Lutton, E., Maître, H., & Lopez-Krahe, J. (1994). Contribution to the determination of vanishing points using Hough transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *16*(4), 430–438.

Mathews, J., & Walker, R. L. (1970). *Mathematical methods of physics*. Menlo Park, CA: Benjamin/Cummings.

Mundy, J. L., & Zisserman, A. (Eds). (1992). *Geometric invariants in computer vision*. Cambridge, MA: MIT Press.

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, *42*(3),145–175.

Ratches, J. A., Walters, C. P., Buser, R. G., & Guenther, B. D. (1997). Aided and automatic target recognition based upon sensory inputs from image forming systems. *IEEE Trans. on PAMI*, *19*(9), 1004–1019.

Shufelt, J. A. (1999). Performance evaluation and analysis of vanishing point detection techniques. *IEEE Trans. on PAMI*, *21*(3), 282–288.

Torr, P., & Zisserman, A. (1998). Robust computation and parameterization of multiple view relations. In *Proceedings of the International Conference on Computer Vision* (pp. 727–732). Los Alamitos, CA: IEEE Computer Society.

Yuille, A. L., & Coughlan, J. M. (2000). Fundamental limits of Bayesian inference: Order parameters and phase transitions for road tracking. *Pattern Analysis and Machine Intelligence*, *22*(2), 160–173.

Yuille, A. L., Coughlan, J. M., Wu, Y-N, & Zhu, S. C. (2001). Order parameters for minimax entropy distributions: When does high level knowledge help? *International Journal of Computer Vision*, *41*(1/2), 9–33.

Zhu, S. C., Lanterman, A., & Miller, M. I. (1998). Clutter modeling and performance analysis in automatic target recognition. In *Proceedings Workshop on Detection and Classification of Difficult Targets* (pp. 477–496). Redstone Arsenal, AL.

Received February 8, 2002; accepted December 2, 2002.

LETTER **Communicated by Michael Jordan**

Kernel-Based Nonlinear Blind Source Separation

Stefan Harmeling
harmeli@first.fhg.de
Andreas Ziehe
ziehe@first.fhg.de
Motoaki Kawanabe
nabe@first.fhg.de
Fraunhofer FIRST.IDA, 12489 Berlin, Germany

Klaus-Robert Müller
klaus@first.fhg.de
Fraunhofer FIRST.IDA, 12489 Berlin, Germany, and University of Potsdam, Department of Computer Science, 14482 Potsdam, Germany

We propose kTDSEP, a kernel-based algorithm for nonlinear blind source separation (BSS). It combines complementary research fields: kernel feature spaces and BSS using temporal information. This yields an efficient algorithm for nonlinear BSS with invertible nonlinearity. Key assumptions are that the kernel feature space is chosen rich enough to approximate the nonlinearity and that signals of interest contain temporal information. Both assumptions are fulfilled for a wide set of real-world applications. The algorithm works as follows: First, the data are (implicitly) mapped to a high (possibly infinite)–dimensional kernel feature space. In practice, however, the data form a smaller submanifold in feature space—even smaller than the number of training data points—a fact that has already been used by, for example, reduced set techniques for support vector machines. We propose to adapt to this effective dimension as a preprocessing step and to construct an orthonormal basis of this submanifold. The latter dimension-reduction step is essential for making the subsequent application of BSS methods computationally and numerically tractable. In the reduced space, we use a BSS algorithm that is based on second-order temporal decorrelation. Finally, we propose a selection procedure to obtain the original sources from the extracted nonlinear components automatically.

Experiments demonstrate the excellent performance and efficiency of our kTDSEP algorithm for several problems of nonlinear BSS and for more than two sources.

Neural Computation **15, 1089–1124** (2003) © 2003 Massachusetts Institute of Technology

1 Introduction

The problem of nonlinear blind source separation (BSS) is a challenging research task, and several methods have been proposed in the literature (Burel, 1992; Valpola, Giannakopoulos, Honkela, & Karhunen, 2000; Lee, Koehler, & Orglmeister, 1997; Lin, Grier, & Cowan, 1997; Yang, Amari, & Cichocki, 1998; Marques & Almeida, 1999; Pajunen, Hyvärinen, & Karhunen, 1996; Pajunen & Karhunen, 1997; Fyfe & Lai, 2000; Taleb & Jutten, 1999; Hyvarinen, Karhunen, & Oja, 2001). Fruitful applications are conceivable in, among others, the fields of telecommunications, array processing, vision, biomedical data analysis, and acoustic source separation where nonlinearities can occur in the mixing process. Nonlinear BSS has recently been applied to data from industrial pulp processing (Hyvarinen et al., 2001).

In nonlinear BSS we observe a mixed signal,

$$\mathbf{x}[t] = \mathbf{f}(\mathbf{s}[t]), \tag{1.1}$$

where $\mathbf{x}[t] := [x_1[t], \ldots, x_n[t]]^\top$ and $\mathbf{s}[t] := [s_1[t], \ldots, s_n[t]]^\top$ are $n \times 1$ column vectors with $t = 1, \ldots, T$ and $\mathbf{f}$ is a nonlinear invertible function from $\Re^n$ to $\Re^n$. In the special case where $\mathbf{f}$ is an $n \times n$ matrix, we regain standard linear BSS (see Hyvarinen et al., 2001; Cardoso, 1998).

Another important special case is the postnonlinear model (PNL),

$$\mathbf{x}[t] = \mathbf{f}(\mathbf{A}\mathbf{s}[t]), \tag{1.2}$$

where $\mathbf{A}$ is a linear mixing matrix and $\mathbf{f}$ is a postnonlinearity that operates component-wise. Given these constraints, the PNL problem can be solved by inverting the one-dimensional nonlinear functions (Taleb & Jutten, 1999; Ziehe, Kawanabe, Harmeling, & Müller, 2001; Achard, Pham, & Jutten, 2001).

Existing algorithmic approaches of the general nonlinear BSS problem have used, for example, self-organizing maps (Pajunen et al., 1996; Lin et al., 1997), extensions of generative topographic mapping (GTM; Pajunen & Karhunen, 1997), neural networks (Burel, 1992; Marques & Almeida, 1999), or Bayesian ensemble learning (Valpola et al., 2000; Iline, Valpola, & Oja, 2001; Lappalainen & Honkela, 2000) to unfold the nonlinearity $\mathbf{f}$. Also, a kernel-based method was tried on simplistic toy signals (Fyfe & Lai, 2000).

Note, however, that these methods are often of high computational cost and, depending on the algorithm, are also prone to run into local minima.

1.1 Problem Formulation. In this work, we focus on the general nonlinear BSS problem like equation 1.1, assuming that the underlying source signals have characteristic time structure, that is, the spectra of the components are pairwise different. An example of a nonlinear mixture is

$$x_1[t] = e^{s_1[t]} - e^{s_2[t]}$$
$$x_2[t] = e^{-s_1[t]} + e^{-s_2[t]}.$$

Note that we are not assuming that $\mathbf{f}$ operates component-wise. The task is to recover the unknown source signals $\mathbf{s}[t]$ using only the mixed signals $\mathbf{x}[t]$.

Hyvärinen and Pajunen (1999) pointed out why in general there are no unique solutions to this problem. However, our method, based on the ingredients of kernel feature spaces, dimension reduction, second-order temporal decorrelation BSS, and a selection procedure (see Figure 1), can recover the sources $\mathbf{s}$ by restricting the space of possible functions used to invert the nonlinearity $\mathbf{f}$ (see section 5.5 for a detailed discussion) and by exploiting the time structure of the unknown sources (see section 3 for a detailed discussion).

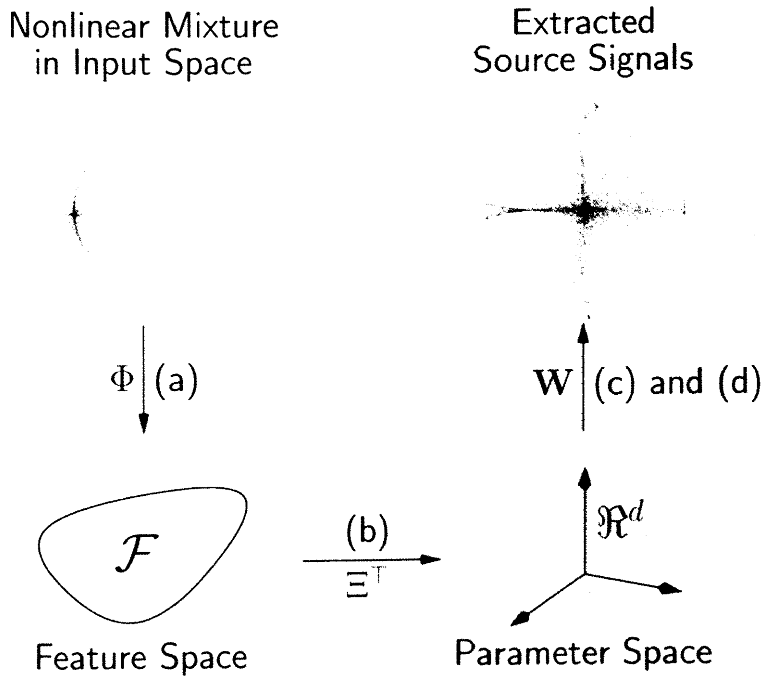

Figure 1: The nonlinear BSS problem is solved in four steps: (a) The data are mapped from input space to feature space, (b) the dimensionality is reduced, (c) second-order temporal decorrelation BSS is used, and (d) an automatic selection procedure is applied.

1.2 Kernelizing Blind Source Separation. Let us first discuss kernel-based learning, which has become a popular technique (Vapnik, 1995; Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2002; Burges, 1998; Schölkopf, Smola, & Müller, 1998; Müller, Mika, Rätsch, Tsuda, & Schölkopf, 2001). The basic idea of kernelizing (see Schölkopf et al., 1998) allows the construction of very powerful nonlinear variants of existing linear scalar product–based algorithms by mapping the data $\mathbf{x}[t]$ $(t = 1, \dots, T)$ implicitly into some kernel feature space $\mathcal{F}$ through some mapping $\Phi\colon \Re^n \to \mathcal{F}$. Performing a simple linear algorithm in $\mathcal{F}$ then corresponds to a nonlinear algorithm in input space. In other words a *linear* BSS in $\mathcal{F}$ would give rise to a *nonlinear* BSS algorithm in input space. All can be done efficiently and never directly but implicitly in $\mathcal{F}$ by using the kernel trick $\mathbf{k}(\mathbf{a}, \mathbf{b}) = \Phi(\mathbf{a}) \cdot \Phi(\mathbf{b})$.[1] However, a straightforward application of the kernel trick to BSS has failed so far for two reasons: applying a linear BSS algorithm in feature space will not necessarily identify the sought-after signals, since there are very likely directions that are also independent but higher-order versions of the original signals, and, in principle, the BSS algorithm has to be applied, after kernelizing, to a T-dimensional problem, which is numerically neither stable nor tractable.[2]

A new aspect that we add in this contribution is to apply first a dimension-reduction step before BSS, since typically the data form a lower-dimensional subspace in $\mathcal{F}$, even much lower than T-dimensional. We therefore propose a mathematical construction, very much inspired by reduced-set methods (Schölkopf et al., 1999), that allows us to adapt to the intrinsic data dimension. In the next step, an orthonormal basis of this low-dimensional submanifold is constructed, which eventually makes the computations of a subsequent BSS algorithm tractable. The subtle difference to reduced-set techniques is that we do not aim to construct a low-dimensional basis for a good classification; rather, we aim for an efficient, that is, low-dimensional, description of the data in $\mathcal{F}$. Note, that we use a BSS algorithm that is based on second-order temporal decorrelation (Ziehe & Müller, 1998; Belouchrani, Meraim, Cardoso, & Moulines, 1997), which is an essential building block of our algorithm. Finally, the sources of interest are automatically selected after the BSS step.

The four ingredients of kernel feature space, dimension reduction, second-order temporal decorrelation BSS, and a selection procedure give rise to an algorithmic solution, called kTDSEP, that is mathematically elegant (see section 2) and efficient, with high performance, as we will see in the experiments on nonlinear mixtures of artificially generated signals and various sound signals (see section 5). A conclusion is given in section 6.

[1] This trick is essential if $\mathcal{F}$ is an infinite-dimensional space.

[2] Note that although $\mathcal{F}$ might be infinite-dimensional, the subspace of $\mathcal{F}$ where the data lie is maximally T-dimensional.

2 Constructing Kernel Feature Spaces of Reduced Dimension

In order to establish a linear problem in feature space that corresponds to some nonlinear problem in input space, we need to specify how to map inputs $\mathbf{x}[1], \dots, \mathbf{x}[T] \in \Re^n$ into feature space $\mathcal{F}$ and how to handle its possibly high dimensionality. Note that $\mathbf{x}[t]$ is scaled down such that its absolute maximum is one. Here we force the signals between -1 and 1 before mapping to $\mathcal{F}$. This will become important for the selection procedure (see section 3).

In the following, we describe two methods that obtain an orthogonal basis in feature space with reduced dimension and explain how to project the data onto this finite-dimensional basis such that BSS techniques can be applied.

There exist a variety of other dimensionality-reduction methods for the kernel setting. Smola and Schölkopf (2000) approximate the kernel matrix by iteratively picking columns of that matrix in a greedy manner. Fine and Scheinberg (2001) use the Sherman-Morrison-Woodbury method and the product-form Cholesky factorization to obtain low-rank kernel representations. Williams and Seeger (2001) employ the Nyström method to a randomly sampled subset of the data that is very similar to the first of our proposed methods (see section 2.1). However, we perform the resampling several times and use in addition the condition numbers of the corresponding kernel matrices to pick a particular subset.

2.1 Finding a Basis via Random Sampling and Clustering. In addition to the input points, consider some further points $\mathbf{v}_1, \dots, \mathbf{v}_d \in \Re^n$ from the same space that will later generate a basis in $\mathcal{F}$. Let us denote the mapped points by $\Phi_\mathbf{x} := [\Phi(\mathbf{x}[1]), \dots, \Phi(\mathbf{x}[T])]$ and $\Phi_\mathbf{v} := [\Phi(\mathbf{v}_1), \dots, \Phi(\mathbf{v}_d)]$.[3] We assume that the columns of $\Phi_\mathbf{v}$ constitute a basis of the column space[4] of $\Phi_\mathbf{x}$, formally expressed as

$$\operatorname{span}(\Phi_\mathbf{v}) = \operatorname{span}(\Phi_\mathbf{x}) \text{ and } \operatorname{rank}(\Phi_\mathbf{v}) = d. \tag{2.1}$$

In section 2.1.2, we will explain how and to what degree this assumption can be fulfilled. Moreover, $\Phi_\mathbf{v}$ being a basis implies that the matrix[5] $\Phi_\mathbf{v}^\top \Phi_\mathbf{v}$ has full rank, and its inverse exists. So now we can define an orthonormal basis (see the empirical kernel map in Schölkopf et al., 1999),

$$\Xi := \Phi_\mathbf{v}(\Phi_\mathbf{v}^\top \Phi_\mathbf{v})^{-1/2}, \tag{2.2}$$

[3] We denote the points of the time series with square brackets (e.g., $\mathbf{x}[t]$) and other points of the input space with subscripts (e.g. $\mathbf{v}_d$).

[4] The column space of $\Phi_\mathbf{x}$ is the space spanned by the column vectors of $\Phi_\mathbf{x}$, written $\operatorname{span}(\Phi_\mathbf{x})$.

[5] The ijth entry of the matrix $\Phi_\mathbf{v}^\top \Phi_\mathbf{v}$ is $\Phi(\mathbf{v}_i)^\top \Phi(\mathbf{v}_j)$.

the column space of which is identical to the column space of $\Phi_{\mathbf{v}}$. Consequently, this basis Ξ enables us to parameterize all vectors that lie in the column space of $\Phi_{\mathbf{x}}$ by some vectors in $\Re^d$. For instance, for vectors $\sum_{i=1}^{T} \alpha_{\Phi i} \Phi(\mathbf{x}[i])$, which we write more compactly as $\Phi_{\mathbf{x}}\alpha_{\Phi}$, and $\Phi_{\mathbf{x}}\beta_{\Phi}$ in the column space of $\Phi_{\mathbf{x}}$ with α_{Φ} and β_{Φ} in $\Re^T$, there exist α_{Ξ} and β_{Ξ} in $\Re^d$ such that $\Phi_{\mathbf{x}}\alpha_{\Phi} = \Xi\alpha_{\Xi}$ and $\Phi_{\mathbf{x}}\beta_{\Phi} = \Xi\beta_{\Xi}$. The orthonormality implies

$$\alpha_{\Phi}^{\top} \Phi_{\mathbf{x}}^{\top} \Phi_{\mathbf{x}} \beta_{\Phi} = \alpha_{\Xi}^{\top} \Xi^{\top} \Xi \beta_{\Xi} = \alpha_{\Xi}^{\top} \beta_{\Xi}, \tag{2.3}$$

which states the remarkable property that the dot product of two linear combinations of the columns of $\Phi_{\mathbf{x}}$ in $\mathcal{F}$ coincides with the dot product in $\Re^d$. By construction of Ξ (see equation 2.2), the column space of $\Phi_{\mathbf{x}}$ is naturally isomorphic (as a vector space) to the $\Re^d$. Moreover, this isomorphism is compatible with the two involved dot products, as was shown in equation 2.3. This implies that all properties regarding angles and lengths can be taken back and forth between the column space of $\Phi_{\mathbf{x}}$ and $\Re^d$. The space spanned by Ξ is called parameter space. Figure 2 pictures our intuition. Usually kernel methods parameterize the column space of $\Phi_{\mathbf{x}}$ in terms of the mapped patterns $\{\Phi(\mathbf{x}[i])\}$, which effectively corresponds to vectors in $\Re^T$. The orthonormal basis from equation 2.2, however enables us to work in $\Re^d$, that is, in the span of Ξ.

2.1.1 Projecting the Input Data onto an Orthonormal Basis. By employing the kernel trick, we can directly map the input data onto the subspace of

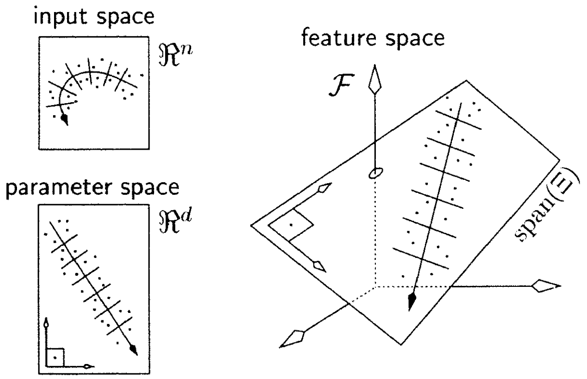

Figure 2: Input data are mapped to some submanifold of $\mathcal{F}$, which is in the span of a d-dimensional orthonormal basis Ξ. Therefore these mapped points can be parameterized in $\Re^d$. The linear directions in parameter space correspond to nonlinear directions in input space.

feature space spanned by the orthonormal basis. The expressions

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

are the entries of a real-valued $d \times d$ matrix $\Phi_{\mathbf{v}}^{\top}\Phi_{\mathbf{v}}$ that can be effectively calculated using the kernel trick. By construction of $\mathbf{v}_1, \ldots, \mathbf{v}_d$, it has full rank and is thus invertible. Similarly, we get

$$\begin{aligned}(\Phi_{\mathbf{v}}^{\top}\Phi_{\mathbf{x}})_{ij} &= \Phi(\mathbf{v}_i)^{\top}\Phi(\mathbf{x}[j]) = \mathbf{k}(\mathbf{v}_i, \mathbf{x}[j]) \\ &\quad \text{with} \quad i = 1, \ldots, d, \quad j = 1, \ldots, T,\end{aligned}$$

which are the entries of the real-valued $d \times T$ matrix $\Phi_{\mathbf{v}}^{\top}\Phi_{\mathbf{x}}$. Using both matrices, we compute finally

$$\begin{aligned}\Psi_{\mathbf{x}}[t] &:= \Xi^{\top}\Phi(\mathbf{x}[t]) = (\Phi_{\mathbf{v}}^{\top}\Phi_{\mathbf{v}})^{-1/2}\Phi_{\mathbf{v}}^{\top}\Phi(\mathbf{x}[t]) \\ &= \begin{bmatrix} \mathbf{k}(\mathbf{v}_1, \mathbf{v}_1) & \cdots & \mathbf{k}(\mathbf{v}_1, \mathbf{v}_d) \\ \vdots & & \vdots \\ \mathbf{k}(\mathbf{v}_d, \mathbf{v}_1) & \cdots & \mathbf{k}(\mathbf{v}_d, \mathbf{v}_d) \end{bmatrix}^{-\frac{1}{2}} \begin{bmatrix} \mathbf{k}(\mathbf{v}_1, \mathbf{x}[t]) \\ \vdots \\ \mathbf{k}(\mathbf{v}_d, \mathbf{x}[t]) \end{bmatrix},\end{aligned} \tag{2.4}$$

which is a real-valued $d \times 1$ vector representing a projected data point. Note that $(\Phi_{\mathbf{v}}^{\top}\Phi_{\mathbf{v}})^{-1/2}$ can be omitted if the subsequent BSS procedure contains a whitening step.

Regarding the computational costs of this projection, we have to evaluate the kernel function $O(d^2 n) + O(dTn)$ times, and equation 2.4 requires $O(d^3)$ multiplications where n denotes the dimension of the input space. Again, note that d is much smaller than T. Furthermore, after projection, the storage requirements are reduced since we do not have to hold the full $T \times T$ kernel matrix but only a $d \times T$ matrix.

2.1.2 Choosing Vectors for the Basis in $\mathcal{F}$. So far, we have assumed as given some points $\mathbf{v}_1, \ldots, \mathbf{v}_d$ that fulfill equation 2.1, and we presented the beneficial properties of our construction. In fact, the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ are roughly analogous to a reduced set in the support vector world (Schölkopf et al., 1999). Note, however, that often we can only approximately fulfill equation 2.1, that is,

$$\operatorname{span}(\Phi_{\mathbf{v}}) \approx \operatorname{span}(\Phi_{\mathbf{x}}), \tag{2.5}$$

for example, for an RBF kernel $\operatorname{span}(\Phi_{\mathbf{x}})$ is T-dimensional, but $\operatorname{span}(\Phi_{\mathbf{v}})$ is by definition d-dimensional (for further discussion, see appendix A; Williams & Seeger, 2000; Bach & Jordan, 2001). Several options exist to achieve this approximation.

The points $\mathbf{v}_1, \ldots, \mathbf{v}_d$ have to be chosen such that in equation 2.4, the inversion of the kernel matrix $\mathbf{K}_{\mathbf{v}}$, whose entries are

$$(\mathbf{K}_{\mathbf{v}})_{ij} := (\Phi_{\mathbf{v}}^{\top}\Phi_{\mathbf{v}})_{ij} = \mathbf{k}(\mathbf{v}_i, \mathbf{v}_j).$$

is numerically stable. We try to find a set of points such that the condition number of its corresponding kernel matrix is below a certain threshold, and if we add one more point, the condition number is above.[6] Since we cannot check all possible combinations, we randomly sample d points (for fixed d) repeatedly, say r, for example, 100 times. Roughly speaking, we perform this sampling for different d until we find d points, $\mathbf{v}_1, \ldots, \mathbf{v}_d$, that are linearly independent in feature space (more precisely: the corresponding condition number is below the threshold), but we cannot find $d+1$ points with the same property (i.e., the condition number for those points is above the threshold). This is done by computing the kernel matrix in $O(d^2 n)$ time, yielding an overall cost of $O(dr)[O(d) + O(d^2 n)] = O(d^3 rn)$ where n denotes the dimension of the input space.

This procedure determines d and points $\mathbf{v}_1, \ldots, \mathbf{v}_d$. Running k-means clustering (with $k = d$), costing $O(Tdn)$, in input space, is another way to pick such points. Our experience shows that both approaches work well as long as d is chosen large enough.

2.2 Finding a Basis via Kernel PCA. Another more direct method to obtain the low-dimensional subspace is kernel PCA (Schölkopf et al., 1998). Such a subspace is optimal with respect to the reconstruction error in feature space; however, computational costs are slightly increased (see Figure 13). For simplicity, we assume that the data are centered in feature space.[7] To perform kernel PCA, we need to find eigenvectors and eigenvalues of the covariance matrix $\frac{1}{T}\Phi_{\mathbf{X}}\Phi_{\mathbf{X}}^{\top}$. Denoting the diagonal matrix with eigenvalues $\lambda_1 \geq, \ldots, \geq \lambda_T$ along the diagonal as $\mathbf{\Lambda}$, the eigenvectors $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_T]$ of the kernel matrix $\frac{1}{T}\Phi_{\mathbf{X}}^{\top}\Phi_{\mathbf{X}}$ fulfill

$$\left(\frac{1}{T}\Phi_{\mathbf{X}}^{\top}\Phi_{\mathbf{X}}\right)\mathbf{E} = \mathbf{E}\mathbf{\Lambda},$$

which immediately implies

$$\left(\frac{1}{T}\Phi_{\mathbf{X}}\Phi_{\mathbf{X}}^{\top}\right)(\Phi_{\mathbf{X}}\mathbf{E}) = (\Phi_{\mathbf{X}}\mathbf{E})\mathbf{\Lambda}.$$

So, $\lambda_1, \ldots, \lambda_T$ are the eigenvalues of $\frac{1}{T}\Phi_{\mathbf{X}}\Phi_{\mathbf{X}}^{\top}$ with corresponding eigenvectors $\Phi_{\mathbf{X}}\mathbf{E}$. Normalizing the first d eigenvectors yields a d-dimensional orthonormal basis,

$$\Xi := \Phi_{\mathbf{X}}\mathbf{E}_d(T\mathbf{\Lambda}_d)^{-1/2},$$

[6] The condition number of a matrix is the ratio between the largest and the smallest singular value.

[7] The kernel matrix $\mathbf{K}$ with entries $\mathbf{k}(\mathbf{x}[i], \mathbf{x}[j])$ can be easily centered (Schölkopf et al., 1998) by $\mathbf{K} \mapsto \mathbf{K} - \mathbf{1}_T\mathbf{K} - \mathbf{K}\mathbf{1}_T + \mathbf{1}_T\mathbf{K}\mathbf{1}_T$, with $\mathbf{1}_T$ being the $T \times T$ matrix with all entries equal to $1/T$.

with $\mathbf{E}_d := [\mathbf{e}_1, \ldots, \mathbf{e}_d]$, $\mathbf{\Lambda}_d$ being the diagonal matrix with $\lambda_1, \ldots, \lambda_d$ along the diagonal and $(T\mathbf{\Lambda}_d)^{-1/2}$ ensuring orthonormality. This basis enables us to parameterize the signals $\Phi(\mathbf{x}[t])$ in feature space as real-valued d-dimensional signals,

$$\begin{aligned}\Psi_{\mathbf{x}}[t] &:= \Xi^{\top}\Phi(\mathbf{x}[t]) = (T\mathbf{\Lambda}_d)^{-1/2}\mathbf{E}_d^{\top}\Phi_{\mathbf{X}}^{\top}\Phi(\mathbf{x}[t]), \\ &= \frac{1}{\sqrt{T}}\begin{bmatrix}\frac{1}{\sqrt{\lambda_1}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sqrt{\lambda_d}}\end{bmatrix}\begin{bmatrix}\mathbf{e}_1^{\top} \\ \vdots \\ \mathbf{e}_d^{\top}\end{bmatrix}\begin{bmatrix}\mathbf{k}(\mathbf{x}[1], \mathbf{x}[t]) \\ \vdots \\ \mathbf{k}(\mathbf{x}[T], \mathbf{x}[t])\end{bmatrix},\end{aligned} \tag{2.6}$$

which are calculated conveniently using the kernel trick.

Since kernel PCA involves solving the eigenvalue problem for a large matrix, whose size depends on the amount of data, $O(T^3)$, we typically apply kernel PCA to a subset of the original data set if T becomes large (Mika, 1998; Schölkopf et al., 1999).

3 Nonlinear Blind Source Separation

Clearly, BSS algorithms cannot be applied directly in full feature space without the proposed reduction step. They would need to solve a T-dimensional BSS problem, which is intractable. A further problem is that manipulating such a $T \times T$ matrix can easily become numerically unstable, and even overfitting might occur (Hyvärinen, Särelä, & Vigário, 1999). In the previous section, we mapped the signals $\mathbf{x}[t]$ from input space onto signals $\Psi_{\mathbf{x}}[t]$ in a $d(\ll T)$–dimensional parameter space (see Figure 1). This was done either by random sampling or k-means clustering using equation 2.4 or by applying kernel PCA together with equation 2.6. Now we are in a situation in which the nonlinear problem in input space has been transformed to a linear problem in parameter space where we can apply linear BSS methods. In particular, we propose to use TDSEP, a second-order BSS technique that relies on time-shifted covariance matrices of the mapped signals $\Psi_{\mathbf{x}}[t]$, thereby exploiting the assumed time structure of the unknown sources (see appendix B for a discussion of why not to use kurtosis-based techniques).

We briefly describe the TDSEP algorithm (details can be found in Ziehe & Müller, 1998; see also Belouchrani et al., 1997). For the signals in parameter space,

$$\Psi_{\mathbf{x}}[t] := \Xi^{\top}\Phi(\mathbf{x}[t]) \in \Re^d,$$

we define symmetrized time-shifted covariance matrices,

$$\begin{aligned}\mathbf{R}_\tau := \frac{1}{2(T-\tau)}\sum_{t=1}^{T-\tau}&((\Psi_{\mathbf{x}}[t] - \mu_\Psi)(\Psi_{\mathbf{x}}[t+\tau] - \mu_\Psi)^{\top} \\ &+ (\Psi_{\mathbf{x}}[t+\tau] - \mu_\Psi)(\Psi_{\mathbf{x}}[t] - \mu_\Psi)^{\top}),\end{aligned}$$

with $\mu_\Psi := \frac{1}{T}\sum_{t=1}^{T}\Psi_{\mathbf{x}}[t]$. Then we find a matrix $\mathbf{W}$ that simultaneously diagonalizes[8] several of these matrices $\mathbf{R}_{\tau_i}$, that is, the matrices

$$\mathbf{W}\mathbf{R}_{\tau_i}\mathbf{W}^\top \quad i = 1, \ldots, m$$

should become approximately diagonal. $\mathbf{W}$ is the sought-after demixing matrix. The extracted d nonlinear components are

$$\mathbf{y}[t] := \mathbf{W}\Psi_{\mathbf{x}}[t] \in \Re^d.$$

4 Selecting from the Extracted Components

Besides the sought-after sources, there are also signals that we are not interested in among the extracted components $\mathbf{y}[t]$. Empirically, these other signals can be explained by higher-order monomials of the sources, as we will see next (with ideas from Harmeling, Ziehe, Kawanabe, Blankertz, & Müller, 2001). These monomials are well motivated for polynomial kernels but are also useful to analyze signals that have been extracted using a gaussian kernel.

4.1 Reconstructing the Extracted Components. For two source signals s_1 and s_2, we call the monomials of these sources up to a certain degree *quasi sources*. For example, the quasi sources up to degree 2 (i.e., where each variable appears up to degree 2) are

$$\mathbf{q}_2 := (s_1^2 s_2^2, s_1^2 s_2, s_1^2, s_1 s_2^2, s_1 s_2, s_1, s_2^2, s_2)^\top.$$

For brevity, we write s_1 instead of $s_1[t]$ (i.e., s_1 is a signal). In general, the quasi sources up to degree m are all monomials of the form $s_1^{m_1} s_2^{m_2}$ for $0 \leq m_1, m_2 < m$. Accordingly, $\mathbf{q}_m$ is the vector containing all those monomials.

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

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

Since for symmetrically distributed signals s_1 and s_2 (with mean zero and variance one), the odd moments are zero,

$$E\{s_1^k\} = 0 \quad \text{if } k \text{ is odd,}$$

two quasi sources $s_1^{k_1}s_2^{m_1}$ and $s_1^{k_2}s_2^{m_2}$ are uncorrelated in most cases:

$$\mathrm{corr}(s_1^{k_1}s_2^{m_1}, s_1^{k_2}s_2^{m_2}) = 0 \quad \text{if } k_1 + k_2 \text{ is odd or } m_1 + m_2 \text{ is odd.} \tag{4.1}$$

This is easily implied from the above equation using the fact that if the sum of two integers is odd, then one of the summands must be odd as well. Therefore, the quasi sources for two signals can be collocated into four groups with no correlations among the groups; for example, for the quasi sources up to degree 2, the four groups are (see Figure 3)

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

We will use these findings to reconstruct the extracted components for an easy example. Consider two sinusoidal source signals $\mathbf{s}[t] = [s_1[t], s_2[t]]^\top$ that are nonlinearly mixed by

$$\mathbf{x}[t] = \mathbf{A}(s_1[t], s_2[t])^\top + \mathbf{c}s_1[t]s_2[t],$$

with

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

(mixture taken from Molgedey & Schuster, 1994). Running our algorithm with a polynomial kernel of degree 4,

$$\mathbf{k}(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^\top \mathbf{b} + 1)^4,$$

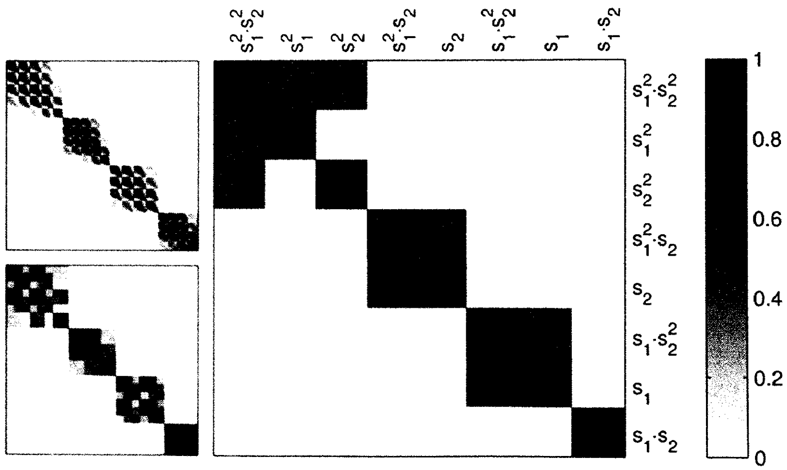

Figure 3: Most quasi sources are pairwise correlated. The right panel shows the covariance matrix of the quasi sources up to degree 2, the lower left panel up to degree 4, and the upper left panel up to degree 8. Note that the quasi sources can always be collocated into four groups.

[8] We use the algorithm described in Cardoso and Souloumiac (1996). See also Cardoso (1999).

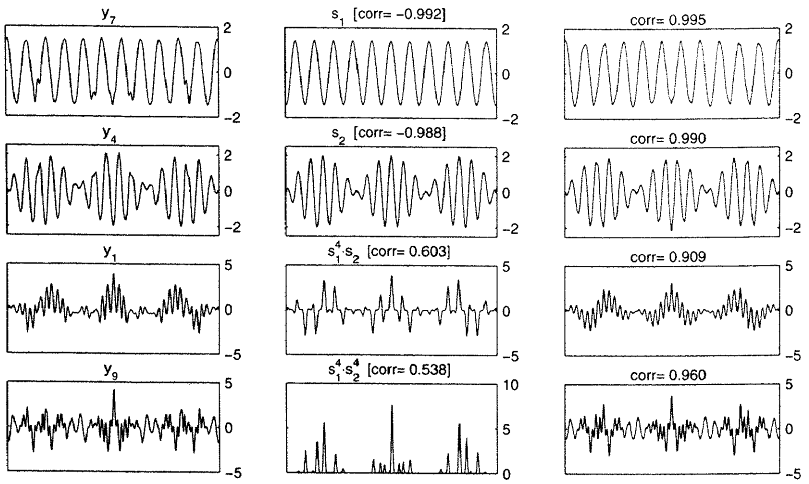

Figure 4: The extracted signals in the left panels (only four are shown) are matched with single quasi sources in the middle panels and combinations of subgroups of quasi sources (right panels).

we have to consider the quasi sources up to degree 4: all possible products of s_1, s_1^2, s_1^3, s_1^4 and their counterparts in s_2. Using equation 4.1, these quasi sources can also be arranged into four groups with no correlations among the groups. As examples, we explain four of the extracted signals using these quasi sources: y_7, y_4, y_1, y_9, shown in the left panels of Figure 4. 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). The same holds for y_1 and the other extracted signals not shown in the figure. Note that for y_7 and y_4, which match s_1 and s_2 reasonably well, more quasi sources do not improve the result notably.

Empirically, we have seen that the extracted components can be explained by linear combinations of higher-order monomials of the sources. This knowledge can be used to suggest several options to select the signals of interest: the signals built by higher-order monomials are very peaky and therefore have, after proper normalization, a lower variance and also a lower description length (Rissanen, 1978) than the signals of interest. However, whether methods based on these intuitions work in practice depends very much on the considered signals. The goal is to identify the sought-after signals s_1 and s_2 among the other extracted signals. If, for example, s_1 has a much higher description length than s_2, there might be a problem: probably s_1^2 will have a large description length as well, and therefore it is more likely that a method based on these principles will prefer s_1^2, instead of s_2. This explains why these selection procedures can fail easily, which is in accordance with our experience of running a number of experiments with different signals.

4.2 Selection by Rerunning the Algorithm. An algorithmic trick that worked well in all our experiments is to apply the algorithm twice. This trick is motivated by work on assessing reliability (Meinecke, Ziehe, Kawanabe, & Müller, 2002, in press; Müller, Vigario, Meinecke, & Ziehe, in press). Intuitively, the idea is to look for the most reliable components among the extracted signals—the components that appear again after reiteration of the algorithm. For this, we repeat the algorithm with the same parameters (kernel choice, d, τ, ...), but instead of sending $\mathbf{x}[t]$ into the feature space, we start with the d-dimensional demixed results $\mathbf{y}[t]$, map them to the feature space, reduce the dimensionality,[9] and demix with TDSEP, which yields $\mathbf{y}'[t]$. The sought-after components of $\mathbf{y}[t]$ are the ones that are matched best by the components $\mathbf{y}'[t]$ of the second run of TDSEP in feature space.

Why does this selection process find the right signals? As we saw in the previous section, most of the undesired signals are linear combinations of peaky higher-order monomials of the source signals, which can have very large values. Before the signals get mapped to feature space, they are scaled such that their absolute maximum is one (see the first paragraph of section 2). This is done by dividing each signal by its absolute maximum. The effect of this rescaling is that very large peaky signals are penalized; their variance is decreased more than the variance of signals that are less peaky. By doing so, we bias the desired signals to appear again with high correlations after another nonlinear demixing. This method works very well. All experiments documented in this article successfully used this selection method. Our experience shows that this selection fails only in cases where the sources are not recovered at all, that is, where the demixing failed, which means that there is nothing to select from.

5 Experiments

Nonlinearities appear in different contexts; for example, amplifier saturation results in difficult nonlinearities. Also, sensors can have nonlinearities, which have a disadvantageous influence on the recorded signals. However,

[9] The kernel function can be used with signals of arbitrary dimension.

real-world signals have the drawback that the ground truth—the true source signals—is not known. Since we want to demonstrate the performance of our algorithm, we consider in this article well-defined, controlled situations where we can compare the results of the algorithm to the true sources.

5.1 Deterministic Artificial Data. In the first experiment, we generate 2000 data points from two sinusoidal signals $\mathbf{s}[t] = [s_1[t], s_2[t]]^{\top}$ that have different frequencies ($s_1[t] = \sin(0.05\pi t)$, $s_2[t] = \sin(0.021\pi t)$ with $t = 1, \ldots, 2000$). These source signals are nonlinearly mixed (see the left panel in the first row of Figure 5) by

$$x_1[t] = e^{s_1[t]} - e^{s_2[t]}$$
$$x_2[t] = e^{-s_1[t]} + e^{-s_2[t]}.$$

We use a polynomial kernel of degree 9,

$$\mathbf{k}(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^{\top}\mathbf{b} + 1)^9,$$

which induces a feature space of all monomials up to degree 9. Applying k-means clustering to 500 randomly chosen input vectors, we determine vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{20}$ in input space, shown as + in the left panel in the first row of Figure 5. Projecting onto the feature space images of these vectors reduces the dimension to 20. As the second step, we apply TDSEP (with time shifts $\tau = 0, \ldots, 7$) to those 20-dimensional mapped signals $\Psi_{\mathbf{x}}[t]$. We obtain 20 components, among which we select two components, as described in the previous section. Their scatter plots and their waveforms are shown in the right panels in the first and third rows of Figure 5. For comparison, we plot in the left panels of the second and fourth rows the results of applying linear TDSEP directly to the nonlinearly mixed signals $\mathbf{x}[t]$. In this simple example, linear TDSEP already reaches a high correlation ($\mathrm{corr}(y_1^{lin}, s_1) = 0.9716$, $\mathrm{corr}(y_2^{lin}, s_2) = 0.9716$) to the true sources, but as we can see in the scatter plots shown in Figure 5, linear BSS fails to recover the right shape, in contrast to our nonlinear method, which recovers the shape of the scatter plot almost perfectly ($\mathrm{corr}(y_{20}, s_1) = 0.9998$, $\mathrm{corr}(y_5, s_2) = 0.9999$). We study the same mixture with two sinusoidal signals that have almost the same frequencies ($s_1[t] = \sin(0.0045\pi t)$, $s_2[t] = \sin(0.005\pi t)$ with $t = 1, \ldots, 2000$). Figure 6 shows the results after running our algorithm with the same parameters as in the previous case. We see that even two signals that have almost the same frequencies are separated.

5.2 Speech Data—Bent. In another experiment, we nonlinearly mix two speech signals $\mathbf{s}[t] = [s_1(t), s_2(t)]^{\top}$ (each with 20,000 samples, sampling rate 8 kHz, each ranging between −1 and +1) by

$$x_1[t] = -(s_2[t] + 1)\cos(\pi s_1[t])$$
$$x_2[t] = 1.5(s_2[t] + 1)\sin(\pi s_1[t]).$$

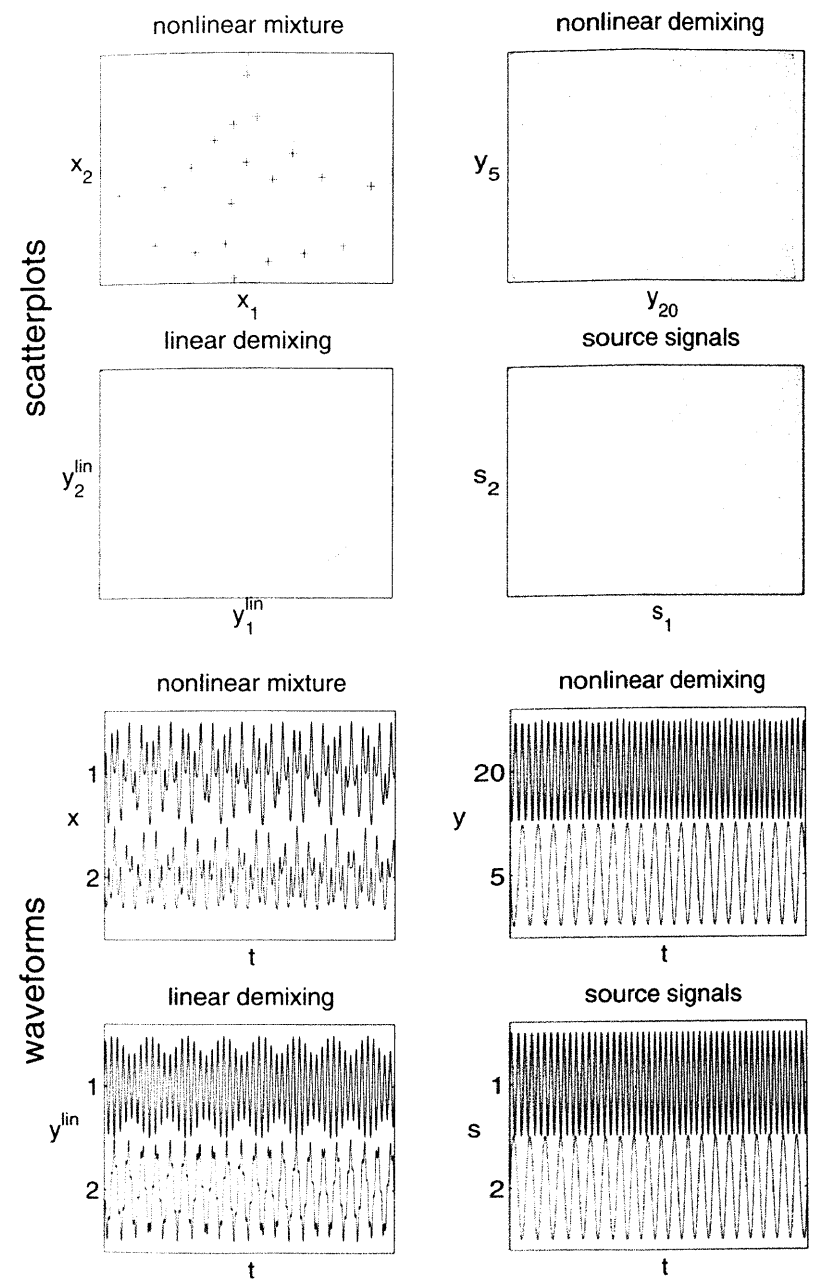

Figure 5: Deterministic artificial data. Scatter plots and waveforms of the nonlinear mixture and the nonlinear demixing (first and third rows) and of linear demixing and the true source signals (second and fourth rows).

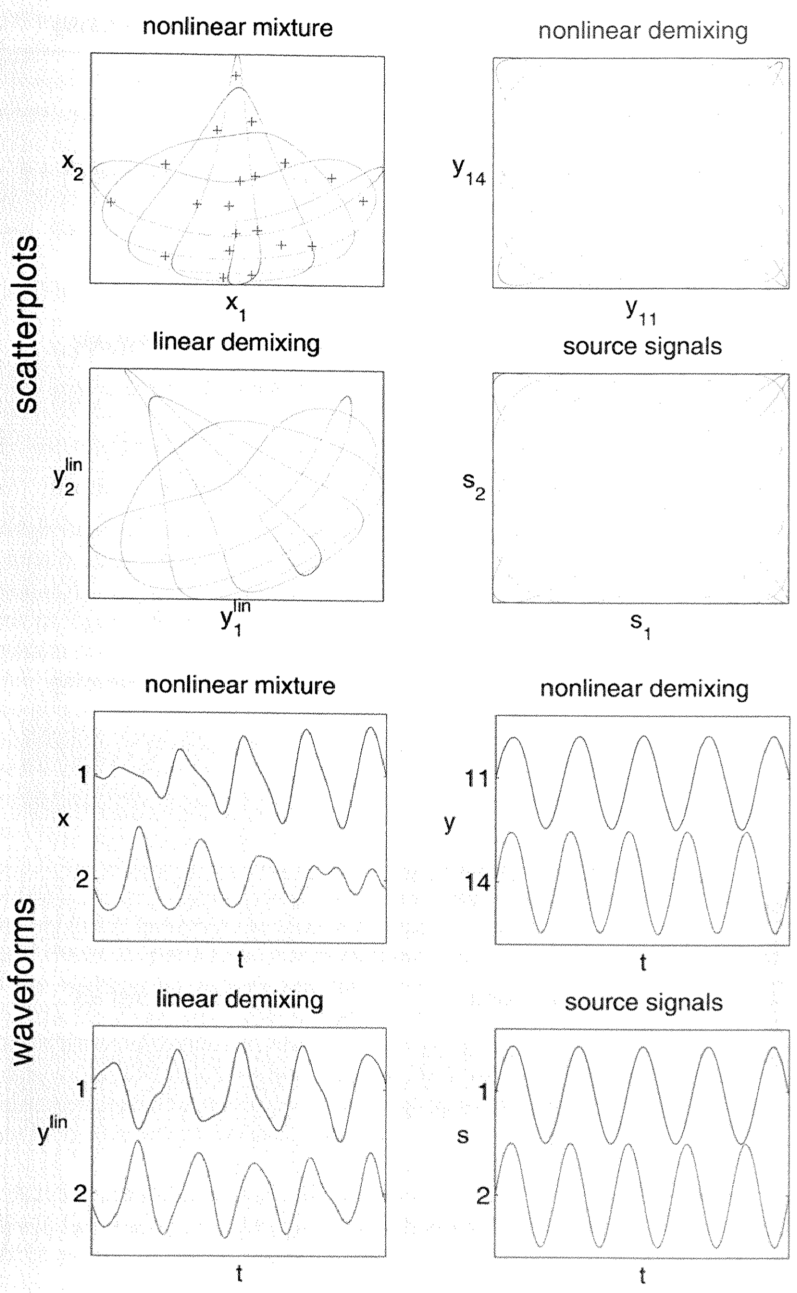

Figure 6: Deterministic artificial data with very close frequencies: Scatter plots and waveforms of the nonlinear mixture and the nonlinear demixing (first and third rows) and of linear demixing and the true source signals (second and fourth rows).

We employ a gaussian radial basis function (RBF) kernel,

$$\mathbf{k}(\mathbf{a}, \mathbf{b}) = e^{-\frac{\|\mathbf{a}-\mathbf{b}\|^2}{2\sigma^2}},$$

which induces a feature space where each direction measures the similarity to one of the training points. We can set $\sigma^2 = \frac{1}{2}$ and use

$$\mathbf{k}(\mathbf{a}, \mathbf{b}) = e^{-\|\mathbf{a}-\mathbf{b}\|^2},$$

without loss of generality, if signals are scaled in an appropriate way. Projecting onto feature space images of the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{20} \in \Re^2$ (depicted as + in the left panel in the first row of Figure 7), which are determined by repeated random sampling, reduces the dimensionality to $d = 20$. Among the 20 signals that we obtain by TDSEP (with time shifts $\tau = 0, \ldots, 7$) we automatically choose with our selection method two signals that turn out to reach very high correlations ($\mathrm{corr}(y_9, s_1) = 0.9768$, $\mathrm{corr}(y_4, s_2) = 0.9923$) with the original source signals. Since the linear method can only shear and rotate the data, it fails to recover the two signals ($\mathrm{corr}(y_2^{lin}, s_1) = 0.8811$, $\mathrm{corr}(y_1^{lin}, s_2) = 0.4091$).

5.3 Speech Data—Twisted. For an even more difficult experiment, we mix the two sound signals from the previous example by

$$x_1[t] = (s_2[t] + 3s_1[t] + 6)\cos(1.5\pi s_1[t])$$
$$x_2[t] = (s_2[t] + 3s_1[t] + 6)\sin(1.5\pi s_1[t]),$$

which twists the sources. The first source is mapped along a spiral around the center, and the second controls the deviation from that spiral. Note that the second source contributes much less to the mixture than the first source. We map the data to a feature space induced by a gaussian RBF kernel, $\mathbf{k}(\mathbf{a}, \mathbf{b}) = e^{-\|\mathbf{a}-\mathbf{b}\|^2}$, and apply kernel PCA to 500 randomly chosen input vectors. We obtain a 25-dimensional subspace of feature space that approximates the high-dimensional manifold in feature space very well. Projecting the mixed signals into that space, we obtain $\Psi_{\mathbf{x}}[t]$ and, finally applying TDSEP (with time shifts $\tau = 0, \ldots, 7$), we recover 25 signals, among which we select the signals of interest automatically. Again, these signals have very high correlations with the true sources ($\mathrm{corr}(y_2, s_1) = 0.9900$, $\mathrm{corr}(y_9, s_2) = 0.9466$), whereas the linear ones do not ($\mathrm{corr}(y_2^{lin}, s_1) = 0.5703$, $\mathrm{corr}(y_1^{lin}, s_2) = 0.0483$). Also, their scatter plot and their waveforms represented in the right panels in the first and third rows of Figure 8 show how well even the second source is found that is hidden as the amplitude of the spiral.

5.4 Analysis of the Cross-Correlations Through Time. To analyze the found signals $\mathbf{y}$ more carefully, we calculated the cross-correlations with quasi sources for different time-lagged versions of the signals.

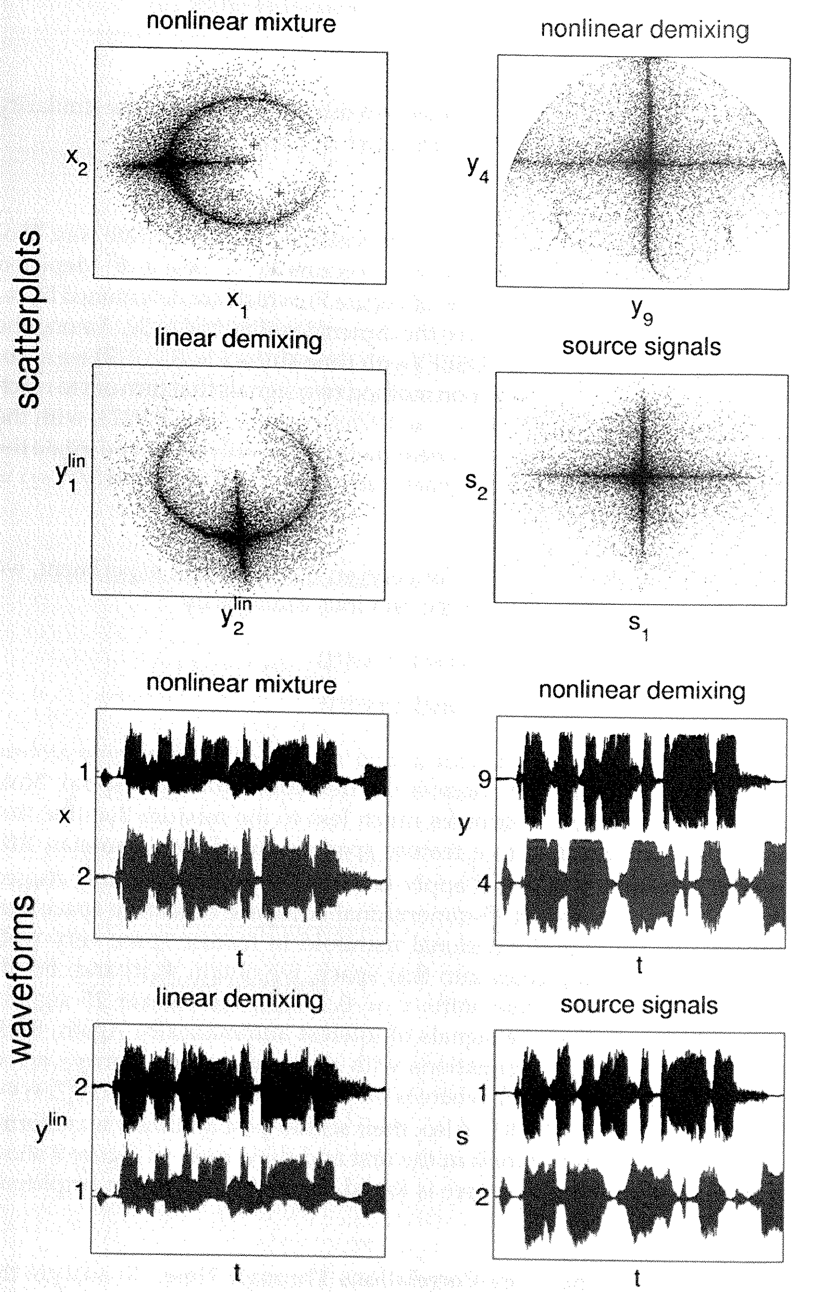

Figure 7: Speech data—bent. Scatter plots and waveforms of the nonlinear mixture and the nonlinear demixing (first and third rows) and of linear demixing and the true source signals (second and fourth rows).

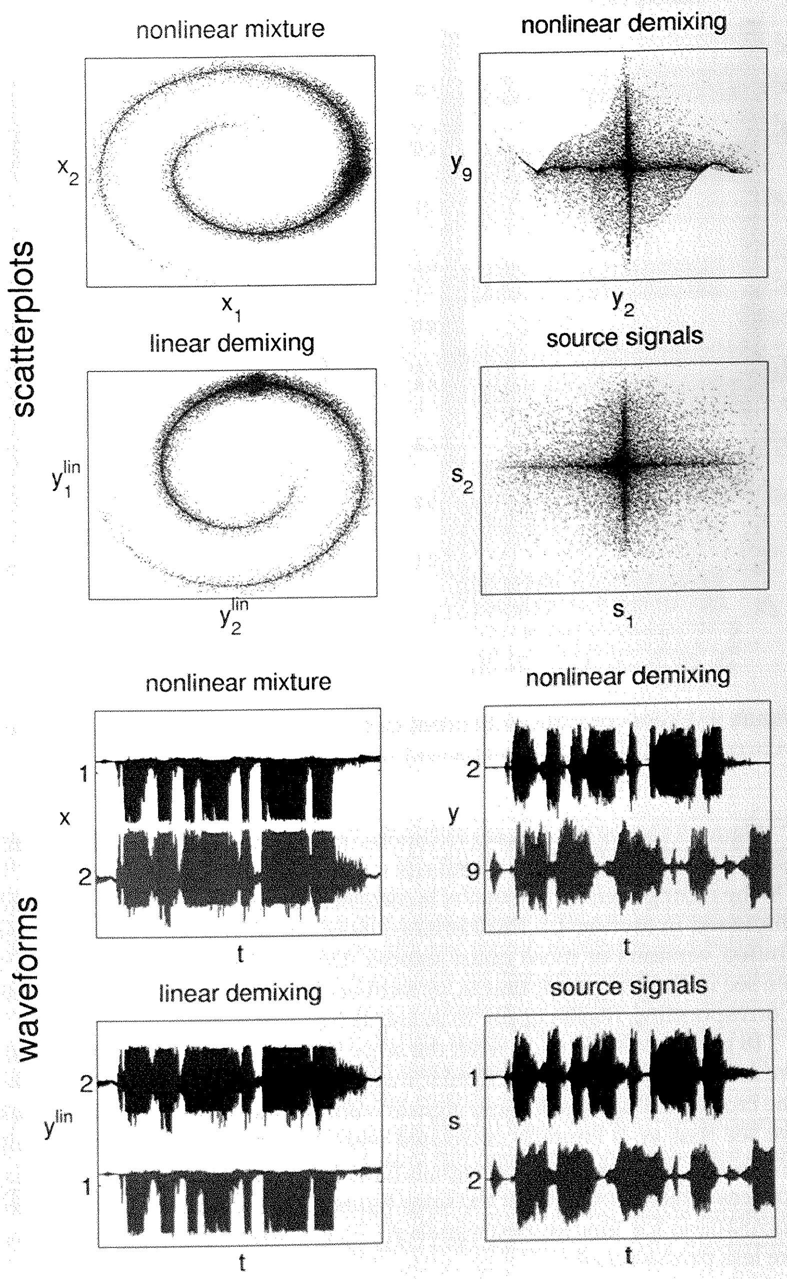

Figure 8: Speech data—twisted. Scatter plots and waveforms of the nonlinear mixture and the nonlinear demixing (first and third rows) and of linear demixing and the true source signals (second and fourth rows).

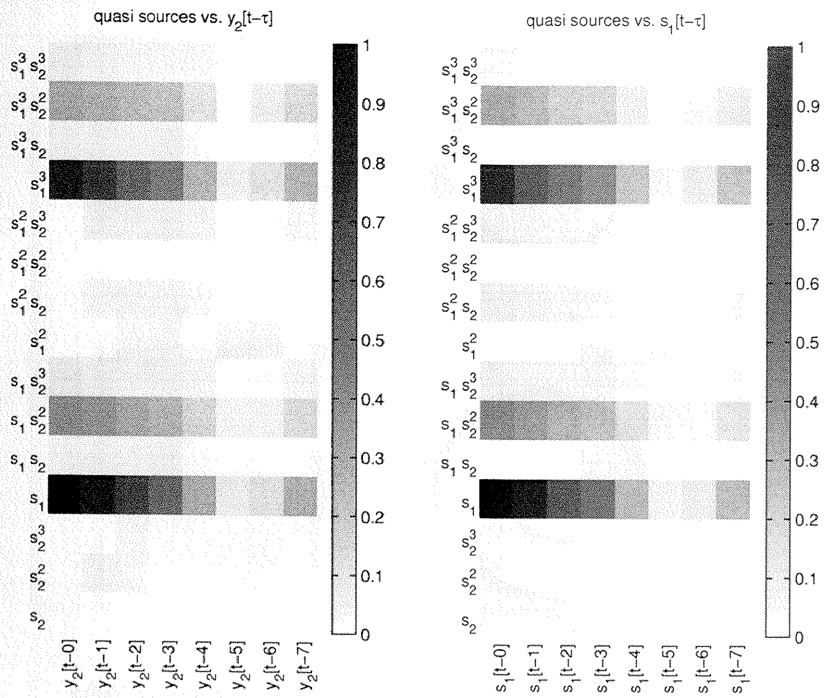

Figure 9: Cross-correlations between (left) the quasi sources and $y_2[t-\tau]$ and between (right) the quasi sources and $s_1[t-\tau]$ for different τ.

Figure 9 shows the cross-correlations of the quasi sources up to degree 3 (see section 4.1) and $y_2[t-\tau]$ and $s_1[t-\tau]$ for different τ (with $\tau = 0, \dots, 7$). On the right panel, we see that s_1 is correlated to $s_1 s_2^2$, s_1^3, and $s_1^3 s_2^2$, as already discussed in section 4.1. Furthermore, there are correlations with the time-shifted versions of those quasi sources. Exactly the same holds for y_2, as we see in the left panel; that is, y_2 recovers s_1, including its time structure. Corresponding results apply to y_9 and s_2 (see Figure 10).

To give some clue as to what the other signals that TDSEP extracted are, we analyzed $y_1, \dots, y_{25}$ in a similar way. The upper panel of Figure 11 shows the cross-correlations of these signals with the quasi sources. For example, we see that y_1 is strongly correlated with s_1^2 or that y_{16} is correlated with $s_1 s_2$, $s_1 s_2^3$, $s_1^3 s_2$, and $s_1^3 s_2^3$. Most signals have close connections to certain quasi sources. The lower panel of the same figure shows the corresponding cross-correlations for time-shifted signals $\mathbf{y}[t-\tau]$. Through time, the correlations are less pronounced.

5.5 Kernel PCA vs. Random Sampling vs. Clustering and the Choice of *d*. In this section, we compare the three proposed dimensionality-reduction methods and discuss the trade-off in choosing the dimensionality of the subspace. We repeat the experiment set out in section 5.3 using different methods for dimensionality reduction—kernel principal component analysis (PCA) versus random sampling versus clustering—and for different subspace dimensionalities d. The results are shown in Figure 12. Overall, it turns out that it does not matter too much for the separation result (measured here as the correlation to the true sources) which of the three reduction methods is used. Kernel PCA has slightly more difficulties for finding the second source for small d. This might seem surprising because kernel PCA is optimal in finding a subspace of the feature space that contains most of the variance of the projected data, so one would expect better performance. The reason is that the PCA criterion does not necessarily optimize for good separation performance.

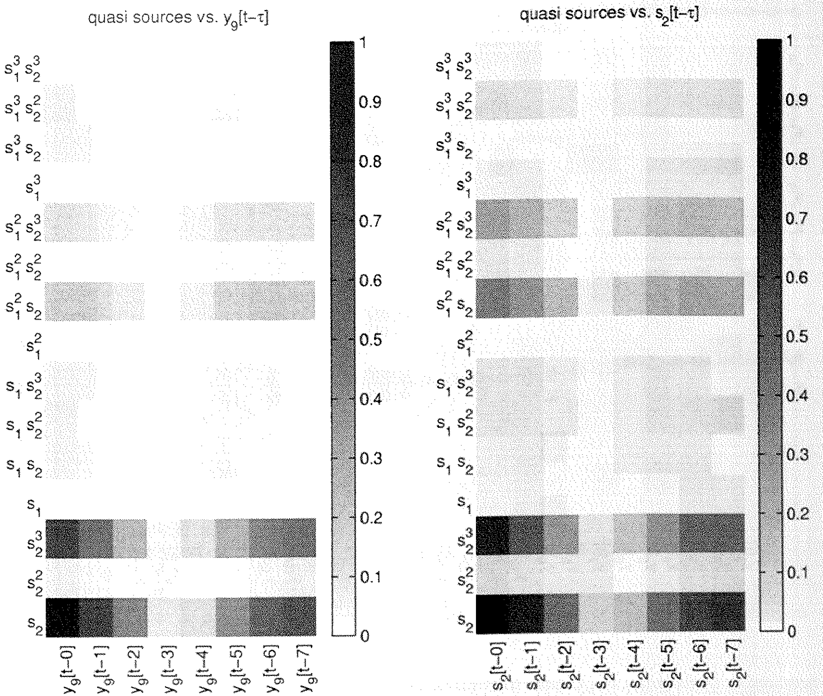

Figure 10: Cross-correlations between (left) the quasi sources and $y_9[t-\tau]$ and between (right) the quasi sources and $s_2[t-\tau]$ for different τ.

Furthermore, we see from the plots that increasing d generally improves the separation performance. But since the running time of TDSEP increases[10]

[10] TDSEP involves simultaneous diagonalization of several $d \times d$ matrices, that is, $O(d^3)$.

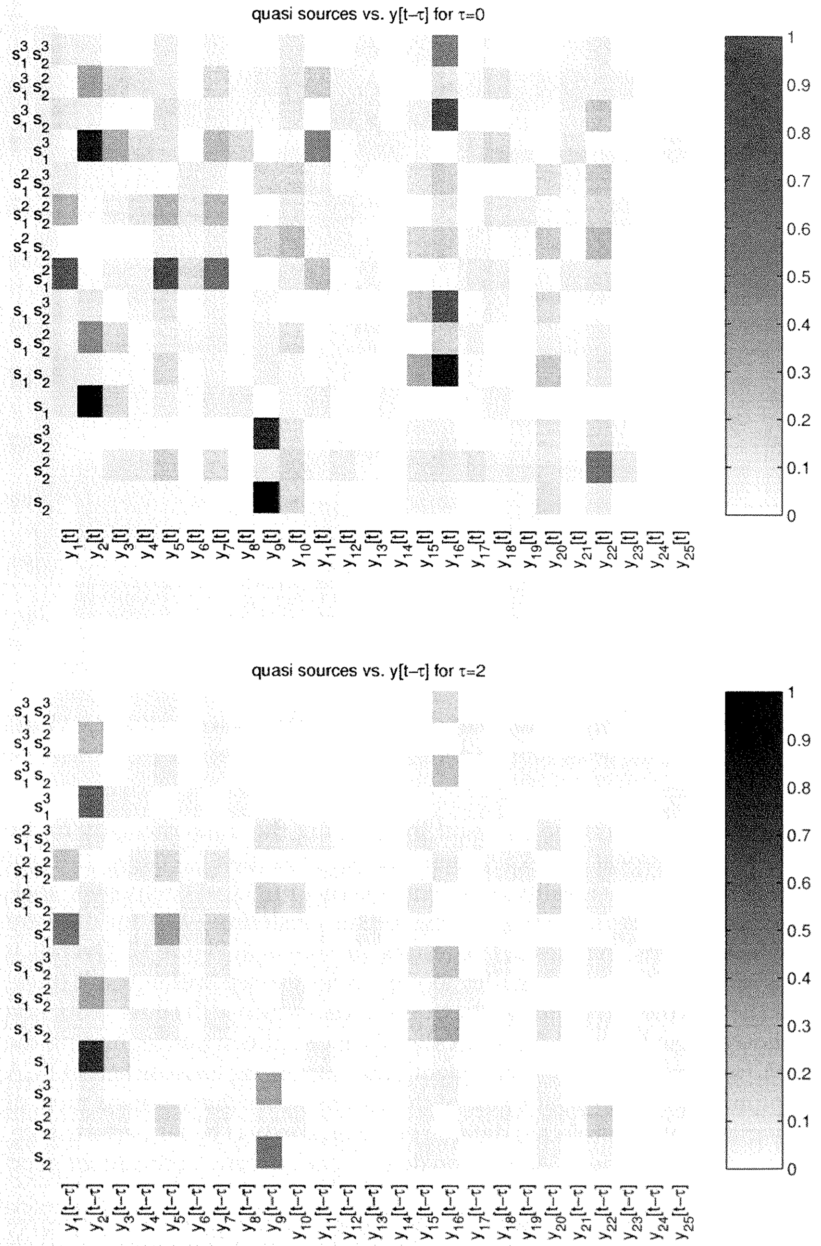

Figure 11: Cross-correlations between the quasi sources and $\mathbf{y}[t-\tau]$ for (top) $\tau = 0$ and (bottom) $\tau = 2$.

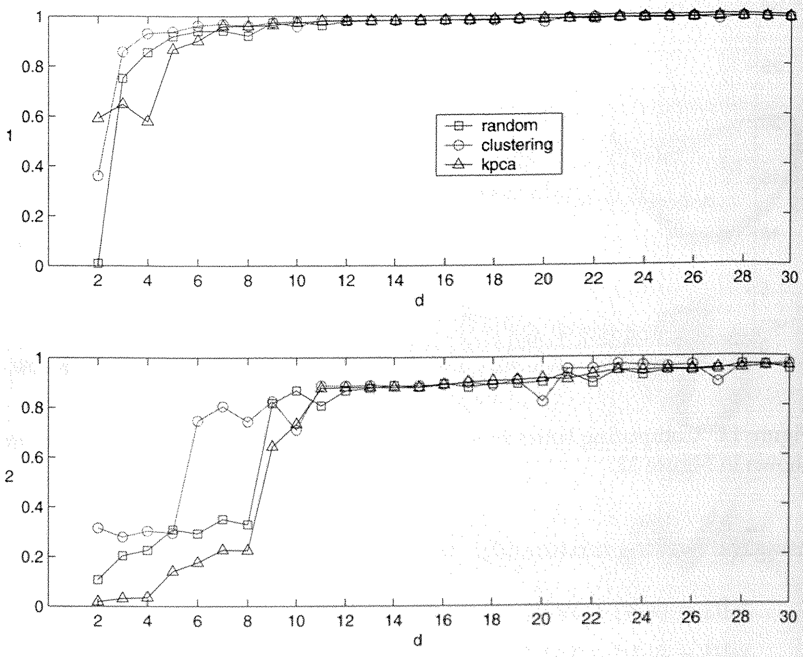

Figure 12: Correlations of the two source signals to the best-matching extracted signals for kernel PCA versus random sampling versus clustering and for different subspace dimensionalities d.

for larger and larger d (see Figure 13), one should try to choose d such that the running time of TDSEP is still tolerable while the subspace is complex enough to demix.

The lower panel of Figure 12 shows some interesting behavior: for small d, the second signal cannot be reconstructed very well. But increasing d to 10 and larger, the subspace has enough complexity to unfold, and thus to recover the second source.

The third column of Figure 14 validates this finding. Shown are the scatter plots for the same experiment, and we see that the biggest improvement happens between $d = 9$ and $d = 12$. The other columns show scatter plots for the two other experiments. Interestingly, the second mixture (in the second column) does not require a very large d; $d = 6$ is enough to recover the sources reasonably well.

5.6 Stochastic Artificial Data. For completeness, we also test our method for stochastic data with short correlation length. We generate 2000 data points from two autoregressive processes of order 3 and mix them

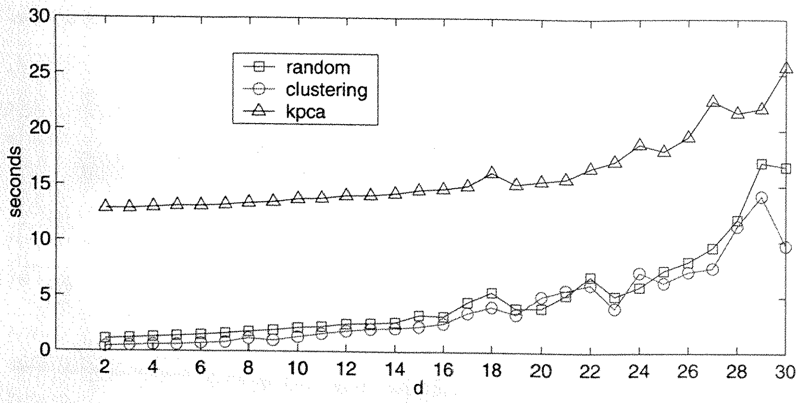

Figure 13: Computing times in seconds for the various runs of the experiment shown in Figure 12.

using the twisting mixture from the previous example:

$$x_1[t] = (s_2[t] + 3s_1[t] + 6)\cos(1.5\pi s_1[t])$$

$$x_2[t] = (s_2[t] + 3s_1[t] + 6)\sin(1.5\pi s_1[t]).$$

We applied kTDSEP with the same parameters as in the previous experiment (kernel PCA applied to 500 randomly chosen input vectors, subspace dimensionality $d = 25$). As in the experiments before, the linear method is not able to uncover the sources ($\mathrm{corr}(y_2^{lin}, s_1) = 0.7893$, $\mathrm{corr}(y_1^{lin}, s_2) = 0.0080$), but our nonlinear method is able ($\mathrm{corr}(y_1, s_1) = 0.9917$, $\mathrm{corr}(y_{10}, s_2) = 0.9370$), as can be seen in Figure 15.

5.7 More Than Two Sources. In the experiments so far, we confined ourselves to mixtures of two sources because they can be nicely visualized. The next experiment demonstrates that our algorithm also works well with more than two sources.

We nonlinearly mix seven audio sources $\mathbf{s}[t] = [s_1[t], \ldots, s_7[t]]^\top$ (piano music, scientific utterance, cembalo music, street noise, cello music, funk music, and political speech, each with 20,000 samples, sampling rate 8 kHz) by two steps:

1. Scale the signals between -1 and 1, that is, they are contained inside the centered hypercube with side length 2. Rotate that cube such that its main diagonal (which has length $2\sqrt{7}$) is aligned with the first axis. This operation can be done by some orthogonal 7×7 matrix $\mathbf{A}$.
2. Rotate around different planes by an angle that depends on the first component of the vector $\bar{\mathbf{s}}[t] = \mathbf{A}\mathbf{s}[t]$, which we denote by $\bar{s}_1[t]$. More

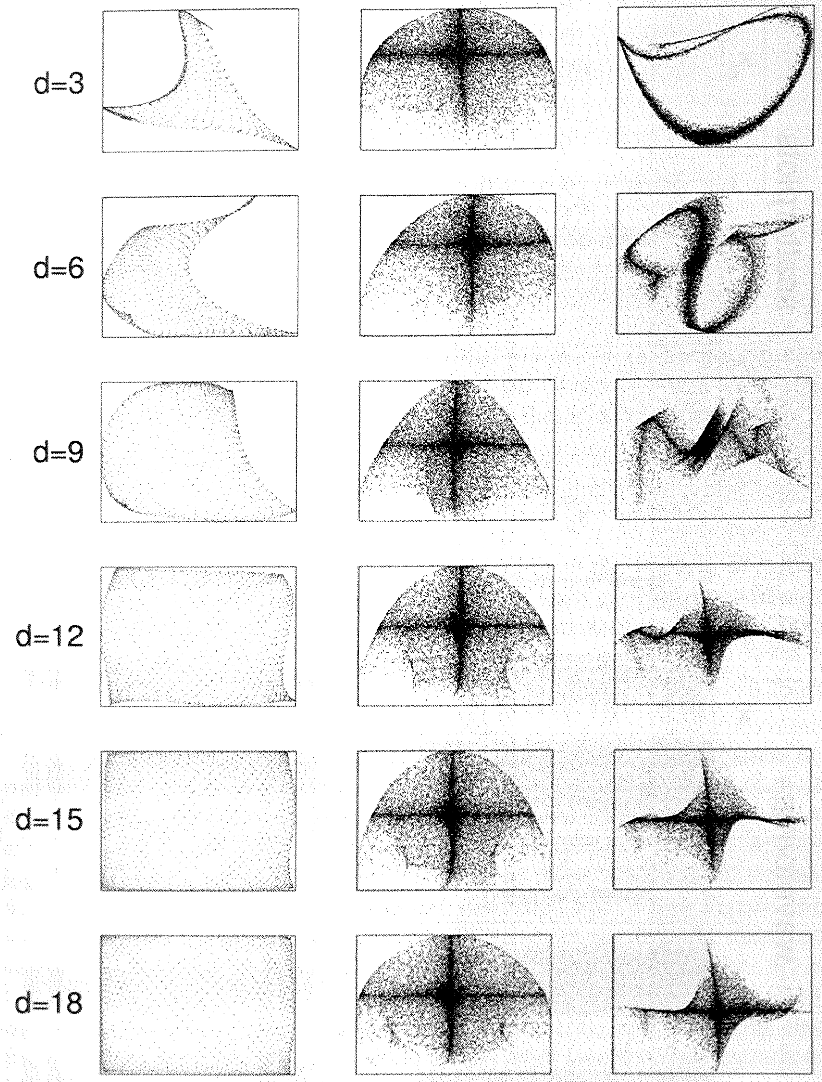

Figure 14: Scatter plots for different values of d: (left) artificial data, (middle) speech data—bent, and (right) speech data—twisted.

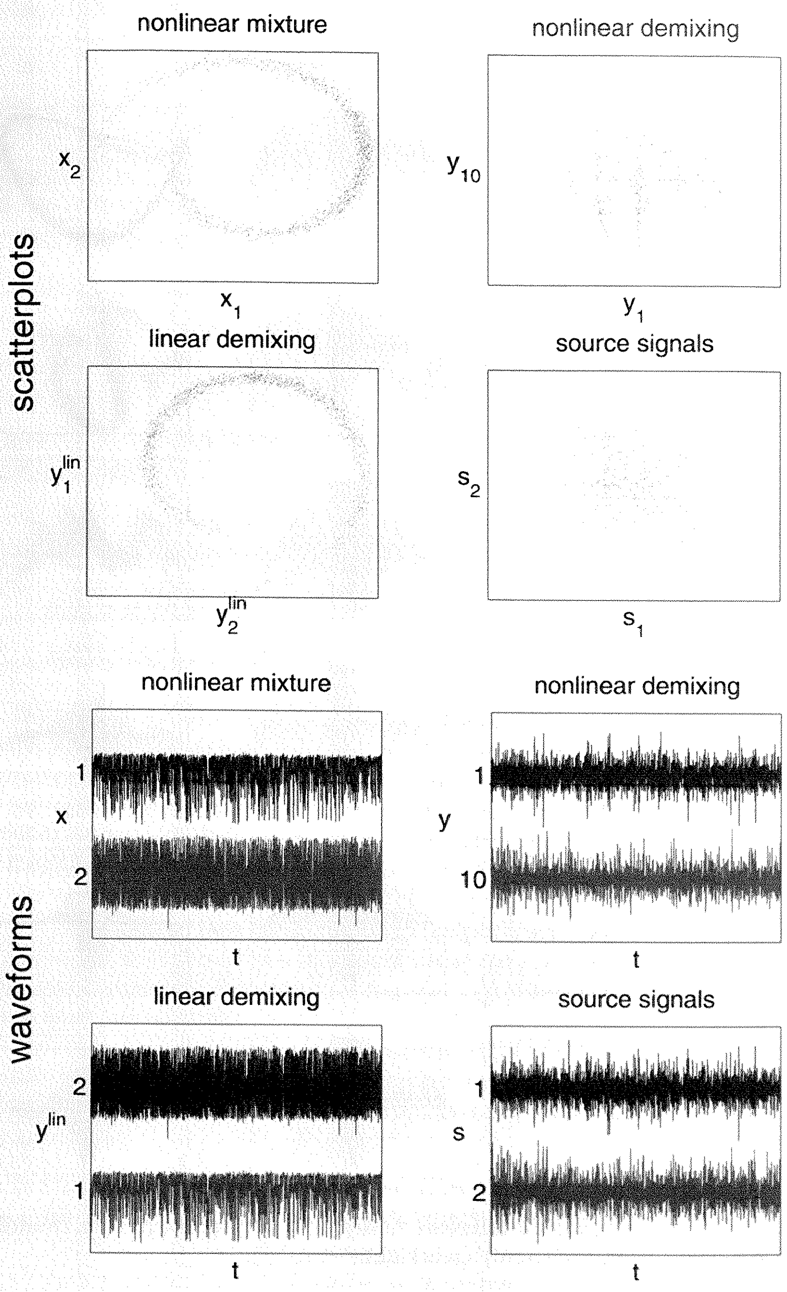

Figure 15: Stochastic artificial data. Scatter plots and waveforms of the nonlinear mixture and the nonlinear demixing (first and third rows) and of linear demixing and the true source signals (second and fourth rows).

precisely, for a vector $\bar{\mathbf{s}}[t]$, we define a 7×7 matrix,

$$\mathbf{B}(\bar{\mathbf{s}}[t]) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \bar{s}_1[t] & 0 & 0 & 0 & 0 \\ 0 & -\bar{s}_1[t] & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{s}_1[t] & 0 & 0 \\ 0 & 0 & 0 & -\bar{s}_1[t] & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \bar{s}_1[t] \\ 0 & 0 & 0 & 0 & 0 & -\bar{s}_1[t] & 0 \end{bmatrix},$$

and use it to define a rotation matrix (employing the matrix exponential),

$$\mathbf{C}(\bar{\mathbf{s}}[t]) = e^{\pi \mathbf{B}(\bar{\mathbf{s}}[t])},$$

that rotates along the second and third planes, along the fourth and fifth planes, and along the sixth and seventh planes by the angle $\pi \bar{s}_1[t]$. Note that $\mathbf{C}(\bar{\mathbf{s}}[t])$ is an orthogonal matrix that is continuous in $\bar{\mathbf{s}}[t]$.

The complete mixture[11] then reads,

$$\mathbf{x}[t] = \mathbf{C}(\mathbf{A}\mathbf{s}[t])\mathbf{A}\mathbf{s}[t].$$

Linear TDSEP is not able to demix $\mathbf{x}[t]$. Listening to the linearly demixed signals reveals that each component contains at least contributions of two sources (and they are distorted as well). Their correlations with the true sources are below 0.82 ($\mathrm{corr}(y_2^{lin}, s_1) = 0.7805$, $\mathrm{corr}(y_1^{lin}, s_2) = 0.7415$, $\mathrm{corr}(y_3^{lin}, s_3) = 0.6663$, $\mathrm{corr}(y_4^{lin}, s_4) = 0.7481$, $\mathrm{corr}(y_5^{lin}, s_5) = 0.7550$, $\mathrm{corr}(y_6^{lin}, s_6) = 0.5929$, $\mathrm{corr}(y_7^{lin}, s_7) = 0.8130$).

For kTDSEP, we apply k-means clustering to 500 randomly chosen input vectors and obtain $d = 40$ vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{40}$. The mapped signals $\Phi(\mathbf{x}[t])$ in the feature space (induced by a gaussian RBF kernel $\mathbf{k}(\mathbf{a}, \mathbf{b}) = \exp(-\|\mathbf{a} - \mathbf{b}\|^2)$) are projected onto the images of those 40 vectors. Applying TDSEP (with time shifts $\tau = 0, \ldots, 30$) to those resulting 40 signals $\Psi_{\mathbf{x}}[t]$ and using the selection procedure described above (see Figure 16), we find the sought-after sources. Those seven nonlinearly demixed signals not only have high correlations with the true sources ($\mathrm{corr}(y_1, s_1) = 0.9432$, $\mathrm{corr}(y_6, s_2) = 0.9496$, $\mathrm{corr}(y_{24}, s_1) = 0.9394$, $\mathrm{corr}(y_{16}, s_2) = 0.9218$, $\mathrm{corr}(y_5, s_1) = 0.9508$, $\mathrm{corr}(y_{22}, s_2) = 0.9142$, $\mathrm{corr}(y_4, s_1) = 0.9402$), but also their waveforms match the true sources very well (see the right panels of Figure 17).

Note that the complexity of the algorithm depends mostly on the choice of d. Even to demix seven nonlinearly mixed sources, the most time-consuming

[11] This mixture is invertible because of the orthogonality and continuity of the matrices involved.

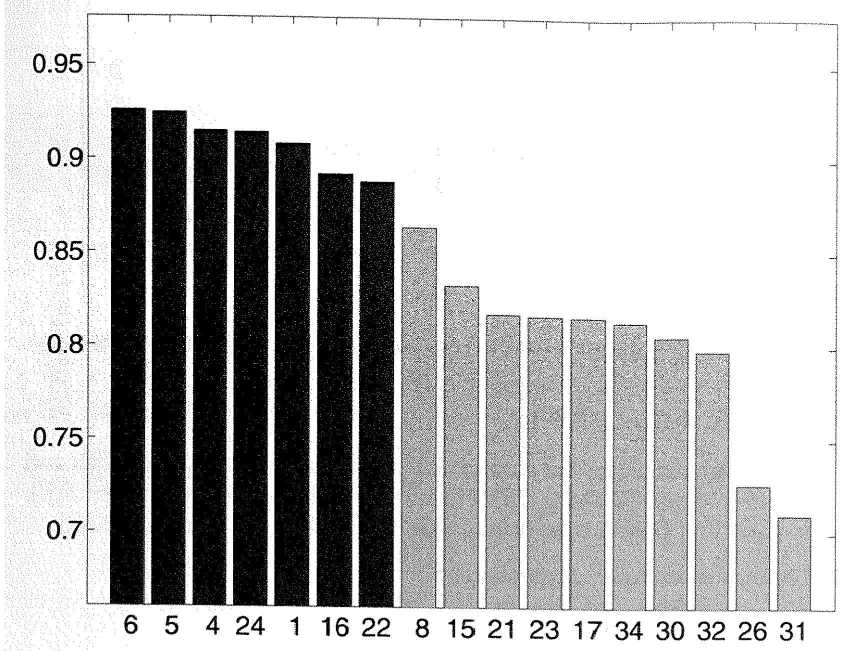

Figure 16: More than two sources: The 17 largest correlations between the demixed signals **y**[t] and the signals that we obtain after applying kTDSEP again to **y**[t]. The numbers at the foot of the bars are the indices of the corresponding signals in **y**[t]. The seven leftmost bars indicate the sought-after sources.

part of the algorithm is simultaneously to diagonalize 31 time-shifted covariance matrices of size 40 × 40, which can be done very fast (Cardoso & Souloumiac, 1996). On a 600 MHz Pentium laptop, the Matlab calculation for this experiment with seven sources took less than 6 minutes.

6 Conclusion

Our work combines three interesting ideas: kernel feature spaces, techniques for dimensionality reduction, and blind source separation. The first two enable us to construct an orthonormal basis of the low-dimensional subspace in kernel feature space $\mathcal{F}$ where the data lie. This technique establishes a highly useful (scalar-product-preserving) isomorphism between the image of the data points in $\mathcal{F}$ and a d-dimensional space $\Re^d$. Moreover, we can acquire knowledge about the intrinsic dimension of the data manifold in $\mathcal{F}$ from the learning process. Furthermore, using this formulation, we tackle the problem of nonlinear BSS from the viewpoint of kernel-based learning. The proposed kTDSEP algorithm allows us to unmix arbitrary invertible nonlinear mixtures with low computational costs. Key to the success of our algorithm are the time correlations exploited by TDSEP; intuitively, they are the glue that provides the coherence for the separated signals.

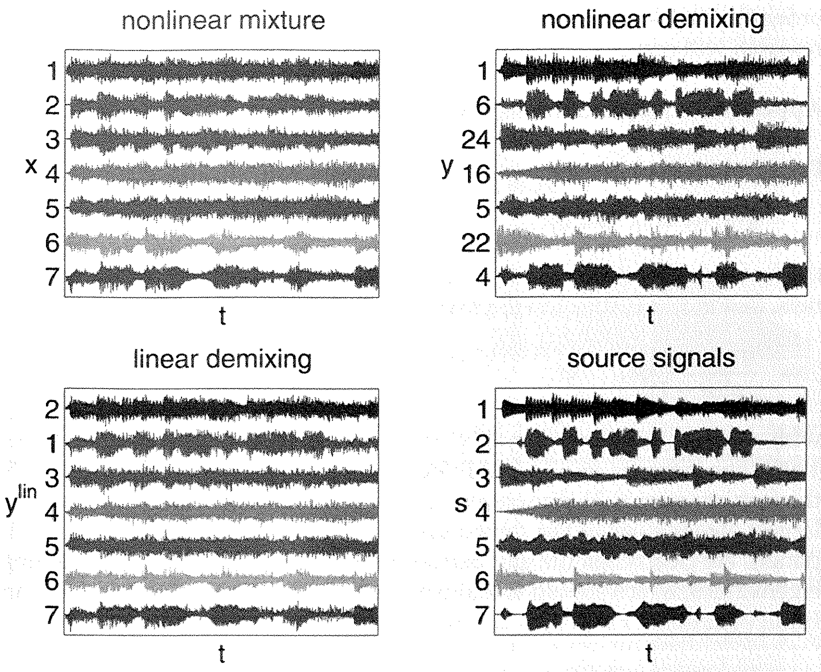

Figure 17: More than two sources: Waveforms of (upper left and right) the nonlinear mixture and the nonlinear demixing and of (lower left and right) the linear demixing and the true source signals.

Experiments on artificially generated signals and various sound signals, also beyond two sources, show that an elegant solution has been found for a challenging problem. Applications where nonlinearly mixed signals can occur are found in the fields of telecommunications, array processing, and biomedical data analysis. Potentially, kTDSEP might contribute to revealing nonlinear phenomena in biological neural systems. Furthermore, our algorithm could be used to provide software-based correction of sensors that have nonlinear characteristics due, for example, to manufacturing errors.

Clearly, kTDSEP is only one algorithm that can perform nonlinear BSS; kernelizing other BSS algorithms can be done following our reasoning.

Recent work by Bach and Jordan (2001) considers a further interesting application of kernel-based methods. In contrast to our algorithm, which provides a nonlinear separation, Bach and Jordan work in the context of *linear* independent component analysis (ICA) and use the kernel trick in order to obtain a clever approximation of the mutual information. Future research will consider combinations of both complementary approaches.

Appendix A: Approximating the Data Manifold in Feature Space

For a polynomial kernel,

$$\mathbf{k}(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^\top \mathbf{b} + c)^p,$$

the feature space is finite-dimensional. For example, for homogeneous kernels, that is, $c = 0$, the dimensionality can be calculated by the formula

$$\frac{(n+p-1)!}{p!(n-1)!},$$

with n being the dimensionality of the mixed signals (taken from Mika, 1998). For instance, at $n = 3$ and for a polynomial kernel of degree $p = 5$, the feature space is 21-dimensional, which can also be seen by plotting the largest eigenvalues of the corresponding kernel matrix (see Figure 18, left panel). We clearly see the gap between the twenty-first and the remaining eigenvalues, which should actually be zero.[12] Obviously, in this case we can fulfill equation 2.1 with $d = 21$. For a gaussian RBF kernel,

$$\mathbf{k}(\mathbf{a}, \mathbf{b}) = e^{-\|\mathbf{a}-\mathbf{b}\|^2},$$

the feature space is infinite-dimensional. However, T data points are always contained in a T-dimensional subspace—the one spanned by the mapped points themselves—but since the corresponding eigenvalues are decaying exponentially fast (see Figure 18, right panel), the data in feature space can be approximated very well with a much lower-dimensional subspace (for more detailed discussions on this issue, see Williams & Seeger, 2000; Bach & Jordan, 2001). Therefore, we can approximate equation 2.1 for suitable d.

Appendix B: Kurtosis-Based ICA in Feature Space?

Exploiting the temporal structure of the source signals is essential for good performance of our method. However, for nonlinear feature extraction, it would be desirable to construct nonlinear kernel ICA methods where the source signals are assumed to be independent in time. We have tried applying standard linear ICA algorithms such as JADE and FastICA (with deflation) in feature space but could not get successful results. One reason might be that general nonlinear ICA problems (see equation 1.1) have nonunique solutions, as explained in Hyvärinen and Pajunen (1999). This indeterminacy in kernel Hilbert space has not been clearly understood yet and is beyond the scope of this article. The other reason is that some of the linear ICA contrast functions are not appropriate in feature space, which we will examine here.

[12] Those eigenvalues are nonzero for numerical reasons.

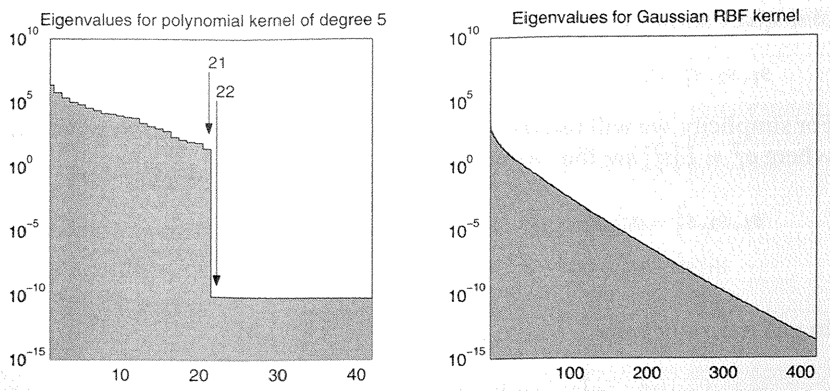

Figure 18: The largest eigenvalues of the kernel matrix on a logarithmic scale: (left) for polynomial kernel of degree 5 and (right) for gaussian RBF kernel.

As an example, we show that by simply applying kurtosis-based ICA methods (deflation) in feature space, we cannot extract the original sources. For this, we consider an example discussed in Harmeling et al. (2001):

$$x_1 = a_{11}s_1 + a_{12}s_2 + b_1 s_1 s_2$$
$$x_2 = a_{21}s_1 + a_{22}s_2 + b_2 s_1 s_2.$$

Let us take a polynomial kernel of order 2,

$$\mathbf{k}(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^\top \mathbf{b} + 1)^2,$$

which introduces a six-dimensional feature space $\mathcal{F}$ represented by

$$x_1, x_2, x_1^2, x_2^2, x_1 x_2, 1.$$

Kurtosis-based ICA methods pick linear mixtures y's of these six basis directions, which are the local maxima of the scaled kurtosis,

$$\tilde{\kappa}(y) = \frac{E[(y-\mu)^4] - 3E[(y-\mu)^2]^2}{E[(y-\mu)^2]^2} = \frac{E[(y-\mu)^4]}{E[(y-\mu)^2]^2} - 3, \tag{B.1}$$

where $\mu = E[y]$. From the bilinear form of the ICA model, we see that any vector in the feature space $\mathcal{F}$ can be represented as a linear combination of nine quasi sources,[13] which are polynomials of s_1 and s_2:

$$s_1, s_2, s_1^2, s_2^2, s_1s_2, s_1^2s_2, s_1s_2^2, s_1^2s_2^2, 1.$$

For simplicity, we will use the following transformed quasi sources instead, where $\sigma_i^2 = E[s_i^2]$ are the variances of the signals:

$$\begin{aligned} &s_1, s_2, s_1^2-\sigma_1^2, s_2^2-\sigma_2^2, s_1s_2, (s_1^2-\sigma_1^2)s_2, \\ &\quad s_1(s_2^2-\sigma_2^2), (s_1^2-\sigma_1^2)(s_2^2-\sigma_2^2), 1. \end{aligned} \tag{B.2}$$

In general, only pairs

$$\alpha_1 s_1 + \alpha_2(s_1^2-\sigma_1^2) \quad \text{and} \quad \beta_1 s_2 + \beta_2(s_2^2-\sigma_2^2), \qquad \alpha_i, \beta_i \in \Re \tag{B.3}$$

are mutually independent.[14] We can show the following proposition, which implies that the kurtosis-based ICA method does not work; that is, it does not give solutions expressed as equation B.3.

Proposition. *Let*

$$y_i(\boldsymbol{\alpha}) = \alpha_1 s_i + \alpha_2(s_i^2-\sigma_i^2), \qquad \alpha_1, \alpha_2 \in \Re, i = 1, 2$$

be an independent signal included in equation B.3. Then inequalities

$$\tilde{\kappa}(y_1(\boldsymbol{\alpha})s_2) \geq \tilde{\kappa}(y_1(\boldsymbol{\alpha})), \qquad \forall \alpha_1, \alpha_2 \in \Re \tag{B.4}$$

$$\tilde{\kappa}(s_1 y_2(\boldsymbol{\alpha})) \geq \tilde{\kappa}(y_2(\boldsymbol{\alpha})), \qquad \forall \alpha_1, \alpha_2 \in \Re \tag{B.5}$$

hold. That is, the kurtosis of the independent signals $y_1(\boldsymbol{\alpha})$ and $y_2(\boldsymbol{\alpha})$ are equal to or smaller than that of

$$\begin{aligned} y_1(\boldsymbol{\alpha})s_2 &= \alpha_1 s_1 s_2 + \alpha_2(s_1^2-\sigma_1^2)s_2, \\ s_1 y_2(\boldsymbol{\alpha}) &= \alpha_1 s_1 s_2 + \alpha_2 s_1(s_2^2-\sigma_2^2), \end{aligned}$$

which are not included in equation B.3.

This proposition means that the original sources s_1 or s_2 or their transformations $y_1(\boldsymbol{\alpha})$ or $y_2(\boldsymbol{\alpha})$ do not have maximum kurtosis, $\tilde{\kappa}$. In other words, independent sources cannot be extracted by applying the deflation method based on kurtosis.

[13] Only a six-dimensional subspace of the space spanned by the nine quasi sources corresponds to the feature space $\mathcal{F}$.

[14] If s_1 or s_2 are subject to some special distributions, it may happen that some other nontrivial pairs become mutually independent (see Hyvärinen & Pajunen, 1999).

Proof. As a special case of the moment inequalities for a random variable z,

$$E[|z|^\alpha]^{\frac{1}{\alpha}} \geq E[|z|^\beta]^{\frac{1}{\beta}}, \qquad \alpha > \beta > 0,$$

the following inequality holds:

$$E[s_i^4] \geq E[s_i^2]^2, \qquad i = 1, 2.$$

Therefore, equation B.4 follows from this inequality easily:

$$\begin{aligned} \tilde{\kappa}(y_1(\boldsymbol{\alpha})s_2) &= \frac{E[y_1(\boldsymbol{\alpha})^4]}{E[y_1(\boldsymbol{\alpha})^2]^2}\frac{E[s_2^4]}{E[s_2^2]^2} - 3 \\ &\geq \frac{E[y_1(\boldsymbol{\alpha})^4]}{E[y_1(\boldsymbol{\alpha})^2]^2} - 3 = \tilde{\kappa}(y_1(\boldsymbol{\alpha})). \end{aligned}$$

The other inequality, equation B.5, or even more general inequalities can be proved in the same way.

Acknowledgments

We thank Benjamin Blankertz, Gunnar Rätsch, Sebastian Mika, and Frank Meinecke for valuable discussions. This work was partly supported by Neurocolt II, the EU project BLISS (IST-1999-14190—BLISS) and DFG (JA 379/9-2, MU 987/1-1). Furthermore, we are grateful to the anonymous reviewers for their highly useful comments. This work is an extension of a conference paper (Harmeling, Ziehe, Kawanabe, & Müller, 2002).

References

Achard, S., Pham, D., & Jutten, C. (2001). Blind source separation in post nonlinear mixtures. In T.-W. Lee (Ed.), *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2001)* (pp. 295–300). San Diego, CA: University of California, San Diego.

Bach, F., & Jordan, M. (2001). *Kernel independent component analysis* (Tech. Rep. UCB/CSD-01-1166). Berkeley: University of California, Berkeley.

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

Burel, G. (1992). Blind separation of sources: A nonlinear neural algorithm. *Neural Networks*, *5*(6), 937–947.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, *2*(2), 121–167.

Cardoso, J.-F. (1998). Blind signal separation: Statistical principles. *Proceedings of the IEEE*, *9*(10), 2009–2025.

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

Cardoso, J.-F., & Souloumiac, A. (1996). Jacobi angles for simultaneous diagonalization. *SIAM J. Mat. Anal. Appl.*, *17*(1), 161–164.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines.* Cambridge: Cambridge University Press.

Fine, S., & Scheinberg, K. (2001). Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, *2*, 243–264.

Fyfe, C., & Lai, P. (2000). ICA using kernel canonical correlation analysis. In P. Pajunen & J. Karhunen (Eds.), *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)* (pp. 279–284). Helsinki, Finland: Helsinki University of Technology.

Harmeling, S., Ziehe, A., Kawanabe, M., Blankertz, B., & Müller, K.-R. (2001). Nonlinear blind source separation using kernel feature spaces. In T.-W. Lee (Ed.), *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2001)* (pp. 102–107). San Diego, CA: University of California, San Diego.

Harmeling, S., Ziehe, A., Kawanabe, M., & Müller, K.-R. (2002). Kernel feature spaces and nonlinear blind source separation. In T. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems, 14.* Cambridge, MA: MIT Press.

Hyvarinen, A., Karhunen, J., & Oja, E. (2001). *Independent component analysis.* New York: Wiley.

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

Hyvärinen, A., Särelä, J., & Vigário, R. (1999). Spikes and bumps: Artefacts generated by independent component analysis with insufficient sample size. In J. F. Cardoso, Ch. Jutten, & Ph. Loubaton (Eds.), *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)* (pp. 425–429). Aussois, France.

Iline, A., Valpola, H., & Oja, E. (2001). Detecting process state changes by nonlinear blind source separation. In T.-W. Lee (Ed.), *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2001)* (pp. 704–709). San Diego, CA: University of California, San Diego.

Lappalainen, H., & Honkela, A. (2000). Bayesian nonlinear independent component analysis by multi-layer perceptrons. In M. Girolami (Ed.), *Advances in independent component analysis* (pp. 93–121). New York: Springer-Verlag.

Lee, T.-W., Koehler, B., & Orglmeister, R. (1997). Blind source separation of nonlinear mixing models. *Neural Networks for Signal Processing VII* (pp. 406–415). Piscataway, NJ: IEEE Press.

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

Marques, G., & Almeida, L. (1999). Separation of nonlinear mixtures using pattern repulsion. In J. F. Cardoso, Ch. Jutten, & Ph. Loubaton (Eds.), *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)* (pp. 277–282). Aussois, France.

Meinecke, F., Ziehe, A., Kawanabe, M., & Müller, K.-R. (2002). Estimating the reliability of ICA projections. In T. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems, 14.* Cambridge, MA: MIT Press.

Meinecke, F., Ziehe, A., Kawanabe, M., & Müller, K.-R. (in press). A resampling approach to estimate the stability of one- or multidimensional independent components. *IEEE Transactions on Biomedical Engineering.*

Mika, S. (1998). *Kernel algorithms for nonlinear signal processing in feature spaces.* Master's thesis, Technical University of Berlin.

Molgedey, L., & Schuster, H. G. (1994). Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, *72*, 3634–3636.

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

Müller, K.-R., Vigario, R., Meinecke, F., & Ziehe, A. (in press). Blind source separation techniques for decomposing event related brain signals. *Bifurcation and Chaos.*

Pajunen, P., Hyvärinen, A., & Karhunen, J. (1996). Nonlinear blind source separation by self-organizing maps. In *Proc. Int. Conf. on Neural Information Processing* (pp. 1207–1210). Hong Kong: Springer-Verlag.

Pajunen, P., & Karhunen, J. (1997). A maximum likelihood approach to nonlinear blind source separation. In W. Gerstner, A. Germond, M. Hasler, & J.-D. Nicoud (Eds.), *Proceedings of the 1997 Int. Conf. on Artificial Neural Networks (ICANN'97)* (pp. 541–546). Lausanne, Switzerland: Springer-Verlag.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*, 465–471.

Schölkopf, B., Mika, S., Burges, C., Knirsch, P., Müller, K.-R., Rätsch, G., & Smola, A. (1999). Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, *10*(5), 1000–1017.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels.* Cambridge, MA: MIT Press.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*, 1299–1319.

Smola, A., & Schölkopf, B. (2000). Sparse greedy matrix approximation for machine learning. In P. Langley (Ed.), *Proc. ICML'00* (pp. 911–918). San Mateo, CA: Morgan Kaufmann.

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

Valpola, H., Giannakopoulos, X., Honkela, A., & Karhunen, J. (2000). Nonlinear independent component analysis using ensemble learning: Experiments and discussion. In P. Pajunen & J. Karhunen (Eds.), *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)* (pp. 351–356). Helsinki, Finland: Helsinki University of Technology.

Vapnik, V. (1995). *The nature of statistical learning theory.* New York: Springer-Verlag.

Williams, C., & Seeger, M. (2000). The effect of the input density distribution on kernel-based classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Williams, C., & Seeger, M. (2001). Using the Nyström method to speed up kernel machines. In V. T. T.G. Dietrich and T. K. Leen (Eds.), *Advances in neural information processing systems, 13* (pp. 682–688). Cambridge, MA: MIT Press.

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

Ziehe, A., Kawanabe, M., Harmeling, S., & Müller, K.-R. (2001). Separation of post-nonlinear mixtures using ACE and temporal decorrelation. In T.-W. Lee (Ed.), *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2001)* (pp. 433–438). San Diego, CA: University of California, San Diego.

Ziehe, A., & Müller, K.-R. (1998). TDSEP—an efficient algorithm for blind separation using time structure. In L. Niklasson, M. Bodén, & T. Ziemke (Eds.), *Proceedings of the 8th International Conference on Artificial Neural Networks* (pp. 675–680). Berlin: Springer-Verlag.

Received February 19, 2002; accepted September 2, 2002.

LETTER Communicated by Raymond Watrous

An Algorithm of Supervised Learning for Multilayer Neural Networks

Zheng Tang
tang@iis.toyama-u.ac.jp
XuGang Wang
xugunw76@hotmail.com
Hiroki Tamura
tamura@iis.toyama-u.ac.jp
Masahiro Ishii
ishii@iis.toyama-u.ac.jp
Faculty of Engineering, Toyama University, 3190 Gofuku Toyama 930-8555, Japan

A method of supervised learning for multilayer artificial neural networks to escape local minima is proposed. The learning model has two phases: a backpropagation phase and a gradient ascent phase. The backpropagation phase performs steepest descent on a surface in weight space whose height at any point in weight space is equal to an error measure, and it finds a set of weights minimizing this error measure. When the backpropagation gets stuck in local minima, the gradient ascent phase attempts to fill up the valley by modifying gain parameters in a gradient ascent direction of the error measure. The two phases are repeated until the network gets out of local minima. The algorithm has been tested on benchmark problems, such as exclusive-or (XOR), parity, alphabetic characters learning, Arabic numerals with a noise recognition problem, and a realistic real-world problem: classification of radar returns from the ionosphere. For all of these problems, the systems are shown to be capable of escaping from the backpropagation local minima and converge faster when using the new proposed method than using the simulated annealing techniques.

1 Introduction

The backpropagation algorithm (Rumelhart, Hinton, & Williams, 1986) is one of the most widely used and effective algorithms for training feedforward neural networks. Over the past decade, backpropagation and its variations have achieved increasing popularity among scientists, engineers, and other professionals as tools for tackling a wide variety of information processing tasks. Unfortunately, the traditional method for training multilayer networks is notoriously slow and unreliable when applied to many practical tasks. Several fast training algorithms, for example, fast second-order training methods, have been proposed (Watrous, 1987; Kramer &