On Independent Component Analysis for Multimedia Signals

Lars Kai Hansen, Jan Larsen and Thomas Kolenda
Department of Mathematical Modelling, Building 321
Technical University of Denmark
DK-2800 Lyngby, Denmark
lkhansen,jl,thko@imm.dtu.dk
URL: http://eivind.imm.dtu.dk

October 21, 1999

Abstract

We discuss the independent component problem within a context of multimedia applications. The literature offers several independent component analysis schemes which can be applied in this context, and each have its own trade-off between flexibility, complexity and computational effort. The specific applications investigated in this chapter comprise modeling of speech/sound, images, and text data.

1 Background

Blind reconstruction of statistically independent source signals from linear mixtures is relevant to many signal processing contexts [1, 6, 8, 9, 22, 24, 36]. With reference to Principal Component Analysis the problem is often referred to as Independent Component Analysis (ICA)[1].

The source separation problem can be formulated as a likelihood formulation, see e.g., [7, 32, 35, 37]. The likelihood formulation is attractive for several reasons. First, it allows a principled discussion of the inevitable priors implicit in any separation scheme. The prior distribution of the source signals can take many forms and *factorizes* in the source index expressing the fact that we look for *independent* sources. Secondly, the likelihood approach allows for direct adaptation of the plethora of powerful schemes for parameter optimization, regularization, and evaluation of supervised learning algorithms. Finally, for the case of linear mixtures

[1]There are a number of very useful ICA Web pages providing links to theoretical analysis, implementations and applications. Follow links from the page http://eivind.imm.dtu.dk/staff/lkhansen/ica.html

without noise, the likelihood approach is equivalent to another popular approach based on information maximization [1, 6, 27].

The source separation problem can be analyzed under the assumption that the sources are either time-independent, or possessing a more general time dependence structure. The separation problem for *autocorrelated* sequences was studied by Molgedey and Schuster [33]. They proposed a source separation scheme based on assumed non-vanishing temporal autocorrelation functions of the independent source sequences evaluated at a specific *time lag*. Their analysis was developed for sources mixed by square, non-singular matrices. Attias and Schreiner derived a likelihood based algorithm for separation of correlated sequences with a frequency domain implementation [2, 3, 4]. The approach of Molgedey and Schuster is particularly interesting as regards computational complexity because, as it forms a non-iterative, constructive solution.

Belouchrani and Cardoso presented a general likelihood approach allowing for *additive noise* and for non-square mixing matrices. They applied the method to separation of sources taking discrete values [7] estimating the mixing matrix using an Estimate-Maximize (EM) approach with both a deterministic and a stochastic formulation. Moulines *et al.* generalized the EM approach to separation of autocorrelated sequences in presence of noise, and they explored a family of flexible source priors based on Gaussian Mixtures [34]. The difficult problem of noisy, overcomplete source models[2] is recently analyzed by Lewicki and Sejnowski within the likelihood framework [28, 31].

In this chapter we study the likelihood approach and entertain two different approaches to the problem: a modified version of the Molgedey-Schuster scheme [15], based on time-correlations, and a novel iterative scheme generalizing the mixing problem to separation of noisy mixtures of time-independent white sources [16]. The Molgedey-Schuster scheme is extended to the undercomplete[3] case, and further inherent erroneous complex number results are alleviated. In the noisy mixture problem we find a maximum posterior estimate for the sources which interestingly turns out to be non-linear in the observed signal. The specific model investigated here is a special case of the general framework proposed by Belouchrani and Cardoso [7], however, we formulate the parameter estimation problem in terms of the Boltzmann learning rule, which allows for a particular transparent derivation of the mixing matrix estimate.

The methods are applied within several multimedia applications, separation of sound, image sequences, and text.

2 Principal and Independent Component Analysis

Principal Component Analysis (PCA) is a very popular tool for analysis of correlated data, like temporal correlated image databases. By PCA the image database

[2]More sources than acquired mixture signals.

[3]More acquired mixture signals than sources.

is decomposed in terms of "eigenimages" that often lend themselves to direct interpretation. A most striking example is face recognition where so-called "eigenfaces" are used as orthogonal preprocessing projection directions for pattern recognition. The Principal Components (the sequence of projections of the image data onto the eigenimages) are also uncorrelated, hence, perhaps the simplest example of independent components [9]. The basic tool for PCA is Singular Value Decomposition (SVD).

Define the observed $M \times N$ signal matrix, representing a multi-channel signal, by

$$\boldsymbol{X} = \{X_{m,n}\} = \{x_m(n)\} = [\boldsymbol{x}(1), \boldsymbol{x}(2), \cdots, \boldsymbol{x}(N)] \tag{1}$$

where M is the number measurements and N is the number of samples. $x_m(n)$, $n = 1, 2, \cdots, N$ is the m'th signal, and $\boldsymbol{x}(n) = [x_1(n), x_2(n), \cdots, x_M(n)]^\top$. In the case of image sequences, M is the number of pixels.

For fixed choice of $P \leq M$, SVD of $\boldsymbol{X}$ reads[4]

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top = \sum_{i=1}^{P} \boldsymbol{u}_i D_{i,i} \boldsymbol{v}_i^\top, \quad X_{m,n} = \sum_{i=1}^{P} U_{m,i} D_{i,i} V_{n,i} \tag{2}$$

where the $M \times P$ matrix $\boldsymbol{U} = \{U_{m,i}\} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_P]$ and the $N \times P$ matrix $\boldsymbol{V} = \{V_{n,i}\} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_P]$ represent the orthonormal basis vectors (i.e., eigenvectors of the symmetric matrices $\boldsymbol{X}\boldsymbol{X}^\top$ and $\boldsymbol{X}^\top\boldsymbol{X}$, respectively). $\boldsymbol{D} = \{D_{i,i}\}$ is a $P \times P$ diagonal matrix of singular values. In terms of independent sources, SVD can identify a set of *uncorrelated* time sequences, the Principal Components: $D_{i,i}\boldsymbol{v}_i$, enumerated by the source index $i = 1, 2, \ldots, P$. That is, we can write the observed signal as a weighted sum of fixed eigenvectors (eigenimages) $\boldsymbol{u}_i$.

However, considering the likelihood for the time correlated source density, we are often interested in a slightly more general separation of image sources that are *independent* in time, but not necessarily orthogonal in space, i.e., we would like to be able to perform a more general decomposition of the signal matrix,

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}, \quad X_{m,n} = \sum_{i=1}^{P} A_{m,i} S_{i,n} \tag{3}$$

where $\boldsymbol{A}$ is a general mixing matrix of dimension $M \times P$ and $\boldsymbol{S}$ is a source data matrix with dimension $P \times N$ consisting of $P \leq M$ independent sources. Finding $\boldsymbol{A}, \boldsymbol{S}$ is often referred to as Independent Component Analysis (ICA), see e.g., [6], [9].

[4]Usual SVD expresses $\boldsymbol{X} = \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{D}}\widetilde{\boldsymbol{V}}^\top$ where $\widetilde{\boldsymbol{U}}$ is $M \times M$, $\widetilde{\boldsymbol{D}}$ is $M \times N$, and $\widetilde{\boldsymbol{V}}$ is $N \times N$. $\boldsymbol{U}$ is the first P columns of $\widetilde{\boldsymbol{U}}$, $\boldsymbol{D}$ is the $P \times P$ upper-left submatrix of $\widetilde{\boldsymbol{D}}$, and $\boldsymbol{V}$ is the first P columns of $\widetilde{\boldsymbol{V}}$.

3 Likelihood Framework for Independent Component Analysis

Reconstruction of statistically independent components/sources from linear mixtures is relevant to many information processing contexts, see e.g., [27] for an introduction and a recent review. We will derive a solution to the source separation based on the likelihood formulation, see e.g., [7, 32, 37]. An additional benefit from working in the likelihood framework is that it is possible to discuss the *generalizability* of the ICA representation, in particular we use the generalization error as a tool for optimizing the complexity of the representation, see also [14, 17].

The noisy mixing model takes the form,

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{\mathcal{E}} \tag{4}$$

where $\boldsymbol{\mathcal{E}}$ is the $M \times N$ noise signal matrix. The noise is supposed to obey a specific zero mean, parameterized stationary probability distribution. The source signals are assumed to be stationary and mutually independent, i.e., $p(s_i(k)s_j(n)) = p(s_i(k))p(s_j(n))$, $\forall\, i,j \in [1;M]$, $\forall\, n,k \in [1;N]$. The properties of the source signals are introduced by a parameterized prior probability density $p(\boldsymbol{S}|\boldsymbol{\psi})$, where $\boldsymbol{\psi}$ are the parameter vector. The likelihood of the parameters of the noise distribution, the parameters of the source distribution and of the mixing matrix is given by,

$$L(\boldsymbol{A},\boldsymbol{\theta},\boldsymbol{\psi}) = p(\boldsymbol{X}|\boldsymbol{A},\boldsymbol{\theta},\boldsymbol{\psi}) = \int p(\boldsymbol{X}-\boldsymbol{A}\boldsymbol{S}|\boldsymbol{\theta})p(\boldsymbol{S}|\boldsymbol{\psi})d\boldsymbol{S} \tag{5}$$

where $p(\boldsymbol{X}-\boldsymbol{A}\boldsymbol{S}|\boldsymbol{\theta}) = p(\boldsymbol{\mathcal{E}}|\boldsymbol{\theta})$ is the noise distribution parameterized by the vector $\boldsymbol{\theta}$. We will assume that the noise can modeled by i.i.d. Gaussian sequences with a common variance $\theta = \sigma^2$,

$$p(\boldsymbol{\mathcal{E}}|\sigma^2) = \frac{1}{(2\pi\sigma^2)^{MN/2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{m=1}^{M}\sum_{n=1}^{N}\varepsilon_m^2(n)\right). \tag{6}$$

We will consider two different assumptions about the independent source distributions leading to different algorithms.

For the *time-independent white source* problem the parameter free source distribution of [32] is deployed,

$$p(\boldsymbol{S}) = \prod_{i=1}^{P} p(\boldsymbol{s}_i) = \frac{1}{\pi^{NP}} \exp\left(-\sum_{n=1}^{N}\sum_{i=1}^{P}\log\cosh s_i(n)\right). \tag{7}$$

where $\boldsymbol{S}^\top = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_P\}$ and $\boldsymbol{s}_i = [s_i(1), s_i(2), \cdots s_i(N)]^\top$. In the *time-correlated* case, it is assumed that the sources are stationary, independent, possess time-autocorrelation, have zero mean, and are Gaussian distributed[5],

$$p(\boldsymbol{S}|\boldsymbol{\psi}) = \prod_{i=1}^{P} p(\boldsymbol{s}_i|\boldsymbol{\psi}_i) = \prod_{i=1}^{P} \frac{1}{(2\pi)^{N/2}\sqrt{\det(\boldsymbol{\Gamma}_{s_i})}} \exp\left(-\frac{1}{2}\boldsymbol{s}_i^\top \boldsymbol{\Gamma}_{s_i}^{-1}\boldsymbol{s}_i\right) \tag{8}$$

[5]By assuming stationarity, we implicitly neglect transient behavior due to initial conditions.

where $\boldsymbol{\psi} = [\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_P]$ and $\boldsymbol{\Gamma}_{s_i} = E[\boldsymbol{s}_i \boldsymbol{s}_i^\top] = \text{Toeplitz}([\gamma_{s_i}(0), \cdots, \gamma_{s_i}(N-1)])$ [6] is the $N \times N$ Toeplitz autocorrelation matrix consisting of autocorrelation function values, $\gamma_{s_i}(m) = E[s_i(n)s_i(n+m)]$, $m = 0, 1, \cdots, N-1$. The autocorrealtion matrix $\boldsymbol{\Gamma}_{s_i}$ is supposed to be parameterized by $\boldsymbol{\psi}_i$.

3.1 Generalization and the Bias-Variance Dilemma

The parameters of our blind separation model are estimated from a finite random sample, and therefore they also are random variables which inherit noise from the data set they were trained on. Within the likelihood formulation the generalization error of a specific set of parameters is given by the average negative log-likelihood[7]

$$\begin{aligned} G(\boldsymbol{A}, \boldsymbol{\theta}, \boldsymbol{\psi}) &= \int -\log L(\boldsymbol{A}, \boldsymbol{\theta}, \boldsymbol{\psi}) \cdot p_*(\boldsymbol{X})\, d\boldsymbol{X} \\ &= \int [-\log \int p(\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}|\boldsymbol{\theta}) p(\boldsymbol{S}|\boldsymbol{\psi})\, d\boldsymbol{S}] \cdot p_*(\boldsymbol{X})\, d\boldsymbol{X}. \end{aligned} \tag{9}$$

where $p_*(\boldsymbol{X})$ is the true distribution of data. The generalization error is a principled tool for model selection. In the context of blind separation the optimal number of sources retained in the model is of crucial interest. We face a typical bias-variance dilemma [13]. If too few components are used, a structured part of the signal will be lumped with the noise, hence leading to a high generalization error because of "lack of fit". On the other hand, if too many sources are used we expect "overfit" since the model will use the additional degrees of freedom to fit non-generic details in the training data. The generalization error in Eq. (9) can be estimated using a test set of data *independent* of the training set[8].

3.2 Noisy Mixing of White Sources

The specific model investigated here is a special case of the general framework proposed by Belouchrani and Cardoso [7], however, we formulate the parameter estimation problem in terms of the Boltzmann learning rule, which allows for a particular transparent derivation of the mixing matrix estimate.

Let us first address the problem of estimating the sources if the mixing parameters are known, i.e., for given $\boldsymbol{A}$ and σ^2. Note that MacKay [32] showed that the gradient descent scheme for the likelihood problem, for vanishing noise variance, is

[6] Toeplitz$(\cdot)$ transforms a vector into a Toeplitz matrix.

[7] Note the close connection between generalizaton error and the Kullback-Leibler Information (KL), as

$$\begin{aligned} \text{KL}(\, p_*(\boldsymbol{X}) : p(\boldsymbol{X}|\boldsymbol{A}, \boldsymbol{\theta}, \boldsymbol{\psi})\,) &= \int \log \frac{p_*(\boldsymbol{X})}{p(\boldsymbol{X}|\boldsymbol{A}, \boldsymbol{\theta}, \boldsymbol{\psi})} p_*(\boldsymbol{X})\, d\boldsymbol{X} \\ &= G(\boldsymbol{A}, \boldsymbol{\theta}, \boldsymbol{\psi}) + \int \log(p_*(\boldsymbol{X})) p_*(\boldsymbol{X})\, d\boldsymbol{X} \end{aligned}$$

[8] That is, we evaluate Eq. (9) on the test data by using $p_*(\boldsymbol{X}) = \delta(\boldsymbol{X} - \boldsymbol{X}_{\text{test}})$ where δ is the Dirac delta function and $\boldsymbol{X}_{\text{test}}$ are the test data.

equivalent to the Bell-Sejnowski rule [6]. Here we want to consider the more general noisy case. We use Bayes formula $p(\boldsymbol{S}|\boldsymbol{X}) \propto p(\boldsymbol{X}|\boldsymbol{S})p(\boldsymbol{S})$ to obtain the posterior distribution of the sources

$$\begin{aligned} p(\boldsymbol{S}|\boldsymbol{X},\boldsymbol{A},\sigma^2) &\propto \exp\left(-\frac{1}{2\sigma^2}\sum_{m=1}^{M}\sum_{n=1}^{N}\varepsilon_m^2(n) - \sum_{i=1}^{P}\sum_{n=1}^{N}\log\cosh s_i(n)\right) \\ &= \exp\left(-\frac{1}{2\sigma^2}\sum_{m=1}^{M}\sum_{n=1}^{N}(\boldsymbol{X}-\boldsymbol{A}\boldsymbol{S})_{m,n}^2 - \sum_{i=1}^{P}\sum_{n=1}^{N}\log\cosh S_{i,n}\right) \end{aligned} \quad (10)$$

The *maximum a posteriori* (MAP) source estimate is found by maximizing this expression w.r.t. $\boldsymbol{S}$ [9], leading to the following non-linear equation to solve iteratively for the MAP estimate $\widehat{\boldsymbol{S}}$,

$$-\boldsymbol{A}^\top\boldsymbol{A}\widehat{\boldsymbol{S}} + \boldsymbol{A}^\top\boldsymbol{X} - \sigma^2\tanh\widehat{\boldsymbol{S}} = \boldsymbol{0}. \quad (11)$$

There are two problems facing in Eq. (11). First, the equation is non-linear – though only weakly non-linear for low noise levels[10]. Second, $\boldsymbol{A}^\top\boldsymbol{A}$, may be ill-conditioned or even singular. A useful rewriting that takes care of potential ill-conditioning of the system matrix leads to the iterative scheme,

$$\widehat{\boldsymbol{S}}^{(j+1)} = \left(\boldsymbol{A}^\top\boldsymbol{A} + \sigma^2\boldsymbol{I}\right)^{-1}\left(\boldsymbol{A}^\top\boldsymbol{X} + \sigma^2\left(\widehat{\boldsymbol{S}}^{(j)} - \tanh(\widehat{\boldsymbol{S}}^{(j+1)})\right)\right) \quad (12)$$

where j denotes the iteration number and $\boldsymbol{I}$ is the identity matrix. This form suggests an approximate solution for low noise levels

$$\begin{aligned} \widehat{\boldsymbol{S}}^{(1)} &= \boldsymbol{S}^{(0)} + \sigma^2\boldsymbol{H}^{-1}\left(\boldsymbol{S}^{(0)} - \tanh\boldsymbol{S}^{(0)}\right), \\ \boldsymbol{S}^{(0)} &= \boldsymbol{H}^{-1}\boldsymbol{A}^\top\boldsymbol{X}, \quad \boldsymbol{H} = \boldsymbol{A}^\top\boldsymbol{A} + \sigma^2\boldsymbol{I}, \end{aligned} \quad (13)$$

exposing the fact that the presence of additive noise turns the otherwise linear separation problem in to a non-linear one. A non-linear source estimate is also found in Lewicki and Sejnowski's analysis of the overcomplete problem [31].

Since the likelihood is of the hidden-Gibbs form we can use a generalized Boltzmann learning rule to find the gradients of the likelihood of the parameters $\boldsymbol{A}$, σ^2. These averages can be estimated in a mean field approximation [16, 38] leading to recursive rules for $\boldsymbol{A}$ and σ^2,

$$\widehat{\boldsymbol{A}} = \boldsymbol{X}\widehat{\boldsymbol{S}}^\top\left(\widehat{\boldsymbol{S}}\widehat{\boldsymbol{S}}^\top + \beta\boldsymbol{I}\right)^{-1}, \quad (14)$$

$$\widehat{\sigma}^2 = \frac{1}{MN}\mathrm{Tr}(\boldsymbol{X}-\widehat{\boldsymbol{A}}\widehat{\boldsymbol{S}})^\top(\boldsymbol{X}-\widehat{\boldsymbol{A}}\widehat{\boldsymbol{S}}). \quad (15)$$

[9] Note in case of zero noise, the posterior expression leads to the expression given in [32], and the solution is obtained by the Bell-Sejnowski algorithm [6].

[10] This expression is the gradient of the exponent of the posterior distribution. A globally convergent iterative solution can be assured if solving by gradient ascent $\nabla\boldsymbol{S} = \eta\cdot\partial\log p(\boldsymbol{S}|\boldsymbol{X},\boldsymbol{A},\sigma^2)/\partial\boldsymbol{S}$, with a sufficiently small step-size, η. Here, however, we aim for a fast approximate solution for $\boldsymbol{S}$.

β is a regularization constant representing the lumped effect of neglected fluctuations in the mean field approach. β is estimated by

$$\beta = \hat{\sigma}^2 \left(1 - \frac{1}{PN} \sum_{i=1}^{P} \sum_{n=1}^{N} \tanh^2 S_{i,n} \right), \tag{16}$$

see [16].

Fluctuation corrections (hence the magnitude of β) can derived in the low noise limit, based on a Gaussian approximation of the likelihood [16].

The overall algorithm then consists in iterating Eq. (13), (14)–(16), (12), (14)–(16), etc. Convergence of the algorithm is discussed in [16].

3.3 Separation based on Time-Correlation

Molgedey and Schuster [33] have proposed a simple non-iterative source separation scheme based on assumed non-vanishing (time) autocorrelation functions of the independent sources which can be Gaussian distributed[11]. Their idea was developed for sources mixed by square, non-singular, $\boldsymbol{A}$ matrices. Here we generalize their approach in three ways:

- Handling the undercomplete case of more mixture signals than sources, i.e., $P \leq M$. In particular, the algorithm is well-suited for cases where $P \ll M$.
- Alleviating inherent erroneous complex valued results.
- Allowing for simultaneously use of more crosscorrelation matrix function values maintaining the simple non-iterative solution.

Define the $M \times M$ crosscorrelation function matrix for the mixture signals

$$\boldsymbol{C}_{\boldsymbol{x}}(\tau) = E\{\boldsymbol{x}(n)\boldsymbol{x}^\top(n+\tau)\} = \{\, i,j \in [1;M] :\, x_i(n)x_j(n+\tau) \,\} \tag{17}$$

where $\tau = 0, \pm 1, \pm 2, \cdots$ is a time-lag and $E\{\cdot\}$ is the expectation operator. Note for $\tau = 0$ we get the usual crosscorrelation matrix, $\boldsymbol{C}_{\boldsymbol{x}}(0) = E\{\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\}$ which is positive semi-defininte. Assume the noise-free model Eq. (3), $\boldsymbol{x}(n) = \boldsymbol{A}\boldsymbol{s}(n)$, where $\boldsymbol{s}(n) = [s_1(n), \cdots, s_P(n)]^\top$, $\boldsymbol{x}(n) = [x_1(n), \cdots, x_M(n)]^\top$ and further that the $M \times P$ mixing matrix has $\mathrm{rank}(\boldsymbol{A}) = P \leq M$. Since $\boldsymbol{C}_{\boldsymbol{x}}(0) = \boldsymbol{A}\boldsymbol{C}_{\boldsymbol{s}}(0)\boldsymbol{A}^\top$ where $\boldsymbol{C}_{\boldsymbol{s}}(0)$ is the $P \times P$ crosscrorrelation matrix for the source signals, and $\mathrm{rank}(\boldsymbol{A}) = P$ then $\mathrm{rank}(\boldsymbol{C}_{\boldsymbol{x}}(0)) = P$. An eigenvalue decomposition of $\boldsymbol{C}_{\boldsymbol{x}}(0)$ reads

$$\boldsymbol{C}_{\boldsymbol{x}}(0) = \boldsymbol{Q}\boldsymbol{L}\boldsymbol{Q}^\top \tag{18}$$

where $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_M]$ is the orthogonal matrix ($\boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{I}$) of eigenvectors $\boldsymbol{q}_i$ and $\boldsymbol{L} = \mathrm{diag}(l_1, \cdots, l_M)$ is the diagonal matrix of eigenvalues $l_1 \leq l_2 \leq \cdots \leq l_P \leq 0$ and $l_{P+1} = l_{P+2} = \cdots = l_M = 0$. Consider projection onto the P-dimensional full rank subspace,

$$\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{Q}}^\top \boldsymbol{x} \tag{19}$$

[11] At most one source is allowed to be white.

where $\widetilde{\boldsymbol{Q}} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_P]$ is the $M \times P$ projection matrix and $\widetilde{\boldsymbol{x}}$ is the $P \times 1$ projected mixture signal vector. Now define *quotient matrix*

$$\boldsymbol{K} = \boldsymbol{C}_{\widetilde{\boldsymbol{x}}}(\tau)\boldsymbol{C}_{\widetilde{\boldsymbol{x}}}^{-1}(0). \tag{20}$$

Since $\boldsymbol{C}_{\widetilde{\boldsymbol{x}}}(\tau) = \widetilde{\boldsymbol{Q}}^\top \boldsymbol{A}\boldsymbol{C}_{\boldsymbol{s}}(\tau)\boldsymbol{A}^\top\widetilde{\boldsymbol{Q}}$ the quotient matrix can be expressed as[12]

$$\boldsymbol{K} = (\widetilde{\boldsymbol{Q}}^\top \boldsymbol{A})\boldsymbol{C}_{\boldsymbol{s}}(\tau)\boldsymbol{C}_{\boldsymbol{s}}^{-1}(0)(\widetilde{\boldsymbol{Q}}^\top \boldsymbol{A})^{-1} \tag{21}$$

According to Appendix A the quotient matrix has the eigenvalue decomposition $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^{-1}$ where $\boldsymbol{\Lambda}$ is a diagonal matrix of real eigenvalues and $\boldsymbol{\Phi}$ are the associated real eigenvectors. Define a permutation matrix[13] $\boldsymbol{P} = [\boldsymbol{e}_{j_1}, \ldots, \boldsymbol{e}_{j_P}]$ where $\boldsymbol{e}_j = \{\delta_{ij}, i \in [1; P]\}$ are P-dimensional unit column vectors and $[j_1, j_2, \cdots, j_P]$ is a permutation of the numbers $[1; P]$. Note that $\boldsymbol{P}\boldsymbol{P} = \boldsymbol{I}$. Further, define a diagonal scaling matrix $\boldsymbol{\Xi} = \mathrm{diag}([\xi_1, \cdots, \xi_P])$ with $\xi_i \neq 0$. Comparing with Eq. (21) shows that eigenvalue decomposition of $\boldsymbol{K}$ can be used to identify the mixing matrix $\boldsymbol{A}$, as shown by:

$$\boldsymbol{K} = (\widetilde{\boldsymbol{Q}}^\top \boldsymbol{A})\boldsymbol{C}_{\boldsymbol{s}}(\tau)\boldsymbol{C}_{\boldsymbol{s}}^{-1}(0)(\widetilde{\boldsymbol{Q}}^\top \boldsymbol{A})^{-1} = \boldsymbol{\Phi}\boldsymbol{\Xi}\boldsymbol{P}\boldsymbol{P}\boldsymbol{\Xi}^{-1}\boldsymbol{\Lambda}\boldsymbol{\Xi}^{-1}\boldsymbol{P}\boldsymbol{P}\boldsymbol{\Xi}\boldsymbol{\Phi}^{-1} \tag{22}$$

where $\boldsymbol{P}$ is a permutation matrix and $\boldsymbol{\Xi}$ a diagonal scaling matrix as defined in Section 2. Consequently,

$$\begin{aligned} \widetilde{\boldsymbol{Q}}^\top \boldsymbol{A} &= \boldsymbol{\Phi}\boldsymbol{\Xi}\boldsymbol{P}, & (23)\\ \boldsymbol{C}_{\boldsymbol{s}}(\tau)\boldsymbol{C}_{\boldsymbol{s}}^{-1}(0) &= \boldsymbol{P}\boldsymbol{\Xi}^{-1}\boldsymbol{\Lambda}\boldsymbol{\Xi}^{-1}\boldsymbol{P}. & (24) \end{aligned}$$

Here we use the fact that $\boldsymbol{C}_{\boldsymbol{s}}(\tau)$ is diagonal due to independence of the sources signals.

Consider measurements of crosscorrelation function matrix for T different τ's and define the extended quotient matrix:

$$\boldsymbol{K}_{\text{ext}} = \sum_{j=1}^{T} \alpha_j \cdot \boldsymbol{C}_{\widetilde{\boldsymbol{x}}}(\tau_j)\boldsymbol{C}_{\widetilde{\boldsymbol{x}}}^{-1}(0) \tag{25}$$

where α_j are scalar weights. Then eigenvalue decomposition of $\boldsymbol{K}_{\text{ext}} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^{-1}$ leads to

$$\begin{aligned} \widetilde{\boldsymbol{Q}}^\top \boldsymbol{A} &= \boldsymbol{\Phi}\boldsymbol{\Xi}\boldsymbol{P}, & (26)\\ \sum_{j=1}^{T} \alpha_j \cdot \boldsymbol{C}_{\boldsymbol{s}}(\tau_j)\boldsymbol{C}_{\boldsymbol{s}}^{-1}(0) &= \boldsymbol{P}\boldsymbol{\Xi}^{-1}\boldsymbol{\Lambda}\boldsymbol{\Xi}^{-1}\boldsymbol{P}. & (27) \end{aligned}$$

The generalized Molgedey-Schuster algorithm for identification of mixing and source signals up to scaling and permutations is thus summarized in the following steps:

[12] Note that $\widetilde{\boldsymbol{Q}}^\top \boldsymbol{A}$ has full rank equal to P.

[13] $\boldsymbol{W}\boldsymbol{P}$ gives a permutation $\boldsymbol{W}$'s columns, whereas $\boldsymbol{P}\boldsymbol{W}$ gives a permutation of the rows.

1. Perform eigenvalue decomposition: $\boldsymbol{C}_{\boldsymbol{x}}(0) = \boldsymbol{Q}\boldsymbol{L}\boldsymbol{Q}^{\top}$.

2. Compute projected mixing signals, $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{Q}}^{\top}\boldsymbol{x}$.

3. Choose α_j and τ_j for $j = 1, 2, \cdots, T$ and compute the extended quotient matrix $\boldsymbol{K}_{\text{ext}}$

4. Perform eigenvalue decomposition: $\boldsymbol{K}_{\text{ext}} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^{-1}$

5. Up to scaling and permutations, the mixing matrix and sources are identified as:

$$\boldsymbol{A} = \tilde{\boldsymbol{Q}}\boldsymbol{\Phi} \tag{28}$$

$$\boldsymbol{S} = (\boldsymbol{A}^{\top}\boldsymbol{A})^{-1}\boldsymbol{A}^{\top}\boldsymbol{X} = \boldsymbol{\Phi}^{-1}\tilde{\boldsymbol{Q}}^{\top}\boldsymbol{X} \tag{29}$$

3.3.1 Estimation of Mixing Matrix and Source Signals

The procedure described above is based on true crosscorrelation function matrices which in practice are estimated from available data. Consider the estimate,

$$\widehat{\boldsymbol{C}}_{\boldsymbol{x}}(\tau) = \frac{1}{2N}\left(\boldsymbol{X}_{\tau}\boldsymbol{X}^{\top} + \boldsymbol{X}\boldsymbol{X}_{\tau}^{\top}\right) \tag{30}$$

where $\boldsymbol{X}_{\tau} = \{x_m(n+\tau)\}$ is the time-shifted data matrix. Here we consider a cyclic permutation by τ time steps, i.e., $\boldsymbol{X}_{\tau} = \{x_m((n+\tau)_N)\}$ where $(\cdot)_N$ denotes the argument modulo N. Eq. (30) respects the fact that the true correlationmatrix funtion $\boldsymbol{C}_{\boldsymbol{x}}(\tau)$ is symmetric.

Consider the SVD of $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\top}$ in Eq. (2) with P selected so that $\boldsymbol{D}$ consists of positive singular values only. When $\boldsymbol{X}_{\tau}$ is formed by cyclic permutation, $\boldsymbol{X}\boldsymbol{X}^{\top} = \boldsymbol{X}_{\tau}\boldsymbol{X}_{\tau}^{\top}$; hence, $\boldsymbol{X}_{\tau} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}_{\tau}^{\top}$ where $\boldsymbol{V}_{\tau}$ is the cyclic permutation of $\boldsymbol{V}$. The $P \times N$ projected mixture signal matrix is $\widetilde{\boldsymbol{X}} = \boldsymbol{U}^{\top}\boldsymbol{X} = \boldsymbol{D}\boldsymbol{V}^{\top}$ and $\widetilde{\boldsymbol{X}}_{\tau} = \boldsymbol{D}\boldsymbol{V}_{\tau}^{\top}$ as $\boldsymbol{U}$ is an estimate of $\tilde{\boldsymbol{Q}}$. The estimated quotient matrix is according to Eq. (20) given by

$$\begin{aligned}
\widehat{\boldsymbol{K}} &= \widehat{\boldsymbol{C}}_{\tilde{\boldsymbol{x}}}(\tau)\widehat{\boldsymbol{C}}_{\tilde{\boldsymbol{x}}}^{-1}(0) \\
&= \frac{1}{2}\left(\widetilde{\boldsymbol{X}}_{\tau}\widetilde{\boldsymbol{X}}^{\top} + \widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}_{\tau}^{\top}\right)\left(\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^{\top}\right)^{-1} \\
&= \frac{1}{2}\boldsymbol{D}\left(\boldsymbol{V}_{\tau}^{\top}\boldsymbol{V} + \boldsymbol{V}^{\top}\boldsymbol{V}_{\tau}\right)\boldsymbol{D}\left(\boldsymbol{D}\boldsymbol{V}^{\top}\boldsymbol{V}\boldsymbol{D}\right)^{-1} \\
&= \frac{1}{2}\boldsymbol{D}\left(\boldsymbol{V}_{\tau}^{\top}\boldsymbol{V} + \boldsymbol{V}^{\top}\boldsymbol{V}_{\tau}\right)\boldsymbol{D}^{-1}.
\end{aligned} \tag{31}$$

The generalized Molgedey-Schuster (MS) ICA algorithm can be summarized, in the following steps:

1. Perform SVD: $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\top}$ with P selected so that all singular values in $\boldsymbol{D}$ are positive. There is an option for *regularization* by discarding some of the smallest singular values causing an reduction of P.

2. Perform eigenvalue decomposition of the estimated quotient matrix[14]

$$\begin{aligned}\widehat{\boldsymbol{K}} &= \frac{1}{2}\boldsymbol{D}(\boldsymbol{V}_\tau^\top \boldsymbol{V} + \boldsymbol{V}^\top \boldsymbol{V}_\tau)\boldsymbol{D}^{-1} \\ &= \hat{\boldsymbol{\Phi}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{\Phi}}^{-1}.\end{aligned} \tag{32}$$

3. Estimate mixing matrix and sources signals:

$$\hat{\boldsymbol{A}} = \boldsymbol{U}\hat{\boldsymbol{\Phi}}, \tag{33}$$

$$\hat{\boldsymbol{S}} = \hat{\boldsymbol{\Phi}}^{-1}\boldsymbol{D}\boldsymbol{V}^\top. \tag{34}$$

4. Crosscorrelation matrix functions of the source signals are estimated as

$$\widehat{\boldsymbol{C}}_{\boldsymbol{s}}(0) = N^{-1}\hat{\boldsymbol{S}}\hat{\boldsymbol{S}}^\top = N^{-1}\cdot\hat{\boldsymbol{\Phi}}^{-1}\boldsymbol{D}^2\hat{\boldsymbol{\Phi}}^{-\top}, \tag{35}$$

$$\widehat{\boldsymbol{C}}_{\boldsymbol{s}}(\tau) = \hat{\boldsymbol{\Lambda}}\widehat{\boldsymbol{C}}_{\boldsymbol{s}}(0). \tag{36}$$

The fact that $\hat{\boldsymbol{\Phi}}$ is non-orthogonal in general implies that $\widehat{\boldsymbol{C}}_{\boldsymbol{s}}(0)$ and $\widehat{\boldsymbol{C}}_{\boldsymbol{s}}(\tau)$ are not diagonal. That is, finite sequence source signals can not be expected to be *uncorrelated*. Unlike PCA, this scheme and other ICA schemes, do not automatically produce a set of uncorrelated features.

3.4 Likelihood

The major advantage of the Molgedey-Schuster algorithm is its non-iterative nature, however, is not directly guaranteed to minimize the likelihood. Still the likelihood is still a convenient tool for understanding the nature of the modeling. Deploying one τ ($T = 1$) is consistent with parameterizing the source distribution $p(\boldsymbol{S}|\boldsymbol{\psi})$ in Eq. (8) using one parameter per source. As more τ's is deployed a more flexible parameterization of the likelihood applies.

The likelihood can be computed in a simple way using Fourier techniques, as will be shown in a forthcoming paper. This also enables computation of validation/generalization error, and consequently a principled way to select optimal τ's aiming at achieving minimum generalization error. However, the discussion is beyond the scope of this chapter.

4 Separation of Sound Signals

In this example the aim is to demonstrate how ICA is applied to separation of sound signals. This could be thought of as a special case of blind signal separation in connection with the *cocktail party problem* illustrated in Figure 1.

The present example deals with speech from 3 persons which are assumed statistically independent. The sampling frequency of the signals is 11025 Hz and they

[14] When $T > 1$ the term $(\boldsymbol{V}_\tau^\top \boldsymbol{V} + \boldsymbol{V}^\top \boldsymbol{V}_\tau)$ is replaced by $\sum_{j=1}^{T} \alpha_j (\boldsymbol{V}_{\tau_j}^\top \boldsymbol{V} + \boldsymbol{V}^\top \boldsymbol{V}_{\tau_j})$.

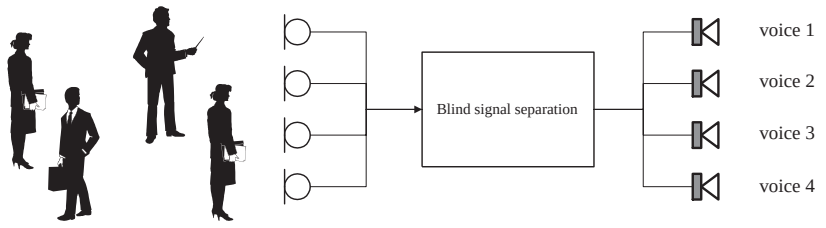

Figure 1: In the cocktail party problem speech from a group of people is recorded by a number of microphones. Without prior knowledge of the dynamics in the voices, how they are mixed, or presence of additional noise sources, the goal is to separate the voice of the individual speakers into different output channels.

consist of 50000 samples each. A linear instantaneous mixing with fixed known 3×3 mixing matrix is deployed and enables a quantitative evaluation of the ICA separation. The source and mixing signals are shown in Figure 2. In general these assumptions would not hold in real world applications due to echo, noise, delay, and different kind of nonlinear effects. In such cases more elaborate source separation is needed, as described e.g., in [2, 3, 4, 10]. In order to evaluate the results of the

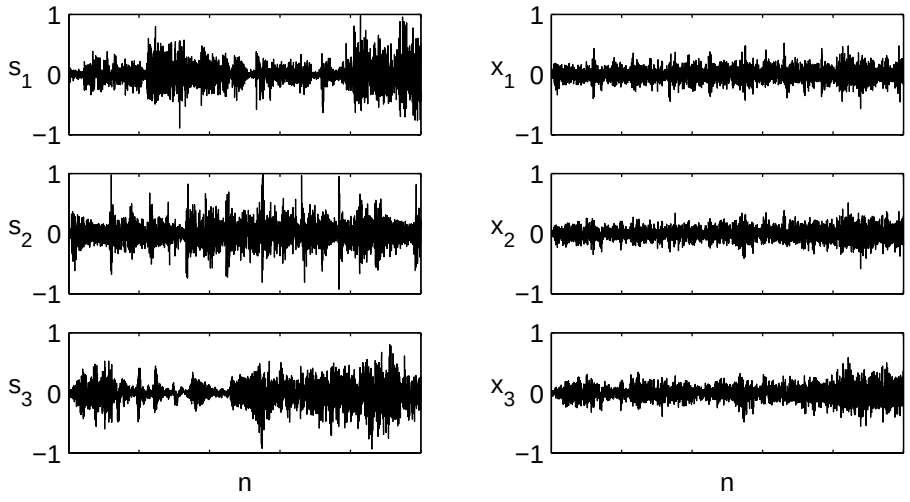

Figure 2: The orginal source sound signals $s_1(n)$, $s_2(n)$, $s_3(n)$ consist of 50000 samples and are assumed to be statistically independent. The mixture signals $x_1(n)$, $x_2(n)$, $x_3(n)$ are linear instantaneous combinations of the source signals.

separation, we consider the so-called *system matrix* defined as

$$\boldsymbol{SM} = (\widehat{\boldsymbol{A}}\widehat{\boldsymbol{C}}_{\boldsymbol{s}}(0)^{1/2})^{-1}\boldsymbol{P}\boldsymbol{A} \tag{37}$$

where $\widehat{\boldsymbol{A}}$ is the estimated mixing matrix, $\boldsymbol{P}$ is a permutation matrix, and $\widehat{\boldsymbol{C}}_{\boldsymbol{s}}(0)$ is the crosscorrelation matrix of the estimated source signals. If the separation is

$$\boldsymbol{SM} = \begin{bmatrix} 0.56 & 0.98 & 0.62 \\ 0.28 & 0.72 & 0.23 \\ 0.18 & 0.50 & 0.06 \end{bmatrix}$$

Table 1: System matrix for the PCA separation of sound signals.

successfully, the system matrix equals the identity matrix.

4.1 Sound Separation using PCA

The principal component analysis described in Section 2 is often used because it is simple and relatively fast. Moreover it offers the possibility of reducing the number of sources by ranking sources according to power (variance). The result of the PCA

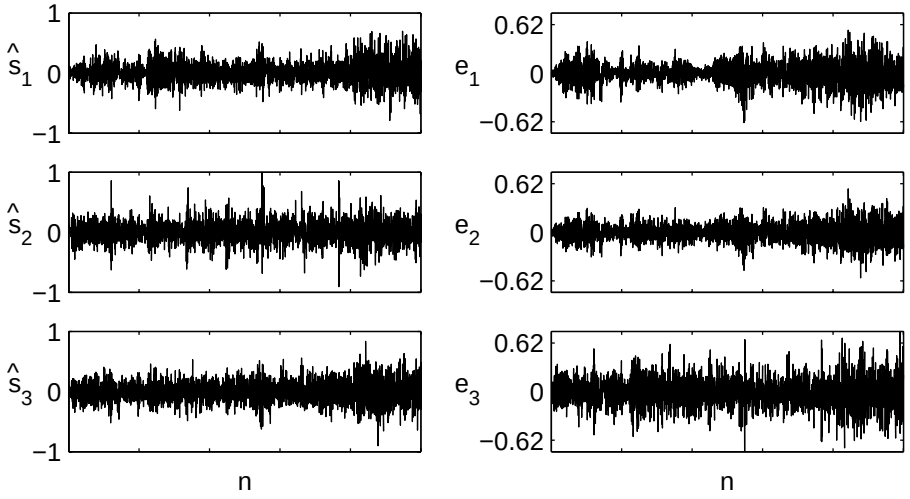

Figure 3: Separated sound source signals using PCA. Right panels show error signals, $e_i(n) = s_i(n) - \widehat{s}_i(n)$.

separation is shown in Figure 3 and the corresponding system matrix in Table 1. Obviously the result is poor when comparing estimated sources to the original sources in Figure 2. This is also confirmed by inspecting the system matrix in Table 1.

4.2 Sound Separation using Molgedey-Schuster ICA

The main advantage of the MS-ICA algorithm is that it is non-iterative, and consequently very fast. A standard $T = 1$ MS was employed, and the choice $\tau = 1$ gave best performance. In Figure 4 the estimated sound signals from the separation are shown. Comparison with original source signals in Figure 2 indicates very good separation. The system matrix in Table 2 and an additional listening test also confirms this result.

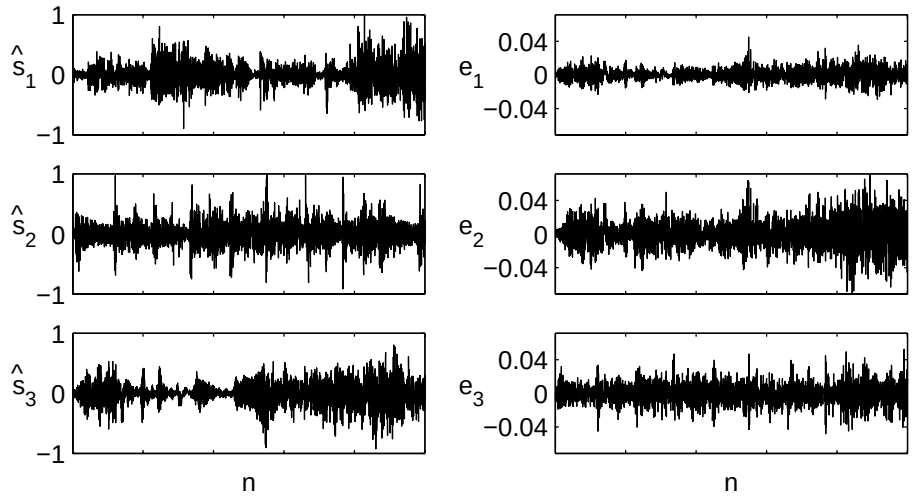

Figure 4: Separated sound source signals using Molgedey-Schuster ICA. Right panels show error signals, $e_i(n) = s_i(n) - \widehat{s}_i(n)$.

$$\boldsymbol{SM} = \begin{bmatrix} 1.00 & 0.02 & 0.03 \\ 0.02 & 1.00 & -0.01 \\ -0.03 & -0.03 & -1.00 \end{bmatrix}$$

Table 2: System matrix for the Molgedey-Schuster ICA separation of sound signals.

4.3 Sound Separation using Bell-Sejnowski ICA

The very commonly used Bell-Sejnowski ICA [6] is equivalent to maximum likelihood with assumptions like those presented in Section 3.2 in the case of zero noise. Bell-Sejnowski ICA iteratively computes an estimate of the mixing matrix by updating proportional to the natural gradient of the likelihood. The step size (gradient parameter) was initially 10^{-4} and a line search was employed using bisection. The algorithm was terminated when the negative log-likelihood was below 10^{-12}. Due to the iterative nature, the algorithm is much more time consuming than the Molgedey-Schuster algorithm.

In Figure 3 and Table 3 the results of the separation are shown. Clearly, the system matrix is closer to the identity matrix than that of Molgedey-Shuster at the expense of increased computational burden.

4.4 Comparison

Table 4 lists the norm of the system matrix deviation from the identity matrix as well as computation time.

Obviously, PCA was out-performed by both ICA algorithms due to very restricted separation capabilities. Both ICA algorithms performed very well. The

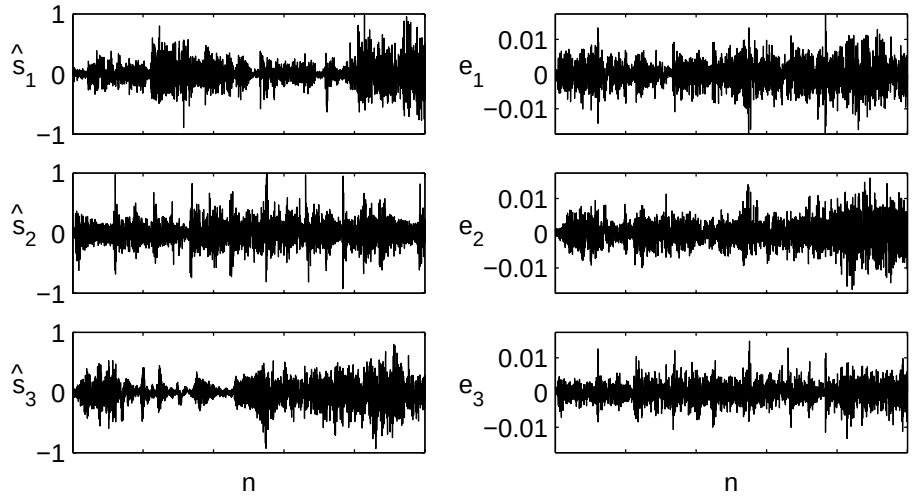

Figure 5: Separated sound source signals using Bell-Sejnowski ICA. Right panels show error signals, $e_i(n) = s_i(n) - \widehat{s}_i(n)$.

$$\boldsymbol{SM} = \begin{bmatrix} 1.00 & -0.01 & 0.01 \\ 0.00 & 1.00 & -0.01 \\ 0.01 & 0.01 & 1.00 \end{bmatrix}$$

Table 3: System matrix for the Bell-Sejnowski ICA separation of sound signals.

major difference is computation time, thus MS-ICA was more than 200 times faster than the BS-ICA. The advantage of the BS-ICA algorithm is that the system matrix can be significantly closer to unity provided sufficient computation time. A hybrid of MS-ICA and BS-ICA in which MS-ICA is used to initialize BS-ICA seems obvious.

By listening to the separated signals it was hardly impossible to tell the difference between the ICA results.

	$\lvert\boldsymbol{SM} - \boldsymbol{I}\rvert$	**Comp. time (sec.)**
PCA	1.21	0.25
MS-ICA	0.05	0.25
BS-ICA 22 iterations	0.05	56.10
BS-ICA 56 iterations	0.01	152.18

Table 4: Norm of the system matrix' deviation from the identity matrix and computation time in seconds. MS-ICA is the Molgedey-Schuster ICA, BS-ICA is Bell-Sejnowksi ICA for 22 and 56 iterations, respectively.

5 Separation of Image Mixtures

Applying ICA to images has been carried out in a number of applications ranging from face recognition to localizing activated areas in the brain, see e.g., [5, 16, 19, 20, 29, 30].

In this section we will illustrate some of the basic features using ICA in contrast to/or in combination with PCA for image segmentation. From a sequence of images, the objective is to extract a sequence images where common features has been separated into different images. In the present case ICA is based on raw images, however, in principle, the segmentation can also be done from features extracted from the images. The simple data set as shown in Figure 6 is used in this example. There are $P = 4$ original source images of $N = 9100$ (91 by 100) pixels rearranged into the $P \times N$ source matrix $\boldsymbol{S}$ so that each row represents an images. The $M \times N$ signal matrix $\boldsymbol{X}$ with $M = 6$ is generated by using the following $M \times P$ mixing matrix

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ -1 & 1 & 0 & 1 \\ 1 & 1 & -2 & 1 \\ -1 & -1 & -2 & 1 \\ 1 & -1 & 0 & 1 \\ -1 & -1 & 0 & 1 \end{bmatrix} \tag{38}$$

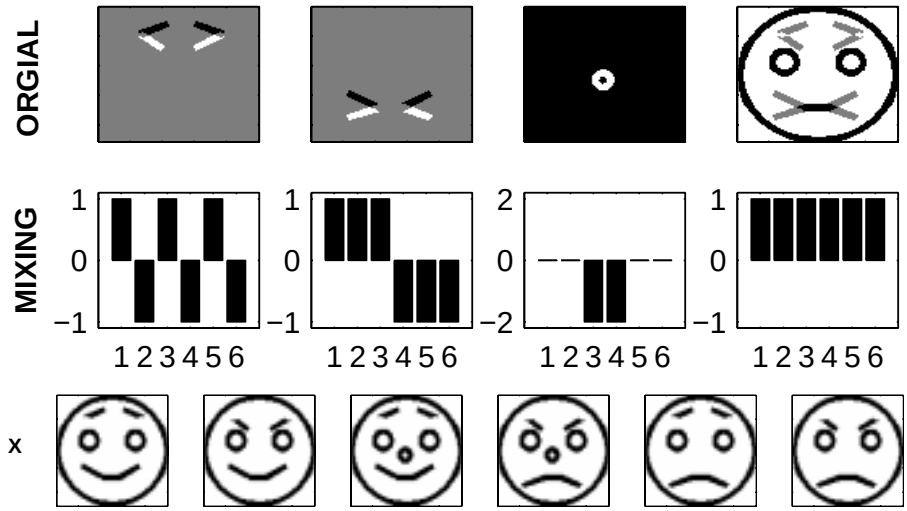

Figure 6: The artificial face data set used for image segmentation. The top row shows the $P = 4$ sources of $N = 9100$ pixels which is multiplied with the mixing $\boldsymbol{A}$ in the middle row to generate the signal matrix $\boldsymbol{X}$ with $M = 6$ components in the bottom row.

5.1 Image Segmentation using PCA

The result of applying PCA to the face data set in shown in Figure 7. The number of non-zero eigenvalues are correctly determined to 4. Notice that eyebrows and mouth position operate in pairs; when the mouth is "smiling" it can not be "sad" and likewise for the eyebrows. PCA is able to detect this behavior but mixes both eyebrows and mouth pieces in source 2 and 3. Further the nose is present in source 1. This is a typical effect in PCA since its decomposition is based on finding the directions with the most variance, which it is not always well-suited for the data.

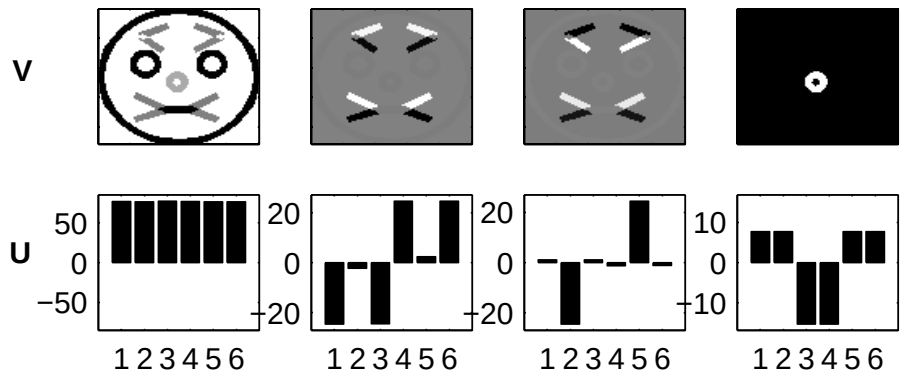

Figure 7: Applying PCA to the artificial face data. The sources $\boldsymbol{V}$ (top row) and corresponding mixing matrix estimate (bottom row). Unfortunately PCA mixes the eyebrows and mouth pieces in source 2 and 3. Further the nose is present in source 1.

5.2 Image Segmentation using Molgedey-Schuster ICA

ICA on images can be perfomed either to the signal matrix $\boldsymbol{X}$ or the transpose $\boldsymbol{X}^\top$. In the first case N = number of pixels, M = number of images in sequence corresponding to assuming independence of pixels in the sources. In this case the sources are images and the mixing matrix is time sequence. In the second case N = number of images in sequence and M = number of pixels corresponding to assuming independence driving time sequence sources. Thus, the mixing matrix corresponds to (eigen)images. This is summarized in Table 5.

The result when assuming pixel-independence (i.e., using $\boldsymbol{X}$ as signal matrix) is shown in Figure 8. The result when assuming time-independence (i.e., using $\boldsymbol{X}^\top$ as signal matrix) is shown in Figure 9.

5.3 Discussion

PCA and ICA used in real images applications often show preference towards ICA. This is mainly because ICA is able to produce a non-orthogonal basis and is not constrained by the variance ranking inherent in PCA. Using PCA as preprocessing

	Signal Matrix	
	$\boldsymbol{X}$	$\boldsymbol{X}^\top$
M	no. of images in seq.	no. of pixels
N	no. of pixels	no. of images in seq.
$\boldsymbol{S}$	images	time-sequence
$\boldsymbol{A}$	time-sequence	images
assumption	pixel-independence	time-independence

Table 5: Two ways of performing ICA on image sequences.

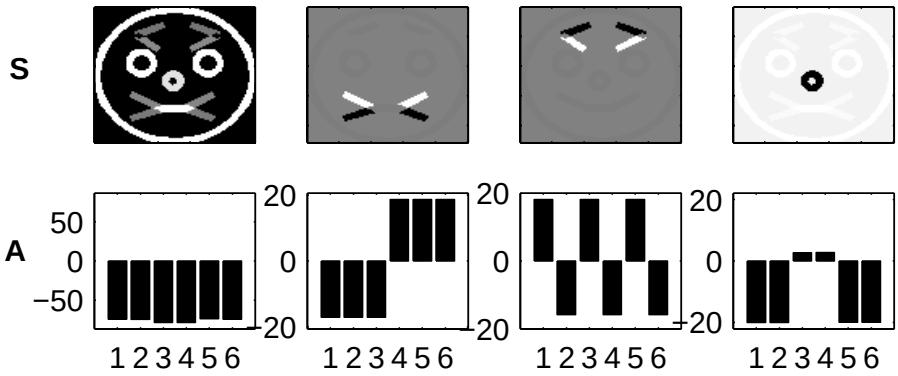

Figure 8: Using MS-ICA on the artificial face data with the *pixel-independence assumption*, i.e., $\boldsymbol{X}$ is the signal matrix. The estimated sources (eigenimages) is shown in the top row and associated mixing matrix (time-sequences) in the bottom row. Unlike PCA in Figure 7, MS-ICA does not mix eyebrows and mouths together, i.e., the sources are almost perfect except for a small problem with the nose component in source 1. Also the mixing matrix $\boldsymbol{A}$ is almost perfect in comparison with Figure 6.

to ICA in order to determine the number of sources has proven successfully [6]. Also the PCA estimate of the mixing matrix can be used as initialization for an iterative ICA scheme like Bell-Sejnowski [6] and the algorithm of Section 3.2. Performing ICA using the Molgedey-Schuster algorithm gives better results than PCA at comparable computational cost.

The choice of *pixel-independence* versus *time-independence* is related to the problem at hand. In the image segmentation problem above pixel-independence gave the best result, however, other cases have shown preference to time-independence, see e.g., [15, 16].

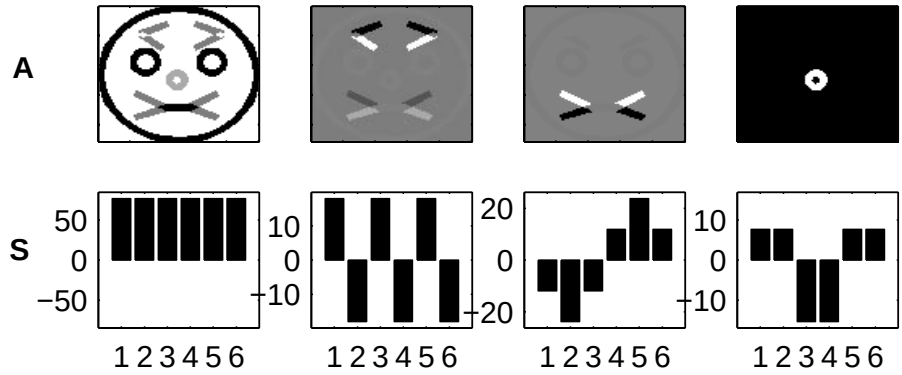

Figure 9: Using MS-ICA on the artificial face data with the *time-independence assumption*, i.e., $\boldsymbol{X}^\top$ as signal matrix. The estimated sources (time-sequences) is shown in the bottom row and associated mixing matrix (eigenimages) in the top row. The mouth is present in both eigenimage 2 and 3, thus producing a slightly worse result than that in Figure 8.

6 ICA for Text Representation

6.1 Text Analysis

The field of text analysis aims at searching for specific information and structure in text data which has emerged rapidly in recent years due to the Internet and other massive text databases. The general way of search and grouping are usually boolean[15] search and query[16] subset selection. These methods are straight forward, however, not based on statistical modeling. Due to the large amount of data any statistical approach has been very difficult, and only in the resent years a serious effort has been carried out.

The general ideal behind many text analysis algorithms are the so called N-gram histograms. The N-gram histogram is based on counting simultaneous occurrence of N words or terms. We consider merely 1-gram histograms as higher order histograms often has large areas of infinitesimal probability mass due to infrequent occurrence of many word combinations. In Figure 10 a 1-gram histogram is shown and will be referenced to as the *term/document matrix*. The term/document matrix can contain features extracted from the documents and can be used as a signal matrix $\boldsymbol{X}$ for PCA and ICA. Recently PCA and ICA has been apply to text analysis [21, 23, 25] and in the following we shall apply both PCA and ICA on the 1-gram histogram using a the MED data set [11]. The MED data set is a commonly studied collection of medical abstracts. It consists of 1033 abstracts of which 30 labels has been assigned to 696 of the documents. The goal is not to compare the performance of ICA to other unsupervised methods, rather the intend is to demonstrate its capability in

[15]Boolean search operates from AND and OR operators.

[16]By making a query a subset of the data are selected. This can e.g., be done by boolean search – often found by SQL statements.

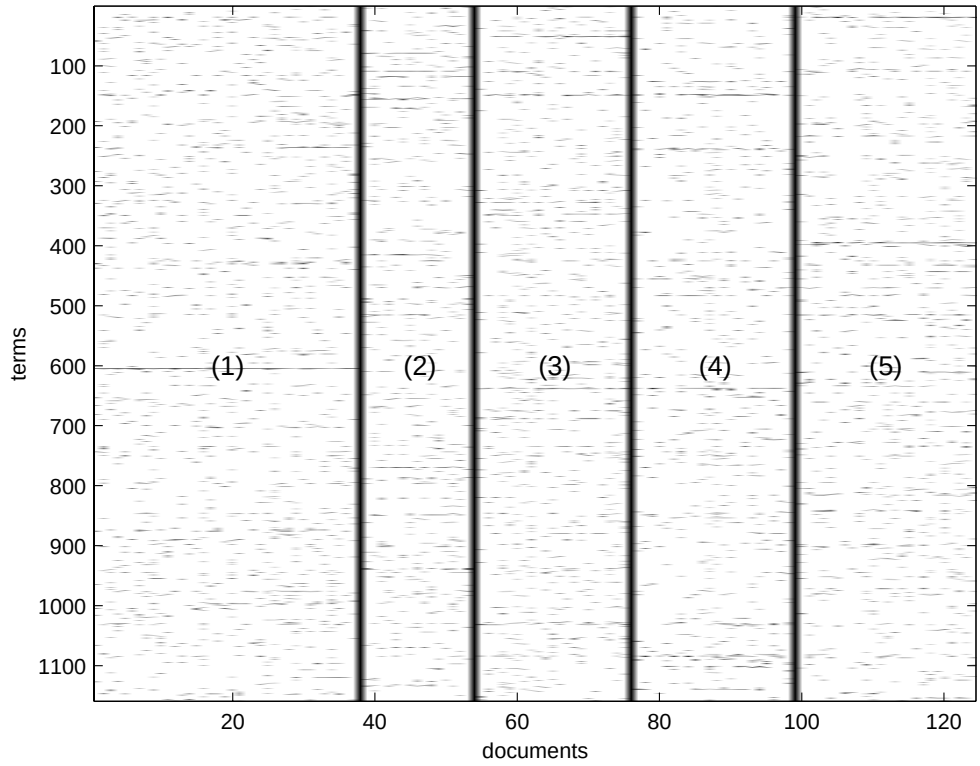

Figure 10: The term/document matrix $\boldsymbol{X}$ is an 1-gram histogram. The rows represent different words/terms appearing in a collection of text documents. In the present study we use $M = 1159$ terms. Each column represents the histogram for a specific document or text group. In the present example, $N = 124$ documents were used.

text analysis. Consequently, we restrict the study to 124 abstracts, i.e., the first 5 groups/classes in the MED data set which can be characterized by the following verbal descriptions:

1. The crystalline lens in vertebrates, including humans.
2. The relationship of blood and cerebrospinal fluid oxygen concentrations or partial pressures. A method of interest is polarography.
3. Electron microscopy of lung or bronchi.
4. Tissue culture of lung or bronchial neoplasms.
5. The crossing of fatty acids through the placental barrier. Normal fatty acid levels in placenta and fetus.

When constructing the histogram term/document matrix, words that occur in more than one abstract where chosen as a term word. In order to facilitate the analysis

commonly used words[17] were removed. 1159 terms remained in the construction of the term/document matrix. In summary, the term/document matrix $\boldsymbol{X}$ is $M = 1159$ by $N = 124$. The ICA algorithm used in this example is the noisy mixing algorithm described in Section 3.2.

6.2 Latent Semantic Analysis – PCA

A classical method for both search and grouping (clustering) is *Latent Semantic Analysis* LSA introduced by [11]. The principle of LSA is to build the term/document matrix and finding a better basis representation using PCA. Consider the SVD $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$ where $\boldsymbol{U}$ contains the eigenvectors of the term covariance matrix $\boldsymbol{X}\boldsymbol{X}^\top$. Likewise $\boldsymbol{V}$ contains the eigenvectors of the document covariance matrix $\boldsymbol{X}^\top\boldsymbol{X}$. $\mathbf{D}$ is the diagonal matrix of increasing singular values equal to square root of the eigenvalues. Paraphrased, $\boldsymbol{U}$ provides relative coordinates for the covariance between different terms and likewise, $\boldsymbol{V}$ relative coordinates for the documents. In Figure 11 the documents are represented by a 3 dimensional PCA basis. A clear data cluster structure is noticed.

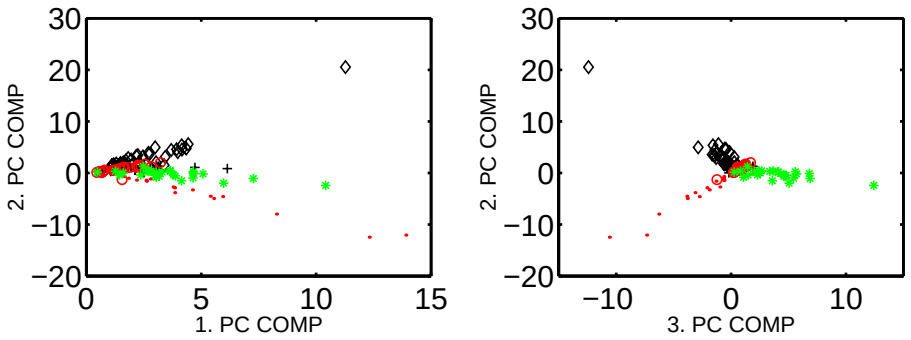

Figure 11: PCA on the term/document matrix. The documents are plotted with different signatures corresponding to the pre-labeling into 5 classes. A clear cluster structure is noticed.

Using clustering techniques the documents can now be clustered into groups of similar meaning. This also enables the characterization new document by projecting on to the identified PCA basis.

6.3 Latent Semantic Analysis – ICA

The objective of ICA in LSA is that it should serve as a clustering algorithm so that different semantic groups is represented by separate independent components. The ICA algorithm produces the mixing matrix $\boldsymbol{A}$ in which each column represents a histogram associated with a specific semantic cluster. The source matrix $\boldsymbol{S}$ expresses the how the documents contribute to the semantic clusters.

[17] A stop word list was defined.

Since we typically face problems with thousands of words in the terms list and a possibly much fewer documents, we face a so-called extremely ill-posed learning problem which can be remedied without loss of generality by PCA projection. The PCA decompose the term/document matrix on eigen-histograms. These eigen-histograms are subject to an orthogonality constraint being eigenvectors to a symmetric real matrix. We are interested in a slightly more general separation of sources that are independent as sequences, but not necessarily orthogonal in the word histogram, i.e., we would like to be able to perform a more general decomposition of the data matrix, corresponding to the model in Eq. (4). Before performing the ICA we can make use of the PCA for simplification of the ICA problem. The approach taken here is similar to the so-called "cure for extremely ill-posed learning" [26] used to simplify supervised learning in short image sequences. We first note that the likelihood, considered as a function of the columns of $\boldsymbol{A}$ (histograms) can be split in two parts. A part, $\boldsymbol{A}_1$, orthogonal to the subspace spanned by the M rows of $\boldsymbol{X}$, and a part $\boldsymbol{A}_2$ situated in the subspace spanned by the N columns of $\boldsymbol{X}$. The first is part trivially minimized for any non-zero configuration of sources by putting $\boldsymbol{A}_1 = \boldsymbol{0}$. It simply does not "couple" to data. The remaining part $\boldsymbol{A}_2$ can be projected onto an N-dimensional hyperplane spanned by the documents. In this way we reduce the high-dimensional separation problem to the separation of a square (projected) data matrix of size $N \times N$. We note that it often may be possible to further limit the dimensionality of the PCA subspace, hence, further reducing the histogram dimensionality M of the remaining problem. Using the "cure for extremely ill-posed learning" method the problem is reduced to a $M = 124$ by $N = 124$ problem without loss of generality. However, we expect that even fewer components are needed for creating a generalizable model. In Figure 12 we show the test and training set errors evaluated on training sets of 104 patterns randomly chosen among the set of 124. The test set consists of the remaining 20 documents in each resample. The generalization error shows a shallow minimum for $P = 4$ independent components, reflecting a bias-variance tradeoff (Section 3.1) as function of the complexity of the estimated mixing matrix. In Figure 13 we show scatterplots in the most variant independent components. While the distribution of documents forms rather well-defined group structure in the PCA scatterplots, clearly the ICA scatterplots are much better axis aligned. We conclude that the non-orthogonal basis found by ICA better "explains" the group structure. To further illustrate this finding we have converted the ICA solution to a pattern recognition device by a simple heuristic. We assign a group label based on the magnitude of the recovered source signal. In Table 6 and 7 we show that this device is quite successful in recognizing the group structure although the ICA training procedure is completely unsupervised. For an ICA with three independent components two are recognized perfectly, and three classes are lumped together. The four component ICA, which is the generalization optimal model, "recognizes" three of the five classes almost perfectly and confuses the two classes 3 and 4. Inspecting the groups we found that the two classes indeed are on very similar topics[18], and investigating classifications for five or more ICA

[18] They both concern medical documents on diseases of the human lungs.

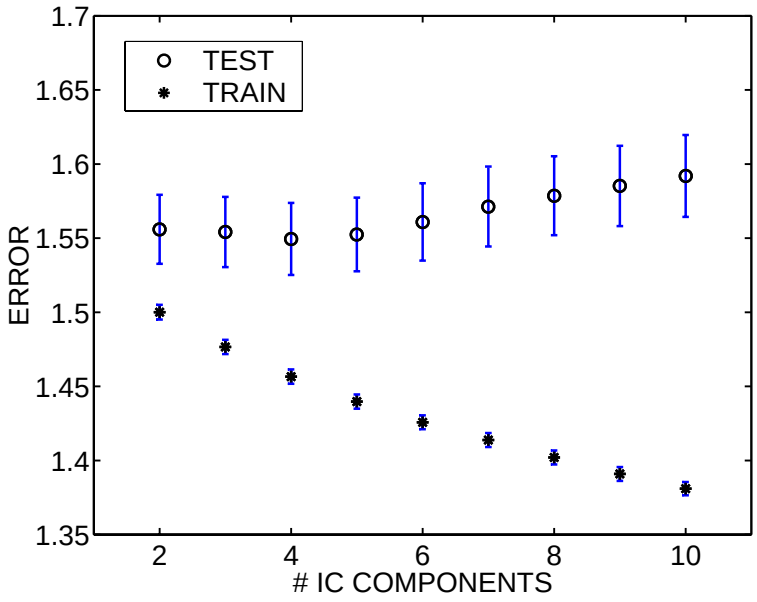

Figure 12: ICA analysis of the MED dataset. Training and test error as a function of the number of sources, or number of components P. The training set consist of 104 documents randomly chosen among the set of 124 possible and the remaining 20 is used for test. The test curve shows a shallow minimum for $P = 4$ components reflecting the bias-variance tradeoff discussed in Section 3.1.

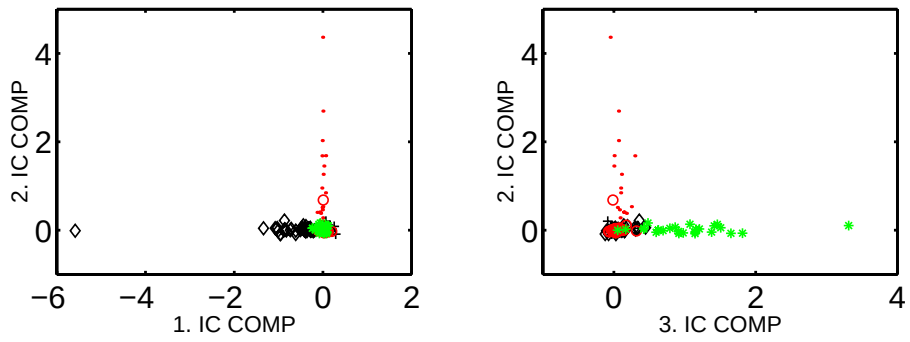

Figure 13: ICA applied on on the term/document matrix. The documents are plotted with different signatures corresponding to the pre-labeling into 5 classes. ICA projects the natural clusters along the basic vectors making them easy to separate.

component did not resolve the ambiguity between them. The ability of the ICA-classifier to identify the topic structure is further illustrated in Figure 14 where we show scatterplots color coded according to ICA classifications. This shows that the ICA is better than PCA based LSI in identifying relevant latent semantic structure. Finally, we inspect the histograms produced by ICA by backprojection using the PCA basis. Thresholding the ICA histograms we find the salient terms for the given

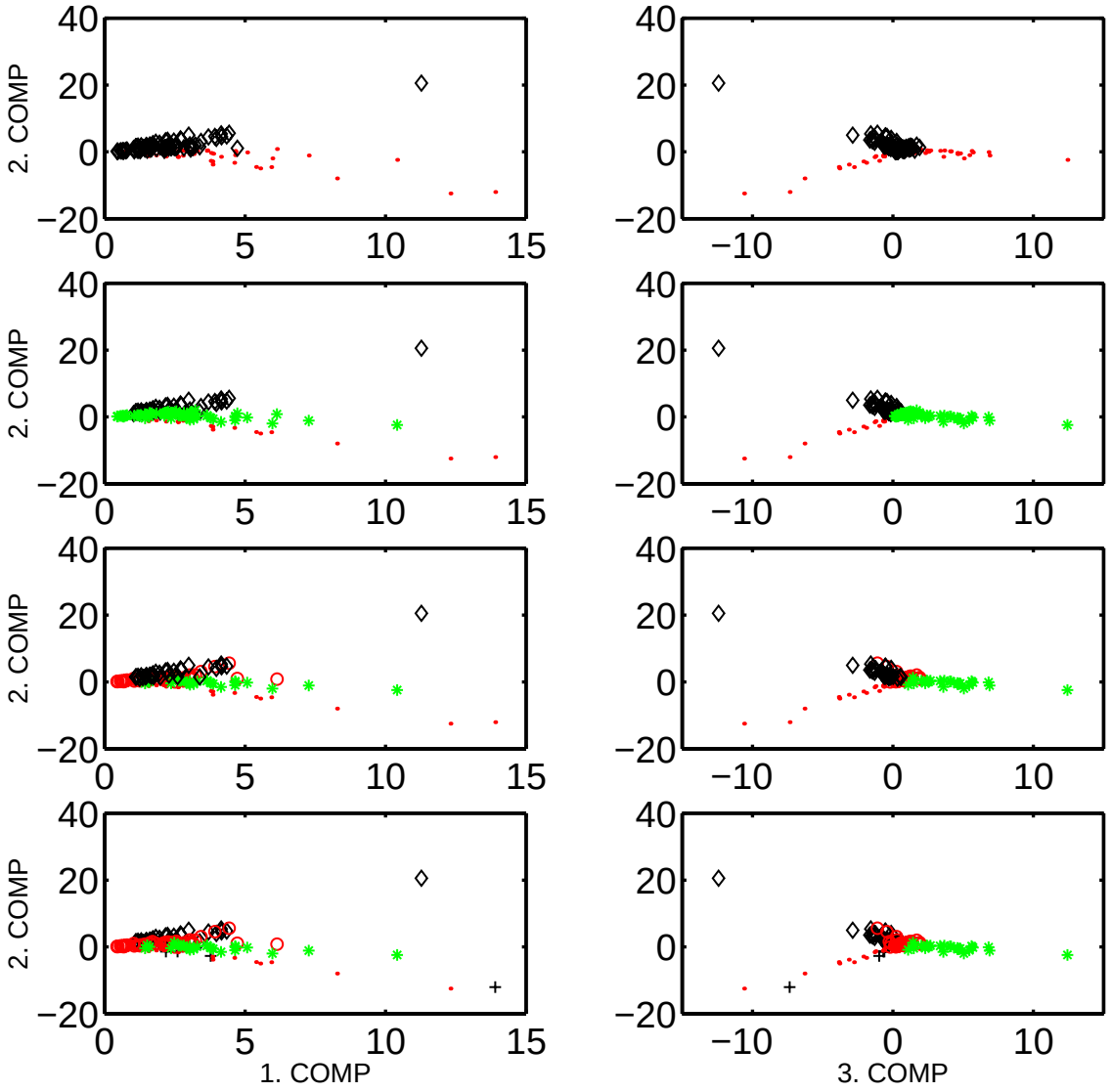

Figure 14: ICA analysis of the MED dataset. The dataset consists of 124 documents in five topics. The source signals recovered in the ICA has been converted to a simple classifier, and we have coded these classes by different colors. From top to bottom we show scatterplots in the principal component representation 1 vs. 2 and 3 vs. 2., with colors signifying the classification proposed by the ICA with 2,3,4,5 independent components respectively.

component. These terms are keywords for the given topic as shown in Tables 6 and 7 and follow nicely the behavior of the confusion matrices.

	Class					Keywords
	1	2	3	4	5	
$\mathbf{IC_1}$	37	0	0	0	0	lens protein
$\mathbf{IC_2}$	0	16	1	1	0	arterial blood cerebral oxygen rise
$\mathbf{IC_3}$	0	0	21	22	26	acid blood cell fatty free glucose insulin

Table 6: Confusion matrix for a simple classifier constructed from the three component ICA. Two of the five MED classes are recovered while the last independent component contains a mixture of the remaining three classes.

	Class					Keywords
	1	2	3	4	5	
$\mathbf{IC_1}$	31	0	0	0	0	lens protein
$\mathbf{IC_2}$	0	16	0	1	0	arterial blood cerebral oxygen rise
$\mathbf{IC_3}$	6	0	22	21	2	alveolar cell lens lung
$\mathbf{IC_4}$	0	0	0	1	24	acid blood fatty free glucose insulin

Table 7: Confusion matrix for a simple classifier constructed from the four component ICA. Three of the five MED classes are recovered, while the remaining two classes are mixed. The two unresolved classes are related by both making reference to the lung physiology.

7 Conclusion

This chapter discussed the use of Independent Component Analysis (ICA) for multimedia applications. In particular we applied ICA to separation of speech signals, segmentation of images, and text analysis/clustering.

A likelihood framework for ICA was presented and enables a unified view of different algorithms. Furthermore this enables formulation of the generalization error, defined as the expected negative log-likelihood on independent examples. The generalization error is a principled tool for model optimization, e.g., number of sources retained in the model.

We focused on two ICA algorithms: separation based on time-correlation and noisy mixing of white sources. In the first case we presented a generalized version of the Molgedey-Schuster algorithm allowing for handling undercomplete problems, alleviating inherent erroneous complex valued result, and simultaenous use of more crosscorrelation measurements while maintaining the simple non-iterative nature of the algorithm. In the noisy mixing case a maximum a posteriori estimate for source estimation was employed, and the the mixing matrix and noise variance were estimated via Boltzmann learning.

Acknowledgment

This work was funded by the Danish Research Councils through the Distributed Multimedia Technologies and Applications within Center for Multimedia and the THOR Center for Neuroinformatics. Andrew Back is acknowledged for valuable discussions concerning the Molgedey-Schuster algorithm.

A Property of the Quotient Matrix

Theorem 1 *The quotient matrix* $\boldsymbol{K} = \boldsymbol{C}_{\widetilde{\boldsymbol{x}}}(\tau)\boldsymbol{C}_{\widetilde{\boldsymbol{x}}}^{-1}(0)$ *has real eigenvalues and eigenvectors, and obtains the eigenvalue decomposition* $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^{-1}$.

Proof $\boldsymbol{C}_{\widetilde{\boldsymbol{x}}}(\tau)$ is symmetric since it can be expressed as $\boldsymbol{C}_{\widetilde{\boldsymbol{x}}}(\tau) = \widetilde{\boldsymbol{Q}}^{\top}\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{s}}(\tau)\boldsymbol{A}^{\top}\widetilde{\boldsymbol{Q}}$. Further, $\boldsymbol{C}_{\widetilde{\boldsymbol{x}}}(0)$ is positive definite, as $\boldsymbol{C}_{\boldsymbol{s}}(0)$ is positive definite. A similarity transform of $\boldsymbol{K}$ is given by

$$\boldsymbol{K}_{\text{sim}} = \boldsymbol{C}_{\widetilde{\boldsymbol{x}}}^{-1/2}(0)\boldsymbol{K}\boldsymbol{C}_{\widetilde{\boldsymbol{x}}}^{1/2}(0) = \boldsymbol{C}_{\widetilde{\boldsymbol{x}}}^{-1/2}(0)\boldsymbol{C}_{\widetilde{\boldsymbol{x}}}(\tau)\boldsymbol{C}_{\widetilde{\boldsymbol{x}}}^{-1/2}(0) \tag{39}$$

$\boldsymbol{K}_{\text{sim}}$ is thus symmetric with real eigenvalues and eigenvectors [18, Theorem 4.1.5], and obtains the eigenvalue decomposition $\boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^{\top}$ where $\boldsymbol{E}$ is the orthogonal ($\boldsymbol{E}^{\top}\boldsymbol{E} = \boldsymbol{I}$) matrix of eigenvectors and $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues. Since $\boldsymbol{K}$ and $\boldsymbol{K}_{\text{sim}}$ are similar they have the exact same eigenvalues, counting multiplicity [18, Corollary 1.3.4]. Finally, using the similarity transform $\boldsymbol{K} = \boldsymbol{C}_{\widetilde{\boldsymbol{x}}}^{1/2}(0)\boldsymbol{K}_{\text{sim}}\boldsymbol{C}_{\widetilde{\boldsymbol{x}}}^{-1/2}(0)$, then $\boldsymbol{K}$ obtains the eigenvalue decomposition $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^{-1}$ where $\boldsymbol{\Phi} = \boldsymbol{C}_{\widetilde{\boldsymbol{x}}}^{1/2}(0)\boldsymbol{E}$.

Q.E.D.

References

[1] S. Amari, A. Cichocki & H.H. Yang: "A New Learning Algorithm for Blind Signal Separation," in D. Touretzky, M. Mozer, and M. Hasselmo (eds.) *Advances in Neural Information Processing Systems 8*, MIT Press: Cambridge MA, pp. 757–763, 1996.

[2] H. Attias & C.E. Schreiner: "Blind Source Separation and Deconvolution: The Dynamic Component Analysis Algorithm," *Neural Computation* vol. 10, pp. 1373–1424, 1998.

[3] H. Attias & C.E. Schreiner: "Blind Source Separation and Deconvolution by Dynamic Component Analysis," in J. Principe *et al.*, *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing VII*, Piscataway, New Jersey: IEEE, pp. 456–465, 1997.

[4] H. Attias & C.E. Schreiner: "Blind Source Separation and Deconvolution: The Dynamic Component Analysis Algorithm," preprint to appear in *Neural Computation*, 1998. Available via `http://keck.ucsf.edu/~hagai/dca.ps`

[5] M.S. Bartlett, H.M. Lades & T.J. Sejnowski: "Idependent Component Representations for Face Recognition," *Proceedings of the SPIE – The International Society for Optical Engineering*, vol. 3299, pp. 528–539, 1998.

[6] A. Bell & T.J. Sejnowski: "An Information-Maximization Approach to Blind Separation and Blind Deconvolution, *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[7] A. Belouchrani & J.-F. Cardoso. "Maximum Likelihood Source Separation by the Expectation-Maximization Technique: Deterministic and Stochastic Implementation, " in *Proc. NOLTA*, pp. 49–53, 1995.

[8] J.-F. Cardoso & A. Soulourniac: "Blind Beamforming for Non Gaussian Signals," IEE Proceedings-F, vol. 140, no. 6, pp. 362–370, 1993.

[9] P. Comon: "Independent Component Analysis: A New Concept," *Signal Processing*, vol. 36, pp. 287–314, 1994.

[10] G. Deco & D. Obradovic: *An Information-Theoretic Approach to Neural Computing*, Springer-Verlag: Berlin Germany, ISBN 0-387-94666-7, 1996.

[11] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman: "Indexing by Latent Semantic Analysis," *Journ. Amer. Soc. for Inf. Science.*, vol. 41, pp. 391–407, 1990.

[12] F. Ehlers and H.G. Schuster: "Blind Separation of Convolutive Mixtures and an Application in Automatic Speech Recognition in a Noisy Environment," *IEEE Transactions on Signal Processing*, vol. 45, no. 10, pp. 2608–2612, Oct. 1997.

[13] S. Geman, E. Bienenstock & R. Doursat, "Neural Networks and the Bias/Variance Dilemma," *Neural Computation*, vol. 4, pp. 1–58, 1992.

[14] L.K. Hansen & J. Larsen *Unsupervised Learning and Generalization.* In Proceedings of the IEEE International Conference on Neural Networks 1996, Washington DC, vol. 1, 1996, pp. 25–30.

[15] L.K. Hansen & J. Larsen: "Soruce Separation in Short Image Sequences using Delayed Correlation," in *Proceedings of NORSIG'98*, Vigsø, Denmark, pp. 253–256, ISBN-87-985750-8-2, June 1998. `ftp://eivind.imm.dtu.dk/dist/1998/hansen.norsig98.ps.Z`.

[16] L.K. Hansen: *Blind Separation of Noisy Mixtures.* Techn. Report, Department of Mathematical Modelling, Techn. Univ. of Denmark, 1998. `http://eivind.imm.dtu.dk/staff/lkhansen/ica.html`.

[17] L.K. Hansen, J. Larsen, F.Å. Nielsen, S.C. Strother, E. Rostrup, R. Savoy, C. Svarer & O.B. Paulson: "Generalizable Patterns in Neuroimaging: How Many Principal Components?" in *NeuroImage*, vol. 9, pp. 534–544, 1999.

[18] R.A. Horn & C.R. Johnson: *Matrix Analysis*, Cambridge University Press: Cambridge UK, 1994.

[19] J. Hurri, A. Hyvärinen, J. Kahunen & E. Oja: "Image Feature Extraction Using Independent Component Analysis," *Proceedings of IEEE Nordic Conference on Signal Processing (NORSIG'96)*, 1996.

[20] A. Hyvärinen, E. Oja, P. Hoyer & J. Hurri: "Image Feature Extraction by Sparse Coding and Independent Component Analysis," *Procceings of Int. Conf. on Pattern Recognition 98*, Brisbane, Australia, pp. 1268–1273, 1998.

[21] C.L. Isbell, Jr. & P. Viola: "Restructuring Sparse High Dimensional Data for Effective Retrieval," *Advances in Neural Information Processing Systems 11*, MIT Press: Cambridge MA, pp. 480–486, 1999.

[22] C. Jutten & J. Herault: "Blind Separation of Sources: An Adaptive Algorithm Based on Neuromimetic Architecture, *Signal Processing*, vol. 24, pp. 1–10, 1991.

[23] T. Kolenda & L.K. Hansen: "Independent Components in Text," *Techn. Report, Dept. of Mathematical Modelling, DTU*, 1999. Available via `ftp://eivind.imm.dtu.dk/dist/kolenda.nips99.ps.gz`

[24] S.Y. Kung & C. Mejuto "Extraction of Independent Components from Hybrid Mixture: KuicNet Learning Algorithm and Applications," *Procedings of IEEE ICASSP98*, vol. 2, pp. 1209–1212, Seattle, USA, May 1998.

[25] T.K. Landauer, D. Laham & P. Foltz: "Learning Human-like Knowledge by Singular Value Decomposition: A Progress Report," *Advances in Neural Information Processing Systems 10*, MIT Press: Cambridge MA, pp. 45–51, 1998.

[26] B. Lautrup, L.K. Hansen I. Law, N. Mørch, C. Svarer & S.C. Strother: "Massive Weight Sharing: A Cure for Extremely Ill-posed Problems," in H.J. Hermanet *et al.*, (eds.) *Supercomputing in Brain Research: From Tomography to Neural Networks*, World Scientific Pub. Corp. pp. 137–148, 1995.

[27] T.-W. Lee, M. Girolami, A.J. Bell & T.J. Sejnowski: "A Unifying Information-theoretic Framework for Independent Component Analysis," Salk Institute preprint, to appear in *International Journal on Computers and Mathematics with Applications*, in press, 1999. Available via `http://www.cnl.salk.edu/~tewon/Public/ijmc99.ps.gz`

[28] T.-W. Lee, M.S. Lewicki, M. Girolami, and T.J. Sejnowski: "Blind Source Separation of More Spurces Than Mixtures Using Overcomplete Representations," *Signal Procesing Letters*, vol. 4, no. 4., April 1999. Available via `http://www.cnl.salk.edu/~tewon/Public/ocica99.ps.gz`

[29] T.-W. Lee, M.S. Lewicki & T.J. Sejnowski: "Unsupervised Classification, Segmentation and De-noising of Images using ICA Mixture Models," submitted for publication.

[30] T.-W. Lee, M.S. Lewicki & T.J. Sejnowski: "ICA Mixture Models for Image Processing," *Proceedings of the 6th Joint Symposium on Neural Computation*, California Institute of Technology, Pasadena, CA, pp. 79–86, 1999.

[31] M.S. Lewicki & T.J. Sejnowski: "Learning Overcomplete Representations" Preprint Salk Institute, *Neural Computation*, in press, 1999. Available via `http://www.cnl.salk.edu/~lewicki/papers/overcomplete.ps.gz`

[32] D. MacKay: "Maximum Likelihood and Covariant Algorithms for Independent Components Analysis," *Draft 3.7*, 1996. Available via `ftp: mroa.cam.ac.uk/hello.ps.gz`

[33] L. Molgedey & H. Schuster: "Separation of Independent Signals using Time-Delayed Correlations," *Physical Review Letters*, vol. 72, no. 23 , pp. 3634–3637, 1994.

[34] E. Moulines, J.-F. Cardoso & E. Gassiat: "Maximum Likelihood for Blind Separation and Deconvolution of Noisy Signals Using Mixture Models, in *Proceedings of ICASSP'97* Munich, Germany, vol. 5, 1997, pp. 3617–3620.

[35] B.A. Olshausen: *Learning Linear, Sparse, Factorial codes* A.I. Memo 1580, Massachusetts Institute of Technology, 1996.

[36] E. Oja: "PCA, ICA, and Nonlinear Hebbian Learning, in *Proceedings of the International Conference on Artificial Neural Networks ICANN-95*, 1995, pp. 89–94.

[37] B.A. Pearlmutter & L.C. Parra: "Maximum Likelihood Blind Source Separation: A Context-Sensitive Generalization of ICA," in M.C. Mozer *et al.* (eds.) *Advances in Neural Information Processing Systems 9*, Cambridge, Massachusetts: MIT Press, 1997, pp. 613–619.

[38] C. Peterson & J.R. Anderson: "Mean Field Theory Learning Algorithm for Neural Networks," *Complex Systems*, vol. 1, pp. 995–1019, 1987.