Journal of Machine Learning Research Submitted 03/04; Revised

Denoising Source Separation

Jaakko Särelä JAAKKO.SARELA@HUT.FI
Neural Networks Research Centre
Helsinki University of Technology
P.O.Box 5400, FI-02015 HUT, Espoo, FINLAND

Harri Valpola HARRI.VALPOLA@HUT.FI
Artificial Intelligence Laboratory
University of Zurich
Andreasstrasse 15, 8050 Zurich, Switzerland
or *Neural Networks Research Centre*
Helsinki University of Technology
P.O.Box 5400, FI-02015 HUT, Espoo, FINLAND

Editor: Michael Jordan

Abstract

A new algorithmic framework called denoising source separation (DSS) is introduced. The main benefit of this framework is that it allows for easy development of new source separation algorithms which are optimised for specific problems. In this framework, source separation algorithms are constucted around denoising procedures. The resulting algorithms can range from almost blind to highly specialised source separation algorithms. Both simple linear and more complex nonlinear or adaptive denoising schemes are considered. Some existing independent component analysis algorithms are reinterpreted within DSS framework and new, robust blind source separation algorithms are suggested. Although DSS algorithms need not be explicitly based on objective functions, there is often an implicit objective function that is optimised. The exact relation between the denoising procedure and the objective function is derived and a useful approximation of the objective function is presented. In the experimental section, various DSS schemes are applied extensively to artificial data, to real magnetoencephalograms and to simulated CDMA mobile network signals. Finally, various extensions to the proposed DSS algorithms are considered. These include nonlinear observation mappings, hierarchical models and overcomplete, nonorthogonal feature spaces. With these extensions, DSS appears to have relevance to many existing models of neural information processing.

Keywords: blind source separation, BSS, prior information, denoising, denoising source separation, DSS, independent component analysis, ICA, magnetoencephalograms, MEG, CDMA

1. Introduction

Over the recent years, source separation of linearly mixed signals has attracted a wide range of researchers. The focus of this research has been on developing algorithms that make minimal assumptions on the underlying process, thus approaching blind source sep-

aration (BSS). Independent component analysis (ICA) (Hyvärinen et al., 2001b) clearly follows this tradition. This blind approach certainly has its assets, giving the algorithms a wide range of possible applications. ICA has been a valuable tool, in particular, in testing certain hypotheses in magnetoencephalogram (MEG) and electroencephalogram (EEG) analysis (*cf.,* Vigário et al., 2000).

Nearly always, however, there is further information due to the experimental setup, other design specifications or cumulated knowledge due to scientific research. For example in biomedical signal analysis (*cf.,* Gazzaniga, 2000, Rangayyan, 2002), careful design of experimental setups provides us with presumed signal characteristics. In man-made technology, such as a CDMA mobile system (*cf.,* Viterbi, 1995), the transmitted signals are even more restricted.

The Bayesian approach provides a sound framework for including prior information into inferences about the signals. Recently, several Bayesian ICA algorithms have been suggested (cf. Knuth, 1998, Attias, 1999, Lappalainen, 1999, Miskin and MacKay, 2001, Choudrey and Roberts, 2001, Højen-Sørensen et al., 2002, Chan et al., 2003). They offer accurate estimations for the linear model parameters. For instance, universal density approximation using mixture of Gaussians (MoG) may be used for the source distributions. Furthermore, hierarchical models can be used for incorporating complex prior information (*cf.,* Valpola et al., 2001). However, the Bayesian approach does not always result in simple or computationally efficient algorithms.

FastICA (Hyvärinen, 1999) provides a set of algorithms for performing ICA based on optimising easily calculatable contrast functions. The algorithms are fast but often more accurate results can be achieved by computationally more demanding algorithms (Giannakopoulos et al., 1999), for example by the Bayesian ICA algorithms. Valpola and Pajunen (2000) analysed the factors behind the speed of FastICA. The analysis suggested that the nonliearity used in FastICA can be interpreted as denoising. Bayesian noise filtering as the nonlinearity resulted in fast Bayesian ICA.

Denoising corresponds to procedural knowledge while in most approaches to source separation, the algorithms are derived from explicit objective functions or generative models. This corresponds to declarative knowledge. Algorithms are procedural, however. Thus declarative knowledge has to be translated into procedural form, which may result in complex and computationally heavy algorithms.

In this paper, we generalise the denoising intepretation by Valpola and Pajunen (2000) and introduce a source separation framework called denoising source separation (DSS). We show that it is actually possible to construct the source separation algorithms around the denoising methods themselves. Fast and accurate denoising will result in a fast and accurate separation algorithm. We suggest that various kinds of prior knowledge can be easily formulated in terms of denoising. In some cases a denoising scheme has been used to post-process the results after separation (*cf.,* Vigneron et al., 2003), but in DSS framework this denoising can be used for the source separation itself.

The paper is organised as follows: After setting the general problem of linear source separation in Sec. 2, we review an expectation-maximisation (EM) algorithm as a solution to a generative linear model and a one-unit version of it (Sec. 2.1). We interpret the nonlinearity as denoising and call this one-unit algorithm DSS. Equivalence of the linear DSS and a power method is shown in Sec. 2.2 and the convergence of the linear DSS is

analysed via the power method. We then proceed in general nonlinear denoising (Sec. 2.3). The applicability of two common extensions of the power method: deflation and spectral shift are discussed in the rest of the section. Section 3 discusses the often implicit objective function in the DSS algorithms, especially in the nonlinear case. We then introduce some practical denoising functions in Sec. 4. These denoising functions are extensively applied to artificial mixtures (Sec. 5.1) and to MEG recordings (Secs. 5.2 and 5.3). We also apply a DSS algorithm to bit-stream recovery in a simulated CDMA network (Sec. 5.4). Finally, in Sec. 6, we discuss extensions to DSS framework and their connections to models of neural information processing.

2. Source separation by denoising

Consider a linear instantaneous mixing of sources:

$$\mathbf{X} = \mathbf{AS} + \boldsymbol{\nu}\,, \tag{1}$$

where

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_M \end{bmatrix}\,, \qquad \mathbf{S} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_N \end{bmatrix}\,. \tag{2}$$

The source matrix $\mathbf{S}$ consists of N sources. Each individual source $\mathbf{s}_i$ consists of T samples, that is, $\mathbf{s}_i = [s_i(1) \ldots s_i(t) \ldots s_i(T)]$. Note that in order to simplify the notation throughout the paper, we have defined each source to be a row vector instead of the more traditional column vector. The symbol t often stands for time, but other possibilities include, *e.g.*, space. For the rest of the paper, we refer to t as time, for convenience. The observations $\mathbf{X}$ consist of M mixtures of the sources, that is, $\mathbf{x}_i = [x_i(1) \ldots x_i(t) \ldots x_i(T)]$. Usually it is assumed that $M \geq N$. The linear mapping $\mathbf{A} = [\mathbf{a}_1\, \mathbf{a}_2 \cdots \mathbf{a}_N]$ consists of the mixing vectors $\mathbf{a}_i = [a_{1i}\, a_{2i} \ldots a_{Mi}]^T$, and is usually called mixing matrix. In the model, there is some Gaussian noise $\boldsymbol{\nu}$, too. The sources, the noise and hence also the mixtures can be assumed to have zero mean without losing generality because the mean can always be removed from the data.

If the sources are assumed i.i.d. Gaussian, this is a general, linear factor analysis model with rotational invariance. There are several ways to fix the rotation, *i.e.*, to separate the original sources, $\mathbf{S}$. Some approaches assume structure for the mixing matrix. If no structure is assumed, the solution to this problem is usually called blind source separation (BSS). Note that this approach is not really blind, since one always needs some information to be able to fix the rotation. One such piece of information is the non-Gaussianity of the sources, which leads to the recently popular ICA methods (*cf.,* Hyvärinen et al., 2001b). The temporal structure of the sources may be used as well as in Tong et al. (1991), Molgedey and Schuster (1994), Belouchrani et al. (1997), Ziehe and Müller (1998), Pham and Cardoso (2001).

The rest of this section is organised as follows: first we review an EM algorithm for source separation and a one-unit-version derived from it in Sec. 2.1. The E- and M-steps have natural interpretations as denoising of the sources and re-estimation of the mixing

vector, respectively, and the derived algorithm provides the starting point for the DSS framework. The convergence of the DSS algorithms for Gaussian sources (linear denoising) are analysed in Sec. 2.2 and in the case of non-Gaussian sources (nonlinear denoising) in Sec. 2.3. Deflation and symmetric method for extracting several sources are reviewed in Sec. 2.4. Section. 2.5 discusses a speedup technique called spectral shift.

2.1 One-unit algorithm for source separation

The EM algorithm (Dempster et al., 1977) is a method for performing maximum likelihood estimation when part of the data is missing. One way to perform EM estimation in case of linear models is to assume that the missing data consists of the sources and the mixing matrix needs to be estimated. In the following, we review one such EM algorithm by Bermond and Cardoso (1999) and a derivation of a one-unit version of it by Hyvärinen et al. (2001b).

The algorithm proceeds by alternating two steps: 1) E-step and 2) M-step. In the E-step, the posterior distribution for the sources is calculated based on the known data and the current estimate of the mixing matrix. In the M-step, the mixing matrix is fit to the new source estimates. In other words:

$$\mathrm{E}-\mathrm{step}:\mathrm{compute}\; q(\mathbf{S}) = p(\mathbf{S}|\mathbf{A},\mathbf{X}) = p(\mathbf{X}|\mathbf{A},\mathbf{S})p(\mathbf{S})/p(\mathbf{X}|\mathbf{A}) \tag{3}$$

$$\mathrm{M}-\mathrm{step}:\mathrm{find}\; \mathbf{A}_{\mathrm{new}} = \mathrm{argmax}_{\mathbf{A}}\, \mathrm{E}_{q(\mathbf{S})}[\log p(\mathbf{S},\mathbf{X}|\mathbf{A})] = \mathbf{C}_{\mathbf{XS}}\mathbf{C}_{\mathbf{SS}}^{-1}, \tag{4}$$

where $p(\mathbf{X}|\mathbf{A},\mathbf{S})$ is considered to be the likelihood of $\mathbf{S}$ and $p(\mathbf{S})$ is the prior for the sources. $p(\mathbf{X}|\mathbf{A})$ is a normalising constant. Furthermore,

$$\mathbf{C}_{\mathbf{XS}} = \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}[\mathbf{x}(t)\mathbf{s}(t)^T|\mathbf{X},\mathbf{A}] = \frac{1}{T}\sum_{t=1}^{T} \mathbf{x}(t)\,\mathrm{E}[\mathbf{s}(t)^T|\mathbf{X},\mathbf{A}] \tag{5}$$

$$\mathbf{C}_{\mathbf{SS}} = \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}[\mathbf{s}(t)\mathbf{s}(t)^T|\mathbf{X},\mathbf{A}], \tag{6}$$

where $\mathbf{x}(t) = [x_1(t)\ \cdots\ x_i(t)\ \cdots\ x_M(t)]^T$ and $\mathbf{s}(t) = [s_1(t)\ \cdots\ s_j(t)\ \cdots\ s_N(t)]^T$ are used to denote the values of all of the mixtures and the sources at the time instance t, respectively.

Many source separation algorithms preprocess the data by normalising the covariance to unit matrix, *i.e.*, $\mathbf{C}_{\mathbf{XX}} = \mathbf{X}\mathbf{X}^T/T = \mathbf{I}$. This is referred to as sphering or whitening and its result is that any signal obtained by projecting the sphered data on any unit vector has zero mean and unit variance. Furthermore, orthogonal projections yield uncorrelated signals. Often sphering is combined with reducing the dimension of the data by selecting a principal subspace which contains most of the energy of the original data.

Because of the indeterminacy of scale in linear models, it is necessary to fix either the variance of sources or the norm of mixing matrix. It is usual to fix the variance of the sources to unity $\mathbf{S}\mathbf{S}^T/T = \mathbf{I}$. Then, assuming that the linear independent-source model holds and there is infinite amount of data, with Gaussian noise, the covariance of the sphered data is $\mathbf{A}\mathbf{S}\mathbf{S}^T\mathbf{A}^T/T + \mathbf{\Sigma}_\nu = \mathbf{A}\mathbf{A}^T + \mathbf{\Sigma}_\nu = \mathbf{I}$, *i.e.*, a unit matrix because of the sphering. If the noise variance is proportional to the covariance of the data that is due to the sources, *i.e.*, $\mathbf{\Sigma}_\nu \propto \mathbf{A}\mathbf{A}^T$, it holds that $\mathbf{A}\mathbf{A}^T \propto \mathbf{I}$, which means that the mixing matrix $\mathbf{A}$ is orthogonal

for sphered data. Furthermore, the likelihood $L(\mathbf{S}) = p(\mathbf{X}|\mathbf{A}, \mathbf{S})$ of $\mathbf{S}$ can be factorised:

$$L(\mathbf{S}) = C \prod_i L_i(\mathbf{s}_i)\,, \tag{7}$$

where the constant C is independent of $\mathbf{S}$. The constant C reflects the fact that likelihoods do not normalise the same way as probability densities. The above factorisation still becomes unique if $L_i(\mathbf{s}_i)$ are appropriately normalised. In the case of linear model with Gaussian noise, a convenient normalisation is to require the maximum of $L_i(\mathbf{s}_i)$ to equal one. The terms can then be shown to equal

$$L_i(\mathbf{s}_i) = \exp\left(-\frac{1}{2}(\mathbf{s}_i - \mathbf{a}_i^{-1}\mathbf{X})\,\mathbf{\Sigma}_{\mathbf{s},\nu}^{-1}\,(\mathbf{s}_i - \mathbf{a}_i^{-1}\mathbf{X})^T\right)\,, \tag{8}$$

where $\mathbf{a}_i^{-1}$ is the ith row vector of $\mathbf{A}^{-1}$ and $\mathbf{\Sigma}_{\mathbf{s},\nu} \propto \mathbf{I}$ is a diagonal matrix with the diagonal elements equalling $\sigma_\nu^2/(\mathbf{a}_i^T\mathbf{a}_i)$.

Since the prior $p(\mathbf{S})$ factorises, too, the sources are independent in the posterior $q(\mathbf{S})$ and the covariance $\mathbf{C_{SS}}$ is diagonal. This means that $\mathbf{C}_{\mathbf{SS}}^{-1}$ reduces to scaling of individual sources in the M-step (4).

Noisy estimates of the sources can be recovered by $\mathbf{S} = \mathbf{A}^{-1}\mathbf{X}$ which is the mode of the likelihood. Since $\mathbf{A}^{-1} \propto \mathbf{A}^T$ and the posterior $q(\mathbf{S})$ depends on the data only through the likelihood $L(\mathbf{S})$, the expectation $\mathrm{E}[\mathbf{S}|\mathbf{X}, \mathbf{A}]$ is a function of $\mathbf{A}^T\mathbf{X}$, or for individual sources, $\mathrm{E}[\mathbf{s}_i|\mathbf{X}, \mathbf{A}] = \mathbf{f}(\mathbf{a}_i^T\mathbf{X})$. In the case of Gaussian source model $p(\mathbf{S})$, this function is linear (further discussion in Sec. 2.2). The expectation can be computed exactly in some other cases, too, *e.g.*, when the source distributions are mixtures of Gaussians (MoG)[1]. In other cases the expectation can be approximated for instance by $\mathrm{E}_{q(\mathbf{S})}[\mathbf{S}] = \mathbf{S} + \epsilon\,\partial \log p(\mathbf{S})/\partial \mathbf{S}$, where the constant ϵ depends on the noise variance.

In the EM algorithm, all the components are estimated simultaneously. However, after presphering it is possible to extract the sources one-by-one (*cf.,* Hyvärinen et al., 2001b, for a similarly derived algorithm):

$$\mathbf{s} = \mathbf{w}^T\mathbf{X} \tag{9}$$

$$\mathbf{s}^+ = \mathbf{f}(\mathbf{s}) \tag{10}$$

$$\mathbf{w}^+ = \mathbf{X}\mathbf{s}^{+T} \tag{11}$$

$$\mathbf{w}_{\mathrm{new}} = \frac{\mathbf{w}^+}{||\mathbf{w}^+||}. \tag{12}$$

In this algorithm, the first step (9) calculates the noisy estimate of one source and corresponds to the mode of the likelihood $p(\mathbf{X}|\mathbf{w}, \mathbf{s})$. It is a convention to denote the mixing vector $\mathbf{a}$, which in this case is also the separating vector, by $\mathbf{w}$. The second step (10) corresponds to the expectation of $\mathbf{s}$ over $q(\mathbf{S})$ and can be seen as denoising based on the model of the sources. Note that $\mathbf{f}(\mathbf{s})$ is a row vector valued function of a row vector argument. The re-estimation step (11) calculates the new ML estimate of the mixing vector and the M-step in Eq. (11) is completed by normalisation (12). This prevents the norm of the mixing vector from diverging. Although this algorithm separates only one component, it

1. MoG as the source distributions would lead to ICA.

has been shown that the original sources correspond to stable fixed points of the algorithm under quite general conditions (*cf.,* Theorem 8.1, Hyvärinen et al., 2001b), provided that the independent source model holds.

In this paper, we interpret the step (10) as denoising. While this intepretation is not novel, it allows for development of new algorithms that are not derived starting from generative models. We call all of the algorithms where Eq. (10) can be intepreted as denoising and that have the form (9)–(12) DSS algorithms.

2.2 Convergence analysis using power method intepretation

In this section, we analyse the convergence of the above derived DSS algorithm. We show that the DSS algorithm converges to the eigenvector of a data matrix that has been filtered and is equivalent to the classical power method for the covariance of the filtered data. This allows us to study the convergence in detail and we will suggest speedups. Some of the speedups where suggested by Valpola and Pajunen (2000) but the present section provides a rigorous convergence analysis for them.

First, let us assume that the source is Gaussian with autocovariance matrix $\mathbf{\Sigma_{ss}}$. The non-Gaussian case is discussed in Sec. 2.3. The prior probability density function for a Gaussian source is given by

$$p(\mathbf{s}) = \frac{1}{\sqrt{|2\pi\mathbf{\Sigma_{ss}}|}} \exp\left(-\frac{1}{2}\mathbf{s}\mathbf{\Sigma_{ss}^{-1}}\mathbf{s}^T\right), \tag{13}$$

where $\mathbf{\Sigma_{ss}}$ is the autocovariance matrix of the source and $|\mathbf{\Sigma_{ss}}|$ is its determinant. Furthermore, as noted in Eq. (8), the likelihood $L(\mathbf{s})$ is an unnormalised Gaussian with the diagonal covariance $\mathbf{\Sigma}_{\mathbf{s},\nu}$:

$$L(\mathbf{s}) = \exp\left(-\frac{1}{2}(\mathbf{s} - \mathbf{w}^T\mathbf{X})\mathbf{\Sigma}_{\mathbf{s},\nu}^{-1}(\mathbf{s} - \mathbf{w}^T\mathbf{X})^T\right). \tag{14}$$

After some algebraic manipulation, the Gaussian posterior is reached:

$$q(\mathbf{s}) = \frac{1}{\sqrt{|2\pi\mathbf{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{s} - \boldsymbol{\mu})\mathbf{\Sigma}^{-1}(\mathbf{s} - \boldsymbol{\mu})^T\right), \tag{15}$$

with mean $\boldsymbol{\mu} = \mathbf{w}^T\mathbf{X}\left(\mathbf{I} + \sigma_\nu^2\mathbf{\Sigma_{ss}^{-1}}\right)^{-1}$, and variance $\mathbf{\Sigma}^{-1} = \frac{1}{\sigma_\nu^2} + \mathbf{\Sigma_{ss}^{-1}}$. Hence, the denoising step (10) becomes

$$\mathbf{s}^+ = \mathbf{f}(\mathbf{s}) = \mathbf{s}\left(\mathbf{I} + \sigma_\nu^2\mathbf{\Sigma_{ss}^{-1}}\right)^{-1} = \mathbf{s}\mathbf{D}, \tag{16}$$

which corresponds to linear denoising. The denoising step in the DSS algorithm $\mathbf{s}^+ = \mathbf{f}(\mathbf{s})$ is thus equivalent to multiplying the current source estimate $\mathbf{s}$ with a constant matrix $\mathbf{D}$.

To gain more intuition for the denoising, it is useful to consider the eigenvalue decomposition of $\mathbf{D}$. It turns out that $\mathbf{D}$ and $\mathbf{\Sigma_{ss}}$ have the same eigenvectors and the eigenvalue decompositions are

$$\mathbf{\Sigma_{ss}} = \mathbf{V}\Lambda_\Sigma\mathbf{V}^T \tag{17}$$

$$\mathbf{D} = \mathbf{V}\Lambda_D\mathbf{V}^T, \tag{18}$$

where $\mathbf{V}$ is an orthonormal matrix with the eigenvectors as colums and Λ is a diagonal matrix with the corresponding eigenvalues on the diagonal. The eigenvalues are related as

$$\lambda_{D,i} = \frac{1}{1 + \frac{\sigma_\nu^2}{\lambda_{\Sigma,i}}} \,. \tag{19}$$

Note that $\lambda_{D,i}$ is a monotonically increasing function of $\lambda_{\Sigma,i}$. Those directions of $\mathbf{s}$ are dampened the most which have the smallest variances according to the prior model of $\mathbf{s}$.

Now, let us pack the different phases of the algorithm (9), (16), (11) together:

$$\mathbf{w}^+ = \mathbf{X}\mathbf{s}^{+T} = \mathbf{X}\mathbf{D}\mathbf{s}^T = \mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{w} \,. \tag{20}$$

By writing $\mathbf{\Lambda}_D = \mathbf{\Lambda}_D^{\frac{1}{2}}\mathbf{\Lambda}_D^{\frac{1}{2}T} = \mathbf{\Lambda}^*\mathbf{\Lambda}^{*T}$ and adding $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ in the middle, we may split the denoising matrix into two parts:

$$\mathbf{D} = \mathbf{D}^*\mathbf{D}^{*T} \,, \tag{21}$$

where $\mathbf{D}^* = \mathbf{V}\mathbf{\Lambda}^*\mathbf{V}^T$. Further, let us denote $\mathbf{Z} = \mathbf{X}\mathbf{D}^*$. This brings the DSS algorithm for estimating one separating vector into the form

$$\mathbf{w}^+ = \mathbf{Z}\mathbf{Z}^T\mathbf{w} \,. \tag{22}$$

This is the classical *power method* (*cf.,* Wilkinson, 1965) implementation for principal component analysis (PCA). Note that $\mathbf{Z}\mathbf{Z}^T$ is the unnormalised covariance matrix. The algorithm converges to the *fixed point* $\mathbf{w}^*$ where

$$\lambda\mathbf{w}^* = \mathbf{Z}\mathbf{Z}^T/T\,\mathbf{w}^* \,, \tag{23}$$

where λ corresponds to the principal eigenvalue of the covariance matrix $\mathbf{Z}\mathbf{Z}^T/T$ and $\mathbf{w}^*$ is the principal direction. The asterisk is used to stress the fact that the estimate is at the fixed point.

The above power method converges to the direction where the variance $||\mathbf{w}^T\mathbf{Z}||^2$ is maximised. This means that the DSS algorithm using denoising $\mathbf{s}^+ = \mathbf{s}\mathbf{D}$, derived from the EM algorithm, is equivalent to PCA applied to $\mathbf{Z}$. The matrix $\mathbf{D}^*$ is similarly intepreted as denoising the data $\mathbf{X}$. Thus one denoising ($\mathbf{D}$) in the DSS algorithm corresponds to another denoising ($\mathbf{D}^*$) in PCA. The algorithms maximise the objective function of the linear DSS:

$$g_{\mathrm{lin},\mathbf{w}}(\mathbf{w}) = \mathbf{w}^T\mathbf{Z}\mathbf{Z}^T\mathbf{w} = \mathbf{w}^T\mathbf{X}\mathbf{D}^*\mathbf{D}^{*T}\mathbf{X}^T\mathbf{w} = \mathbf{s}\mathbf{D}\mathbf{s}^T = \mathbf{s}\mathbf{f}^T(\mathbf{s}) = g_{\mathrm{lin},\mathbf{s}}(\mathbf{s}) \,. \tag{24}$$

The last step provides an intuitive explanation for the algorithm: it maximises the amount of the signal that gets through in the denoising. For this reason, we often write the objective function as the function of the source estimate $\mathbf{s}$, instead of the mixing vector $\mathbf{w}$. We usually omit the subscript $\mathbf{s}$ when there is no danger of confusion. The fixed point equation (23) can also be expressed in terms of $\mathbf{s}$:

$$\lambda\mathbf{s}^* = \mathbf{f}_{\mathrm{lin}}(\mathbf{s}^*) - \mathbf{s}_{\mathrm{orth}} \,, \tag{25}$$

where $\mathbf{s}_{\mathrm{orth}}$ is orthogonal to $\mathbf{X}$ and is therefore removed by the re-estimation step (11).

The operation of linear DSS algorithm is depicted in Fig. 1. Figure 1a shows two sources that have been mixed into Fig. 1b. After whitening in Fig. 1c, the basis is roughly

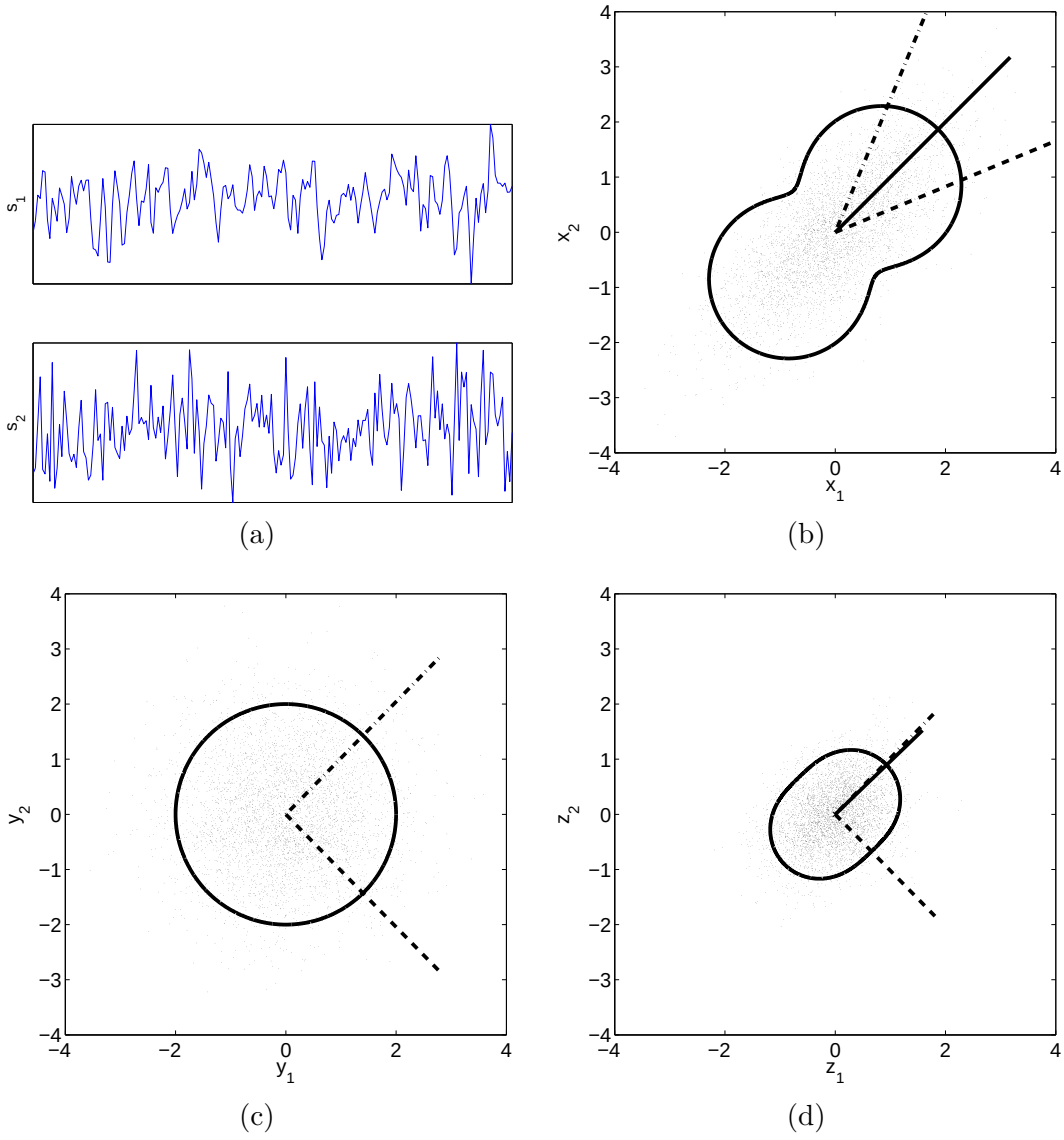


Figure 1: (a) Original sources, (b) scatter-plot of the mixtures, (c) whitened data $\mathbf{X}$ and (d) denoised data $\mathbf{Z} = \mathbf{XD}$. The dashed lines depict the mixing vectors and the solid lines the largest eigenvector. The curves denote the standard deviation of the projection of the data in different directions.

orthogonal. The first source has a somewhat slower temporal evolution and low-pass filtering retains more of that signal. This is evident in Fig. 1d which shows the denoised data and the first eigenvector.

In the following, we analyse the convergence and its speed in the linear DSS algorithms. Because of the equivalence between the DSS (using denoising $\mathbf{D}$) and PCA (using denoising $\mathbf{D}^*$), the properties of the linear DSS algorithm can be analysed via the power method. For a more comprehensive discussion on the power method, *cf.,* Wilkinson (1965).

The power method converges to the eigenvector corresponding to the principal eigenvalue provided that the largest two eigenvalues do not have identical values. Let us consider one iteration of the algorithm. The current source estimate $\mathbf{s}$ can be expressed as a weighted sum of the source estimates $\mathbf{s}_i^*$ at the fixed points:

$$\mathbf{s} = \sum_i c_i \mathbf{s}_i^* \,, \tag{26}$$

where c_i is the contribution of source estimate $\mathbf{s}_i^*$ in the current source estimate. Now rewriting the denoising step (10) for the weighted sum (26) yields:

$$\mathbf{s}^+ = \mathbf{f}_{\text{lin}} \left(\sum_i c_i \mathbf{s}_i^* \right) = \sum_i c_i \mathbf{f}_{\text{lin}}(\mathbf{s}_i^*) = \mathbf{s}_{\text{orth}} + \sum_i c_i \lambda_i \mathbf{s}_i^* \,, \tag{27}$$

where λ_i is the ith eigenvalue. This shows that after n iterations the relative contributions of the fixed points change from $\frac{c_i}{c_j}$ into $\frac{c_i \lambda_i^n}{c_j \lambda_j^n}$.

If there are two fixed points $\mathbf{s}_i^*$ and $\mathbf{s}_j^*$ that have identical eigenvalues $\lambda_i = \lambda_j$, the linear DSS cannot separate between the two. This means, for instance, that it is not possible to separate Gaussian sources that have identical autocovariance matrices, *i.e.*, $\mathbf{\Sigma}_{\mathbf{s}_i \mathbf{s}_i} = \mathbf{\Sigma}_{\mathbf{s}_j \mathbf{s}_j}$ or in other words sources whose time structures do not differ. Otherwise, as long as $c_i \neq 0$, the algorithm converges globally to the source with the largest eigenvalue.

The speed of convergence in the power method (hence in linear DSS) depends linearly on the log-ratio of the largest (absolute) eigenvalues $\log |\lambda_1|/|\lambda_2|$, where $|\lambda_1| \geq |\lambda_2| \geq |\lambda_i|$, $i = 3, \ldots, N$. Note that absolute values of the eigenvalues have been used. While the eigenvalues are usually positive, there are cases where negative eigenvalues may exist, for instance in the case of complex data or when using the so called *spectral shift*, which is discussed in Sec. 2.5.

2.3 Convergence analysis of the nonlinear DSS

In the previous section, it was shown that a Gaussian model for the sources leads to linear denoising and a DSS algorithm whose convergence is global and can be analysed via the eigenvalues. However, denoising is not restricted to linear operations. A non-Gaussian source model yields an E-step (3) which corresponds to nonlinear denoising. Moreover, the correspondence between the denoising and the underlying source model is less obvious. Median filtering (Kuosmanen and Astola, 1997) is an example of a denoising which is not easily derived from a generative model but which can be efficient in practice, both computationally and in terms of the quality of the result. The main benefit of the DSS framework is that it is possible to use many kinds of denoising functions $\mathbf{f}(\mathbf{s})$ and the user

can taylor the denoising for the problem at hand. However, not all denoising functions are equally good, and it is useful to understand how the denoising function affects the performance. In this section, we discuss the convergence properties of the nonlinear DSS algorithms.

The convergence of nonlinear DSS can be analysed by a similar approach as was used for the linear case (27):

$$\mathbf{s}^+ = \mathbf{f}\left(\sum_i c_i \mathbf{s}_i^*\right) = \mathbf{s}_{\text{orth}} + \sum_i c_i \lambda_i(\mathbf{s}) \mathbf{s}_i^* \,. \tag{28}$$

Note that we have defined a local eigenvalue $\lambda_i(\mathbf{s})$ that depends on the current source estimate, in contrast to the constant eigenvalue λ_i in the linear case. The above equation has not been "derived". Rather, it is the definition of the local eigenvalues which play a similar role in the convergence as the global eigenvalues in the linear case. In general, it holds that the eigenvalues are constant if and only if the denoising function is linear.

Source separation can be far more efficient in the nonlinear than in the linear case. If the ICA model holds and there is an infinite amount of data, the sources can usually be separated even in the linear case because minute differences in the eigenvalues of the sources are sufficient for separation. In practice, the separation is based on a finite number of samples and the ICA model only holds approximately. The separation quality is in general much worse if the "true" eigenvalues are close to each other.

In the nonlinear case, the eigenvalues depend on the source estimate. It is possible to view the nonlinear denoising as linear denoising which is constantly adapted to the source estimate. This means that different sources can have locally the largest eigenvalue. If the adaptation is consistent, *i.e.*, $\lambda_i(\mathbf{s})$ grows monotonically with c_i, all stable fixed points correspond to the original sources. In general, the separation quality is the best and convergence is fastest when $\lambda_i(\mathbf{s}_i^*)$ is very large compared to $\lambda_j(\mathbf{s}_j^*)$ with $j \neq i$.

Sometimes it may be sufficient to separate a signal subspace. Then it is enough for the denoising function to make the eigenvalues corresponding to this subspace large compared to the rest but the eigenvalues do not need to differ within the subspace. If full separation is required, it is important to make sure that the denoising not only removes noise (eigenvalues corresponding to noise are small) but other signals, too (eigenvalues corresponding to $\lambda_j(\mathbf{s}_j^*)$ are small).

2.4 Deflation

The classical power method has two common extensions: deflation and spectral shift. They are readily available for the linear DSS since it is equivalent to the power method applied to filtered data via Eq. (21). It is also relatively straightforward to apply them in the nonlinear case.

Linear DSS algorithms converge globally to the source whose eigenvalue has the largest magnitude. Nonlinear DSS algorithms may have several fixed points but even then it is useful to guarantee that the algorithm converges to a source estimate which has not been extracted yet. Deflational method is a procedure which allows one to estimate several sources by iteratively applying DSS algorithm several times. The convergence to previously

extracted sources is prevented by making their eigenvalues zero: $\mathbf{w}_{\text{orth}} = \mathbf{w} - \mathbf{A}\mathbf{A}^T\mathbf{w}$ (Luenberger, 1969), where $\mathbf{A}$ now contains the already estimated mixing vectors.

Note that in this deflation scheme, it is possible to use different kinds of denoising procedures when the sources differ in characteristics. This will be discussed in more detail in Sec. 3.2. Also, if more than one source is estimated simultaneously, the symmetric orthogonalisation methods proposed for symmetric FastICA (Hyvärinen, 1999) can be used.

2.5 Spectral shift

As discussed in Sec. 2.2, the matrix multiplication (22) in the power method does not promote the largest eigenvalue effectively compared to the second largest eigenvalue if they have comparable values. The convergence speed in such cases can be increased by so called spectral shift[2] (Wilkinson, 1965) which modifies the eigenvalues without changing the fixed points. At the fixed point of the linear DSS,

$$\lambda\mathbf{w}^* = \mathbf{X}\mathbf{f}^T(\mathbf{s}^*)/T\,. \tag{29}$$

If the denoising function is multiplied by a scalar, the convergence of the algorithm does not change in any way because the scaling will be overruled by the normalisation step (12). All eigenvalues will be scaled but their ratios, which are what count in convergence, are not affected.

Adding a multiple of $\mathbf{s}$ into $\mathbf{f}(\mathbf{s})$ does not affect the fixed points because $\mathbf{X}\mathbf{s}^T \propto \mathbf{w}$. This does affect the ratios of the eigenvalues and hence the convergence speed. In summary, $\mathbf{f}(\mathbf{s})$ can be replaced by

$$\alpha(\mathbf{s})[\mathbf{f}(\mathbf{s}) + \beta(\mathbf{s})\mathbf{s}]\,, \tag{30}$$

where $\alpha(\mathbf{s})$ and $\beta(\mathbf{s})$ are scalars. The multiplier $\alpha(\mathbf{s})$ is overruled by the normalisation step (12) and has no effect on the algorithm. The term $\beta(\mathbf{s})\mathbf{s}$ is turned into $T\beta(\mathbf{s})\mathbf{w}$ in the re-estimation step (9) and does affect the convergence speed but not the fixed points (however, it can turn a stable fixed point unstable or vice versa). This is because all eigenvalues are shifted by $\beta(\mathbf{s})$:

$$\mathbf{X}[\mathbf{f}(\mathbf{s}^*) + \beta(\mathbf{s}^*)\mathbf{s}^*]^T/T = \lambda\mathbf{w}^* + \beta(\mathbf{s}^*)\mathbf{w}^* = [\lambda + \beta(\mathbf{s}^*)]\mathbf{w}^* \tag{31}$$

This spectral shift modifies the ratios of the eigenvalues and the ratio of the two largest eigenvalues becomes[3] $|[\lambda_1 + \beta(\mathbf{s})]/[\lambda_2 + \beta(\mathbf{s})]| > |\lambda_1/\lambda_2|$, provided that $\beta(\mathbf{s})$ is negative but not much smaller than $-\lambda_2$. This can greatly accelerate convergence.

For very negative $\beta(\mathbf{s})$, some eigenvalues will become negative. In fact, if $\beta(\mathbf{s})$ is small enough, the absolute value of the originally smallest eigenvalue will exceed that of the originally largest eigenvalue. Iterations of linear DSS will then minimise the eigenvalue rather than maximise it.

We suggest that it is often reasonable to shift the eigenvalue corresponding to Gaussian signal $\boldsymbol{\nu}$ to zero. Some eigenvalues may then become negative and the algorithms can converge to fixed points corresponding to these eigenvalues rather than the positive ones.

2. The set of the eigenvalues is often called eigenvalue spectrum.
3. Since the denoising operation presumably preserves some of the signal and noise, it is reasonable to assume that all eigenvalues are originally positive.

In many cases, this is perfectly acceptable because, as will be further discussed in Sec. 3.3, any deviation from Gaussian eigenvalue is indicative of signal. A side effect of a negative eigenvalue is that the estimate $\mathbf{w}$ changes its sign at each iteration. This is not a problem but needs to be kept in mind when determining the convergence.

Since the convergence of the nonlinear DSS is governed by local eigenvalues, the spectral shift needs to be adapted to the changing local eigenvalues to achieve optimal convergence speed. In practice the eigenvalue of a Gaussian signal can be estimated by linearising $\mathbf{f}(\mathbf{s})$ around the current source estimate $\mathbf{s}$:

$$\mathbf{f}(\mathbf{s}+\Delta\mathbf{s}) \approx \mathbf{f}(\mathbf{s}) + \Delta\mathbf{s}\mathbf{J}(\mathbf{s}) \tag{32}$$

$$\lambda_\nu(\mathbf{s}) \approx \frac{\mathbf{f}(\mathbf{s}+\epsilon\boldsymbol{\nu}) - \mathbf{f}(\mathbf{s})}{\epsilon}\boldsymbol{\nu}^T/T \approx \frac{\epsilon\boldsymbol{\nu}\mathbf{J}(\mathbf{s})}{\epsilon}\boldsymbol{\nu}^T/T = \boldsymbol{\nu}\mathbf{J}(\mathbf{s})\boldsymbol{\nu}^T/T \tag{33}$$

$$\beta(\mathbf{s}) = E[-\lambda_\nu(\mathbf{s})] \approx -\operatorname{tr}\mathbf{J}(\mathbf{s})/T \tag{34}$$

The last step follows from the fact that the elements of $\boldsymbol{\nu}$ are mutually uncorrelated and have zero mean and unit variance. Here $\mathbf{J}(\mathbf{s})$ denotes the Jacobian matrix of $\mathbf{f}(\mathbf{s})$ computed at $\mathbf{s}$. For linear denoising $\mathbf{J}(\mathbf{s}) = \mathbf{D}$ and hence β does not depend on $\mathbf{s}$. If denoising is instantaneous, *i.e.*, $\mathbf{f}(\mathbf{s}) = [f_1(s(1))\, f_2(s(2)) \ldots]$, the shift can be written as $\beta(\mathbf{s}) = -\sum_t f_t'(s(t))/T$. This is the spectral shift used in FastICA (Hyvärinen, 1999), but it has been justified as an approximation to Newton's method and our analysis thus provides a novel interpretation.

Sometimes the spectral shift turns out to be either too modest or too strong, leading to slow convergence or lack of convergence, respectively. For this reason, we suggest a simple stabilisation rule: instead of updating $\mathbf{w}$ into $\mathbf{w}_{\text{new}}$ defined by (12), it is updated into

$$\mathbf{w}_{\text{adapted}} = \operatorname{orth}(\mathbf{w} + \gamma\Delta\mathbf{w}) \tag{35}$$

$$\Delta\mathbf{w} = \mathbf{w}_{\text{new}} - \mathbf{w}\,, \tag{36}$$

where γ is the step size and the orthogonalisation has been added in case several sources are to be extracted. Originally $\gamma = 1$, but if the consequtive steps are taken in nearly opposite directions, *i.e.*, the angle between $\Delta\mathbf{w}$ and $\Delta\mathbf{w}_{\text{old}}$ is greater than 179°, then $\gamma = 0.5$ for the rest of the iterations. A stabilised version of FastICA has been proposed by Hyvärinen (1999) as well and procedure similar to the one above has been used. The different speedup techniques considered above, and some additional ones, are studied further by Valpola and Särelä (2004).

Sometimes there are several signals with similar large eigenvalues. It may then be impossible to use spectral shift to accelerate their separation significantly because of small eigenvalues that would assume very negative values exceeding the signal eigenvalues in magnitude. In that case, it may be beneficial to first separate the subspace of the signals with large eigenvalues from the smaller ones. Spectral shift will then be useful in the signal subspace.

3. Approximation for the objective function

The virtue of DSS framework is that it allows one to develop procedural source separation algorithms without referring to an exact objective function or a generative model. However,

in many cases an approximation of the objective function is nevertheless useful. In this section, we propose such an approximation (Sec. 3.1) and discuss its uses, including monitoring (Sec. 3.2) and acceleration of convergence (Sec. 3.3) as well as analysis of separation results (Sec. 3.4).

3.1 Derivation of the approximation

As shown in Sec. 2.2, Eq. (24), the objective function corresponding to linear denoising $\mathbf{f}(\mathbf{s}) = \mathbf{s}D$ is $g(\mathbf{s}) = \mathbf{s}D\mathbf{s}^T$, given that D is a symmetric matrix. This can be written as $g(\mathbf{s}) = \mathbf{s}\,\mathbf{f}_{\text{lin}}^T(\mathbf{s})$. This formula is exact for linear DSS and we propose it as an approximation $\hat{g}$ for the objective function for nonlinear DSS as well:

$$\hat{g}(\mathbf{s}) = \mathbf{s}\,\mathbf{f}^T(\mathbf{s})\,. \tag{37}$$

There is, however, an important caveat to be made. Note that Eq. (30) includes the scalar functions $\alpha(\mathbf{s})$ and $\beta(\mathbf{s})$. This means that functionally equivalent DSS algorithms can be implemented with slightly different denoising functions $\mathbf{f}(\mathbf{s})$ and while they would converge exactly to the same results, the approximation (37) might yield completely different values. In fact, by tuning $\alpha(\mathbf{s})$, $\beta(\mathbf{s})$ or both, the approximation $\hat{g}(\mathbf{s})$ could be made to yield any desired function $h(\mathbf{s})$ which need not have any correspondance to the true $g(\mathbf{s})$.

Due to $\alpha(\mathbf{s})$ and $\beta(\mathbf{s})$, it seems virtually impossible to write down a simple approximation of $g(\mathbf{s})$ that could not go wrong with a malevolent choice of $\mathbf{f}(\mathbf{s})$. In the following, however, we argue that Eq. (37) is in most cases a good approximation and it is usually easy to check whether it behaves as desired—yields values which are monotonic in SNR. If it does not, $\alpha(s)$ and $\beta(s)$ can be easily tuned to correct this.

Let us first check what would be the DSS algorithm maximising $\hat{g}(\mathbf{s})$. Obviously, the approximation is good if the algorithm turns out to use a denoising similar to $\mathbf{f}(\mathbf{s})$. The following Lagrange equation holds at the optimum:

$$\nabla_{\mathbf{w}}[\hat{g}(\mathbf{s}) - \boldsymbol{\xi}^T\mathbf{h}(\mathbf{w})] = 0, \tag{38}$$

where $\mathbf{h}$ denotes the constraints under which the optimisation is performed and $\boldsymbol{\xi}$ are the corresponding Lagrange multipliers. In this case, we are constraint to unit scale projections, *i.e.*, $h(\mathbf{w}) = \mathbf{w}^T\mathbf{w} - 1 = 0$, and it thus follows

$$\mathbf{X}\nabla_{\mathbf{s}}\hat{g}^T(\mathbf{s}) - 2\xi\mathbf{w} = 0. \tag{39}$$

Substituting 2ξ with the appropriate normalising factor which guarantees $|\mathbf{w}| = 1$ results in the following fixed point:

$$\mathbf{w} = \frac{\mathbf{X}\nabla_{\mathbf{s}}\hat{g}^T(\mathbf{s})}{||\mathbf{X}\nabla_{\mathbf{s}}\hat{g}^T(\mathbf{s})||}. \tag{40}$$

Substituting $\mathbf{s} = \mathbf{w}^T\mathbf{X}$ yields

$$\mathbf{w}^+ = \mathbf{X}[\mathbf{f}^T(\mathbf{s}) + \mathbf{J}^T(\mathbf{s})\mathbf{s}^T]\,, \tag{41}$$

where $\mathbf{J}$ is the Jacobian of $\mathbf{f}$. This should conform with the corresponding steps (10) and (11) in the nonlinear DSS which uses $\mathbf{f}(\mathbf{s})$ for denoising. This is true if the two terms in the square brackets have the same form, *i.e.*, $\mathbf{f}(\mathbf{s}) \propto \mathbf{s}\,\mathbf{J}(\mathbf{s})$.

As expected, in the linear case the two algorithms are exactly the same because the Jacobian is a constant matrix and $\mathbf{f}(\mathbf{s}) = \mathbf{s}\mathbf{J}$. The denoised sources are also proportional to $\mathbf{s}\,\mathbf{J}(\mathbf{s})$ in some special nonlinear cases, for instance, when $\mathbf{f}(\mathbf{s}) = \mathbf{s}^n$.

3.2 Negentropy ordering

The approximation (37) can be readily used for monitoring the convergence of DSS algorithms. It is also easy to use it for ordering the sources based on their SNR if several sources are estimated using DSS with the same $\mathbf{f}(\mathbf{s})$. However, simple ordering based on Eq. (37) is not possible if different denoising functions are used for different sources.

In these cases it is useful to order the source estimates by their negentropy which is a normalised measure of structure in the signal. Differential entropy H of a random variable is a measure of disorder and is dependent on the variance of the variable. Negentropy is a normalised quantity measuring the difference between the differential entropy of the component and a Gaussian component with the same variance. Negentropy is zero for the Gaussian distribution and non-negative for all distributions since among the distributions with a given variance, the Gaussian distribution has the highest entropy.

Calculation of the differential entropy assumes the distribution to be known. Usually this is not the case and the estimation of the distributions is often difficult and computationally demanding. Following Hyvärinen (1998), we approximate the negentropy $N(\mathbf{s})$ by

$$N(\mathbf{s}) = H(\boldsymbol{\nu}) - H(\mathbf{s}) \approx \eta_g[\hat{g}(\mathbf{s}) - \hat{g}(\boldsymbol{\nu})]^2\,, \tag{42}$$

where $\boldsymbol{\nu}$ is a normally distributed variable. The reasoning behind Eq. (42) is that $\hat{g}(\mathbf{s})$ carries information about the distribution of $\mathbf{s}$. If $\hat{g}(\mathbf{s})$ equals $\hat{g}(\boldsymbol{\nu})$, there is no evidence of the negentropy to be greater than zero, so this is when $N(\mathbf{s})$ should be minimised. A Taylor series expansion of $N(\mathbf{s})$ w.r.t. $\hat{g}(\mathbf{s})$ around $\hat{g}(\boldsymbol{\nu})$ yields the approximation (42) as the first non-zero term.

Comparison of signals extracted with different optimisation criteria presumes that the weighting constants η_g are known. We propose that η_g can be calibrated by generating a signal with a known, nonzero negentropy. Negentropy ordering is most useful for signals which have a relatively poor SNR—the signals with a good SNR will most likely be selected in any case. Therefore we choose our calibration signal to have SNR of 0 dB, *i.e.*, it contains equal amounts of signal and noise in terms of energy: $\mathbf{s}_s = (\boldsymbol{\nu} + \mathbf{s}_{\text{opt}})/\sqrt{2}$, where $\mathbf{s}_{\text{opt}}$ is a pure signal having no noise. It obeys fully the signal model implicitly defined by the corresponding denoising function $\mathbf{f}$. Since $\mathbf{s}_{\text{opt}}$ and $\boldsymbol{\nu}$ are uncorrelated, $\mathbf{s}_s$ has unit variance. The entropy of $\boldsymbol{\nu}/\sqrt{2}$ is

$$H(\boldsymbol{\nu}/\sqrt{2}) = H(\boldsymbol{\nu}) + \log 1/\sqrt{2} = H(\boldsymbol{\nu}) - 1/2 \log 2\,. \tag{43}$$

Since the entropy can only increase by adding a second, independent signal $\mathbf{s}_{\text{opt}}$, $H(\mathbf{s}_{\mathbf{s}}) \geq H(\boldsymbol{\nu}) - 1/2 \log 2$. It thus holds $N(\mathbf{s}_s) = H(\boldsymbol{\nu}) - H(\mathbf{s}_s) \leq 1/2 \log 2$. One can usually expect that $\mathbf{s}_{\text{opt}}$ has a lot of structure, *i.e.*, its entropy is low. Then its addition to $\boldsymbol{\nu}/\sqrt{2}$ does not significantly increase the entropy. It is therefore often reasonable to approximate

$$N(\mathbf{s}_s) \approx 1/2 \log 2 = 1/2 \,\text{bit}, \tag{44}$$

where we chose base-2 logarithm yielding bits. Depending on $\mathbf{s}_{\text{opt}}$, it may also be possible to compute the negentropy of $N(\mathbf{s}_s)$ exactly. This can then be used instead of the approximation (44).

The coefficients η_g in Eq. (42) can now be solved by requiring that the approximation (42) yields Eq. (44) for $\mathbf{s}_s$. This results in

$$\eta_g = \frac{1}{2(\hat{g}(\mathbf{s}_s) - \hat{g}(\boldsymbol{\nu}))^2}\text{bit} \tag{45}$$

and finally, substitution of the approximation of the objective function (37) and Eq. (45) into Eq. (42) yields the calibrated approximation of the negentropy:

$$N(\mathbf{s}) \approx \frac{\left[\mathbf{s}\,\mathbf{f}^T(\mathbf{s}) - \boldsymbol{\nu}\,\mathbf{f}^T(\boldsymbol{\nu})\right]^2}{2\left[\mathbf{s}_s\,\mathbf{f}^T(\mathbf{s}_s) - \boldsymbol{\nu}\,\mathbf{f}^T(\boldsymbol{\nu})\right]^2}\text{ bit.} \tag{46}$$

3.3 Spectral shift revisited

In Sec. 2.5, we suggested that a reasonable spectral shift is to move the eigenvalue corresponding to a Gaussian signal $\boldsymbol{\nu}$ to zero. This leads to minimising $g(\mathbf{s})$, when the largest absolute eigenvalue is negative. It does not seem very useful to minimise $g(\mathbf{s})$, a function that measures the SNR of the sources, but as we saw with negentropy and its approximation (42), values $g(\mathbf{s}) < g(\boldsymbol{\nu})$ are, in fact, indicative of signal. A reasonable selection for β is thus $-\lambda_\nu$ given by (34) which leads linear DSS to extremise $g(\mathbf{s}) - g(\boldsymbol{\nu})$ or, equivalently, to maximise the negentropy approximation (42).

A well known example where the spectral shift by the eigenvalue of a Gaussian signal is useful is the mixture of both super- and sub-Gaussian distributions. DSS algorithm designed for super-Gaussian distributions would lead to $\lambda > \lambda_\nu$ for super-Gaussian and $\lambda < \lambda_\nu$ for sub-Gaussian distributions, λ_ν being the eigenvalue of the Gaussian signal. By shifting the eigenvalue spectrum by $-\lambda_\nu$, the most non-Gaussian distributions will result in the largest absolute eigenvalues regardless of whether the distribution is super- or sub-Gaussian. By using the spectral shift it is therefore possible to extract both super- and sub-Gaussian distributions with a denoising scheme which is designed for one type of distributions only.

Consider for instance $\mathbf{f}(\mathbf{s}) = \tanh\mathbf{s}$ which can be used as denoising for sub-Gaussian signal while, as will be further discussed in Sec. 4.2.3, $\mathbf{s} - \tanh\mathbf{s} = -(\tanh\mathbf{s} - \mathbf{s})$ is a suitable denoising for super-Gaussian signals. This shows that depending on the choice of β, DSS can find either sub-Gaussian ($\beta = 0$) or super-Gaussian ($\beta = -1$) sources. With the FastICA spectral shift (34), β will always lie in the range $-1 < \beta \leq \tanh^2 1 - 1 \approx -0.42$. In general, β will be closer to -1 for super-Gaussian sources which shows that FastICA is able to adapt its spectral shift to the source distribution.

3.4 Detection of overfitting

In exploratory data analysis DSS is very useful for giving a better insight to the data using a linear factor model. However, it is possible that DSS extracts structures that are not actually present in the data but are generated by the denoising function, *i.e.*, the results may be due to overfitting.

Overfitting in ICA has been extensively studied by Särelä and Vigário (2003). It was observed that it typically results in signals that are mostly inactive, except for a single

spike. In DSS the outlook of the overfitted results depends on the denoising criterion. The results of exploratory DSS should thus be treated with a healthy amount of scepticism.

To detect an overfitted result, one should know how it looks like. As a first approximation, DSS can be performed with same amount of i.i.d. Gaussian data. Then all the results present cases of overfitting. Even better characterisation of the overfitting results can be obtained by mimicking the actual data characteristics as well as possible. In that case it is important to make sure that the structure assumed by the signal model has been broken. Both the Gaussian overfitting test and the more advanced test are used throughout the experiments in Secs. 5.2–5.3.

Note that in addition to visual test, the methods described above provide us with a quantitative measure as well. Using the negentropy approximation (46), we can set a threshold under which the sources are very likely overfits and do not carry much real structure. In the simple case of linear DSS, the negentropy can be approximated easily using the corresponding eigenvalue.

4. Denoising functions in practice

DSS is a framework for designing source separation algorithms. The idea is that the algorithms differ mainly in the denoising function $\mathbf{f}(\mathbf{s})$ while the other parts of the algorithm remain mostly the same. Denoising is useful as such and therefore there is a wide literature of sophisticated denoising methods to choose from (*cf.,* Anderson and Moore, 1979). Moreover, one usually has some knowledge about the signals of interest and thus possesses the information needed for denoising. In fact, quite often the signals extracted by BSS techniques would be post-processed to reduce noise in any case (*cf.,* Vigneron et al., 2003). In the DSS framework, the available denoising methods can be directly applied to source separation, producing better results than purely blind techniques. There are also very general noise reduction techniques such as wavelet denoising (Donoho et al., 1995, Vetterli and Kovacevic, 1995) or median filtering (Kuosmanen and Astola, 1997) which can be applied in exploratory data analysis.

In this section, we discuss denoising functions ranging from simple but powerful linear ones to sophisticated nonlinear ones with the goal of inspiring others to try out their own denoising methods. The range of applicability of the examples spans from cases where the knowledge about the signals is relatively specific to almost blind source separation where very little is assumed about the signal characteristics. Many of the denoising functions discussed in this section are applied in experiments in Section 5.

Before proceeding to examples of denoising functions, we note that it is usually not crucial for the denoising to be very exact. Otherwise DSS would not be very useful because one would only get what is asked from the algorithm in terms of the denoising function. Fortunately, this is not the case: Assuming that the signals are recoverable by linear projections from the observations, it is enough for the denoising function $\mathbf{f}(\mathbf{s})$ to remove more noise than signal (*cf.,* Hyvärinen et al., 2001b, Theorem 8.1). This is because the re-estimation steps (11) and (12) constrain the source $\mathbf{s}$ to the subspace spanned by the data. Even if the denoising discards parts of the signal or creates nonexistent signals, re-estimation steps restore them.

In practice, the observations contain noise which does not fully disappear by any linear projection and then the quality of the separated signals depends on the accuracy of denoising. If there is no detailed knowledge about characteristics of the signals to start with, it is useful to bootstrap the denoising functions. This can be achieved by starting with relatively general signal characteristics and then tuning the denoising functions based on analyses of the structure in the noisy signals extracted in the first phase. In fact, some of the nonlinear DSS algorithms can be regarded as linear DSS algorithms where a linear denoising function is adapted to the sources, leading to nonlinear denoising.

4.1 Detailed linear denoising functions

In this section we consider several detailed, simple, but powerful, linear denoising schemes. We introduce the denoisings using the denoising matrix, $\mathbf{D}$ when feasible. We consider effective implementation of the denoisings as well.

The eigenvalue decomposition (18) shows that any denoising in linear DSS can be implemented as an orthonormal rotation followed by a point-wise adjustment of the samples and rotation to the original space. The eigenvalue decomposition of the denoising matrix $\mathbf{D}$ often offers a good intuition to the denoising function as well as a practical means of its implementation.

4.1.1 On/off-denoising

Consider designed experiments, *e.g.*, in fields of psychophysics or biomedicine. It is usual to control them by having periods of activity and non-activity. In such experiments the denoising can be simply implemented by

$$\mathbf{D} = \mathrm{diag}(\mathbf{b})\,, \tag{47}$$

where $\mathbf{D}$ refers to the linear denoising matrix in Eq. (10) and

$$\mathbf{b} = \begin{cases} 1, \text{for the active parts} \\ 0, \text{for the inactive parts} \end{cases} \tag{48}$$

This amounts to multiplying the source estimate $\mathbf{s}$ by a binary mask[4], where ones represent the active parts and zeroes the non-active parts. Notice that this masking procedure actually satisfies $\mathbf{D} = \mathbf{D}\mathbf{D}^T$. This means that DSS is equivalent to the PCA applied to denoised $\mathbf{Z} = \mathbf{X}\mathbf{D}$ even with exactly the same filtering. In practice this DSS algorithm could be implemented by PCA applied to the active parts of the data while the sphering stage would still involve the whole data.

4.1.2 Denoising based on the frequency content

If, on the other hand, signals are characterised by having certain frequency components, one can transform the source estimate by DCT, mask the spectrum, *e.g.*, with a binary mask, and inverse transform to obtain the denoised signal:

$$\mathbf{D} = \mathbf{V}\mathbf{\Lambda}_D\mathbf{V}^T\,, \tag{49}$$

4. By masking we refer to point-wise multiplication of a signal or a transformation of a signal.

where $\mathbf{V}$ is the transform, $\mathbf{\Lambda}_D$ is the matrix with the mask on its diagonal, and $\mathbf{V}^T$ is the inverse transform. The transform $\mathbf{V}$ can be implemented for example with the Fourier transform[5] or discrete cosine transform (DCT). After the transform, the signal is filtered using the diagonal matrix $\mathbf{\Lambda}$, *i.e.*, by a point-wise adjustment of the frequency bins. Finally the signal is inverse transformed using $\mathbf{V}^T$. In the case of linear time-invariant (LTI) filtering, the denoising characteristics are manifested only in the diagonal matrix, while the transforming matrix $\mathbf{V}$ portrays a constant rotation. When this is the case, the algorithm can be further simplified by imposing the transformation on the sphered data, $\mathbf{X}$. Then the iteration can be performed in the transformed basis. This trick has been exploited in the first experiment of Sec. 5.2.

4.1.3 Spectrogram denoising

Often a signal is well characterised by what frequencies occur at what times. This is evident, *e.g.*, in oscillatory activity in the brain where oscillations often occur in bursts. An example of source separation in such data is studied in Sec. 5.2. The time-frequency behaviour can be described by calculating discrete cosine transform (DCT) in short windows in time. This results in a combined time and frequency representation, spectrogram, where the masking can be applied.

There is a known dilemma in the calculation of the spectrogram: detailed description of the frequency content does not allow detailed information of the activity in time and vice versa. In other words, large amount of different frequency bins T_f will result in small amount of time locations T_t. Wavelet transforms (Donoho et al., 1995, Vetterli and Kovacevic, 1995) have been suggested to overcome this problem. There an adaptive or predefined basis, different from the pure sinusoids used in Fourier transform or DCT, is used to divide the resources of time and frequency behaviour optimally in some sense. Another possibility is to use so called multitaper technique Percival and Walden (1993, Ch. 7).

Here we apply an overcomplete-basis approach related to the above methods. Instead of having just one spectrogram, we use several time-frequency analyses with different T_t's and T_f's. Then the new estimate of the projection $\mathbf{w}^+$ is achieved by summing the new estimates $\mathbf{w}_i^+$ of each of the time-frequency analyses: $\mathbf{w}^+ = \sum_i \mathbf{w}_i^+$.

4.1.4 Denoising of quasiperiodic signals

As a final example of denoising based on detailed source characteristics, consider Fig. 2a. There a source estimate $\mathbf{s}$ has been reached. The apparent quasiperiodic structure of the signal can be used to perform DSS to get a better estimate. The denoising proceeds as follows:

1. Estimate the locations of the peaks of the current source estimate $\mathbf{s}$ (Fig. 2b).
2. Chop each period from peak to peak.
3. Dilate each period to a fixed length L (linearly or nonlinearly).

5. Note that the eigenvalue decomposition contains real rotations instead of complex, but Fourier transform is usually seen as a complex tranformation. To keep the theory simple, we consider real Fourier transform where the corresponding sine and cosine terms have been separated in different elements.

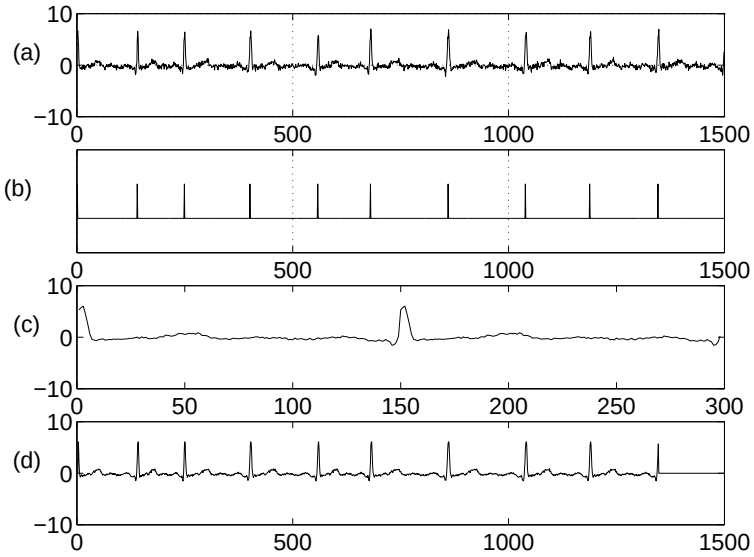


Figure 2: *a) Current source estimate* $\mathbf{s}$ *of a quasiperiodic signal b) Peak estimates c) Average signal* s_{ave} *d) Denoised source estimate* $\mathbf{s}^+$.

4. Average the dilated periods (Fig. 2c).

5. Let the denoised source estimate $\mathbf{s}^+$ be a signal where each period has been replaced by the averaged period dilated back into the original length (Fig. 2d).

The denoised signal $\mathbf{s}^+$ in Fig 2d show significantly better SNR compared to the original source estimate $\mathbf{s}$, in Fig. 2a.

This averaging is a form of linear denoising since it can be implemented as matrix multiplication. Furthermore, it presents another case in addition to the binary masking, where DSS is equivalent to the power method even with exactly the same filtering. It would not be easy to see from the denoising matrix $\mathbf{D}$ itself that $\mathbf{D} = \mathbf{D}\mathbf{D}^T$. However, this becomes evident should one consider the averaging of source estimate $\mathbf{s}^+$ (Fig. 2d) that is already averaged.

Note that there are cases where chopping from peak to peak does not guarantee the best result. This is especially true when the periods do not span the whole section from peak to peak, but there are parts where the response is silent. Then there is a need to estimate the lengths of the periods separately.

4.2 Denoising based on estimated signal variance

In the previous section, several denoising schemes were introduced. In all of them, the details of the denoising were assumed to be known. It is as well possible to estimate the denoising specifications from the data. This makes the denoising nonlinear or adaptive. In this section we consider a particular ICA algorithm in the DSS framework, suggesting modifications which improve separation results and robustness.

4.2.1 Kurtosis based ICA

Consider one of the best known BSS approaches, ICA by optimisation of the sample kurtosis of the sources. The objective function is then $g(\mathbf{s}) = \sum s^4(t)/T - 3\left(\sum s^2(t)/T\right)^2$. Since the source variance has been fixed to unity, we can simply use $g(\mathbf{s}) = \sum s^4(t)/T$ and derive the function $\mathbf{f}(\mathbf{s})$ from gradient ascend. This yields $\nabla_{\mathbf{s}} g(\mathbf{s}) = 4/T\,\mathbf{s}^3$, where $\mathbf{s}^3 = [s^3(1)\, s^3(2)\, \ldots]$. Selecting $\alpha(\mathbf{s}) = T/4$ and $\beta(\mathbf{s}) = 0$ in Eq. (30) then result in

$$\mathbf{f}(\mathbf{s}) = \mathbf{s}^3\,. \tag{50}$$

This implements an ICA algorithm with nonlinear denoising. So far, we have not referred to denoising, but a closer examination of Eq. (50) reveals that one can, in fact, interpret $\mathbf{s}^3$ as being $\mathbf{s}$ masked by $\mathbf{s}^2$, the latter being a somewhat naïve estimate of signal variance and thus relating to SNR.

Kurtosis as an objective function is notorious for being prone to overfitting and producing very spiky source estimates (Särelä and Vigário, 2003, Hyvärinen, 1998). For illustration of this consider Fig. 3. There one iteration of DSS using kurtosis based denoising is shown. Assume that via some means source estimate shown in Fig. 3a has been reached. The source seems to contain increased activity in three portions (around time instances 1000, 2300 and 6000). It as well contains a peak roughly at time instance 4700. The signal variance estimate, *i.e.*, the mask is shown in Fig. 3b. While it has boosted somewhat the broad activity compared to the silent parts, the magnification of the peak is far greater. Thus the denoised source estimate $\mathbf{s}^+$ (Fig 3c) has nearly nothing else than the peak. The new source estimate $\mathbf{s}_{\text{new}}$, based on the new projection $\mathbf{w}_{\text{new}}$, is a clear spike having little left of the broad activity.

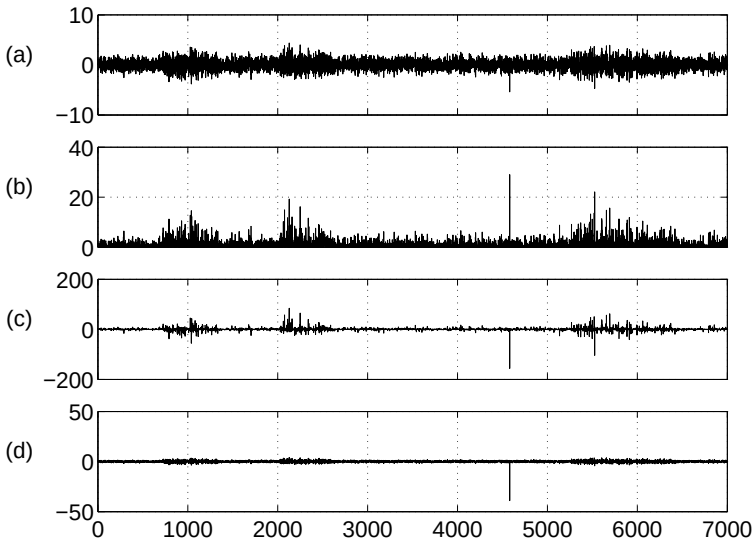


Figure 3: *a) Source estimate* $\mathbf{s}$ *b) Mask* $s^2(t)$ *c) Denoised source estimate* $\mathbf{s}^+ = \mathbf{f}(\mathbf{s}) = \mathbf{s}^3$ *d) Source estimate corresponding to the re-estimated* $\mathbf{w}_{\text{new}}$.

The denoising interpretation suggests that the failure to extract the broad activity is due to a poor estimate of SNR.

4.2.2 Better estimate for the signal variance

Let us now consider a related but better founded estimate. Assuming that $\mathbf{s}$ is composed of Gaussian noise with a constant variance σ_n^2 and Gaussian signal with non-stationary variance $\sigma_s^2(t)$, the maximum-a-posteriori (MAP) estimate of the signal is

$$s^+(t) = s(t)\frac{\sigma_s^2(t)}{\sigma_{\text{tot}}^2(t)}\,, \tag{51}$$

where $\sigma_{\text{tot}}^2(t) = \sigma_s^2(t) + \sigma_n^2$ is the total variance of the observation.

The kurtosis based DSS (50) can be obtained from this MAP estimate if the signal variance is assumed to be far smaller than the total variance. In that case it is reasonable to assume σ_{tot}^2 to be constant and $\sigma_s^2(t)$ can be estimated by $s^2(t) - \sigma_n^2$. Subtraction of σ_n^2 does not affect the fixed points as it can be embedded in the term $\beta(\mathbf{s}) = -\sigma_n^2$ in Eq. (30). Likewise, division by $\sigma_{\text{tot}}^2(t)$ is absorbed by $\alpha(\mathbf{s})$.

Comparison of Eq. (51) and Eq. (50) immediately suggests improvements to the kurtosis based DSS. For instance, it is clear that if $s^2(t)$ is large enough, it is not reasonable to assume that $\sigma_s^2(t)$ is small compared to $\sigma_n^2(t)$. Instead, the mask should saturate for large $s^2(t)$. This already improves robustness against outliers and alleviates the tendency to produce spiky source estimates.

We suggest the following improvements over kurtosis based denoising function (50):

1. The estimates of signal variance and total variance are based on several observations. The rationale of smoothing is the assumption of smoothness of the signal variance. In practice this can be achieved by low-pass filtering the time, frequency or time-frequency description of $s^2(t)$ yielding the approximation of total variance.

2. The noise variance is likewise estimated from data. It should be some kind of soft minimum of the estimated total variances because the estimate can be expected to have random fluctuations. We suggest the following formula:

   $$\sigma_n^2 = C\left(\exp\left\{E\left[\log\left(\sigma_{\text{tot}}^2(t) + \sigma_n^2\right)\right]\right\} - \sigma_n^2\right)\,. \tag{52}$$

   The noise variance σ_n^2 appears on both sides of the equation, but at the right-hand side, it appears only to prevent rare small values of σ_{tot}^2 from spoiling the estimate. Hence, we used the previously estimated value on the right-hand side. The constant C is tuned such that the formula gives a consistent estimate of the noise variance if the source estimate is, in fact, nothing but Gaussian noise.

3. The signal variance should be close to the estimate of the total variance minus the estimate of the noise variance. Since a variance cannot be negative and the estimate of the total variance has fluctuations, we use a formula which yields zero only when the total variance is zero but which asymptotically approaches $\sigma_{\text{tot}}^2(t) - \sigma_n^2$ for large values of the total variance:

   $$\sigma_s^2(t) = \sqrt{\sigma_{\text{tot}}^4(t) + \sigma_n^4} - \sigma_n^2\,. \tag{53}$$

As an illustration of these improvements consider Fig. 4 where one iteration of DSS using the MAP estimate is shown. The first two subplots (Fig. 4a and b) are identical to the ones using kurtosis based denoising. In Fig. 4c, the variance estimate is smoothed using low-pass filtering. Note that the broad activity has been magnified when compared to the spike around time instance 4700. The noise level σ_n^2, calculated using Eq. (52), is shown in dashed line. Corresponding masking (Fig. 4d) results in a denoised source estimate using Eq. (51), shown in Fig. 4e. Finally, the new source estimate $\mathbf{s}_{\text{new}}$ is shown after five iterations of DSS in Fig. 4f. DSS using the MAP-based denoising has clearly removed a considerable amount of background noise as well as the lonely spike.

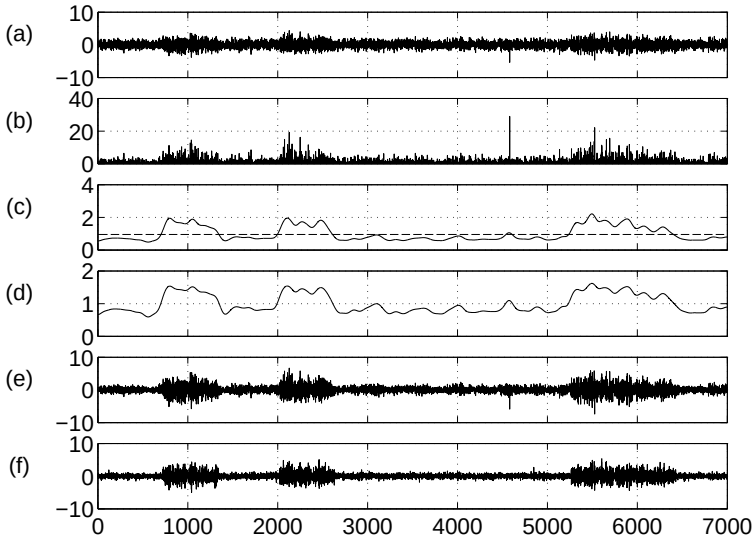


Figure 4: *a) Source estimate* $\mathbf{s}$ *b)* $s^2(t)$*c) Smoothed total variance with the noise level in dashed line d) Denoising mask e) Denoised source estimate* $\mathbf{s}^+$ *f) Source estimate after five iterations of DSS.*

The exact details of these improvements are not crucial, but we wanted to show that the denoising interpretation of Eq. (50) can carry us quite far. The above estimates plugged into Eq. (51) yield a DSS algorithm which is far more robust against overfitting, does not produce the spiky signal estimates and in general yields signals with better SNRs than kurtosis.

Despite the merits of the DSS algorithm described above, there is still one problem with it. While the extracted signals have excellent SNR, they do not necessarily correspond to independent sources, *i.e.*, the sources may remain mixed. This is because there is nothing in the denoising which could discard other sources. Using the notation in (28), $\lambda_i(\mathbf{s}_i^*)$ is much larger than λ_ν as it should but $\lambda_j(\mathbf{s}_i^*)$ are large, too, which means that the iterations do not remove the contribution of the weaker sources efficiently.

Assume, for instance, that two sources have clear-cut and non-overlapping times of strong activity ($\sigma_s^2(t) \gg 0$) and remain silent for most of the time ($\sigma_s^2(t) = 0$). Suppose that one source is present for some time at the beginning of the data and another at the

end. If the current source estimate is a mixture of both, the mask will have values close to one at the beginning and at the end of the signal. Denoising can thus clean the noise from the signal estimate, but it cannot decide between the two sources.

In this respect, kurtosis actually works better than DSS based on the above improvements. This is because the mask never saturates and small differences in the strengths of the relative contributions of two original sources in the current source estimate will be amplified. The problem only occurs in the saturated regime of the mask and we therefore suggest a simple modification of the MAP estimate (51):

$$\mathbf{f}_t(\mathbf{s}) = s(t)\frac{\sigma_s^{2\mu}(t)}{\sigma_{\mathrm{tot}}^2(t)}\,, \tag{54}$$

where μ is a constant slightly greater or equal to one. Note that this modification is usually needed at the beginning of the iterations only. Once the source estimate is dominated by one of the original sources and the contributions of the other sources fall closer to the noise level, the values of the mask are smaller for the other original sources possibly still present in the estimated source.

Another approach is based on the finding that the orthogonalisation of the mixing vectors $\mathbf{A}$ cancels only the linear correlation between different sources. Higher-order correlations may still exist. For instance, the variances of different sources can be correlated. Schwartz and Simoncelli (2001) have suggested that variances may be decorrelated by a divisive procedure, in contrast to the orthogonalisation of $\mathbf{A}$, a subtractive procedure. Then it is necessary to estimate explicitly the correlation between one source and the other sources in the current source estimate. The estimate of the total variance can be based on $\sigma_{\mathrm{tot}}^2(t) = \sigma_s^2(t) + \sigma_n^2 + \sigma_{\mathrm{others}}^2(t)$, where $\sigma_{\mathrm{others}}^2(t)$ stands for the estimate of total leakage of variance from the other sources. This approach has been further pursued by Valpola and Särelä (2004).

The problems related to kurtosis are well known and several other improved nonlinear functions $\mathbf{f}(\mathbf{s})$ have been proposed. However, some aspects of the above denoising, especially smoothing of the total-variance estimate $s^2(t)$, have not been suggested previously although they arise quite naturally from the denoising interpretation.

4.2.3 Tanh-nonlinearity interpreted as saturated variance estimate

A popular replacement of the kurtosis-based nonlinearity (50) is the hyperbolic tangent $\tanh(\mathbf{s})$ operating point-wise for the sources. It is generally considered to be more robust against overfitted and spiky source estimates than kurtosis. By selecting $\alpha(\mathbf{s}) = -1$ and $\beta(\mathbf{s}) = 1$, we arrive at

$$\mathbf{f}_t(\mathbf{s}) = s(t) - \tanh[s(t)] = s(t)\left(1 - \frac{\tanh[s(t)]}{s(t)}\right)\,. \tag{55}$$

Now the term multiplying $s(t)$ can be interpreted as a mask related to SNR. Unlike the naïve mask $s^2(t)$ resulting from kurtosis, the tanh-based mask (55) saturates, though not very fast.

The variance based mask (54) with the improvements considered above offers a new intepretation for the robustness of the tanh-mask. Parameter values $\sigma_n^2 = 1$ and $\mu = 1.08$

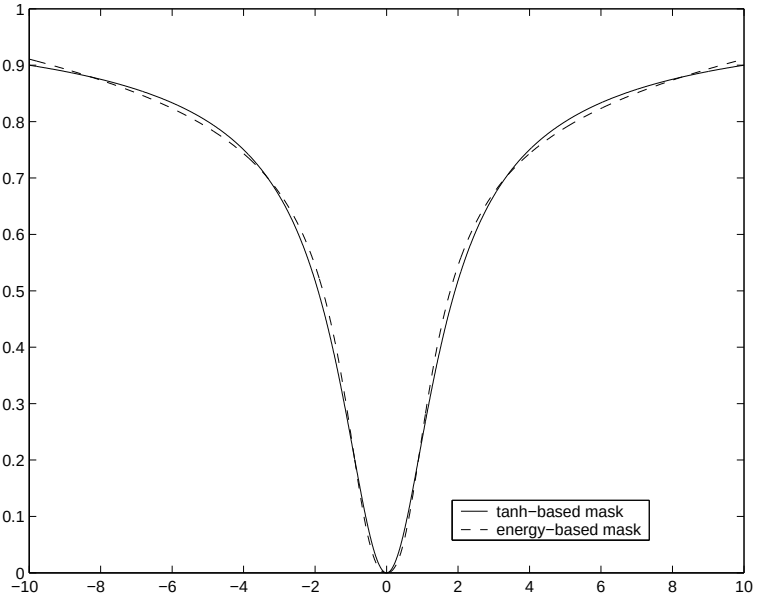


Figure 5: *The tanh-based denoising mask* $1-\tanh(s)/s$ *is shown together with the variance-based denoising mask proposed here. The parameters in the proposed mask were* $\sigma_n^2 = 1$ *and* $\mu = 1.08$*. We have scaled the proposed mask to match the scale of the tanh-based mask.*

give an excellent fit between the masks as shown in Fig. 5. The advantages of the denoising we propose are that σ_n^2 can be tuned to the source estimate, μ can be controlled during the iterations and the estimate of the signal variance can be smoothed. These features contribute to the resistance against overfitting and spiky source estimates.

4.3 Other denoising functions

There are cases where the system specification itself suggests some denoising schemes. One such case, CDMA transmission, is described in Sec. 5.4. Another example is source separation with a microphone array combined with speech recognition. Many speech recognition systems rely on generative models which can be readily used to denoise the speech signals.

Often it would be useful to be able to separate the sources online, *i.e.*, in real time. Since there exists online sphering algorithms (*cf.,* Douglas and Cichocki, 1997, Oja, 1992), real time DSS can be considered as well. One simple case of online denoising is presented by MA filters. However, such online filters typically are not symmetric and thus have no definite objective function (see Sec. 3.1) resulting in potentially unstable DSS. Consider, for example, a case of two harmonic oscillatory sources. It has a rotational invariance in a space defined by the corresponding sine-cosine pair. Batch DSS algorithms would converge to some particular rotation, but non-symmetric on-line denoising by $\mathbf{f}(s(t)) = s(t-1)$ would not converge at all. Thus, in the case of online DSS, denoising would be best kept symmetric.

Sometimes the sources can be grouped to form interesting subspaces. This could happen, *e.g.*, when all the sources are not independent of each others, but there exists anyway subspaces that are mutually independent. It may be desirable to use the information in all sources $\mathbf{S}$ for denoising any particular source $\mathbf{s}_i$. This leads to the following denoising function: $\mathbf{s}_i^+ = \mathbf{f}_i(\mathbf{S})$. Some form of subspace rules can be used to guide the extraction of interesting subspaces in DSS. It is possible to further relax the independence criterion at the borders of the subspaces. This can be achieved by incorporating a neighbourhood denoising rule in DSS, resulting in a topographic ordering of the sources. One such topographic rule was used in topographic ICA (Hyvärinen et al., 2001a).

It is possible to combine various denoising functions when the sources are characterised by more than one type of structure. Note that the combination order might be crucial for the outcome. This is simply because, in general, $\mathbf{f}_i\left(\mathbf{f}_j(\mathbf{s})\right) \neq \mathbf{f}_j\left(\mathbf{f}_i(\mathbf{s})\right)$ where $\mathbf{f}_i$ and $\mathbf{f}_j$ present two different linear or nonlinear denoisings. As an example, consider the combination of the linear on/off-mask (47) and (48), and the nonlinear variance-based mask (54): the noise estimation becomes significantly more accurate when the on/off-masking is performed only after the nonlinear denoising.

Finally, a source might be almost completely known. Then it is possible to apply a detailed matched filter to estimate the mixing coefficients or the noise level. Detailed matched filters have been used in Sec. 5.1 to get an upper limit of the SNRs of the source estimates.

4.4 Spectral shift and approximation of the objective function with mask-based denoisings

In Sec. 3.1 it was mentioned that a DSS algorithm may work pefrectly fine but (37) may still fail to approximate the true objective function if $\alpha(\mathbf{s})$ and $\beta(\mathbf{s})$ are not selected suitably. As an example, consider the masking-based denoisings where denoising is implemented by multiplying the source point-wise by a mask. Without loss of generality, it can be assumed that the data has been rotated with $\mathbf{V}$ and the masking operated directly on the source. According to Eq. (37), $g(\mathbf{s}) = \sum_t s^2(t)m(t)$, where $m(t)$ is the mask. If the mask is constant w.r.t. $\mathbf{s}$, denoising is linear and Eq. (37) is an exact formula, but let us assume that the mask is computed based on the current source estimate $\mathbf{s}$.

In some cases it may be useful to normalise the mask and this could be implemented in several ways. Some possiblities that may come to mind are to normalise the maximum value or the sum of squared values of the mask. While this type of normalisation has no effect on the behaviour of DSS, it can render the approximation (37) useless. This is because a maximally flat mask usually corresponds to a source with a low SNR. However, after normalisation, the sum of values in the mask would be greatest for a maximally flat mask and this tends to produce high values of the approximation of $g(\mathbf{s})$ conflicting with the low SNR.

As a simple example, consider the mask to be $m(t) = s^2(t)$. This corresponds to the kurtosis-based denoising (50). Now the sum of squared values of the mask is $\sum s^4(t)$, but so is $\mathbf{s}\mathbf{f}^T(\mathbf{s})$. If the mask were normalised by dividing by the sum of squares, the approximation (37) would always yield a constant value of one, totally independent of $\mathbf{s}$.

A better way of normalising a mask is to normalise the sum of the values. Then Eq. (37) should always yield approximately the same value if the mask and source estimate are unrelated, but the value would be greater for cases where the magnitude of the source is correlated with the value of the mask. This is usually a sign of a structured source and consequently a high SNR.

The above normalisation also has the benefit that the eigenvalue of a Gaussian signal can be expected to be roughly constant. Assuming that the mask $m(t)$ does not depend very much on the source estimate, the Jacobian matrix $\mathbf{J}(\mathbf{s})$ of $\mathbf{f}(\mathbf{s})$ is roughly diagonal with $m(t)$ as the elements on the diagonal. The trace of $\mathbf{J}(\mathbf{s})$ needed for the estimate of the eigenvalue of a Gaussian signal in (34) is then $\sum_t m(t)$ and the appropriate spectral shift is

$$\beta = -\frac{1}{T}\sum_t m(t)\,. \tag{56}$$

The spectral shift can thus be approximated to be constant due to the normalisation.

5. Experiments

In this section we demostrate the separation capabilities of the algorithms presented earlier. First, in Sec. 5.1, we separate artificial signals with different DSS schemes, some of which can be implemented by FastICA (1998), Hyvärinen (1999). Furthermore, we compare the results to one standard ICA algorithm, JADE (1999), Cardoso (1999). In Secs. 5.2–5.3 linear and nonlinear DSS algorithms are applied extensively in the study of magnetoencephalograms (MEG). Finally, in Sec. 5.4, recovery of CDMA signals is demonstrated. In each experiment after the case of artificial sources, we first discuss the nature of the expected underlying sources. Then we describe this knowledge in the form of denoising.

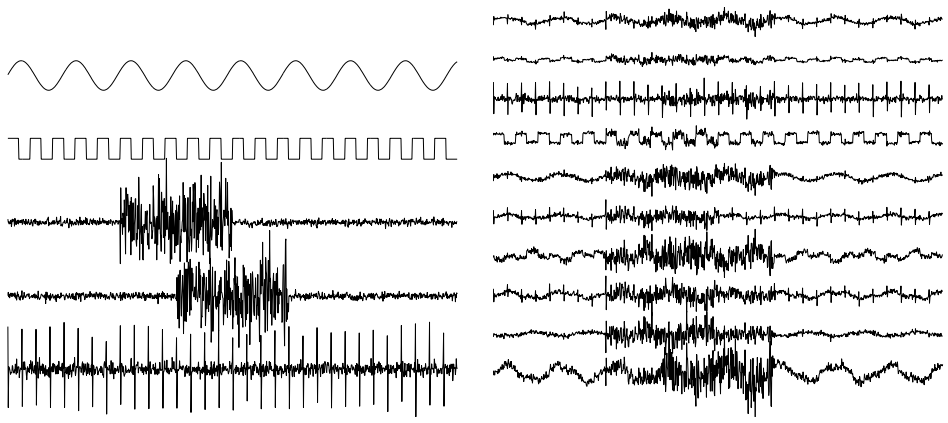
5.1 Artificial signals

Artificial signals were mixed to compare different DSS schemes and JADE (Cardoso, 1999). Ten mixtures of the five sources were produced and independent white noise was added with different SNRs ranging from nearly noiseless mixtures of 50dB to -10dB, a very noisy case. The original sources and the mixtures are shown in Figs. 6a and 6b respectively. The mixtures shown have SNR of 50 dB.

5.1.1 Linear denoising

In this section, we show how the simple linear denoising schemes described in Sec. 4.1 can be used to separate the artificial sources. These schemes require prior knowledge about the source characteristics.

The base frequencies of the first two signals were assumed to be known. Thus two band-pass filtering masks were constructed around these base frequencies. The third and fourth source estimates were known to have periods of activity and non-activity. Third was known to be active in the second quadrant and the fourth a definite period in the latter half. They were denoised using binary masks in time domain. Finally, the fifth source had a known quasi-periodic repetition rate and was denoised using the averaging procedure described in Sec. 4.1.4 and Fig. 2. Since all the five denoisings are linear, five separate filtered data



(a) (b)

Figure 6: *(a) Five artificial signals with simple frequency content (signals #1 and #2), simple on/off non-stationarity in time domain (signals #3 and #4) or quasi-periodicity (signal #5). (b) Ten mixtures of the signals in (a).*

sets were produced and PCA was used to recover the principal components. The separation results are described in Sec. 5.1.3 together with the results of other DSS schemes and JADE.

5.1.2 Nonlinear exploratory denoising

In this section, we describe an exploratory source separation of the artificial signals. One author of this paper gave the mixtures to the other author whose task was to separate the original signals. The author did not receive any additional information, so he was forced to apply a blind approach. He chose to use the masking procedure based on the instantaneous variance estimate, described in Sec. 4.2. To enable the separation of both sub- and super-Gaussian sources in the MAP-based signal-variance-estimate denoising, he used the spectral shift (56). To ensure convergence, he used the 179-rule to control the step size γ (35). Finally, he did not smooth $s^2(t)$ but used it directly as the estimate of the total instantaneous variance $\sigma^2_{\text{tot}}(t)$.

Based on the separation results of the variance-based DSS, he further devised specific masks for each of the source. He chose to denoise the first source in frequency domain with a strict band-pass filter around the main frequency. The author decided to denoise the second source by a simple denoising function $\mathbf{f}(\mathbf{s}) = \text{sign}(\mathbf{s})$. This makes quite an accurate signal model though it neglects the behaviour of the source in time. The third and fourth signal seemed to have periods of activity and non-activity. He found an estimate for the active periods by inspecting the instantaneous variance estimates $\mathbf{s}^2$, and devised simple binary masks. The last signal seemed to consist of alternating positive and negative peaks with fixed inter-peak-interval as well as some additive Gaussian noise. The signal model was tuned to model the peaks only.

5.1.3 Separation results

In this section, we compare the separation results of the linear denoising (Sec. 5.1.1), variance-based denoising and adapted denoising (Sec 5.1.2) to other DSS algorithms. In particular, we compare to the popular denoising schemes $\mathbf{f}(\mathbf{s}) = \mathbf{s}^3$ and $\mathbf{f}(\mathbf{s}) = \tanh(\mathbf{s})$, suggested for use with FastICA (1998). We compare to JADE (Cardoso, 1999) as well. During sphering in JADE, the number of dimensions were either reduced ($n = 5$) or all the ten dimensions were kept ($n = 10$).

We restrained from using deflation in all the different DSS schemes to avoid suffering from cumulative errors in separation of the first sources. Instead one source was extracted with each of the masks several times using different initial vector $\mathbf{w}$ until five sufficiently different source estimates were reached (see Himberg and Hyvärinen, 2003, Meinecke et al., 2002, for further possibilities along these lines). Deflation was only used if no estimate could be found for all the 5 sources. This was often the case for poor SNR under 0dB.

To get some idea of statistical significance of the results, each algorithm was used to separate the sources ten times with the same mixtures, but different measurement noises. The average SNRs of the sources are depicted in Fig. 7. The straight line above all the DSS schemes represents the optimal separation. It is achieved by calculating the unmixing matrix explicitly using the true sources.

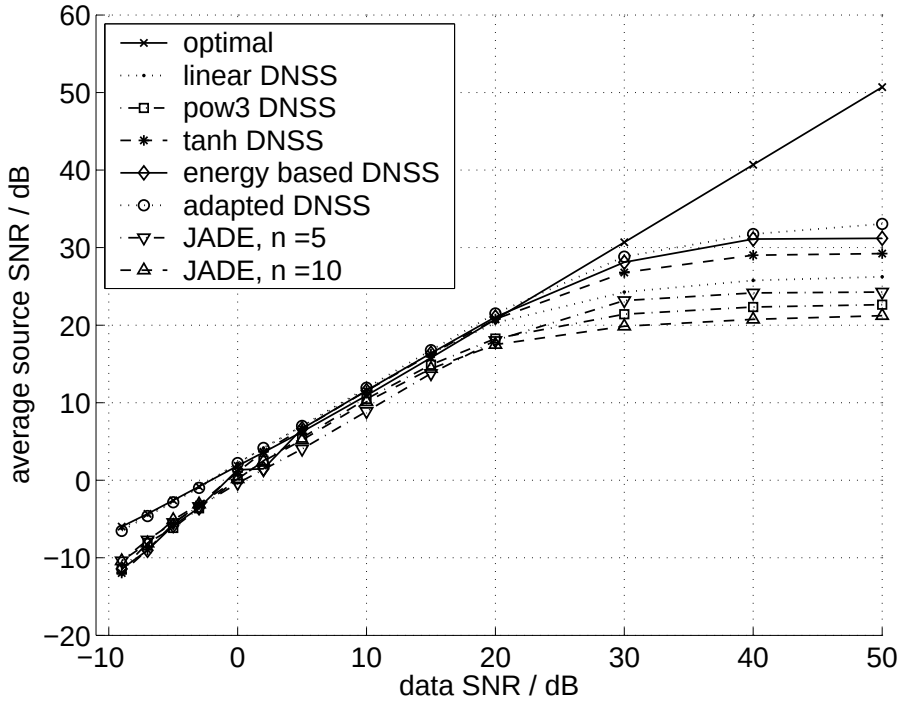


Figure 7: *Average SNRs for the estimated sources averaged over 10 runs.*

With outstanding SNR (> 20 dB), linear DSS together with JADE and kurtosis-based DSS seem to perform worst, while the other, nonlinear DSS approaches: tanh-based, sophis-

ticated variance estimate and the adapted one seem to perform better. The gap between these groups is more than two standard deviations of the 10 runs, making the difference statistically significant. In practice the difference in performance probably does not matter.

With moderate SNRs (between 0 and 20 dB), all algorithms perform quite alike. With poor SNR (< 0 dB), the upper group consist of the linear and adapted DSS as well as the optimal one and the lower group consists of the blind approaches. This seems reasonable, since it makes sense to rely more on prior knowledge when the data is very noisy.

5.2 Exploratory source separation in rhythmic MEG data

In biomedical research it is usual to design detailed experimental frameworks to examine interesting phenomena. Hence it offers a nice field of application for both blind and specialised DSS schemes. In the following we test the developed algorithms in signal analysis of magnetoencephalograms (MEG, Hämäläinen et al., 1993). MEG is a completely non-invasive brain imaging technique measuring the magnetic fields on scalp caused by syncronous activity in the cortex.

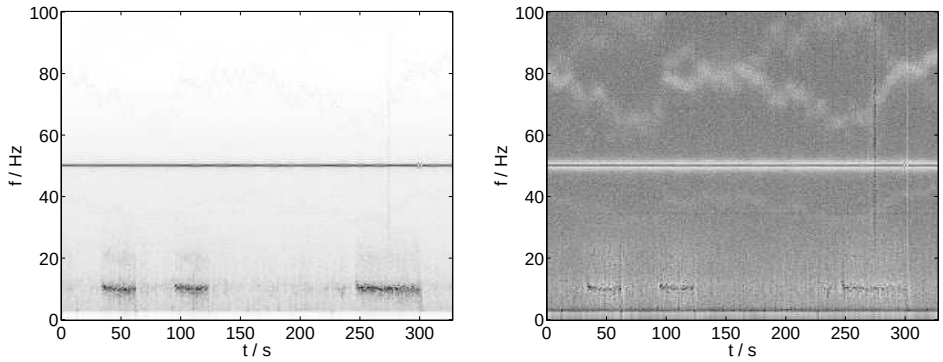
Since the early EEG and MEG recordings, cortical electromagnetic rhythms have played an important role in clinical research, *e.g.*, in detection of various brain disorders, and in studies of development and aging. It is believed that the spontaneous rhythms, in different parts of the brain, form a kind of resting state that allows for quicker responses to stimuli by those specific areas. For example deprivation of visual stimuli by closing one's eyes induces so called α-rhythm on the visual cortex, characterised by a strong 8–13 Hz frequency component. For a more comprehensive discussion regarding EEG and MEG, and their spontaneous rhythms, *cf.,* Niedermeyer and Lopes da Silva (1993), Hämäläinen et al. (1993).

In this paper, we examine an MEG experiment where the subject is asked to relax by closing her eyes (producing α-rhythm). There is also a control state where the subject has her eyes open. The data has been sampled with $f_s = 200$ Hz, and there are $T = 65536$ time samples giving total of more than 300 seconds of measurement. The magnetic fields are measured using a 122-channel MEG device. Some source separation results of this data have been reported by Särelä et al. (2001). Prior to any analysis, the data is high-pass filtered with cut-off frequency of 1 Hz, to get rid of the dominating very low frequencies.

5.2.1 Denoising in rhythmic MEG

Examination of the average spectrogram in Fig. 8a reveals clear structures indicating the existence of several, presumably distinct, phenomena. The burst-like activity around 10 Hz and the steady activity at 50 Hz dominate the data, but there seem to be some weaker phenomena as well, *e.g.*, on higher frequencies than 50 Hz. To amplify these, we not only sphere the data spatially but temporally as well. This temporal decorrelation actually makes the separation harder, but enables the finding of the weaker phenomena. The normalised and filtered spectrogram is shown in Fig. 8b.

The spectrogram data seems well suited for demonstrating the exploratory data analysis use of DSS. As some of the sources seem to have quite steady frequency content in time, but others changing in time, we used two different time-frequency-analyses as described in Sec. 4.1.3 with lengths of the spectra $T_f = 1$ and $T_f = 256$. The first spectrogram is then actually the original frequency-normalised and filtered data with only time information.



(a) (b)

Figure 8: *(a) Averaged spectrogram of all the 122 MEG channels. (b) Frequency normalised spectrogram.*

We apply the several noise reduction principles based on the estimated variance of the signal and the noise discussed in Sec. 4.2. Specifically, the power spectrogram of the source estimate is smoothed over time and frequency using 2-D convolution with Gaussian windows. The standard deviations of the Gaussian windows were $\sigma_t = 8/\pi$ and $\sigma_f = 8/\pi$. After this, the instantaneous estimate of the source variance is found using Eq. (53). Then we get the denoised source estimate using Eq. (54) together with the spectral shift (56). Initially we have set $\mu = 1.3$. This is then decreased by 0.1 every time DSS has converged, until $\mu < 1$ is reached. Finally, the new projection vector is calculated using the stabilised version (35), (36) with the 179-rule in order to ensure convergence.

5.2.2 Separation results

The separated signals, depicted in Fig. 9, include several interesting sources. Due to poor contrast in Fig. 9, we show enhanced and smoothed spectrograms of selected interesting, but low contrast, components (#1, #2, #3 and #18) in Fig. 10. First of all, there exist several sources with α-activity (#1, #4 and #7 for example). The second and 5th source are clearly related to the power-line. The third source depicts an interesting signal caused probably by some anomaly in either the measuring device itself or its physical surroundings. In source #18, there is another, presumably artefactual source, composed of at least two steady frequencies around 70 Hz.

The DSS approach described above seems to be reliable and fast: the temporal decorrelation of the data enabled the finding of very weak sources and yet we found several clear α-sources as well. Valpola and Särelä (2004) have further studied the convergence speed, reliability and stability of DSS with various speedup methods, such as the spectral shift used in FastICA. Convergence speed exceeding standard FastICA by 50 % was reported.

Though quite a clear separation of the sources was achieved, some cross-talk between the signals remains. Better SNR and less talk would probably be achieved by tuning the denoising to the characteristics of each different signal group. In the next section we show

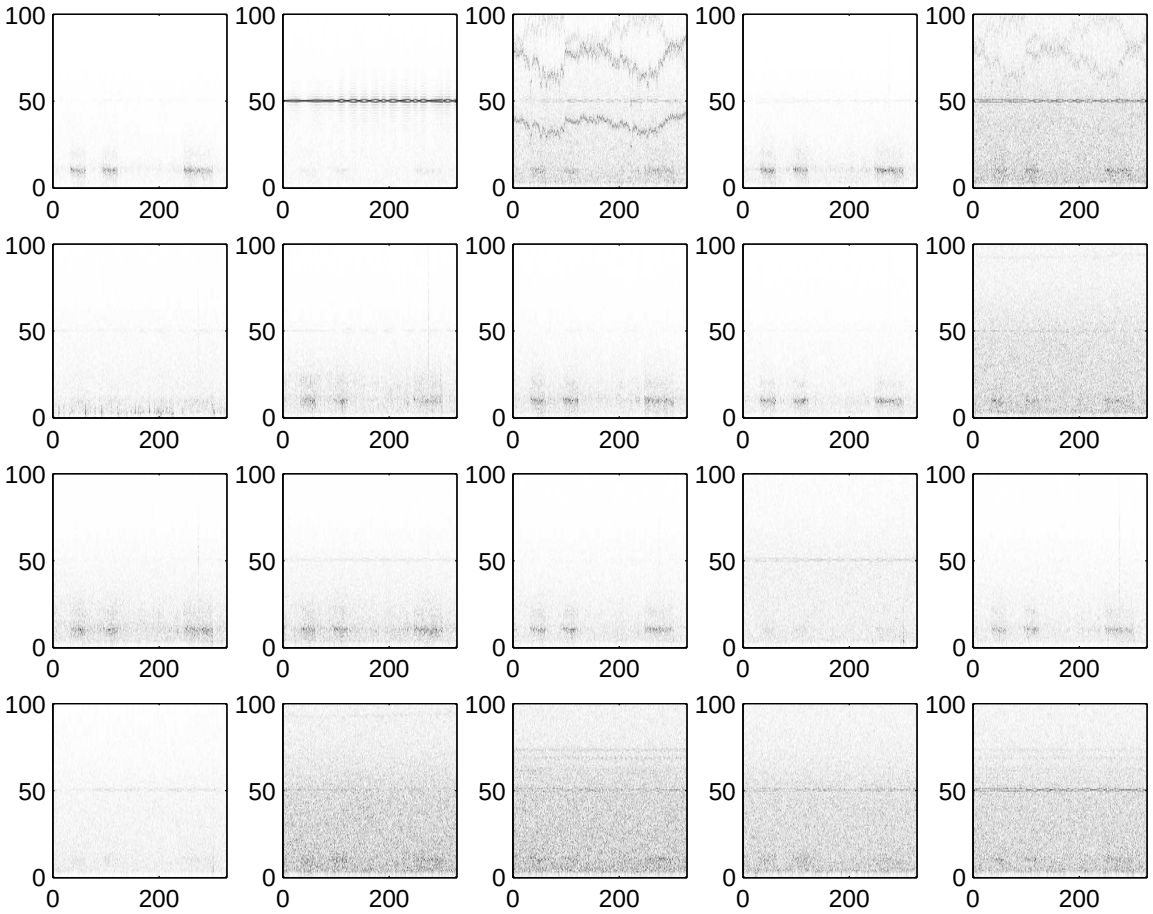


Figure 9: *Spectrograms of the extracted components (comps. 1–5 on the topmost row)*

that with specific knowledge it is possible to find even very weak phenomena in MEG data using DSS.

5.3 Adaptive extraction of cardiac subspace in MEG

Cardiac activity causes magnetic fields as well. Sometimes these are strongly reflected in MEG and can pose a serious problem for the signal analysis of the neural phenomena of interest. In this data, however, the cardiac signals are not visible to the naked eye. Thus, we want to demonstrate the capability of DSS to extract some very weak cardiac signals, using detailed prior information in an adaptive manner.

5.3.1 Denoising of the cardiac subspace

A clear QRS complex, which is the main electromagnetic pulse in the cardiac cycle, can be extracted from the MEG data using standard BSS methods, such as kurtosis- or tanh-based denoising. Due to its sparse nature, this QRS signal can be used to estimate the places of

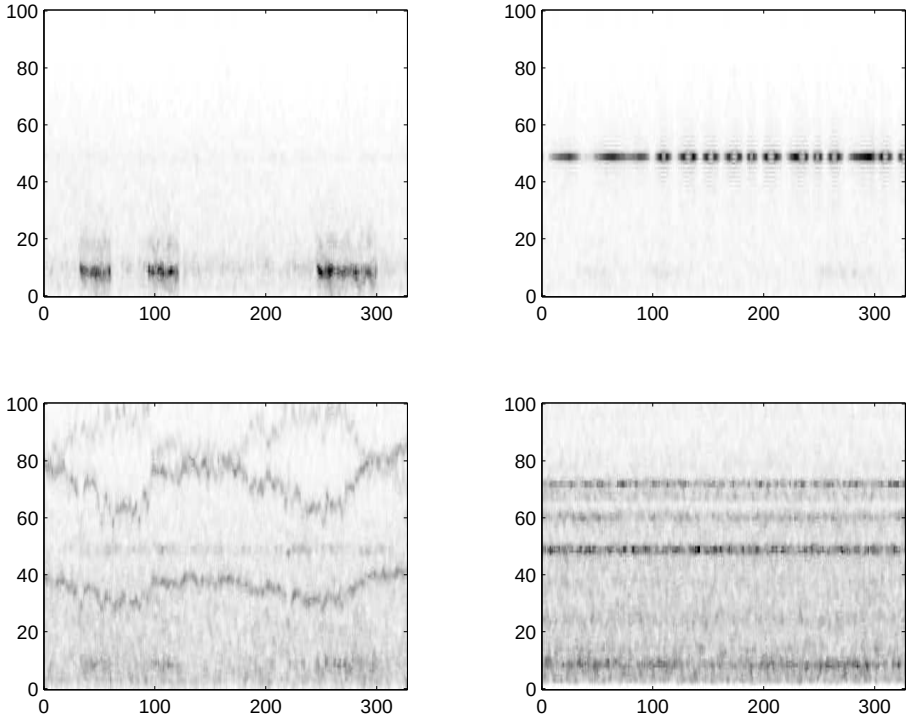


Figure 10: *Enhanced and smoothed spectrograms of the selected components (correspond to sources #1, #2, #3 and #18 in Fig. 9)*

the heart beats. With the places known, we can guide further search using the averaging DSS, as described in Sec. 4.1. Every now and then, we re-estimate the QRS onsets needed for the averaging DSS.

When the estimation of the QRS locations has been stabilised, a subspace that compose of signals having activity phase-locked to the QRS complexes can be extracted.

5.3.2 Separation results

Figure 11 depicts five signals averaged around the QRS complexes, found using the procedure above[6]. The first signal presents a very clear QRS complex, whereas the second one contains the small P and the T waves. An interesting phenomenon is found in the third signal: there is a clear peak at the QRS onset, which is followed by a slow attenuation phase. We presume that it originates from some kind of relaxing state.

Two other heart related signals were also extracted. They both show a clear deflection during the QRS compex, but have as well significant activity elsewhere. These two signals might present a case of overfitting, contemplated in Sec. 3.4. To test this hypothesis, we performed DSS using the same procedure and the same denoising function, but for time-reversed data. As the estimated QRS onsets will then be misaligned, the resulting signals

6. For clarity, two identical cycles of averaged heart beats are always shown.

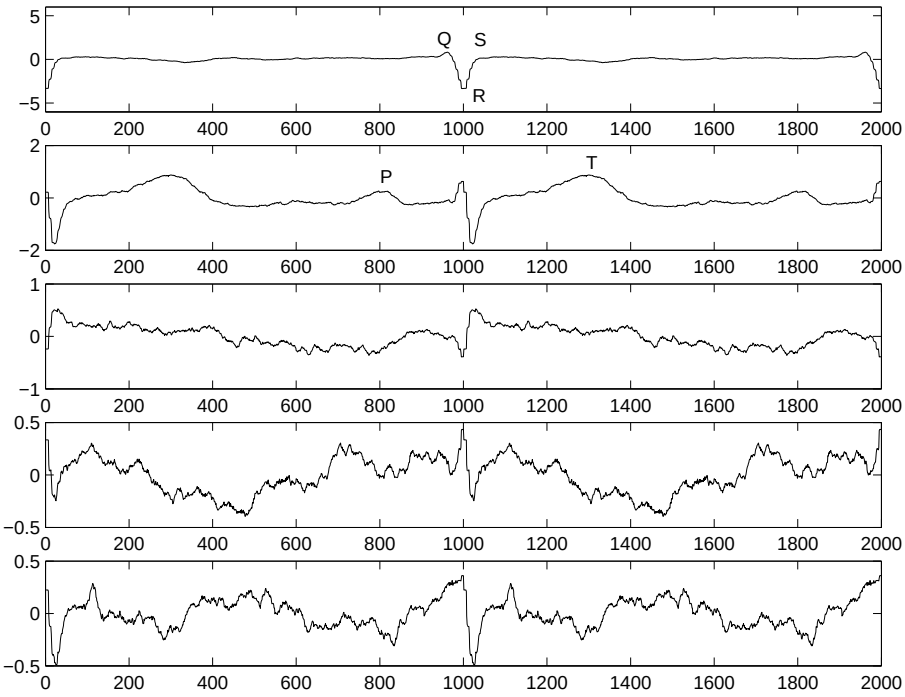


Figure 11: *Averages of three heart related signals and presumably two overfitting results.*

should be pure overfits. The results are shown in Fig. 12. The eigenvalues corresponding to the QRS-complex and the second signal having the P and T waves are approximately 10 times higher than the principal eigenvalue of the reversed data. Thus they clearly exhibit some real structure in the data, as already expected. The eigenvalues corresponding to the last three signals are comparable to the principal eigenvalue of the reversed data, the two largest being somewhat greater. It is reasonable to expect that all the three carry some real structure as there is a nonzero correlation between the first two signals having the main cardiac responses and the overfitted component corresponding to the largest eigenvalue from the reversed data. In the three other signals, there probably occurs some overfitting as well, since the signals have similar structures as the last two signals of the actual subspace experiment shown in Fig. 11.

It is worth noticing that even the strongest component of the cardiac subspace is rather weakly present in the original data. The other components of the subspace are hardly detectable without advanced methods beyond blind source separation. This clearly demonstrates the power that DSS can provide for an exploring researcher.

5.4 Signal recovery in CDMA

Mobile systems constitute another important signal processing application area, in addition to biomedical signal processing. There are several ways to allow multiple users to use the same communication channel, one being modulation scheme called code-division-multiple-

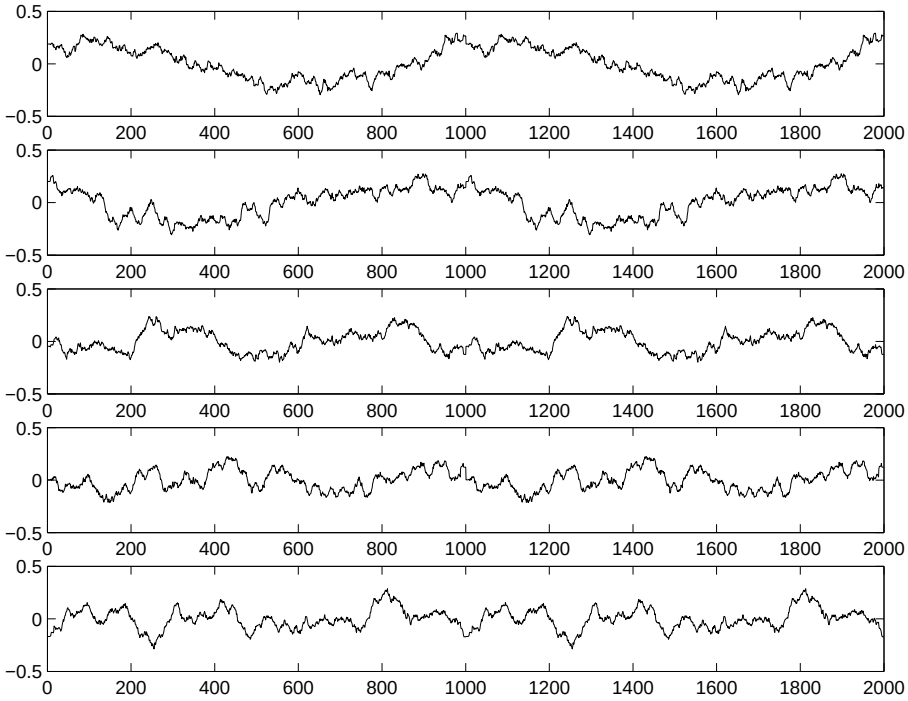


Figure 12: *Averages of five signals from the cardiac control experiment, showing clear overfittings.*

access (CDMA, Viterbi, 1995). In this section we consider bit-stream recovery in a simplified simulation of a CDMA network.

In CDMA each user has a unique signature quasiorthogonal to the signatures of the other users. The user codes each complex bit[7] which he sends using this signature. This coded bit stream is transmitted through the communication channel, where it is mixed with the signals of the other transmitters. The mixture is infected with some noise as well, due to multi-path propagation, doppler shifts, interfering signals, etc.

To recover the sent bit stream, receiver decodes the signal with the known signature. Ideally then, the result would be ones and zeros repeated number of times corresponding to the signature length. In practice, noise and other interefering signals cause variation and the bits are usually extracted by majority voting.

If there are multiple paths a particular bit stream is sent to the receiver or the transmitter and receiver have multiple antennas, so called RAKE procedure can be used: The path coefficients are estimated based on the so called pilot bit streams that are fixed known bit streams and sent frequently by the transmitter. Different bit streams are then summed together before the majority voting. In RAKE-ICA (Raju and Ristaniemi, 2002), ICA is

7. Here a scheme called QAM is used. There two bits are packed into one complex bit by making a 90° phase shift in the other bit.

used to blindly separate the desired signal from the interference of other users and noise. This yields better results in the majority voting.

5.4.1 Denoising of CDMA signals

We know that the original bit stream should consist of repeated coding signatures convoluted by the original complex bits. First the bit stream is decoded using a standard detection algorithm. The denoised signal is then the recoding of the decoded bit stream.

This DSS approach is nonlinear. If the original bit-stream estimate is very inaccurate, *e.g.*, due to serious interference of other users or external noise, the nonlinear approach might get stuck in deficient local minimum. To prevent this, we first initialise by running a simpler, linear DSS. There we only exploit the fact that the signal should consist of repetitions of the signature multiplied by a complex number. The nonlinearity of the denoising is gradually increased in the first iterations.

5.4.2 Separation results

We sent 100 blocks of 200 complex bits. The sent bits were mixed using the streams of 15 other users. For simplicity we set all the path delays to zero. The signal-to-noise-ratio (SNR) varied from -10 to 15 dB. The length of the spreading signature was 31. The mixtures were measured using three antennas. We did not consider multi-path propagation.

Figure 13 sums up the results of CDMA experiments. The comparison to the RAKE algorithm shows that DSS performs better in all situations except in the highest SNR, where RAKE is slightly better. Note that RAKE needs the pilot bits to estimate the mixing while DSS does not need them. The better performance of DSS for low SNR is explained by the fact that DSS in effect actively cancels disturbing signals while RAKE ignores them.

CDMA bit streams consist of known headers that are necessary for standard CDMA techniques to estimate several properties of the transmission channel. In DSS framework, these become useless and can be replaced by proper signal, thus increasing the channel capacity. In addition, bits defined by the actual data such as error-correcting or check bits allow an even better denoising of the desired stream. Furthermore, it is possible to take multipath propagation into account using several delayed versions of the received signal. This should then result in a kind of averaging denoising when proper delay is used analogous to the multi-resolution spectrogram DSS described in Sec. 4.1.3. In case of moving transmitters and receivers, DSS may exploit the Doppler effect.

6. Discussion

In this paper, we developed several DSS algorithms. Moreover, DSS offers a promising framework for developing additional extensions. In this section, we first summarise the extensions that have already been mentioned in previous sections and then discuss some auxiliary extensions, as well.

We discussed online learning strategy in Sec. 4.3, where we noted that asymmetric online denoising may lead to nonconvergent DSS algorithms. However, symmetric denoising procedures performing similar functions may easily be generated.

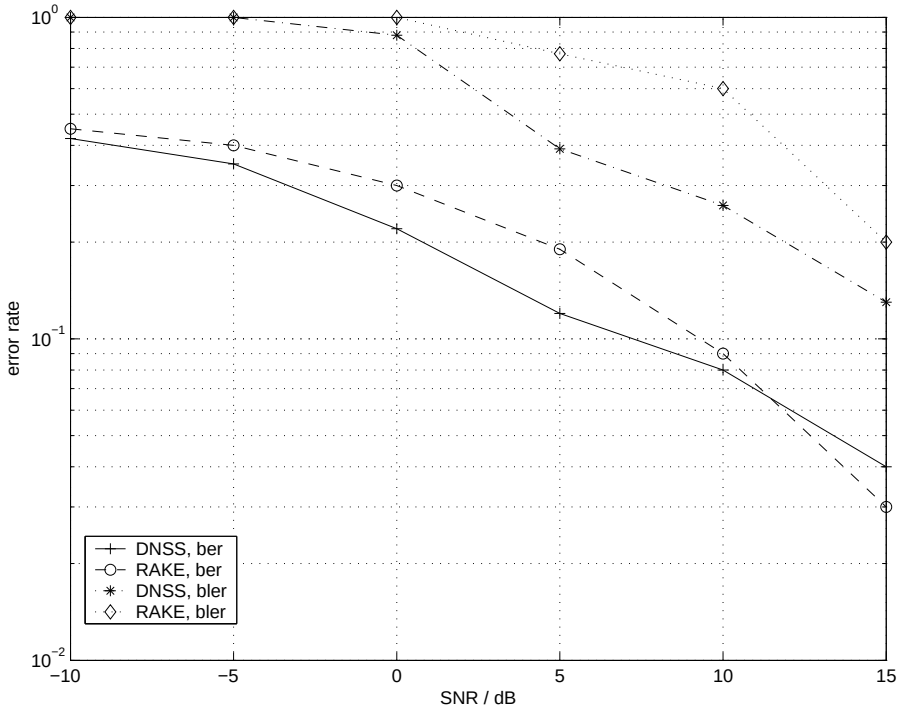


Figure 13: *Bit- and block-error rates for different SNRs for DSS and RAKE.*

We also noted that the masking based on the instantaneous variance in Sec. 4.2 may have problems in separating the actual sources, though it effectively separates the noise subspace from the signal subspace. We proposed a simple modification to magnify small differences between the variance estimates of different sources. Furthermore, we noted that a better founded alternative is to consider explicitly the leakage of variance between the signals. Then the variances of the signals can be decorrelated using similar techniques as suggested by Schwartz and Simoncelli (2001). This idea has been pursued further in the DSS framework (Valpola and Särelä, 2004), making the variance-based masking a very powerful approach to source separation. Furthermore, the variance-based mask saturates on large values. This reduces the tendency to suffer from outliers. However, data values that differ utterly from other data points probably carry no interesting information at all. Even more robustness would then be achieved if the mask would start to decrease on large enough values.

In this paper, we usually considered the sources to have one-dimensional structure, which is used to implement the denoising. We already applied successfully two-dimensional denoising techniques for the spectrograms. Furthermore, it was mentioned in Sec. 2 that the index t of different samples $\mathbf{s}(t)$ might refer as well to space as to time. In space it becomes natural to apply filtering in 2D or even in 3D. For example, the astrophysical ICA (Funaro et al., 2003) would clearly benefit from multidimensional filtering.

Source separation is not the only application of ICA-like algorithms. Another, important field of application is feature extraction. ICA has been used for example in extraction

of features from natural images, similar to those that are found in the primary visual cortex (Olshausen and Field, 1996). It is reasonable to consider DSS extensions that have been suggested in the field of feature extraction as well. For instance, until now we have only considered extraction of multiple components by forcing the projections to be orthogonal. However, nonorthogonal projections resulting from overcomplete representations provide some clear advantages, especially in sparse codes (Földiák, 1990), and may be found useful in DSS framework as well.

Throughout this paper, we have considered linear mapping from the sources to the observations but nonlinear mappings can be used, too. One such approach is slow feature analysis (SFA, Wiskott and Sejnowski, 2002) where the observations are first expanded nonlinearly and sphered. The expanded data is then high-pass filtered and projections minimising the variance are estimated. Due to the nonlinear expansion, it is possible to stack several layers of SFA on top of each others to extract higher-level slowly changing features, resulting in hierarchical SFA.

Interestingly, SFA is directly related to DSS. Instead of minimising the variance after high-pass filtering as in SFA, it is also possible to maximise the variance after low-pass filtering. SFA is thus equivalent to DSS with nonlinear data expansion and low-pass filtering as denoising. This is similar to earlier proposals, *e.g.*, by Földiák (1991).

There are several possibilities for the nonlinear feature expansion in hierarchical DSS. For instance kernel PCA (Schölkopf et al., 1998), sparse coding or liquid state machines (Maass et al., 2002) can be used.

Parga and Rolls (1998) proposed that recurrent activity in cortical circuits might mediate the low-pass filtering. Considering this activity as contextual input suggests that context could be used for denoising more generally, even when the context does not change slowly. Such contextual information has been considered, *e.g.*, by (Becker and Hinton, 1992, Deco and Schürmann, 2000). In particular, Deco and Schürmann (2000) have shown that top-down bias combined with local lateral competition accounts for many of the characteristics of covert attention. In their model, attention emerges because the influence of the representations activated at the higher levels propagates downward the hierarchy. Deco and Rolls (2004) showed that it is possible to learn the features needed for the model but top-down weights were only used for attention. Interpreting the top-down bias as denoising suggests that the same mechanism could account both for learning features and for emergent attention.

The above mentioned connections to the human information processing have been further studied by Valpola (2004). There emergence of complex-cell-like features were achieved by lateral contextual input.

The hierarchical DSS can be used in a fully supervised setting by fixing the activations of the topmost layer to target outputs. Supervised learning often suffers from slow learning in deep hierarchies because the way information is represented gradually changes in the hierarchy. It is therefore difficult to use the information about the target output for learning the layers close to the inputs. The benefit of hierarchical DSS is that learning on lower levels is not only dependent on the information propagated from the target output because the context includes lateral or delayed information from the inputs. In this approach, the mode of learning shifts smoothly from mostly unsupervised learning to mostly supervised

learning from the input layer towards the output layer. A similar mixture of supervised and unsupervised learning has been suggested by Körding and König (2001).

7. Conclusion

The work in linear source separation has concentrated on blind approaches to fix the rotational ambiguity left by the factor analysis model. Usually, however, there would be additional information to find the rotation either more efficiently or more accurately. In this paper we developed an algorithmic framework called denoising source separation (DSS). We showed that denoising can be used for source separation and that the results are often better than with blind approaches. The better the denoising is, the better the results are. Furthermore, many blind source separation techniques can be interpreted as DSS algorithms using very general denoising principles. In particular, we showed that FastICA is a special case of DSS which also implies that DSS can be computationally very efficient.

The main benefit of DSS framework is that it allows for easy development of new source separation algorithms which are optimised for the specific problem at hand. There is a wide literature on signal denoising to choose from and in some cases denoising would be used for post-processing in any case. All the tools needed for DSS are then readily available.

In the experimental section, we demonstrated DSS in various source separation tasks. We showed how denoising can be adapted to the observed characteristics of signals extracted with denoising based on vague knowledge. From MEG signals, we were able to extract very accurately subspaces such as the α-subspace or the very weak components of the cardiac subspace. DSS also proved to be able to recover CDMA signals better than the standard RAKE technique under poor SNR.

Finally, we discussed potential extensions of DSS. It appears that DSS offers a sound basis for developing hierarchical, nonlinear feature extraction methods and the connections to cortical models of attention and perception suggest a promising starting point for future work.

8. Acknowledgements

This work is funded by the Academy of Finland, under the project New information processing principles, and by European Commission, under the project ADAPT (IST-2001-37137).

We would like to show gratitude to Dr. Ricardo Vigário for the fruitful discussions concerning the method in general as well as the MEG experiments in detail and Dr. Aapo Hyvärinen for the method itself and its connections to ICA. We would like to thank as well Mr. Karthikesh Raju for his suggestions and help concerning the CDMA experiments. Our sincere thanks are also to the editor and the anonymous referees for their thorough inspection of the article. Finally, we would like to thank prof. Erkki Oja for his comments on the draft version of this manuscript.

References

B. D. Anderson and J. B. Moore. *Optimal filtering*. Prentice-Hall, 1979.

H. Attias. Independent factor analysis. *Neural Computation*, 11(4):803–851, 1999.

S. Becker and G. E. Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355:161 – 163, 1992.

A. Belouchrani, K. Abed Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique based on second order statistics. *IEEE Trans. on S.P.*, 45(2):434–44, 1997.

O. Bermond and J.-F. Cardoso. Approximate likelihood for noisy mixtures. In *Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation, ICA'99*, pages 325–330, Aussois, France, Jan. 11-15, 1999.

J.-F. Cardoso. High-order contrasts for independent component analysis. *Neural computation*, 11(1):157 – 192, 1999.

K.-L. Chan, T.-W. Lee, and T. J. Sejnowski. Variational Bayesian learning of ICA with missing data. *Neural Computation*, 15 (8):1991–2011, 2003.

R. A. Choudrey and S. J. Roberts. Flexible Bayesian independent component analysis for blind source separation. In *Proc. Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, pages 90–95, San Diego, USA, 2001.

G. Deco and E. T. Rolls. A neurodynamical cortical model of visual attention and invariant object recognition. *Vision research*, 44:621 – 642, 2004.

G. Deco and B. Schürmann. A hierarchical neural system with attentional top-down enhancement of the spatial resolution for object recognition. *Vision research*, 40:2845 – 2859, 2000.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statistical Society, Series B (Methodological)*, 39 (1):1–38, 1977.

D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard. Wavelet shrinkage: asymptopia? *Journal of the Royal Statistical Society ser. B*, 57:301–337, 1995.

S. C. Douglas and A. Cichocki. Neural networks for blind decorrelation of signals. *IEEE Trans. Signal Processing*, 45(11):2829 – 2842, 1997.

FastICA. The FastICA MATLAB package. 1998. Available at `http://www.cis.hut.fi/projects/ica/fastica/`.

P. Földiák. Forming sparse representations by local anti-hebbian learning. *Biological Cybernetics*, 64:165 – 170, 1990.

P. Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3: 194–200, 1991.

M. Funaro, E. Oja, and H. Valpola. Independent component analysis for artefact separation in astrophysical images. *Neural networks*, 16(3 – 4):469 – 478, 2003.

M. S. Gazzaniga, editor. *The New Cognitive Neurosciences*. A Bradford book/MIT Press, 2nd edition, 2000.

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.

M. Hämäläinen, R. Hari, R. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa. Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of Modern Physics*, 65:413–497, 1993.

J. Himberg and A. Hyvärinen. Icasso: software for investigating the reliability of ica estimates by clustering and visualization. In *Proc. 2003 IEEE workshop on neural networks for signal processing (NNSP'2003)*, pages 259–268, Toulouse, France, 2003.

P. A.d.F.R Højen-Sørensen, O. Winther, and L. K. Hansen. Mean-field approaches to independent component analysis. *Neural Computation*, 14:889–918, 2002.

A. Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in Neural Information Processing 10 (Proc. NIPS'98)*, pages 273–279. MIT Press, 1998.

A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. on Neural Networks*, 10(3):626–634, 1999.

A. Hyvärinen, P. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13(7):1525–1558, 2001a.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*. Wiley, 2001b.

JADE. The JADE MATLAB package. 1999. Available at `http://www.tsi.enst.fr/icacentral/Algos/cardoso/`.

K. H. Knuth. Bayesian source separation and localization. In A. Mohammad-Djafari, editor, *SPIE'98 Proceedings: Bayesian Inference for Inverse Problems*, pages 147–158, San Diego, USA, 1998.

P. Kuosmanen and J. T. Astola. *Fundamentals of nonlinear digital filtering*. CRC press, 1997.

K. P. Körding and P. König. Neurons with two sites of synaptic integration learn invariant representations. *Neural Computation*, 13:2823 – 2849, 2001.

H. Lappalainen. Ensemble learning for independent component analysis. In *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, pages 7–12, Aussois, France, 1999.

D. G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, 1969.

W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531 – 2560, 2002.

F. Meinecke, A. Ziehe, M. Kawanabe, and K.-R. Müller. A resampling approach to estimate the stability of one- and multidimensional independent components. *IEEE Trans. Biom. Eng.*, 49(12):1514 – 1525, 2002.

J. Miskin and David J. C. MacKay. Ensemble learning for blind source separation. In S. Roberts and R. Everson, editors, *Independent Component Analysis: Principles and Practice*, pages 209–233. Cambridge University Press, 2001.

J. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Phys. Rev. Lett.*, 72:541–557, 1994.

E. Niedermeyer and F. Lopes da Silva, editors. *Electroencephalography. Basic principles, clinical applications, and related fields*. Baltimore: Williams & Wilkins, 1993.

E. Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5:927–935, 1992.

B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

N. Parga and E. T. Rolls. Transform invariant recognition by association in a recurrent network. *Neural Computation*, 10(6):1507 – 1525, 1998.

D. B. Percival and W. T. Walden. *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge University Press, Cambridge, UK, 1993.

D.-T. Pham and J.-F. Cardoso. Blind separation of instantaneous mixtures of non stationary sources. *IEEE Trans. on Signal Processing*, 49:1837–1848, 2001.

K. Raju and T. Ristaniemi. ICA-RAKE switching for jammer cancellation in DS-CDMA array systems. In *Proc. of the IEEE Int. Symposium on Spread Spectrum Techniques and Applications (ISSSTA)*, pages ?? – ??, Prague, September 2002.

R. M. Rangayyan. *Biomedical signal analysis: A case-study approach*. IEEE Press Series in Biomedical Engineering, 2002.

J. Särelä and R. Vigário. Overlearning in marginal distribution-based ICA: analysis and solutions. *Journal of Machine Learning Research*, 4 (Dec):1447–1469, 2003.

Jaakko Särelä, Harri Valpola, Ricardo Vigário, and Erkki Oja. Dynamical factor analysis of rhythmic magnetoencephalographic activity. In *Proc. Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, pages 451–456, San Diego, USA, 2001.

B. Schölkopf, S. Mika, A. Smola, Gunnar Rätsch, and K.-R. Müller. Kernel PCA pattern reconstruction via approximate pre-images. In *Proc. 8th Int. Conf. on Artificial neural networks (ICANN'98)*, pages 147 – 152, Skövde, 1998.

O. Schwartz and E. P. Simoncelli. Natural signal statistics and sensory gain control. *Nature Neuroscience*, 4(8):819 – 825, 2001.

L. Tong, V. Soo, R. Liu, and Y. Huang. Indeterminacy and identifiability of blind identification. *IEEE Trans. on Circuits and Systems*, 38:499–509, 1991.

H. Valpola. Behaviourally meaningful representations from normalisation and context-guided denoising. Technical report, Artificial Intelligence Laboratory, Department of Information Technology, University of Zurich, 2004. Available at Cogprints: `http://cogprints.ecs.soton.ac.uk/archive/00003633/`.

H. Valpola and P. Pajunen. Fast algorithms for Bayesian independent component analysis. In *Proceedings of the second international workshop on independent component analysis and blind signal separation, ICA'00*, pages 233–238, Espoo, Finland, 2000.

H. Valpola, T. Raiko, and J. Karhunen. Building blocks for hierarchical latent variable models. In *Proc. 3rd Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, pages 710–715, San Diego, USA, 2001.

H. Valpola and J. Särelä. Accurate, fast and stable denoising source separation algorithms. In *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA'04)*, Granada, Spain, 2004.

M. Vetterli and J. Kovacevic. *Wavelets and subband coding*. Prentice-Hall, 1995.

R. Vigário, J. Särelä, V. Jousmäki, M. Hämäläinen, and E. Oja. Independent component approach to the analysis of EEG and MEG recordings. *IEEE transactions on biomedical engineering*, 47(5):589–593, 2000.

V. Vigneron, A. Paraschiv-Ionescu, A. Azancot, O. Sibony, and C. Jutten. Fetal electrocardiogram extraction based on non-stationary ICA and wavelet denoising. In *Proceedings of ISSPA 2003*, Paris (France), July 2003.

A. J. Viterbi. *CDMA : Principles of Spread Spectrum Communication*. Wireless Info Networks Series. Addison-Wesley, 1995.

J. H. Wilkinson. *The algebraic eigenvalue problem*. Monographs on numerical analysis. Clarendon press, London, 1965.

L. Wiskott and T. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14:715 – 770, 2002.

A. Ziehe and K.-R. Müller. TDSEP — an effective algorithm for blind separation using time structure. In *Proc. int. conf. at neural networks (ICANN'98)*, pages 675–680, Skövde, Sweden, 1998.