NONLINEAR BLIND SOURCE SEPARATION USING KERNEL FEATURE SPACES

Stefan Harmeling[1*], *Andreas Ziehe*[1], *Motoaki Kawanabe*[1], *Benjamin Blankertz*[1], *Klaus-Robert Müller*[1,2]

[1]GMD FIRST.IDA, Kekuléstr. 7, 12489 Berlin, Germany
[2]University of Potsdam, Am Neuen Palais 10, 14469 Potsdam,Germany
{harmeli,ziehe,kawanabe,blanker,klaus}@first.gmd.de

ABSTRACT

In this work we propose a kernel-based blind source separation (BSS) algorithm that can perform nonlinear BSS for general invertible nonlinearities. For our kTDSEP algorithm we have to go through four steps: (i) adapting to the intrinsic dimension of the data mapped to feature space $\mathcal{F}$, (ii) finding an orthonormal basis of this submanifold, (iii) mapping the data into the subspace of $\mathcal{F}$ spanned by this orthonormal basis, and (iv) applying temporal decorrelation BSS (TDSEP) to the mapped data. After demixing we get a number of irrelevant components and the original sources. To find out which ones are the components of interest, we propose a criterion that allows to identify the original sources. The excellent performance of kTDSEP is demonstrated in experiments on nonlinearly mixed speech data.

1. INTRODUCTION

Linear blind source separation has been successful in various applications ([12, 5, 4, 7, 1, 2, 19, 10, 25, 9]). Recently a new line of research has emerged that focuses on nonlinear mixings. It has so far only been applied to industrial pulp data [9], but a large class of applications where nonlinearities can occur in the mixing process are conceivable, e.g. in the fields of telecommunications, array processing, biomedical data analysis (EEG, MEG, EMG, . . .) and acoustic source separation. Various methods have been proposed for solving nonlinear mixings, e.g. self-organizing maps [17, 13], extensions of GTM [18], neural networks [23, 14], ensemble learning [21] or correlation maximization using ACE [24]. Note, that most methods except [24] use high computational cost and depending on the algorithm are prone to run into local minima. The simplest scenario is the so-called post-nonlinear BSS

$$\mathbf{x}[t] = \mathbf{f}(\mathbf{A}\mathbf{s}[t]), \tag{1}$$

where $\mathbf{x}[t]$ and $\mathbf{s}[t]$ are $n \times 1$ column vectors, $\mathbf{A}$ is an $n \times n$ matrix and $\mathbf{f}$ is a nonlinear function that operates componentwise [20].

The general nonlinear BSS problem, that we will address in this paper, has an even more challenging setup. Here, the mixing model reads

$$\mathbf{x}[t] = \mathbf{f}(\mathbf{s}[t]) \tag{2}$$

and $\mathbf{f}$ is an (at least approximately invertible) nonlinear function from $\Re^n$ to $\Re^n$. First algorithms for this problem[1] that are based on the idea of kernel based learning (cf. e.g. [22, 6, 16]) were only tried on toy signals [8]. The difference between our kTDSEP algorithm and [8] lies mainly in the manner and the superior efficiency in which the kernel feature space is constructed and used for unmixing (our approach considers temporal decorrelation). This eventually allows to demix large, real-world data sets that are nonlinearly mixed according to Eq. (2).

Let us first introduce the basic ideas of kernel based methods that are needed for our algorithm. For input vectors $\mathbf{x}[t] \in \Re^n$ $(t = 1 \ldots T)$ from an input space a kernel function $\mathbf{k} : \Re^n \times \Re^n \to \Re$ that fulfills certain conditions (cf. [16]) induces a mapping $\Phi : \Re^n \to \mathcal{F}$ into some feature space $\mathcal{F}$ such that the dot product for points in the image of Φ can be simply calculated using the kernel function (often called the kernel trick),

$$\mathbf{k}(\mathbf{x}[i], \mathbf{x}[j]) = \Phi(\mathbf{x}[i]) \cdot \Phi(\mathbf{x}[j]). \tag{3}$$

By using linear algorithms in feature space, nonlinear problems in input space can be solved efficiently and elegantly. To solve nonlinear BSS problems one could apply along these lines a linear BSS algorithm to the mapped data in feature space. This would give us some direction $\mathbf{w} \in \mathcal{F}$ that corresponds to a nonlinear direction in input space. Such a direction is parameterized, as usual for kernel methods, by a $T \times 1$ vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_T)^\top \in \Re^T$ such that

$$\mathbf{w} = \Phi_{\mathbf{x}}\boldsymbol{\alpha} = \sum_{j=1}^{T} \alpha_j \Phi(\mathbf{x}[j]) \in \mathcal{F},$$

[*] To whom correspondence should be addressed. The authors thank Gunnar Rätsch and Sebastian Mika for valuable discussions. This work was partly supported by the EU project (IST-1999-14190 – BLISS) and DFG (JA 379/9-1, MU 987/1-1).

[1] Note, that in fact, it is only possible to extract the sources up to an arbitrary invertible transformation (cf. [11]).

where $\Phi_{\mathbf{x}}$ is the matrix with the column vectors $\Phi(\mathbf{x}[1])$, ... , $\Phi(\mathbf{x}[T])$. Using the kernel trick (Eq. (3)) we can calculate the real valued $T \times T$ matrix

$$\Phi_{\mathbf{x}}^{\top}\Phi_{\mathbf{x}} = (\mathbf{k}(\mathbf{x}[i], \mathbf{x}[j]))_{ij} \quad \text{where } i, j = 1 \dots T$$

which we use to compute the signal that corresponds to the nonlinear direction $\mathbf{w}$

$$y[t] = \mathbf{w}^{\top}\Phi(\mathbf{x}[t]) = \boldsymbol{\alpha}^{\top}\Phi_{\mathbf{x}}^{\top}\Phi(\mathbf{x}[t]) = \sum_{j=1}^{d} \alpha_j \mathbf{k}(\mathbf{x}[j], \mathbf{x}[t])$$

without having actually to specify the mapping Φ. However, T is the number of samples and since T is for BSS problems quite large, such a parameterization leads to feasibility and stability problem. In this paper we introduce a new algorithm that overcomes these problems and that performs nonlinear BSS.

2. A NEW ALGORITHM FOR NONLINEAR BSS

The image of the input space $\Re^n$ under Φ is a manifold that is contained in a d dimensional subspace of $\mathcal{F}$. The key for our algorithm is to find an orthonormal basis for this subspace that enables us to parameterize the signals in feature space efficiently with vectors in a d dimensional parameter space $\Re^d$ (cf. Fig. 1). Based on TDSEP that uses temporal decorrelation (cf. [25, 3]) we use this orthonormal basis to construct a new nonlinear BSS algorithm. This new algorithm is denoted as kTDSEP (kernel TDSEP). kTDSEP requires four steps:

(i) We start with determining d: randomly choose d input vectors $\mathbf{v} := \mathbf{v}_1, \dots, \mathbf{v}_d$ from $\mathbf{x}[1], \dots, \mathbf{x}[T]$ and check whether the columns of the matrix $\Phi_{\mathbf{v}} := (\Phi(\mathbf{v}_1)$, ... ,$\Phi(\mathbf{v}_d))$ form a maximally independent system in $\mathcal{F}$ (i.e. whether they form a basis for the image of the input space under Φ). In order to do that we calculate the rank of the real-valued $d \times d$ matrix

$$\Phi_{\mathbf{v}}^{\top}\Phi_{\mathbf{v}} = (\mathbf{k}(\mathbf{v}_i, \mathbf{v}_j))_{ij} \quad \text{for } i, j = 1, \dots, d.$$

We repeat this random sampling process with varying d until we have found a d such that there are d input vectors $\mathbf{v}$ for which the matrix $\Phi_{\mathbf{v}}^{\top}\Phi_{\mathbf{v}}$ has full column rank, i.e. has rank d, and we can not find $d+1$ input vectors $\mathbf{v}$ for which the associated matrix $\Phi_{\mathbf{v}}^{\top}\Phi_{\mathbf{v}}$ has full column rank as well[2].

(ii) Next we define an orthonormal basis for the subspace of $\mathcal{F}$ that contains the image of Φ. Either use random sampling like in (i) or use k-means clustering to obtain d input vectors $\mathbf{v}$ for which the matrix $\Phi_{\mathbf{v}}^{\top}\Phi_{\mathbf{v}}$ has full column rank. Note, that it is not important that the chosen vectors $\mathbf{v}$ are among the input vectors $\mathbf{x}[1], \dots, \mathbf{x}[T]$. With the images $\Phi_{\mathbf{v}}$ of these vectors we construct an orthonormal basis,

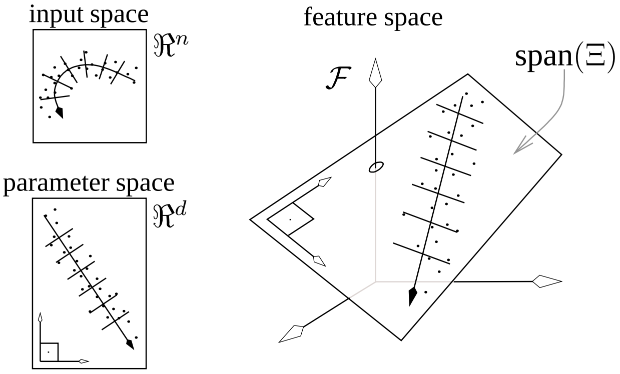

Fig. 1: Input data are mapped to some submanifold of $\mathcal{F}$ which is the span of some d-dimensional orthonormal basis Ξ. Therefore these mapped points can be parameterized in $\Re^d$. The linear directions in parameter space correspond to nonlinear directions in input space.

$$\Xi := \Phi_{\mathbf{v}}(\Phi_{\mathbf{v}}^{\top}\Phi_{\mathbf{v}})^{-\frac{1}{2}}.$$

This basis enables us to parameterize the subspace that contains the mapped input vectors in feature space with vectors from a d dimensional parameter space $\Re^d$ as we will see in the next step.

(iii) After scaling the observed signals $\mathbf{x}[t]$ such that their absolute maximum is smaller than one (later in this section we will see why this is useful) we employ this basis to map the input signals $\mathbf{x}[t]$ to real-valued d dimensional signals $\Psi(\mathbf{x}[t])$ in parameter space,

$$\Psi(\mathbf{x}[t]) := \Xi^{\top}\Phi(\mathbf{x}[t]) = (\Phi_{\mathbf{v}}^{\top}\Phi_{\mathbf{v}})^{-\frac{1}{2}}\Phi_{\mathbf{v}}^{\top}\Phi(\mathbf{x}[t])$$
$$= ((\mathbf{k}(\mathbf{v}_i, \mathbf{v}_j))_{ij})^{-\frac{1}{2}}(\mathbf{k}(\mathbf{v}_i, \mathbf{x}[t]))_i \quad \text{for } i, j = 1, \dots, d.$$

Note that by construction $(\Phi_{\mathbf{v}}^{\top}\Phi_{\mathbf{v}})^{-\frac{1}{2}}$ is an invertible real valued $d \times d$ matrix and $\Phi_{\mathbf{v}}^{\top}\Phi(\mathbf{x}[t])$ is a real valued $d \times 1$ vector. Both are computed using the kernel trick (Eq. (3)) without explicitely specifying the mapping $\Phi : \Re^n \to \mathcal{F}$.

(iv) Finally, we apply temporal decorrelation (TDSEP, [25]) to $\Psi(\mathbf{x}[t])$ which gives us d linear directions in parameter space that correspond to d nonlinear directions in input space. The solutions are parameterized by a real-valued $d \times d$ matrix $\boldsymbol{\alpha} \in \Re^{d \times d}$. The corresponding demixed signals are simply the product of $\boldsymbol{\alpha}$ and $\Psi(\mathbf{x}[t])$,

$$\mathbf{y}[t] := \boldsymbol{\alpha}\Psi(\mathbf{x}[t]) \in \Re^d.$$

[2] Clearly, this is not possible for all kernel functions. However, throughout this paper we consider only polynomial kernels.

Most of these signals are irrelevant. To pick the signals of interest we use a heuristic: by ensuring $-1 < \mathbf{x}[t] < 1$ we influence the variance of the unwanted signals because the latter contain higher order versions of the source signals as we will see in the next section. Therefore, after normalizing all signals (such that they have zero mean and their absolute maximum is one) the demixed source signals are the ones with the highest variance.

3. ANALYSIS OF A TOY EXAMPLE

To give some clue how our algorithm works we take a detailed look at a toy example: for

$$\mathbf{A} = \begin{bmatrix} -1.2173 & -1.1283 \\ -0.0412 & -1.3493 \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} -0.2611 \\ 0.9535 \end{bmatrix}$$

let $\mathbf{x}[t] = \mathbf{A}(s_1[t], s_2[t])^\top + \mathbf{b}s_1[t]s_2[t]$ be a simple nonlinear mixture (taken from [15]). For the kernel function $\mathbf{k}(\mathbf{x},\mathbf{y}) = (\mathbf{x}^\top\mathbf{y}+1)^2$, a polynomial kernel of degree 2, we can explicitly write down the mapping Φ from input space to feature space (cf. [16]),

$$\Phi(\mathbf{x}) = (x_1^2,\ x_1x_2,\ \sqrt{2}x_1,\ x_2^2,\ \sqrt{2}x_2)^\top.$$

Note, that we omitted the dimension in which $\Phi(\mathbf{x})$ is constant. Since the feature space is $\Re^5$ we do not have to consider an orthonormal basis Ξ. Denote by

$$\mathbf{q} := (s_1^2s_2^2,\ s_1^2s_2,\ s_1^2,\ s_1s_2^2,\ s_1s_2,\ s_1,\ s_2^2,\ s_2)^\top$$

the monomials of the source signals that appear as linear combinations in the feature space. We call these monomials quasi sources. A simple calculation gives us a real-valued 5×8 matrix $\mathbf{C} = \mathbf{D}\,\mathbf{C}_0$, where

$$\mathbf{C}_0^\top = \begin{pmatrix} b_1^2 & b_1b_2 & 0 & b_2^2 & 0 \\ 2a_{11}b_1 & a_{21}b_1{+}a_{11}b_2 & 0 & 2a_{21}b_2 & 0 \\ a_{11}^2 & a_{11}a_{21} & 0 & a_{21}^2 & 0 \\ 2a_{12}b_1 & a_{22}b_1{+}a_{12}b_2 & 0 & 2a_{22}b_2 & 0 \\ 2a_{11}a_{12} & a_{22}a_{11}{+}a_{12}a_{21} & b_1 & 2a_{21}a_{22} & b_2 \\ 0 & 0 & a_{11} & 0 & a_{21} \\ a_{12}^2 & a_{12}a_{22} & 0 & a_{22}^2 & 0 \\ 0 & 0 & a_{12} & 0 & a_{22} \end{pmatrix}$$

with $\mathbf{A} = (a_{ij})$, $\mathbf{b} = (b_i)$ and $\mathbf{D} = \operatorname{diag}(1, \sqrt{2}, \sqrt{2}, 1, \sqrt{2})$ such that we can expand the mixture in feature space linearly in terms of the quasi sources,

$$\Phi(\mathbf{x}[t]) = \mathbf{C}\,\mathbf{q}[t]. \tag{4}$$

At first view, this situation looks like a mixture of an overcomplete basis that might hardly be solved by TDSEP. Fortunately, most quasi sources are pairwise correlated: for two independent signals s_1 and s_2 the correlation between arbitrary monomials in s_1 and s_2 is

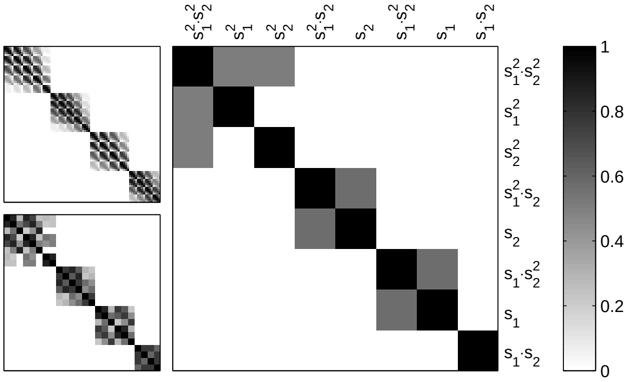

Fig. 2: Most quasi sources are pairwise correlated; the middle panel shows the covariance matrix of the quasi sources resulting from a polynomial kernel of degree 2, the lower left panel for degree 4 and the upper left panel for degree 8. Note, that the quasi sources can always be collocated into four groups.

$$\begin{aligned} &\operatorname{corr}(s_1^{k_1}s_2^{m_1}, s_1^{k_2}s_2^{m_2}) = \\ &\quad \frac{\operatorname{cov}(s_1^{k_1}s_2^{m_1}, s_1^{k_2}s_2^{m_2})}{\prod_{i=1,2}\sqrt{\operatorname{var}(s_1^{k_i}s_2^{m_i})}} = \\ &\frac{\mathrm{E}\{s_1^{k_1+k_2}\}\,\mathrm{E}\{s_2^{m_1+m_2}\} - \mathrm{E}\{s_1^{k_1}\}\,\mathrm{E}\{s_1^{k_2}\}\,\mathrm{E}\{s_2^{m_1}\}\,\mathrm{E}\{s_2^{m_2}\}}{\prod_{i=1,2}\sqrt{\mathrm{E}\{s_1^{2k_i}\}\,\mathrm{E}\{s_2^{2m_i}\} - (\mathrm{E}\{s_1^{k_i}\}\,\mathrm{E}\{s_2^{m_i}\})^2}}. \end{aligned}$$

Since the moments of normally distributed signals s_1 and s_2 are (with mean zero and variance one)

$$\mathrm{E}\{s_1^k\} = \begin{cases} 1 \cdot 3 \cdots (k-1) & \text{if } k \text{ is even} \\ 0 & \text{if } k \text{ is odd} \end{cases}$$

we get for such signals

$$\operatorname{corr}(s_1^{k_1}s_2^{m_1}, s_1^{k_2}s_2^{m_2}) = 0 \tag{5}$$

if $k_1 + k_2$ is odd or $m_1 + m_2$ is odd. Therefore the quasi sources can be collocated into four groups with no correlations between the groups; e.g. for a polynomial of degree 2 the four groups are (cf. Fig. 2),

$$\{s_1^2s_2^2, s_1^2, s_2^2\},\ \{s_1^2s_2, s_2\},\ \{s_1s_2^2, s_1\},\ \{s_1s_2\}.$$

Consequently, the mixture in Eq. (4) is not overcomplete.

Next we describe the signals that our algorithm extracts. Consider two sinusoidal source signals $(s_1, s_2)^\top$ that are nonlinearly mixed using the above mixture. For a polynomial kernel function of degree 4,

$$\mathbf{k}(\mathbf{x},\mathbf{y}) = (\mathbf{x}^\top\mathbf{y}+1)^4,$$

there are twenty-four quasi sources: all possible products of s_1, s_1^2, s_1^3, s_1^4 and their counterparts in s_2. Using Eq. (5)

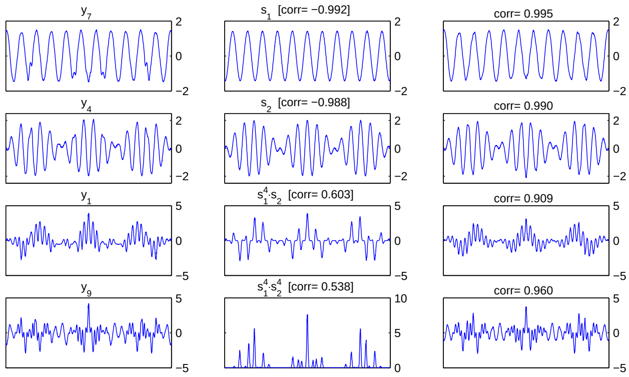

Fig. 3: The extracted signals in the left panels (only four shown) are tried to be matched with single quasi sources in the middle panels and combinations of subgroups of quasi sources (right panels).

these quasi sources can also be arranged into four groups with no correlations between the groups. Applying kTDSEP with that kernel function we computed $d = 15$ (dimension of the parameter space) and extracted all fifteen signals. Now we try to explain those signals using the quasi sources that belong to the used kernel. Four of the extracted signals (y_7, y_4, y_1, y_9) are shown in the left panels of Fig. 3. The middle panels show the best matching quasi sources. Note, that the true sources, s_1 and s_2, have a very high correlation to their left neighbors, y_7 and y_4, respectively. The other extracted signals, y_1 and y_9, do not have a very high correlation to any of the quasi source signals: the best fits, $s_1^4 s_2$ and $s_1^4 s_2^4$, are plotted in the two lower middle panels. The extracted signals can better be explained with linear combinations of subsets of mutually correlated quasi sources. Therefore, we combined all quasi sources that are correlated with $s_1^4 s_2^4$ to reconstruct y_9. The result is shown in the lower right panel which reaches a good fit (corr $= 0.960$), similarly for y_1 and the other not shown extracted signals. Note, that for y_7 and y_4 that matched s_1 and s_2 already reasonably well more quasi sources do not improve the result notably.

It remains the question why the sought-after source signals appear so well among the extracted signals without much interference from their correlated quasi sources. The answer has two parts: to begin with, s_1 and s_2 usually have the largest variance among the other quasi sources of their respective groups. When this is not the case (e.g. for very large b_1 and b_2 in our mixture) our algorithm can fail. Secondly, we experienced in our experiments problems if $\mathbf{x}$ is not scaled between -1 and 1. We think the reason for this behavior is that by scaling $\mathbf{x}$ between -1 and 1 we assure that the higher order monomials introduced by most of the components of Φ have smaller variance than the components containing x_1 and x_2 and hereby also favorably influencing the ratio between the variances of s_1 and s_2 and other quasi sources. Note, that this implies that kTDSEP is not scale invariant which is, however, no real problem since we can always ensure $-1 < \mathbf{x} < 1$.

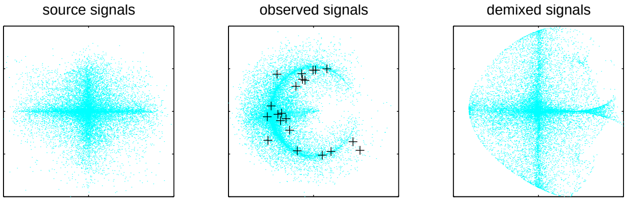

Fig. 4: The left panel shows a scatterplot of the source signals, the middle panel a scatterplot of the nonlinearly mixed signals and the right panel the unmixed extracted components that were chosen by calculating the variance of the normalized signals.

4. EXPERIMENT WITH SPEECH DATA

Consider two speech signals (with 16,000 samples, sampling rate 8 kHz, each ranging between -1 and $+1$) that are nonlinearly mixed by

$$\begin{aligned} x_1[t] &= -(s_2[t] + 1)\cos(\pi s_1[t]) \\ x_2[t] &= 1.5\,(s_2[t] + 1)\sin(\pi s_1[t]). \end{aligned}$$

This mixture is highly nonlinear (cf. Fig. 4; it transforms polar into Cartesian coordinates), but kTDSEP succeeds because s_2 appears linearly in x_1 and s_1 appears linearly in x_2 (to see this expand cosine and sine into their series). Linear TDSEP fails to extract both signals: the second source (that appears as the radius in the mixture) can linearly not be reconstructed. We applied kTDSEP with a polynomial kernel of degree 5,

$$\mathbf{k}(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^5,$$

calculated $d = 21$ to be the dimensionality of the parameter space and obtained the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{21} \in \Re^2$ by k-means clustering. These points are marked as $+$ in the middle panel of Fig. 4. An application of TDSEP to the twenty-one dimensional parameter space yields nonlinear components whose projections to the input space are depicted in Fig. 5. Note, that the third and the eighth extracted signals reach very high correlations with s_1 and s_2 (corr $= 0.963$, corr $= 0.989$). To select these two signals among the twenty-one extracted components in an unsupervised manner we use the above mentioned heuristic approach that calculates the variances of the normalized signals (mean equal to zero and absolute maximum equal to one). In Fig. 5 the right column shows a horizontal bar plot of these variances. The two signals of interest are clearly highlighted through their large variances.

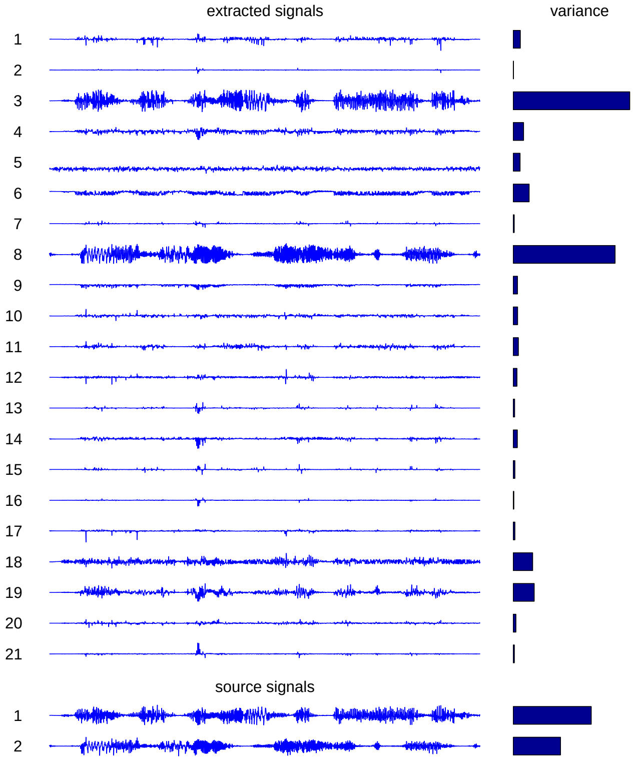

Fig. 5: On the left side are the extracted components and the source signals; the horizontal bars on the right side indicate the variance of the corresponding signals after normalization (mean is zero and absolute maximum is one). The third and the eighth signals are clearly highlighted through their large variances.

5. CONCLUSION

This paper proposes the kTDSEP algorithm for nonlinear BSS based on support vector kernels. It follows a series of steps: first we map the data into kernel feature space $\mathcal{F}$ where we try to compute the intrinsic dimension d of the mapped data. Then we construct an orthonormal basis of this d dimensional submanifold in $\mathcal{F}$ and apply temporal decorrelation BSS (TDSEP). The rationale behind this is that the mapping to feature space is constructed such that the nonlinear separation in input space becomes a (simple) *linear* separation in $\mathcal{F}$. Note, that as we are using the kernel trick (Eq. (3)) we can avoid to work directly in $\mathcal{F}$. Afterwards, TDSEP makes use of the *temporal glue* in the original sources and extracts d signals from which we pick the components of interest by employing a variance based criterion. A set of experiments on toy and speech signals underline that an elegant algorithm has been found to a challenging problem.

Applications where nonlinearly mixed signals occur are perceived e.g. in the fields of telecommunications, array processing, biomedical data analysis and acoustic source separation. In fact, our algorithm would allow a software-based correction of sensors that have nonlinear characteristics, e.g. due to manufacturing errors. Clearly, kTDSEP is only one BSS algorithm that can perform nonlinear BSS; kernelizing other ICA/BSS algorithms will be left for future work.

6. REFERENCES

[1] S.-I. Amari, A. Cichocki, and H.H. Yang. A new learning algorithm for blind source separation. In *Advances in Neural Information Processing Systems 8*, pages 757–763. MIT Press, 1996.

[2] A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[3] A. Belouchrani, K. Abed Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique based on second order statistics. *IEEE Trans. on Signal Processing*, 45(2):434–444, 1997.

[4] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non Gaussian signals. *IEE Proceedings-F*, 140(6):362–370, 1993.

[5] P. Comon. Independent component analysis—a new concept? *Signal Processing*, 36:287–314, 1994.

[6] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

[7] G. Deco and D. Obradovic. Linear redundancy reduction learning. *Neural Networks*, 8(5):751–755, 1995.

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

[9] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, 2001.

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

[11] A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.

[12] C. Jutten and J. Hérault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[13] J. K. Lin, D. G. Grier, and J. D. Cowan. Faithful representation of separable distributions. *Neural Computation*, 9(6):1305–1320, 1997.

[14] G. Marques and L. Almeida. Separation of nonlinear mixtures using pattern repulsion. In *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, pages 277–282, Aussois, France, 1999.

[15] L. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72:3634–3636, 1994.

[16] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

[17] P. Pajunen, A. Hyvärinen, and J. Karhunen. Nonlinear blind source separation by self-organizing maps. In *Proc. Int. Conf. on Neural Information Processing*, pages 1207–1210, Hong Kong, 1996.

[18] P. Pajunen and J. Karhunen. A maximum likelihood approach to nonlinear blind source separation. In *Proceedings of the 1997 Int. Conf. on Artificial Neural Networks (ICANN'97)*, pages 541–546, Lausanne, Switzerland, 1997.

[19] P. Pajunen and J. Karhunen, editors. *Proc. of the 2nd Int. Workshop on Independent Component Analysis and Blind Signal Separation, Helsinki, Finland, June 19-22, 2000*. Otamedia, 2000.

[20] A. Taleb and C. Jutten. Source separation in post-nonlinear mixtures. *IEEE Trans. on Signal Processing*, 47(10):2807–2820, 1999.

[21] H. Valpola, X. Giannakopoulos, A. Honkela, and J. Karhunen. Nonlinear independent component analysis using ensemble learning: Experiments and discussion. In *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, pages 351–356, Helsinki, Finland, 2000.

[22] V.N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, New York, 1995.

[23] H. H. Yang, S.-I. Amari, and A. Cichocki. Information-theoretic approach to blind separation of sources in non-linear mixture. *Signal Processing*, 64(3):291–300, 1998.

[24] A. Ziehe, M. Kawanabe, S. Harmeling, and K.-R. Müller. Separation of postnonlinear mixtures using ACE and temporal decorrelation. submitted to ICA 2001.

[25] A. Ziehe and K.-R. Müller. TDSEP—an efficient algorithm for blind separation using time structure. In *Proc. Int. Conf. on Artificial Neural Networks (ICANN'98)*, pages 675–680, Skövde, Sweden, 1998.