mean signals [12], and can be carried out without changing mutual information (or independencies) between the variables

$$t_i = \rho_{t_i s_i} s_i, \quad I(\rho_{ii} X, Y) = I(X, Y) \tag{25}$$

where $\rho_{t_i s_i}$ is correlation between t_i and s_i. Thus if the given data have been sphered to have zero mean and unit variance, and the cross-dependence matrix $\mathbf{G}_{xy}$ can be diagonalized with mappings $\mathbf{s} = \mathbf{U}^T\mathbf{x}$ and $\mathbf{t} = \mathbf{V}^T\mathbf{y}$ then the mapping from $\mathbf{x}$ to $\mathbf{y}$ is

$$\mathbf{W} = \mathbf{V} diag(\rho_{t_1 s_1}, \rho_{t_2 s_2}, \ldots, \rho_{t_N s_N}) \mathbf{U}^T \tag{26}$$

This method can be seen as an extension of linear mean-square error optimization which assumes Gaussian distributions. If a cross-correlation matrix $\mathbf{C}_{xy}$ is used as a cross-dependence matrix $\mathbf{G}_{xy}$, then the resulting mapping becomes

$$\mathbf{W} = \mathbf{V} diag(\rho_{t_1 s_1}, \rho_{t_2 s_2}, \ldots, \rho_{t_N s_N}) \mathbf{U}^T = \mathbf{V}\mathbf{S}\mathbf{U}^T = \mathbf{C}_{xy}^T \tag{27}$$

It can be seen that this is exacly the same as given by linear mean-square error minimization for sphered data. With a bit of calculus one can see that minimization of the mean-square error criterion

$$\mathrm{E}\{\frac{1}{2}||\mathbf{W}\mathbf{x} - \mathbf{y}||^2\} \tag{28}$$

yields the optimal solution

$$\mathbf{W} = \mathbf{C}_{yx}\mathbf{C}_x^{-1} = \mathbf{C}_{xy}^T \tag{29}$$

where the last step follows from the whitening (sphering) of the data $\mathbf{x}$: $\mathbf{C}_x = \mathbf{C}_x^{-1} = \mathbf{I}$. So if diagonalization of a cross-dependence matrix removes most of the correlations between different components of the involved vectors, then Method 3 minimizes the mean-square error and at the same time it tries to take non-Gaussian properties of distributions into account.

In our tests we preprocessed the data with PCA to have zero mean and unit variance, and then used the cross-dependence matrix

$$\mathbf{G}_{xy} = \mathrm{E}\{\tanh(\mathbf{x})\mathbf{y}^T + \mathbf{x}\tanh(\mathbf{y})^T\} \tag{30}$$

which was iteratively diagonalized with SVD. Brief tests indicated that this method has about same minimum mean-square error (28) when $\mathbf{y} = \mathbf{A}\mathbf{x} + \epsilon$. But sometimes the method performed considerably better than the standard pseudoinverse based least-square error minimization (29) when the output vectors $\mathbf{y}$ were generated from $\mathbf{x}$ with two different matrices $\mathbf{y} = \mathbf{A}\mathbf{x}$ and $\mathbf{y} = \mathbf{B}\mathbf{x}$.

IV. Measuring the dependence

Theoretically, a suitable measure of the dependence between any two continuous scalar random variables x and y is their mutual information [14], [17]

$$I_{xy} = \int_{-\infty}^{+\infty} p_x(x) \log \frac{p_x(x)}{p_y(y)} dxdy \tag{31}$$

where $p_x(x)$ and $p_y(y)$ denote the probability density functions of x and y, respectively. The mutual information can easily be generalized for vector-valued random variables $\mathbf{x}$ and $\mathbf{y}$. It is actually the Kullback-Leibler divergence (information) between x and y, and measures the distance between the probability densities $p_x(x)$ and $p_y(y)$ [14], [17].

Mutual information I_{xy} is strictly speaking not a proper distance measure because it is not symmetric for x and y. But it has the following important theoretical property: Mutual information is always nonnegative, and it is zero if and only if x and y are statistically independent. The more dependent they are the larger is their mutual information I_{xy}.

While mutual information is in some sense a theoretically ideal dependence measure, it cannot usually be applied in practice. The basic reason is that it is very difficult to reliably estimate the tails of the distributions $p_x(x)$ and $p_y(y)$ [9], [17]. Therefore, one must resort to some kind of approximations (see for example [17], [4]) or to other simpler dependence measures.

A review of dependence measures related to tests of independence in statistics can be found in the paper [26]. However, such tests are not necessarily most suitable in context with ICA, because they typically make specific assumptions on the distributions of the variables to be studied (for example, Gaussianity).

In ICA and blind source separation (BSS) [17], [4], measures of statistical dependence have been developed and studied in several papers. Bach and Jordan [3] have introduced contrast functions based on canonical correlations in a reproducing kernel Hilbert space. They have shown that these contrast functions are related to mutual information

and have desirable mathematical properties as measures of statistical dependence. Their ideas have recently been developed further in [10], where two new kernel-based functionals are introduced for measuring the degree of independence of random variables.

Another way is to use characteristic functions for defining statistical independence and for measuring dependence. This approach has been studied in [6], [7], leading to three criteria for ICA. Dependence measures can be based either on approximating mutual information using the characteristic function or on applying a moment generating function. Furthermore, simpler quadratic measures for estimating dependence have been developed in [1], [27].

We have tested several of these methods experimentally using simple test cases of three statistically independent source signals. It turned out that the method based on moment generating function performed best in the sense that the difference between the cases of independent and more or less dependent signals was the largest. However, also the other tested methods gave qualitatively correct results. That is, more independent variables provided better values of the respective performance index than more dependent ones.

Accordingly, we chose the method based on moment generating function [6], [7] for measuring dependence in our experiments. In the following, we explain this dependence measure in more detail.

The moment generating function method is based on estimation of the expectation

$$\mathrm{E}[\exp(\mathbf{w}^T\mathbf{x})] = \mathrm{E}[\exp(\sum_{i=1}^{n} w_i x_i)] \tag{32}$$

over the components $x_1, x_2, \ldots, x_n$ of the data vector $\mathbf{x}$. Here $\mathbf{w}$ is the weight vector whose components $w_1, w_2, \ldots, w_n$ define some linear combination of the components of $\mathbf{x}$. Clearly, if $\mathbf{w}^T$ is one of the row vectors of the inverse $\mathbf{A}^{-1}$ of the square mixing matrix $\mathbf{A}$ in the standard linear ICA model (1), $\mathbf{w}^T\mathbf{x}$ becomes one of the independent components s_j [17]. On the other hand, if the components x_i of $\mathbf{x}$ in (32) are statistically independent, Eq. (32) decouples into

$$\mathrm{E}[\exp(\mathbf{w}^T\mathbf{x})] = \prod_{i=1}^{n} \mathrm{E}[\exp(w_i x_i)] \tag{33}$$

Based on this observation, one can estimate for two scalar random variables x_i and x_j the quantity

$$\begin{aligned} d_{x_i x_j}(w_i, w_j) &= \{\mathrm{E}[\exp(w_i x_i + w_j x_j)] \\ &- \mathrm{E}[\exp(w_i x_i)]\mathrm{E}[\exp(w_j x_j)]\}^2 \end{aligned} \tag{34}$$

This is always nonnegative, and becomes zero when the variables x_i and x_j are independent. The moments and moment generating function do not uniquely define the variables x_i and x_j, but a large correspondence implies that the functions are similar.

In the experiments, we measured the independence of a two-dimensional random variable by computing the function [6], [7]

$$\begin{aligned} I_{x_i x_j}[\mathbf{w}] &= d_{x_i x_j}(w_i, w_j) d_{x_i x_j}(-w_i, -w_j) \\ &+ d_{x_i x_j}(-w_i, w_j) d_{x_i x_j}(w_i, -w_j) \end{aligned} \tag{35}$$

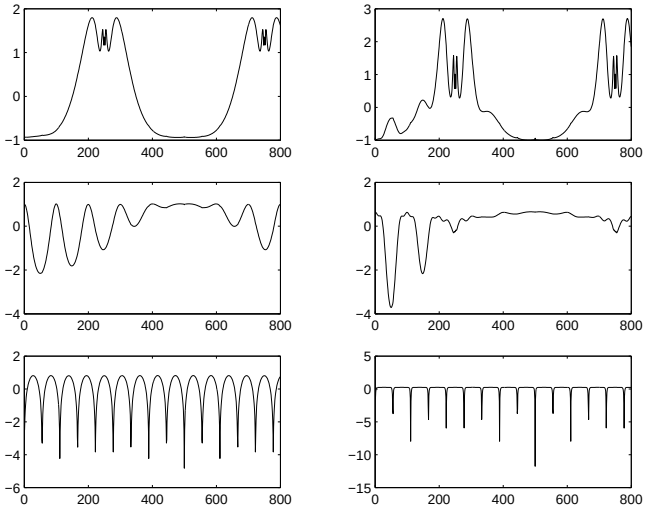

Fig. 1. The source signals $\mathbf{s}(t)$ (left) and $\mathbf{t}(t) = \mathbf{f}(\mathbf{s}(t))$ (right) used to generate input data $\mathbf{x}(t)$ and $\mathbf{y}(t)$. The nonlinearity used in generating the source signal pairs was $f(s) = s^3 - 0.5s^2$. The mean and variance of the generated source signals were set to zero and one.

This is a positive, real-valued function measuring the dependence. We generated this function only at the point $\mathbf{w} = (1, 1)$. Finally, the quality of the found solution was assessed by computing the quantity

$$J(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{n} I_{x_i y_i}(1,1)}{\sum_{i=1}^{n}\sum_{j\neq i}^{n} I_{x_i y_j}(1,1)} \tag{36}$$

The higher the value of $J(\mathbf{x}, \mathbf{y})$, the more dependent $\mathbf{x}$ and $\mathbf{y}$ are. Here t denotes the index of the sample. This is a measure of goodness which tries to take into account both independence and dependence between the targeted pairs $x_i(t), y_i(t)$ and non-pairs $x_i(t), y_j(t)$, $j \neq i$ of the signals.

V. Experimental results

A. Artificially generated data

First, we present some experimental results for artificially generated data. Such data are useful in testing various methods, because the correct results are known, enabling computation of performance or error measures and comparisons.

The original source signals were as follows:

$$\begin{aligned} s_1(t) &= -\mathrm{sinc}(\log|\cos(20\pi t)|) \\ s_2(t) &= \mathrm{sinc}(\cos(10\pi t)\sin(100\pi t)) \\ s_3(t) &= \log|\sin(180\pi t) + 0.01| \\ t_1(t) &= f(s_1) \\ t_2(t) &= f(s_2) \\ t_3(t) &= f(s_3) \end{aligned} \tag{37}$$

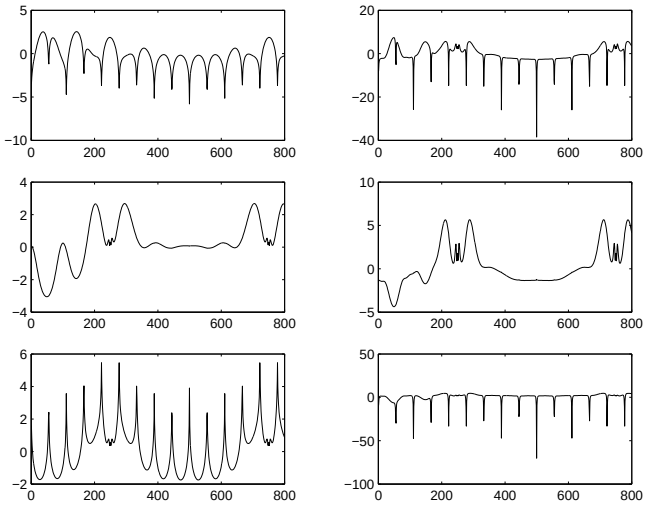

Fig. 2. The generated input data $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$ (left) and $\mathbf{y}(t) = \mathbf{B}\mathbf{t}(t)$ (right).

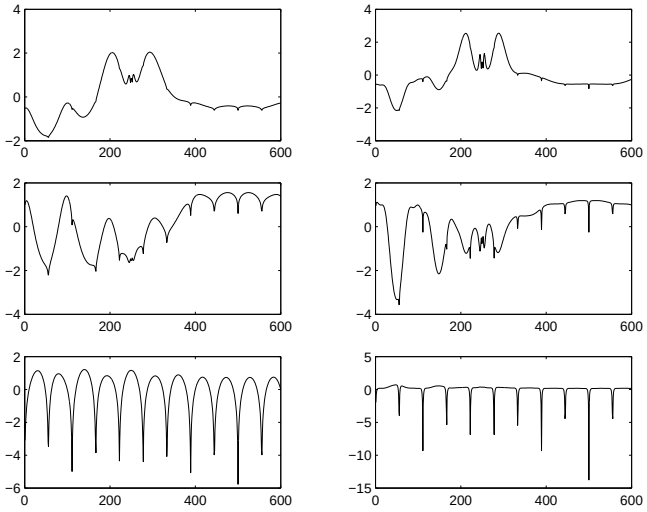

Fig. 3. The jointly dependent sources found using singular value decomposition.

The function $\mathrm{sinc}(t) = \sin(t)/t$ for $t \neq 0$, and 1 for $t = 0$. The sources $s_1(t)$, $s_2(t)$, and $s_3(t)$ are our own and actually not statistically independent. These sources have been depicted in Figure 1.

The first data set $\mathbf{x}(t)$ was obtained by directly mixing original sources $\mathbf{s}(\mathbf{t})$ with a randomly chosen non-singular mixing matrix $\mathbf{A}$. The second related data set $\mathbf{y}(t)$ was generated by first applying the nonlinearity

$$f(s(t)) = [s(t)]^3 - 0.5[s(t)]^2 \tag{38}$$

to the sources $s(t)$, which were then mixed using another randomly chosen non-singular mixing matrix $\mathbf{B}$, yielding $\mathbf{y}(t) = \mathbf{B}f(\mathbf{s}(t))$. The data vectors $\mathbf{x}(t)$ and $\mathbf{y}(t)$ generated in this way are shown in Figure 2.

The jointly dependent sources estimated using different

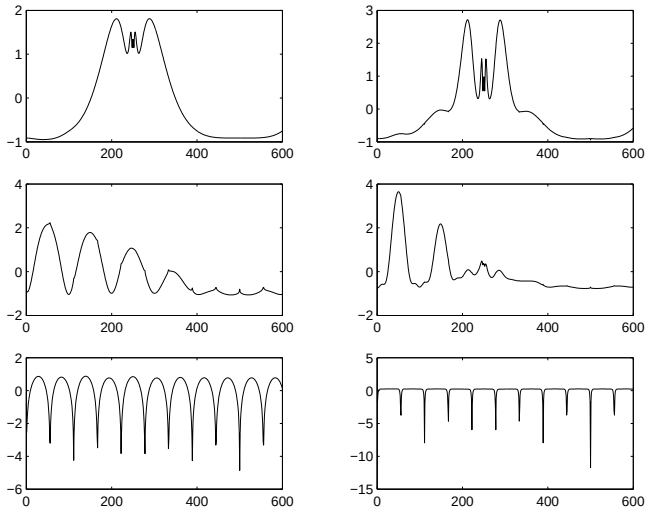

Fig. 4. The jointly dependent sources found using Method 1.

methods have been depicted in Figures 3, 4, and 5. More specifically, Figure 3 shows the sources estimated by the singular value decomposition (4) and (6): $\mathbf{s} = \mathbf{U}^T\mathbf{x}$, $\mathbf{t} = \mathbf{V}^T\mathbf{y}$. Figure 4 depicts the sources provided by our first method (18)-(20), and Figure 5 the sources given by the second method (21)-(24). A visual inspection of the results suggests that the proposed novel methods 1 and 2 perform somewhat better than linear singular value decomposition in this example. This is confirmed in Figure 6, which shows the performance measure $J(\mathbf{x}, \mathbf{y})$ in (36) for the estimated jointly dependent 6 sources. Method 1 (Algorithm 6 in the figure) attains consistently somewhat higher values than SVD (Algorithm 1 in the figure), while our second Method 2 (Algorithm 5) performs for the 2nd and 6th source better than SVD and for the remaining sources about equally well. Figure 7 illustrates the average performances of the algorithms using the moment generating function.

The results were qualitatively similar for the other nonlinearities tried in our simulations. They included the absolute value $f(s(t)) = |s(t)|$ (cf. [2]), a sinusoidal type nonlinearity $f(s(t)) = \sin(s(t))\cos(s(t))$, and a nonlinearity which strongly breaks the signal: $f(s(t)) = \mathrm{floor}[s(t)]\sin(s(t))$, where the function $\mathrm{floor}(t)$ means rounding t to the nearest integer which has a lower or equal value. Hence for example $\mathrm{floor}(5.1) = 5$ and $\mathrm{floor}(-3.3) = -4$. In particular our second method was consistently among the three best ones of the 13 methods tried for all error measures. Our first method performed the best when a simple correlation between the best matching pairs of two variables or characteristic function was used as an error measure.

B. Application to cryptographic data

In these experiments, we tried to find out the dependent corresponding components from texts and their encrypted versions. The texts were taken from the data sets made available by Project Gutenberg [11] in ASCII form. We picked up 4 books, with each ASCII letter at the same

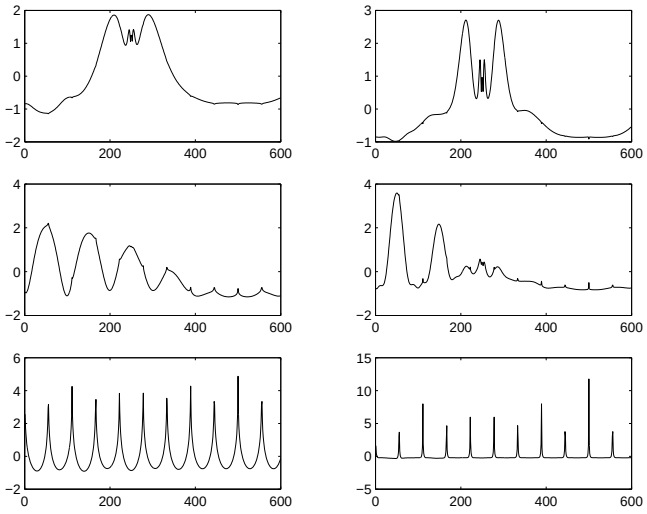

Fig. 5. The jointly dependent sources found using Method 2.

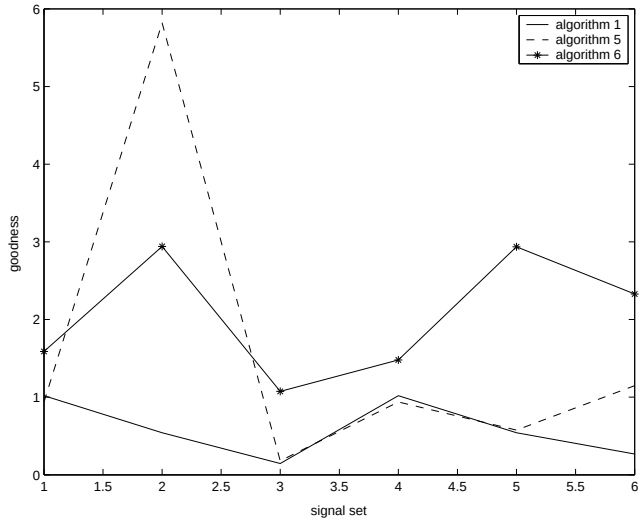

Fig. 6. Performances (goodness-of-fit measures) of the singular value decomposition (algorithm 1), Method 1 (algorithm 6), and Method 2 (algorithm 5) for the artificially generated data sets $\mathbf{x}(t)$ and $\mathbf{y}(t)$ of 6 sources. The nonlinearity used was (38), and the performance was estimated using the moment generating function.

position (that is, the index of the letter in question starting from the beginning of the text) in the books corresponding to one component of a $4-$dimensional vector. There were 288048 such vectors $\mathbf{x}(t)$ $(t = 1, 2, \ldots, 288048)$. The encrypted corresponding vectors $\mathbf{y}(t)$ $(t = 1, 2, \ldots, 288048)$ were generating by applying a 128-bit AES encoding [25] separately to each ASCII letter appended by zeros, so that each letter contained 128 bits. The encrypted 128 bit long numbers were transformed to floating point numbers having 96 bits, which we further approximated by a 32-bit floating point number in our MATLAB experiments. The source signals were preprocessed so that their mean was zero and variance unity.

We tried to estimate the source vectors $\mathbf{s}$ and $\mathbf{t}$ as well

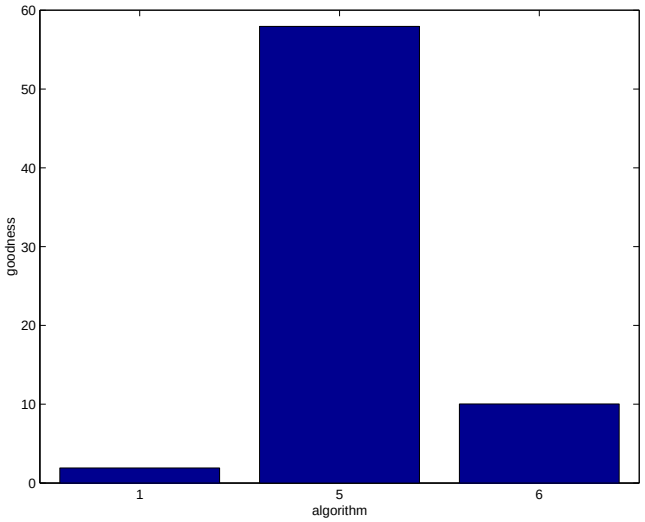

Fig. 7. Average performances of the SVD (algorithm 1), Method 2 (algorithm 5), and Method 1 (algorithm 6) for the nonlinearity (38), estimated using the moment generating function.

as the mixing matrices $\mathbf{A}$ and $\mathbf{B}$ in the model (7) using the methods developed in the previous section. That is, we tried to determine these quantities so that the joint information between the corresponding components of $\mathbf{s}$ and $\mathbf{t}$ is maximized. After this, we computed the connectivity matrix $\mathbf{M}$, defined by

$$\mathbf{t} = \mathbf{M}\mathbf{s} = \mathbf{A}^{-1}\mathbf{B}\mathbf{s} \tag{39}$$

assuming that the mixing matrix $\mathbf{A}$ is square and of full rank and hence invertible.

Encryption aims at blurring or mixing the information contents of a message as much as possible, so that it cannot be identified any more from the encrypted version [25]. Thus the goal of encryption is a kind of opposite to what ICA and BSS methods aim at. It is realistic to expect that the elements of the connectivity matrix $\mathbf{M}$ have higher absolute values when the encrypted message is strongly related to the original text.

We tried several algorithms for this problem. A general conclusion on these experiments is that the performance of the suggested algorithms for estimating jointly dependent components gradually improves. Roughly speaking, they start to perform appropriately when the number of the elements in the vectors $\mathbf{x}(t)$ and $\mathbf{y}(t)$ nears $200,000$. As an example, consider Method 1. It could connect correctly two components of the vectors $\mathbf{x}(t)$ and $\mathbf{y}(t)$ when the number of sample vectors was $t = 180000$, and all four components for $t = 262440$ and $t = 288048$. Some other algorithms performed slightly better, being able to find out all the four dependent components already when $t = 180000$. This holds especially for Method 3, which was on an average the best performing of the methods we tested in this problem.

For identifying the connected components, we used the following heuristics. Consider the absolute values $|m_{ij}|$ of the elements m_{ij} of the connectivity matrix $\mathbf{M}$. Find the

element having the largest absolute value, and mark the corresponding components having the indices $i = i_{max}$ and $j = j_{max}$ connected. Continue the procedure by finding the element of the matrix $\mathbf{M}$ having the next largest absolute value and different indices $i \neq i_{max}$ and $j \neq j_{max}$, and connect the corresponding conponents. The procedure is continued until all the components of the vectors $\mathbf{s}$ and $\mathbf{t}$ have been connected. Of course, connecting takes place so that only one element on each row and column of the matrix $\mathbf{M}$ is selected. That is, each of the components of the source vector $\mathbf{s}$ is connected to one of the component of the source vector $\mathbf{t}$.

When the number t of data vectors increased, not only the found connected components gradually became the correct ones. Also the absolute values of correct elements in the matrix $\mathbf{M}$ increased, and large erroneous values decreased. The methods using ICA for preprocessing were quite slow, because ICA was not usually able to find an independent group of source signals.

The final value of the matrix $\mathbf{M}$ for $t = 288048$ data vectors is shown in (40) for the best performing Method 3. For clarity, we have omitted the common multiplying factor 10^{36} from the elements of $\mathbf{M}$. From the results (40), one can without doubt deduce the correct jointly dependent corresponding source pairs. The corresponding largest elements of the matrix $\mathbf{M}$ on its each row and column have been boldfaced in (40).

$$\mathbf{M} = \begin{bmatrix} 0.138 & -0.225 & 0.939 & \mathbf{2.889} \\ -1.446 & 1.329 & \mathbf{2.269} & 1.039 \\ 0.330 & \mathbf{2.797} & -1.212 & -0.050 \\ \mathbf{2.719} & 0.428 & 1.410 & 0.514 \end{bmatrix} \quad (40)$$

VI. Concluding remarks

In this paper, we have presented first result on some novel methods for finding mutually corresponding dependent components from two different but related data sets. Our methods generalize cross-correlation analysis based on singular value decomposition to take into account higher-order statistics in a similar manner as in ICA. The data model is rather simple, and could be generalized in several ways. A natural extension would be to allow a more flexible model than pairs of dependent components independent of other such pairs, see for example [15], [16], [19]. Experimental results demonstrating the usefulness of proposed methods have been presented both for artificially generated and realistic cryptography data.

Acknowledgements

This research has been funded by the Academy of Finland under its Center of Excellence Programme.

References

[1] S. Achard, D. Pham, and C. Jutten. Quadratic dependence measure for nonlinear blind sources separation. In *Proc. of the 4th Int. Symp. on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 263–268, Nara, Japan, April 2003.

[2] S. Akaho, Y. Kiuchi, and S. Umeyama. MICA: multidimensional independent component analysis. In *Proc. of the 1999 Int. Joint Conf. on Neural Networks (IJCNN'99)*, pages 927–932, Washington, DC, USA, July 1999. IEEE Press.

[3] F. Bach and M. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, July 2002.

[4] A. Cichocki and S.-I. Amari. *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. Wiley, New York, 2002.

[5] K. Diamantaras and S.-Y. Kung. *Principal Component Neural Networks: Theory and Applications*. Wiley, New York, 1996.

[6] J. Eriksson, A. Kankainen, and V. Koivunen. Novel characteristic function based criteria for ICA. In *Proc. of the 3rd Int. Conf. on Independent Component Analysis and Blind Signal Separation (ICA2001)*, pages 108–113, San Diego, California, USA, December 2001.

[7] J. Eriksson and V. Koivunen. Characteristic-function-based independent component analysis. *Signal Processing*, 83:2195–2208, 2003.

[8] C. Fyfe and P. Lai. ICA using kernel canonical correlation analysis. In *Proc. of the 2nd Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, pages 279–284, Helsinki, Finland, June 2000.

[9] X. Giannakopoulos, J. Karhunen, and E. Oja. Experimental comparison of neural algorithms for independent component analysis and blind separation. *Int. J. of Neural Systems*, 9(2):651–656, 1999.

[10] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *J. of Machine Learning Research*, 6:2075–2129, December 2005.

[11] Project Gutenberg. Url: http://www.promo.net/pg/index.html.

[12] M. Hayes. *Statistical Digital Signal Processing and Modelling*. Wiley, New York, 1996.

[13] S. Haykin. *Modern Filters*. Macmillan, New York, 1989.

[14] S. Haykin. *Neural Networks - A Comprehensive Foundation*. Prentice Hall, 2nd edition, 1998.

[15] A. Hyvärinen and P. Hoyer. Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, 2000.

[16] A. Hyvärinen, P. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13:1527–1558, 2001.

[17] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, New York, 2001.

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

[19] J. Karhunen, F. Meinecke, H. Valpola, and A. Ziehe. Final technical report on BSS models and methods for non-independent BSS. Technical report, European Joint Project BLISS, Deliverable D21, 2003. URL: http://www.lis.inpg.fr/pages_perso/bliss/.

[20] J. Karhunen and E. Oja. New methods for stochastic approximation of truncated Karhunen-Loeve expansions. In *Proc. of the 6th Int. Conf. on Pattern Recognition*, pages 550–553, Munich, Germany, October 1982.

[21] J. Koetsier, D. MacDonald, D. Charles, and C. Fyfe. Exploratory correlation analysis. In *Proc. of the 10th European Symp. on Artificial Neural Networks (ESANN2002)*, pages 483–488, Bruges, Belgium, April 2002.

[22] P. Lai and C. Fyfe. A neural implementation of canonical correlation analysis. *Neural Networks*, 12:1391–1397, 1999.

[23] K. Mardia, J. Kent, and J. Bibby. *Multivariate Analysis*. Academic Press, London, 1979.

[24] E. Oja. *Subspace Methods for Pattern Recognition*. Research Studies Press, Letchworth, Hertfordshire, England, 1983.

[25] D. Stinson. *Cryptography: Theory and Practice*. Chapman and Hall, CRC Press, 2002.

[26] D. Tjostheim. Measures of dependence and tests of independence. *Statistics*, 28:249–284, 1996.

[27] K. Waheed and F. Salam. A data-derived quadratic independence measure for adaptive blind source recovery in practical applications. In *Proc. of the 45th IEEE Int. Midwest Symp. on Circuits and Systems*, volume 3, pages 473–476, Tulsa, Oklahoma, USA, August 2002.