Multiscale Framework For Blind Source Separation

Pavel Kisilev*
Michael Zibulevsky*
Yehoshua Y. Zeevi*

*Department of Electrical Engineering, Technion, Haifa 32000, Israel
*Columbia University, New York, NY 10027, USA

Abstract

We consider the problem of blind separation of sources from a set of their linear mixtures. It was discovered recently, that exploiting the sparsity of sources, appropriately represented according to some signal dictionary, dramatically improves the quality of separation. In this study we take advantage of the properties of multiscale transforms, such as wavelet or wavelet packets, to decompose signals into sets of local features with various degrees of sparsity. We study how the separation error is affected by the sparsity of decomposition coefficients, and by the misfit between the probabilistic model of these coefficients and their actual distribution. Our error estimator, based on the Taylor expansion of the quasi-ML function, is used in selection of the best subsets of coefficients, utilized in turn for further separation. The performance of the algorithm is verified on noise-free and noisy data. Experiments with simulated signals, musical sounds and images demonstrate significant improvement of separation quality over previously reported results.

Keywords: Blind Source Separation, Multiscale transforms, Maximum Likelihood, Wavelets

1. Introduction

In a variety of communication and signal sensing applications, crosstalk or mixing of several source signals occurs. The so-called Blind Source Separation (BSS) is concerned with scenarios where an $N$-channel sensor signal $x(\xi)$ is generated by $M$ unknown scalar source signals $s_m(\xi), m = 1, ..., M$, linearly mixed by an unknown $N \times M$ mixing, or crosstalk, matrix $A$, and possibly corrupted by additive noise $n(\xi)$:

$$x(\xi) = As(\xi) + n(\xi).$$ (1)

The independent variable $\xi$ represents either time, spatial coordinates in the case of images, or spatio-temporal variables in the case of video sequences. The BSS problem is to estimate the mixing matrix $A$ and, thereby, the $M$-dimensional source signal $s(\xi)$.

One promising application of blind source separation in 2D is encountered in hyperspectral imaging [1], wherein images of a body surface are taken at several wavelengths. If several chemical compounds are present on a surface, the image at each wavelength represents a weighted sum of fingerprints of the unknown concentrations of the various compounds, with weights determined by radiation spectra of each compound. The problem is to recover the unknown concentrations and spectra.

The assumption of statistical independence of the source components $s_m(\xi)$ leads to the Independent Component Analysis (ICA) [2], [3]. A more powerful assumption is sparsity of
the decomposition coefficients, when the sources are properly represented [4], [5]. Sparsity means that only a small number of coefficients differ significantly from zero. In particular, let each $s_m(\xi)$ have a sparse representation of its decomposition coefficients $c_{mk}$, obtained by means of a signal dictionary of functions $\varphi_k(\xi)$:

$$s_m(\xi) = \sum_k c_{mk} \varphi_k(\xi). \quad (2)$$

The functions $\varphi_k(\xi)$ are called atoms or elements of the dictionary. These elements do not have to be linearly independent, and instead may form an overcomplete dictionary, for example, wavelet-related dictionaries (wavelet packets, stationary wavelets, etc., see for example [6], [7], [8]). The corresponding representation of the mixtures, according to the same signal dictionary, is:

$$x_m(\xi) = \sum_k y_{mk} \varphi_k(\xi), \quad (3)$$

where $y_{mk}$ are the decomposition coefficients of the mixtures. From (1) and (2), in the noise-free case, the relation between the decomposition coefficients of the mixtures and the decomposition coefficients of the sources, is

$$y_k = Ac_k.$$ 

Here, vectors $y_k$ and $c_k$ are constructed from the $k$-th coefficients of mixtures and of sources, respectively. Note, that the relation between decomposition coefficients of the mixtures and the sources is exactly the same relation as in the original domain of signals, where $x_\xi = As_\xi$. Unmixing of the sources is then performed in the transform domain, that is, using the decomposition coefficients of the mixtures $y_k$ instead of the mixtures $x_\xi$.

The property of sparsity often yields much better source separation than standard ICA, and can work well even with more sources than mixtures [9]. In many cases there are distinct groups of coefficients, wherein sources have different sparsity properties. The proposed multiscale, or multiresolution, approach to the BSS is based on selecting only a subset of features, or coefficients, $Y = \{y_k : k \in \Omega\}$, which is best suited for separation, with respect to the sparsity of coefficients and to the separability of sources’ features. In our experiments we use the Wavelet Packet (WP) transform that reveals the structure of signals, wherein several subsets of the WP coefficients have significantly better sparsity and separability than others.

We also investigate how the separation error is affected by the sparsity of a particular subset of the decomposition coefficients, and by the misfit between the probabilistic model of these coefficients and their actual distribution. Since the actual separation errors are not tractable in practice, we propose to use a method for estimation of the separation error, based on the Taylor expansion of the quasi log-likelihood function. The obtained estimates are used for selection of the best subset of coefficients, i.e. the subset that leads to the lowest estimated error. After the new data set is formed from this best subset, one uses it in the separation process, that can be accomplished by any of the standard ICA algorithms or by clustering.

The performance of our approach is verified on noise-free and noisy data. Our experiments with 1D signals and images demonstrate that the proposed method significantly
improves separation quality, as compared to the results obtained by using sparsity of a complete set of decomposition coefficients.

2. Sparse source separation

Sparse sources can be separated by each one of several techniques, for example, by approaches based on the maximum likelihood (ML) considerations ([10], [11], [12]), or by approaches based on geometric considerations ([13], [14], [15]). In the former case, the algorithm estimates the unmixing matrix \( W = A^{-1} \), while in the later case the output is the estimated mixing matrix. In both cases, these matrices can be estimated only up to a column permutation and a scaling factor [16].

2.1 Quasi Maximum Likelihood BSS

In this section, we discuss the ML and the quasi ML solution of the BSS problem, based on the data in the domain of decomposition coefficients. The term quasi indicates that the proposed hypothetical density is used instead of the true one (see discussion below).

Let \( Y \) be the features, or new data matrix of dimension \( M \times K \), where \( K \) is the number of features, or data points, and the coefficients \( y_k \)'s form the columns of \( Y \). Note, that, in general, the rows of \( Y \) can be formed from either the samples of sensor signals, that is, mixtures, or, as in our setting, from their decomposition coefficients. In the latter case, \( Y = \{ y_k : k \in \Omega_{jn} \} \), where \( \Omega_{jn} \) are the subsets indexed on the WP tree, as explained in the section on the Multinode analysis. We are interested in the maximum likelihood estimate of \( A \) given the data \( Y \).

2.1.1 Quasi log-likelihood function

We assume that the coefficients \( c_{mk} \) are i.i.d. random variables with the joint probability density function (pdf)

\[
    f_C = p(C) = \prod_{m,k} p(c_{mk}),
\]

where \( p(c_{mk}) \) is of an exponential type:

\[
    f_{c_{mk}} = p(c_{mk}) = N_q \exp\{-\nu(c_{mk}, q)\}.
\]

The normalization constant \( N_q \) is omitted in the further calculations, since it has no effect on the maximization of the log-likelihood function. In a particular case wherein

\[
    \nu(c_{mk}, q) = |c_{mk}|^q / q,
\]

and \( q < 1 \), the above distribution is widely used for modeling sparsity [17], [18]. For \( q = 0.5 \div 1 \), it approximates rather well the empirical distributions of wavelet coefficients of natural signals and images [19]. A smaller \( q \) corresponds to a distribution with greater sparsity.

Let \( W \equiv A^{-1} \) be the unmixing matrix to be estimated. Taking into account that \( Y = AC \), we arrive at the standard expression of the ICA log-likelihood, but with respect
to the decomposition coefficients:

\[
L_W(Y) = K \log |\det W| - \sum_{m=1}^{M} \sum_{k=1}^{K} \nu([WY]_{mk}, q).
\]

In the case wherein \(\nu(\cdot, q) = |\cdot|^q / q\), the second term in the above log-likelihood function is not convex for \(q < 1\), and non-differentiable, and, therefore, is difficult to optimize. Furthermore, the parameter \(q\) of the true pdf is usually unknown in practice, and estimation of this parameter along with the estimation of the unmixing matrix is a difficult optimization problem. Therefore, it is convenient to replace the actual \(\nu(\cdot)\) with its hypothetical substitute, a smooth, convex approximation of the absolute value function, for example \(\tilde{\nu}(c_{mk}) = \sqrt{c_{mk}^2 + \zeta}\), with \(\zeta\) being a smoothing parameter. This approximation has a minor effect on the separation performance, as indicated by our numerical results. The corresponding quasi log-likelihood function is

\[
\tilde{L}_W(Y) = K \log |\det W| - \sum_{m=1}^{M} \sum_{k=1}^{K} \tilde{\nu}([WY]_{mk}). \tag{4}
\]

### 2.1.2 Natural Gradient algorithm update

Maximization of \(\tilde{L}_W(Y)\) with respect to \(W\) can be solved efficiently by several methods, for example the Natural Gradient (NG) algorithm [20], [21], or, equivalently, by the Relative Gradient [22], as implemented in the ICA/EEG Matlab toolbox [23].

The derivative of the quasi log-likelihood function with respect to the matrix parameter \(W\):

\[
\frac{\partial \tilde{L}_W(y)}{\partial W} = (KI - \tilde{\psi}(c)c^T)(W^T)^{-1}, \tag{5}
\]

where \(c\) is the ‘column stack’ version of \(C\), and \(\tilde{\psi}(c) = [\tilde{\psi}_1(c_1) ... \tilde{\psi}_{MK}(c_{MK})]^T\), where \(\tilde{\psi}_{mk}(c_{mk}) \equiv -(\log f_{c_{mk}})'\) are the so-called score functions. Note, that, in our case, \(\psi_{mk}(c_{mk}) = \tilde{\nu}'(c_{mk})\). The learning rule of the Natural Gradient algorithm is given by

\[
\Delta W = \frac{\partial \tilde{L}_W(y)}{\partial W} W^T W = (KI - \tilde{\psi}(c)c^T)W.
\]

In the original Natural Gradient algorithm, as implemented in [23], the above update equation is expressed in terms of the non-linearity, which has the form of the cumulative density function (cdf) of the hypothetical source distribution. The built-in non-linearity is the logistic function, \(\tilde{g}_s = 1/(1 + \exp(-s))\). The relation of the non-linearity to the score function, is as follows: since \(\tilde{f}_c = \tilde{g}'_c\), we have, by differentiation:

\[
\tilde{\psi}(c) \equiv -(\log \tilde{f}_c)' = \frac{\tilde{f}_c'}{\tilde{f}_c} = -\frac{\tilde{g}'_c}{\tilde{g}_c}. \tag{6}
\]

The score function for the above \(\tilde{g}_s\), is \(\tilde{\psi}_g = -1 + 2\tilde{g}_s\). The corresponding function \(\tilde{\nu}(\cdot)\) is, as before, a kind of smooth approximation of the absolute value function, with a smoothing parameter of order 1. In order to adapt the algorithm to our purposes, we use hypothetical density with \(\tilde{\nu}(c_{mk}) = \sqrt{c_{mk}^2 + \zeta}\), and the corresponding score function is \(\tilde{\psi}(c_{mk}) = c_{mk}/\sqrt{c_{mk}^2 + \zeta}\).
2.2 Clustering-based BSS

In the case of geometry based methods separation of sparse sources can be achieved by clustering along orientations of data concentration in the $N$-dimensional space wherein each column $y_k$ of the matrix $Y$ represents a data point and $N$ is the number of mixtures. Let us consider a two-dimensional noiseless case, wherein two source signals, $s_1(t)$ and $s_2(t)$, are mixed by a $2 \times 2$ matrix $A$, arriving at two mixtures $x_1(t)$ and $x_2(t)$. In this case, the data matrix is constructed from these mixtures $x_1(t)$ and $x_2(t)$. An example of a scatter plot of two sparse mixtures $x_1(t)$ versus $x_2(t)$ is shown in Figure 1. If only one source, say $s_1(t)$, was present, the sensor signals would be

$$
x_1(t) = a_{11}s_1(t)$$
$$
x_2(t) = a_{21}s_1(t)
$$

and the data points at the scatter diagram of $x_2$ versus $x_1$ would belong to the straight line placed along the vector $[a_{11}a_{21}]^T$. The same thing happens, when two sparse sources are present. In this sparse case, at each particular index where a sample of the first source is large, there is a high probability, that the corresponding sample of the second source is small, and the point at the scatter diagram still lies close to the mentioned straight line. The same arguments are valid for the second source. As a result, data points are concentrated around two dominant orientations, which are directly related to the columns of $A$.

Source signals are rarely sparse in their original domain. In contrast, their decomposition coefficients (2) usually show high sparsity. Therefore, we construct the data matrix $Y$ from the decomposition coefficients of mixtures (3), rather than from the mixtures themselves, and the above discussion is valid.

In order to determine orientations of scattered data, we project the data points onto the surface of a unit sphere by normalizing corresponding vectors, and then apply a standard clustering algorithm. This clustering approach works efficiently even if the number of sources is greater than the number of sensors.

Our clustering procedure can be summarized as follows:

1. Form the feature matrix $Y$, by putting samples of the sensor signals or (subset of) their decomposition coefficients into the corresponding rows of the matrix;

2. Normalize feature vectors: $y_k = y_k / \|y_k\|_2$, in order to project data points onto the surface of a unit sphere, where $\|\cdot\|_2$ denotes the $l_2$ norm;

Before normalization, it is reasonable to remove data points with a very small norm, since these very likely to be crosstalk-corrupted by small coefficients from others’ sources.
3. Move data points to a half-sphere, e. g. by forcing the sign of the first coordinate \( y^1_k \) to be positive: if \( y^1_k < 0 \) then \( y_k = -y_k \).

Without this operation each set of linearly (that is, along a line) clustered data points would yield two clusters on opposite sides of the sphere.

4. Estimate cluster centers by using a clustering algorithm. The coordinates of these centers will form the columns of the estimated mixing matrix \( \hat{A} \):

We used Fuzzy C-means (FCM) clustering algorithm [24] as implemented in Matlab Fuzzy Logic Toolbox.

2.3 Sources recovery

The estimated unmixing matrix \( \hat{W} = \hat{A}^{-1} \) can be obtained by either clustering, or by the quasi-ML approach along with the Natural Gradient (we will call it simply Natural Gradient), or by other algorithms, which are applied to either the complete data set, or to some subsets of data (see the subsequent section). In any case, this matrix and, therefore, the sources, can be estimated only up to a column permutation and a scaling factor [16]. The sources are recovered in their original domain by

\[
\hat{s}(t) = \hat{W}x(t).
\]

It should be stressed, that if the above clustering approach is used, the estimation of sources is not restricted to the case of square mixing matrices, although the sources recovery is more complicated in the rectangular cases.

3. Multiscale BSS

To provide intuitive insight into the practical implications of our main ideas, we first discuss an example of separation of 1D block functions, that are piecewise constant, with random amplitude and duration of each constant piece (Figure 2).

3.1 Motivating example: sparsity of random blocks in the Haar basis

It is known, that the Haar wavelet basis provides compact representation of block functions. Let us take a close look at the Haar wavelet coefficients at different resolution levels \( j=0,1,...,J \). Wavelet basis functions at the finest resolution level \( j=J \) are obtained by translation of the Haar mother wavelet:

\[
\varphi_{J}(t) = \begin{cases} 
1 & \text{if } 0 \leq t < \frac{1}{2} \\
-1 & \text{if } \frac{1}{2} \leq t < 1 \\
0 & \text{otherwise}
\end{cases}
\]

Taking the scalar product of a function \( s(t) \) with the wavelet \( \varphi_j(t - \tau) \), we produce a finite differentiation of the function \( s(t) \) at the point \( t = \tau \). This means that the number of non-zero coefficients at the finest resolution for a block function will correspond roughly to the number of jumps of this function. Proceeding to the next, coarser resolution level, we have the wavelet \( \varphi_{J-1}(t) = \{ \frac{1}{2}, \text{if } 0 \leq t < 1; -\frac{1}{2}, \text{if } 1 \leq t < 2; 0 \text{ otherwise}\} \). At this
level, the number of non-zero coefficients still corresponds to the number of jumps, but the total number of coefficients at this level is halved, and so is the sparsity. If we further proceed to coarser resolutions, we will encounter levels where the support of a wavelet $\varphi_j(t)$ is comparable to the typical distance between jumps in the function $s(t)$. In this case, most of the coefficients are expected to be nonzero, and, therefore, sparsity will fade away.

To demonstrate how this influences accuracy of a blind source separation, we randomly generated two block-signal sources (Figure 2, two upper plots.), and mixed them by the crosstalk matrix

$$A = \begin{pmatrix} 0.8321 & 0.6247 \\ -0.5547 & 0.7809 \end{pmatrix}.$$  

Resulting sensor signals, or mixtures, $x_1(t)$ and $x_2(t)$ are shown in the two lower plots of Figure 2. The scatter plot of $x_1(t)$ versus $x_2(t)$ does not exhibit any visible distinct orientations (Figure 3, left). Similarly, in the scatter plot of the wavelet coefficients at the lowest resolution distinct orientations are hardly detectable (Figure 3, middle). In contrast, the scatter plot of the wavelet coefficients at the highest resolution (Figure 3, right) depicts two distinct orientations, which correspond to the columns of the mixing matrix.

Since a cross-talk matrix $A$ is estimated only up to a column permutation and a scaling factor, in order to measure the separation accuracy, we normalize the original sources $s_m(t)$ and their corresponding estimated sources $\tilde{s}_m(t)$. The averaged (over sources) normalized squared error (NSE) is then computed as:

$$NSE = \frac{1}{M} \sum_{m=1}^{M} \left( \frac{\|s_m - \tilde{s}_m\|}{\|s_m\|} \right)^2.$$  

(6)
In the noise-free case, this error is equal to the averaged residual cross-talk error (CTE): 

\[ CTE = \frac{1}{M} \sum_{m=1}^{M} \sum_{l=1}^{M} \frac{(\hat{A}^{-1}A)_{ml}^2}{\sum_{l=1}^{M} (\hat{A}^{-1}A)_{ml}^2} - \left( \max\{ (\hat{A}^{-1}A)_{m} \} \right)^2, \]  

where \( \max\{ (\hat{A}^{-1}A)_{m} \} \) is the largest element in the \( m \)-th raw of the matrix \( \hat{A}^{-1}A \).

Resulting separation errors for block sources are presented in the lower part of Figure 3. The largest error (11%) is obtained on the raw data, and the smallest (0.002%) — on the wavelet coefficients at the highest resolution, which have the best sparsity. Using all wavelet coefficients yields intermediate sparsity and performance.

### 3.2 Multinode representation

Our choice of a particular wavelet basis and of the sparsest subset of coefficients was obvious in the above example: it was based on knowledge of the structure of piecewise constant signals. For sources having oscillatory components (like sounds or images with textures), other systems of basis functions, such as wavelet packets [7], [25], trigonometric function libraries [26], or multiwavelets [8] might be more appropriate. In particular, the wavelet packet library consists of the triple-indexed family of functions:

\[ \varphi_{j,i,n}(t) = 2^{j/2} \varphi_n(2^j t - i), \quad j, i \in \mathbb{Z}, \ n \in \mathbb{N}. \]  

where \( j, i \) are the scale and shift parameters, respectively, and \( n \) is the frequency-like parameter. [Roughly speaking, \( n \) is proportional to the number of oscillations of a mother wavelet \( \varphi_n(t) \).] These functions form a binary tree whose nodes are indexed by the depth of the level \( j \) and the node number \( n = 0, 1, 2, 3, ..., 2^j - 1 \) at the specified level \( j \). This same indexing is applied to corresponding subsets of wavelet packet coefficients (Figure 4), and is used for the scatter diagrams in the section on experimental results.
3.3.1 Heuristic selection of best nodes

Usually, it is difficult to decide in advance which nodes, i.e. subsets of data, contain the sparsest sets of coefficients. Furthermore, there may be a situation wherein only one, or, more generally, not all of the sources 'live', that is, represented in one of the subspaces (at a
Figure 5: Wavelet Packets (WP) based multinode data structure: the example with mixture of 2 Flutes. Each node depicts scatter plot of WP coefficients of the sensor signals, wherein the coefficients are taken from the corresponding nodes of the WP tree (Figure 4).

particular node). In this case, the corresponding scatter plot will reveal only one or several dominant orientation. Such a node still can be very useful for separation if the points on the scatter plot are well concentrated.

One can apply the following heuristic approach for choosing the appropriate nodes [27]. First, for every node of the tree, we apply our clustering algorithm, and compute a measure of clusters’ distortion. In our experiments we used a standard global distortion, the mean squared distance of data points to the centers of their own, closest, clusters (here again, the weights of the data points can be incorporated):

$$d = \sum_{k=1}^{K} \min_m \| u_m - y_k \|_2,$$

where $K$ is the number of data points, $u_m$ is the $m$-th centroid coordinates, $y_k$ is the $k$-th data point coordinates, and $\|.\|_2$ is the $l_2$ norm, which is a sum-of-squares distance. Second, we choose a few best nodes with the minimal distortion, combine their coefficients into one data set and apply a separation algorithm, clustering or Natural Gradient, to these data. In the case of the wavelet packets, where two sets of children basis functions span the same subspace as their parent, we should check that there is no redundancy in the set of the best nodes, so that no children and their parents are present at the same time in the final set used for separation. For images, there are four children subsets of functions for each parent set, and therefore, each set of parent coefficients is split into four subsets of coefficients: approximations, horizontal, vertical and diagonal details.
Alternatively, the following iterative approach can be used. First, we apply the Wavelet Transform to original data, and apply a standard separation technique to these data in the transform domain. As such separation technique, we can use either the Natural Gradient, our clustering approach, or simply apply some optimization algorithm to minimize the corresponding log-likelihood function. This provides us with an initial estimate of the unmixing matrix $\mathbf{W}$ and with the estimated source signals. Then, at each iteration of the algorithm, we apply a multinode representation (for example, WP, or trigonometric library of functions) to the estimated sources, and calculate a measure of sparsity for each subset of coefficients. For example, one can consider the $l_1$ norm of the coefficients $\sum_{m,k} |c_{mk}|$, its modification $\sum_{m,k} \log |c_{mk}|$, or some other entropy-related measures.

Finally, we combine a new data set from the subsets with the highest sparsity (in particular, we can take for example, the best 10% of coefficients), and apply some separation algorithm to the new data. The iteration of the algorithm is completed. This process can be repeated till the convergence is achieved.

### 3.3.2 Error estimator

When signals have a complex nature, the heuristic approach may not be as robust as desired, and the error-related statistical quantities must be estimated. We use the following approach. First, we apply the Wavelet Transform to original data, and apply a standard separation technique (Natural Gradient, clustering, or optimization algorithm to minimize the log-likelihood function) to these data in the transform domain. Second, given the initial estimate of $\mathbf{W}$, and the subsets of data (coefficients of mixtures) for each node, we estimate the corresponding error variance, as described below. Finally, we choose a few best nodes (or, simply, the best one) with small estimated errors, combine their coefficients into one data set, and apply a separation algorithm to these data. In the rest of this section, we focus on the issues related to the estimation of error variance.

A robust estimate of the error covariance matrix can be derived using the second order Taylor expansion of the log-likelihood function. Let $\mathbf{W}_* = \mathbf{A}^{-1}$ be the exact solution of the BSS problem (1); it satisfies

$$
\mathbf{W}_* = \arg \max E \left[ \hat{L}_\mathbf{W}(\mathbf{Y}) \right].
$$

Further, let $\mathbf{W}_o$ be the estimate of $\mathbf{W}_*$, based on the particular realization of data, $\mathbf{Y}_o$, that is,

$$
\mathbf{W}_o = \arg \max \hat{L}_\mathbf{W}(\mathbf{Y}_o).
$$

Note, that $\nabla \hat{L}_{\mathbf{W}_o}(\mathbf{Y}_o) = 0$, while $\nabla \hat{L}_{\mathbf{W}_*}(\mathbf{Y}_o) \neq 0$ (though, $E \left[ \nabla \hat{L}_{\mathbf{W}_*}(\mathbf{Y}) \right] = 0$). We want to estimate the error, $\Delta \mathbf{W} = \mathbf{W}_* - \mathbf{W}_o$.

Using the linearization of $\nabla \hat{L}_\mathbf{W}(\mathbf{Y}_o)$ around $\mathbf{W}_o$,

$$
\nabla \hat{L}_{\mathbf{W}_o+\Delta \mathbf{W}}(\mathbf{Y}_o) \simeq \nabla \hat{L}_{\mathbf{W}_o}(\mathbf{Y}_o) + \nabla^2 \hat{L}_{\mathbf{W}_o}(\mathbf{Y}_o) \Delta \mathbf{W}
$$

we obtain

$$
\Delta \mathbf{W} \simeq \mathbf{H}^{-1} \nabla \hat{L}_{\mathbf{W}_o}(\mathbf{Y}_o),
$$
where \( \mathbf{H} = \nabla^2 \tilde{L}_\mathbf{w}_o (\mathbf{Y}_o) \) is the Hessian. Therefore, the error covariance matrix, whose diagonal elements correspond to the error variances of sources, can be approximated as

\[
E [ \Delta \mathbf{W} \Delta \mathbf{W}^T ] \simeq \mathbf{H}^{-1} \mathbf{\Sigma} (\mathbf{H}^{-1})^T,
\]

where

\[
\mathbf{\Sigma} = E \left[ \nabla \tilde{L}_\mathbf{w}_o (\mathbf{Y}) \nabla \tilde{L}_\mathbf{w}_o (\mathbf{Y})^T \right]
\]

is the covariance matrix of the gradient vector.

Further, denote by \( \nabla \tilde{L}_k \) the column stack vector of the \( k \)-th data point gradient

\[
\nabla \tilde{L}_k = \frac{\partial \tilde{L}_\mathbf{w}(\mathbf{c}_k)}{\partial \mathbf{w}} = (\mathbf{I} - \tilde{\psi}(\mathbf{c}_k)\mathbf{c}_k^T)(\mathbf{W}^T)^{-1},
\]

obtained from (5). Then, since

\[
\nabla \tilde{L}_\mathbf{w}(\mathbf{Y}) = \sum_{k=1}^{K} \nabla \tilde{L}_k,
\]

the covariance matrix \( \mathbf{\Sigma} \) can be estimated from \( K \) data points as:

\[
\hat{\mathbf{\Sigma}} = K \hat{\mathbf{\Gamma}},
\]

where

\[
\hat{\mathbf{\Gamma}} = \frac{1}{K} \sum_{k=1}^{K} \left[ \nabla \tilde{L}_k \nabla \tilde{L}_k^T \right].
\]

The Hessian matrix can be calculated either analytically, or via numerical evaluation (see [28] for details). In the latter case, its \( j \)-th column is approximated by:

\[
H_j(\mathbf{w}) \simeq \frac{\nabla L(\mathbf{w} + \xi \mathbf{e}_j) - \nabla L(\mathbf{w})}{\xi}
\]

where \( \mathbf{w} \) is a column stack version of \( \mathbf{W} \), \( \xi \) is a small constant and \( \mathbf{e}_j = [0, 0, ..., 1, 0, ..., 0]^T \) is the vector with the only nonzero entry at the \( j \)-th position.

The diagonal element \( \hat{\varepsilon}_{ii}^2 \) in the error covariance matrix above (9) represents an estimate of the squared error introduced to the reconstructed \( i \)-th source by cross-talks from other sources. The analytic expression for the mean square relative contamination of the reconstructed \( i \)-th source by the \( j \)-th source, for the quasi-ML estimator, is given by [10]:

\[
\hat{\varepsilon}_{ii}^2 = \frac{1}{K} \frac{\alpha_i^2 \alpha_j^2 (\beta_i^2 / \alpha_i^2 + \beta_j^2 - 2 \beta_i \beta_j / \alpha_i \alpha_j)}{1 - \alpha_i \alpha_j / \beta_i \beta_j},
\]

where \( K \) is the number of data samples, and, parameters \( \alpha \) and \( \beta \) are dependant on the source-related expectations. In our case, \( \alpha \) and \( \beta \) are coefficient-related:

\[
\alpha_i = \frac{E[\tilde{\psi}_i(\mathbf{c}_i)\mathbf{c}_i]}{E[\tilde{\psi}_i(\mathbf{c}_i)]E[\mathbf{c}_i^2]}, \quad \beta_i = \frac{E[\tilde{\psi}_i^2(\mathbf{c}_i)\mathbf{c}_i]}{\sqrt{E[\tilde{\psi}_i^2(\mathbf{c}_i)]E[\mathbf{c}_i^2]}}
\]

Note, that the expectation operator \( E[\cdot] \) is with respect to the true pdf \( f_{\mathbf{C}} \), while \( \tilde{\psi}(\mathbf{c}) = -(\log \tilde{f}_{\mathbf{C}})^\prime \) is dependent on the hypothetical pdf \( \tilde{f}_{\mathbf{C}} \).
4. Experiments

For reasons given in the previous section, the hypothetical density is used in the expression of the log-likelihood function (4). In our experiments we use the function \( \nu(c_{mk}, \zeta) = \sqrt{c_{mk}^2 + \zeta} \), as a smooth approximation of the absolute value function.

4.1 Numerical results: theoretic and estimated separation errors

In the following experiment, we study the effect of using the quasi log-likelihood instead of the likelihood function on the theoretical error performance. For this purpose, we evaluate expectations in (11) via numerical calculation of the corresponding integrals. Then, for various values of the smoothness parameter \( \zeta \), we calculate the error variance according to (10) with the sample size \( K = 10^4 \), as a function of the true density shape parameter, \( q^* \). The results of this calculation are presented in Figure 6. Note, that the error variance drops dramatically as the sparseness of sources increases. This figure shows also that using hypothetical density has a negligible effect on the theoretical error.  

![Figure 6: Theoretical cross-talk error variance: Using the true density (\( q_{th} = q^* \)) vs. the hypothetical density (\( q_{th} = 1, \zeta = 0, 10^{-2}, 10^{-4} \)) for the ML estimator.](image)

4.1.1 Synthetic exponential sources

In the following experiment, we generate two source signals with \( 10^4 \) samples each, drawn from the exponential distribution \( f_s \sim \exp\{-\nu(s, q)\} \) with \( q^* = 0.5 \). These two signals are mixed together with a 2x2 matrix of normally distributed random numbers. Consequently, the matrix is normalized for the purpose of the error calculation. In order to separate this
signals, we apply the following algorithms: 1) the original Natural Gradient with the built-in non-linearity, as implemented in the ICA/EEG Matlab toolbox [23]; 2) the modified Natural Gradient with the non-linearity corresponding to our quasi log-likelihood function, with parameters \( q_{th} = 1 \), and \( \zeta = 10^{-4} \), that is, using \( \nu(c_{mk}, \zeta) = \sqrt{c_{mk}^2 + 10^{-4}} \); 3) the Matlab function \( \text{fminu} \) (an implementation of the BFGS Quasi-Newton optimization method [29]), applied to optimize our quasi log-likelihood function with the above parameter values. The following error-related quantities were calculated: 1) the error variance \( \hat{\varepsilon}^2 \), estimated from data according to (10) by evaluation of corresponding expectations via averaging; 2) the error variances calculated according to (9); these are evaluated for each one of the above three algorithms; 3) actual squared separation errors of the algorithms; 4) theoretical error variance, calculated as described above. Table 1 summarizes the results of the above experiment; the corresponding values are averaged over the trials. The smallest actual separation error is achieved by using the \( \text{fminu} \) function. The modified Natural Gradient outperforms the original one, as expected, since its hypothetical pdf is ‘closer’ to the true pdf of sources. Also, the proposed error estimate (9) is closer to the theoretical error variance than the estimate according to (10).

<table>
<thead>
<tr>
<th></th>
<th>fminu</th>
<th>Mod. NG</th>
<th>Orig. NG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual sq. error</td>
<td>8e-5</td>
<td>1.2e-4</td>
<td>4.1e-4</td>
</tr>
<tr>
<td>Error var. estimate (10)</td>
<td>1.6e-4</td>
<td>3.8e-4</td>
<td>5e-4</td>
</tr>
<tr>
<td>Error var. estimate (9)</td>
<td>7.8e-5</td>
<td>7.8e-5</td>
<td>2e-4</td>
</tr>
<tr>
<td>Theoretical error variance</td>
<td>6.6e-5</td>
<td>6.6e-5</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1: Actual squared cross-talk errors and the error variance estimates

### 4.1.2 Natural sources

In the following experiment we apply the \( \text{fminu} \) function (as described above) to estimate \( W \) from each one of 22 data subsets formed by the WP coefficients of mixture of natural images. In Figure 7, we depict the following quantities, for each one of the 22 data subsets: the actual squared separation error and its estimate according to (9). The first subset corresponds to the complete set of the Wavelet Transform (WT) coefficients; the other subsets are indexed on the WP tree. Our ‘error predictor’ provides quite accurate estimates of the actual errors. Note, that for the best, the 5-th, subset, the separation error and its estimate are smaller by several orders, as compared to the corresponding quantities of the 1-st set, the complete set of the WT coefficients.

### 4.2 Separation of simulated and real signals and images

The proposed blind separation method based on the wavelet-packet representation, was evaluated by using several types of signals. We have already discussed the relatively simple example of a random block signal. The second type of signal is a frequency modulated (FM) sinusoidal signal. The carrier frequency is modulated by either a sinusoidal function (FM signal) or by random blocks (BFM signal). The third type is a musical recording of flute sounds. Finally, we apply our algorithm to images. An example of such images is presented...
Figure 7: The actual squared cross-talk error \([-\cdot]\), and its estimate (9) \([\ldots]\), evaluated for each one of 22 data sets formed by the corresponding subsets of WP coefficients.

in Figure 8. Source images and their mixtures are shown at the upper two sets of plots, and the estimated images are shown in the lower two plots.

In order to compare accuracy of our multiscale BSS method with that attainable by standard methods, we form the following feature sets: (1) raw data, (2) Short Time Fourier Transform (STFT) coefficients (in the case of 1D signals), (3) Wavelet Transform coefficients (4) Wavelet packet coefficients at the best nodes found by the proposed error estimator (9), while using various wavelet families with different smoothness (haar, db-4, db-8). In the case of image separation, we used the Discrete Cosine Transform (DCT) instead of STFT, and the sym4 and sym8 mother wavelet instead of db-4 and db-8, when using wavelet transform and wavelet packets.

4.2.1 Understanding scatter plot and PDF diagrams

Let us consider an example of image separation from two mixtures of two sources (Figure 8). Figure 9 shows corresponding scatter plots of the wavelet packet coefficients of mixtures. These scatter plots correspond to the various nodes of the wavelet packet tree. The upper left scatter plot, marked with 'C', corresponds to the complete set of coefficients at all nodes. The rest are the scatter plots of sets of coefficients indexed on a wavelet packet tree. Generally speaking, the more distinct the two dominant orientations appear on these plots, the more precise is the estimation of the mixing matrix, and, therefore, the better is the quality of separation. Note, that only two nodes (the most left ones in the second from the bottom row) show clear orientations. These nodes will most likely be selected by the algorithm for further estimation process.

Figure 10 shows distributions of angles (orientations) formed by points on the corresponding scatter plots of the wavelet packet coefficients at various nodes. Here, again, the sharper are the picks of a distribution, the better is the separation.
4.2.2 Separation from noise-free mixtures

Table 2 summarizes results of experiments in which we applied the (original) Natural Gradient to each noise-free feature set. In the case of the WP features, the best subset was selected, using the proposed error estimator (9). In these experiments, we compared the quality of separation of deterministic signals by calculating NSE’s, or residual crosstalk errors, according to (6). In the case of random block and BFM signals, we performed 100 Monte-Carlo simulations and calculated the normalized mean-squared errors (NMSE) for the above feature sets.

From Table 2 it is clear that using our best nodes method, implemented with the proposed error estimator, outperforms all other feature sets, including complete set of wavelet coefficients, for each type of signals. Similar improvements were achieved by using the FCM algorithm along with the heuristic data subset selection. In the case of the random block signals, using the Haar wavelet function for the wavelet packet representation yields a better separation than using some smooth wavelet, e. g. db-8. The reason is that these block signals, that are not natural signals, have a sparser representation in the case of the Haar
Figure 9: Scatter plots of the wavelet packet (WP) coefficients of mixtures of two images; subsets are indexed on the WP tree.

Figure 10: Distributions of angles (orientations) characterizing the scatter diagrams of the WP coefficients of mixtures of two images.

wavelets. In contrast, as expected, natural signals such as the Flute’s signals are better represented by smooth wavelets, that in turn provide a better separation. This is another
Table 2: Normalized squared cross-talk errors [%]: Applying Natural Gradient-based separation to raw data and decomposition coefficients in various domains. In the case of wavelet packets (WP), the best nodes, selected by our error predictor, were used.

4.2.3 Separation from noisy mixtures

In order to verify the performance of our method in presence of noise, we added various types of noise (white Gaussian and salt&pepper) to three mixtures of three images at various SNR’s. Table 3 summarizes these experiments in which we applied our approach along with the separation via the modified Natural Gradient algorithm.

Table 3: Performance of the algorithm in presence of various sources of noise in mixtures: Normalized mean-squared (NSE) and cross-talk (CTE) errors for image separation, applying our multiscale adaptive approach along with the Natural Gradient based separation.

An example of ‘difficult’ source separation from noisy mixtures with Gaussian noise is shown in Figure 11. It turns out that the ideas used in wavelet based signal denoising (see for example [30] and references therein), are applied to signal separation from noisy mixtures. In particular, in case of white Gaussian noise, the noise energy is uniformly distributed over all wavelet coefficients at various scales. Therefore, at sufficiently high signal-to-noise energy ratios (SNR), the large coefficients of the signals are only slightly distorted by the noise coefficients. As a result, the presence of noise has a minor effect on the estimation of
the unmixing matrix (see the CTE entries in Table 3). Note, that the NSE entries reflect the noise energy passed to the reconstructed sources from the mixtures. Our algorithm provides reasonable separation quality (CTE of about 4%) for SNR’s of about 10 and higher. On the contrast, the Natural Gradient algorithm applied to the noisy mixtures themselves, failed completely to separate source images, arriving at CTE of 47% even in the case of SNR=30. We should stress here that, although our adaptive best nodes method performs reasonably well even in the presence of noise, it is not supposed to further denoise the reconstructed images. The post-filtering can be achieved by some denoising method, after initial source signals are separated. For example, in Figure 11, a simple wavelet denoising method from [30] was applied to separated images.

5. Conclusions

The proposed method improves the separation quality by utilizing the structure of signals projected onto a proper space, wherein certain subsets of the wavelet packet coefficients depict significantly better sparsity and, therefore, contribute to better separability than others. Other multiresolution representations based on wavelet-type, multi-wavelets, and nonseparable two-dimensional wavelets, can most likely provide better results in specific cases of image subspaces. The approach depicted in this study using the specific example of wavelet packets can then be adapted to the other representations in a straightforward manner, by constructing the appropriate multinode tree. In addition, the block partitioning of the image structure into subsets of projected data, can also be incorporated into the proposed multinode framework of BSS, to better exploit the localized spatial-varying properties of images as well as those of the imaging system.

Numerical and simulation results demonstrate that sparsity of decomposition coefficients has a major effect on the separation quality. In contrast, using the hypothetical log-likelihood function, that is, a smooth approximation of the likelihood derived from the actual pdf of sources, only slightly degrades the performance.

The error variance estimator, based on the Taylor expansion of the quasi log-likelihood function, provides an efficient way for selection of the best subset(s) of coefficients and, in fact, provides a reasonable metric for the vaguely defined concept of optimal selection of subsets of coefficients. Once this subset is selected, the mixing matrix is estimated using only this new subset of coefficients, by the Natural Gradient algorithm or by clustering.

Experiments with both one- and two-dimensional simulated and natural signals demonstrate that multinode sparse representations significantly improves the efficiency and quality of blind source separation.

Acknowledgments

We thank Arkadi Nemirovski for useful suggestions. This work is supported in part by the Ollendorff Minerva Center, by the Israeli Ministry of Science. The research of YYZ is currently supported also by the Medical Imaging Group, Dept. of Biomedical Eng. Columbia University and by the ONR-MURI Grant N00014-01-1-0625.
References


Figure 11: Applying Multiscale BSS to noisy mixtures: three source images (1st row), their mixtures (2nd row), the mixtures with additive white Gaussian noise (3d row), separated images (4th row), and post-filtered separated images (5th row)