Kernel-Based Nonlinear Independent Component Analysis

Kun Zhang and Laiwan Chan*

Department of Computer Science and Engineering,
The Chinese University of Hong Kong
Shatin, Hong Kong
{kzhang,lwchan}@cse.cuhk.edu.hk

Abstract. We propose the kernel-based nonlinear independent component analysis (ICA) method, which consists of two separate steps. First, we map the data to a high-dimensional feature space and perform dimension reduction to extract the effective subspace, which was achieved by kernel principal component analysis (PCA) and can be considered as a pre-processing step. Second, we need to adjust a linear transformation in this subspace to make the outputs as statistically independent as possible. In this way, nonlinear ICA, a complex nonlinear problem, is decomposed into two relatively standard procedures. Moreover, to overcome the ill-posedness in nonlinear ICA solutions, we utilize the minimal nonlinear distortion (MND) principle for regularization, in addition to the smoothness regularizer. The MND principle states that we would prefer the nonlinear ICA solution with the mixing system of minimal nonlinear distortion, since in practice the nonlinearity in the data generation procedure is usually not very strong.

1 Introduction

Independent component analysis (ICA) aims at recovering independent sources from their mixtures, without knowing the mixing procedure or any specific knowledge of the sources. In particular, in this paper we consider the general nonlinear ICA problem. Assume that the observed data $\mathbf{x} = (x_1, \cdots, x_n)^T$ are generated from an independent random vector $\mathbf{s} = (s_1, \cdots, s_n)^T$ by a nonlinear transformation $\mathbf{x} = \mathcal{H}(\mathbf{s})$, where $\mathcal{H}$ is an unknown real-valued n-component mixing function. (For simplicity, it is usually assumed that the number of observable variables equals that of the original independent variables.) The general nonlinear ICA problem is to find a mapping $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^n$ such that $\mathbf{y} = \mathcal{G}(\mathbf{x})$ has statistically independent components.

In the general nonlinear ICA problem, in order to model arbitrary nonlinear mappings, one may need to resort to a flexible nonlinear function approximator, such as the multi-layer perceptron (MLP) network or the radius basis function (RBF) network, to represent the nonlinear separation system $\mathcal{G}$ or the mixing system $\mathcal{H}$ (see,

* This work was partially supported by a grant from the Research grants Council of the Hong Kong Special Administration Region, China.

M.E. Davies et al. (Eds.): ICA 2007, LNCS 4666, pp. 301–308, 2007.

e.g. [1]). In such a way, parameters at different locations of the network are adjusted simultaneously. This would probably slow down the learning procedure.

Kernel-based methods has also been considered for solving the nonlinear blind source separation (BSS) problem [5,10].[1] These methods exploit the temporal information of sources for source separation, and do not enforce mutual independence of outputs. In [5], the data are first implicitly mapped to high-dimensional feature space, and the effective subspace in feature space is extracted. TD-SEP [13], a BSS algorithm based on temporal decorrelation, is then performed in the extracted subspace. Denote by d the reduced dimension. This method produces d outputs and one needs to select from them n outputs, as an estimate of the original sources. This method produces successful results in many experiments. However, a problem is that its outputs may not contain the estimate of the original sources, due to the effect of spurious outputs. Moreover, this method may fail if some sources lack specific time structures.

In this paper we propose a kernel-based method to solve nonlinear ICA. The separation system $\mathcal{G}$ is constructed using the kernel methods, and unknown parameters are adjusted by minimizing the mutual information between outputs y_i. The first step of this method is similar to that in [5], and kernel principal component analysis (PCA) is adopted to construct the feature subspace of reduced dimension. In the second step we solve a linear problem—we adjust the $n \times d$ linear transformation matrix $\mathbf{W}$ to make the outputs statistically independent. As stated in [5], standard linear ICA algorithms do not work here. We derive the algorithm for learning $\mathbf{W}$, which is in a similar form to the traditional gradient-based ICA algorithm.

We then consider suitable regularization conditions with which the proposed kernel-based nonlinear ICA leads to nonlinear BSS. In the general nonlinear ICA problem, although we do not know the form of the nonlinearity in the data generation procedure, fortunately, the nonlinearity in the generation procedure of natural signals is usually not strong. Hence, provided that the nonlinear ICA outputs are mutually independent, we would prefer the solution with the mixing transformation as close as possible to linear. This information, formulated as the minimal nonlinear distortion (MND) principle [12], can help to reduce the indeterminacies in solutions of nonlinear ICA greatly. MND and smoothness are incorporated for regularization in the kernel-based nonlinear ICA.

2 Kernel-Based Nonlinear ICA

Kernel-based learning has become a popular technique, in that it provides an elegant way to tackle nonlinear algorithms by reducing them to linear ones in some feature space $\mathcal{F}$, which is related to the original input space $\mathbb{R}^n$ by a possibly nonlinear map Φ. Denote by $\mathbf{x}^{(i)}$ the ith sample of $\mathbf{x}$. The dot products of the form $\Phi(\mathbf{x}^{(i)}) \cdot \Phi(\mathbf{x}^{(j)})$ can be computed using kernel representations $k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \Phi(\mathbf{x}^{(i)}) \cdot \Phi(\mathbf{x}^{(j)})$. Thus, any linear algorithm formulated in terms of dot products can be made nonlinear by making use of the kernel trick, without

[1] Note that kernel ICA [3] actually performs *linear* ICA with the kernel trick.

knowing explicitly the mapping Φ. Unfortunately, ICA could not be kernelized directly, since it can not be carried out using dot products.

However, the kernel trick can still help to perform nonlinear ICA, in an analogous manner to the development of kTDSEP, which is a kernel-based algorithm for nonlinear BSS [5]. Kernel-based nonlinear ICA involves two separate steps. The first step is the same as that in kTDSEP: the data are implicitly mapped to a high-dimensional feature space and its effective subspace is extracted. As a consequence, the nonlinear problem in input space is transformed to a linear one in the reduced feature space. In the second step, a linear transformation in the reduced feature space is constructed such that it produces n statistically independent outputs. In this way nonlinear ICA is performed faithfully, without any assumption on the time structure of sources.

Many techniques can help to find the effective subspace in feature space $\mathcal{F}$. Here we adopt kernel PCA [11], since the subspace it produces gives the smallest reconstruction error in feature space. The effective dimension of feature space, denoted by d, can be determined by inspection on the eigenvalues of the kernel matrix. Let $\mathbf{x}$ be a test point, and let $\tilde{k}(\mathbf{x}^{(i)}, \mathbf{x}) = \tilde{\Phi}(\mathbf{x}^{(i)}) \cdot \tilde{\Phi}(\mathbf{x})$, where $\tilde{\Phi}$ denotes the centered image in feature space. The pth centered nonlinear principal component of $\mathbf{x}$, denoted by z_p, is in the form (for details see [11]):

$$z_p = \sum_{i=1}^{T} \tilde{\alpha}_{pi} \tilde{k}(\mathbf{x}^{(i)}, \mathbf{x}) \tag{1}$$

Let $\mathbf{z} = (z_1, \cdots, z_d)^T$. It contains all principal components of the images $\Phi(\mathbf{x})$ in feature space. Consequently, in the following we just need find a $n \times d$ matrix $\mathbf{W}$ which makes the components of

$$\mathbf{y} = \mathbf{W}\mathbf{z} \tag{2}$$

as statistically independent as possible.

2.1 Can Standard Linear ICA Work in Reduced Feature Space?

As claimed in [5], applying standard linear ICA algorithms, such as JADE [4] and FastICA [6], to the signals $\mathbf{z}$ does not give successful results. In our problem, z_p, $p = 1, \cdots, d$, are nonlinear mixtures of only n independent sources, and we aim at *transforming z_p to n signals* (generally $n \ll d$) *which are statistically independent*, with a linear transformation. But standard ICA algorithms, such as the natural gradient algorithm [2] and JADE, assume that $\mathbf{W}$ is square and invertible and try to extract d independent signals from z_i. So they can not give successful results in our problem.

Although FastICA, which aims at maximizing the nongaussianity of outputs, can be used in a deflationary manner, its relation to maximum likelihood of the ICA model or minimization of mutual information between outputs is established when the linear ICA model holds and $\mathbf{W}$ is square and invertible [7]. When the linear ICA model does not hold, just like in our problem, nongaussianity of outputs does not necessarily lead to the independence between them. In fact, if we apply FastICA in a deflationary manner to z_i, the outputs y_i will be

extremely nongaussian, but they are not necessarily mutually independent. The extreme nongaussianity of y_i is because theoretically, with a properly chosen kernel function, by adjusting the ith row of $\mathbf{W}$ the mapping from $\mathbf{x}$ to y_i covers quite a large class of continuous functions.

2.2 Learning Rule

Now we aim to adjust $\mathbf{W}$ in Eq. 2 to make the outputs y_i as independent as possible. This can be achieved by minimizing the mutual information between y_i, which is defined as $I(\mathbf{y}) = \sum_{i=1}^{n} H(y_i) - H(\mathbf{y})$ where $H(\cdot)$ denotes the differential entropy. Denote by $\mathbf{J}$ the Jacobian matrix of the transformation from $\mathbf{x}$ to $\mathbf{y}$, i.e. $\mathbf{J} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}}$, and by $\mathbf{J}_1$ the Jacobian matrix of the transformation from $\mathbf{x}$ to $\mathbf{z}$, i.e. $\mathbf{J}_1 = \frac{\partial z}{\partial x}$.[2] Due to Eq. 2, one can see $\mathbf{J} = \mathbf{W} \cdot \mathbf{J}_1$. We also have $H(\mathbf{y}) = H(\mathbf{x}) + E \log |\det \mathbf{J}|$. Consequently,

$$I(\mathbf{y}) = \sum_{i=1}^{n} H(y_i) - H(\mathbf{y}) = -\sum_{i=1}^{n} \log p_{y_i}(y_i) - E \log |\det(\mathbf{W} \cdot \mathbf{J}_1)| - H(\mathbf{x})$$

As $H(\mathbf{x})$ does not depend on $\mathbf{W}$, the gradient of $I(\mathbf{y})$ w.r.t. $\mathbf{W}$ is

$$\frac{\partial I(\mathbf{y})}{\partial \mathbf{W}} = E[\boldsymbol{\psi}(\mathbf{y}) \cdot \mathbf{z}^T] - E[\mathbf{J}^{-T} \cdot \mathbf{J}_1^T] \tag{3}$$

where $\boldsymbol{\psi}(\mathbf{y}) = (\psi_1(y_1), \cdots, \psi_n(y_n))^T$ with ψ_i being the score function of p_{y_i}, defined as $\psi_i = -(\log p_{y_i})' = -\frac{p'_{y_i}}{p_{y_i}}$. $\mathbf{W}$ can then be adjusted according to Eq. 3 with the gradient-based method. Note that the gradient in Eq. 3 is in a similar form to that in standard ICA, and the only difference is that the second term becomes $-E[\mathbf{W}^{-T}]$ in standard ICA[3].

In standard ICA, we can obtain correct ICA results even if the estimation of the densities p_{y_i} or the score functions ψ_i is not accurate. But in the nonlinear case, they should be estimated accurately. We use the mixture of 5 Gaussian's to model p_{y_i}. After each iteration of Eq. 3, parameters in the Gaussian mixture model are adjusted by the EM algorithm to adapt the current outputs y_i.

3 With Minimum Nonlinear Distortion

Solutions to nonlinear ICA always exist and are highly non-unique [8]. In fact, in the general nonlinear ICA problem, nonlinear BSS is impossible without additional prior knowledge on the mixing model [9]. Smoothness of the mapping

[2] $\mathbf{J}_1$ is involved in the obtained update rule Eq. 3. Since $\tilde{k}(\mathbf{x}^{(i)}, \mathbf{x}) = \tilde{\Phi}(\mathbf{x}^{(i)}) \cdot \tilde{\Phi}(\mathbf{x}) = k(\mathbf{x}^{(i)}, \mathbf{x}) - \frac{1}{T}\sum_{p=1}^{T} k(\mathbf{x}^p, \mathbf{x}) - \frac{1}{T}\sum_{q=1}^{T} k(\mathbf{x}^{(i)}, \mathbf{x}^{(q)}) + \frac{1}{T^2}\sum_{p=1}^{T}\sum_{q=1}^{T} k(\mathbf{x}^{(p)}, \mathbf{x}^{(q)})$. We have $\frac{\partial \tilde{k}(\mathbf{x}^{(i)}, \mathbf{x})}{\partial \mathbf{x}} = \frac{\partial k(\mathbf{x}^{(i)}, \mathbf{x})}{\partial \mathbf{x}} - \frac{1}{T}\sum_{p=1}^{T} \frac{\partial k(\mathbf{x}^{(p)}, \mathbf{x})}{\partial \mathbf{x}}$. According to Eq. 1, the pth row of $\mathbf{J}_1$ is then $\frac{\partial z_p}{\partial \mathbf{x}} = \sum_{i=1}^{T} \tilde{\alpha}_{pi} \frac{\partial \tilde{k}(\mathbf{x}^{(i)}, \mathbf{x})}{\partial \mathbf{x}}$. This can be easily calculated and saved in the first step of our method for later use.

[3] Assuming that $\mathbf{W}$ is square and invertible, the natural gradient ICA algorithm is obtained multiplying the right-hand side of $\frac{\partial I(\mathbf{y})}{\partial \mathbf{W}}$ by $\mathbf{W}^T\mathbf{W}$ [2]. However, as $\mathbf{W}$ in Eq. 2 is $n \times d$, the natural gradient for $\mathbf{W}$ could not be derived in this simple way.

$\mathcal{G}$ provides a useful regularization condition to lead nonlinear ICA to nonlinear BSS [1]. But it seems not sufficient, as shown by the counterexample in [9].

In this paper, in addition to the smoothness regularization, we exploit the "minimal nonlinear distortion" (MND) principle [12] for regularization of nonlinear ICA. MND has exhibited quite good performance for regularizing nonlinear ICA, when the nonlinearity in the data generation procedure is not very strong [12]. The objective function to be minimized thus becomes

$$J(\mathbf{W}) = I(\mathbf{y}) + \lambda_1 R_1(\mathbf{W}) + \lambda_2 R_2(\mathbf{W}) \tag{4}$$

where R_1 denotes the regularization term for achieving MND, R_2 is that for enforcing smoothness, and λ_1 and λ_2 are corresponding regularization parameters.

3.1 Minimum Nonlinear Distortion

MND states that, under the condition that the separation outputs y_i are mutually independent, we prefer the nonlinear mixing mapping $\mathcal{H}$ that is as close as possible to linear. So R_1 is a measure of "closeness to linear" of $\mathcal{H}$. Given a nonlinear mapping $\mathcal{H}$, its deviation from the affine mapping $\mathbf{A}^*$, which fits $\mathcal{H}$ best among all affine mappings $\mathbf{A}$, is an indicator of its "closeness to linear" or the level of its nonlinear distortion. Mean square error (MSE) is adopted to measure the deviation, since it greatly facilitates the following analysis. Let $\mathbf{x}^* = (x_1^*, \cdots, x_n^*)^T = \mathbf{A}^*\mathbf{y}$. R_1 can be defined as the total MSE between x_i and x_i^* (here we assume that both $\mathbf{x}$ and $\mathbf{y}$ are zero-mean):

$$\begin{aligned} &R_1 = E\{(\mathbf{x}-\mathbf{x}^*)^T(\mathbf{x}-\mathbf{x}^*)\}\ \text{, where} \\ &\mathbf{x}^* = \mathbf{A}^*\tilde{\mathbf{y}},\ \text{and}\ \mathbf{A}^* = \arg_{\mathbf{A}}\min E\{(\mathbf{x}-\mathbf{A}\mathbf{y})^T(\mathbf{x}-\mathbf{A}\mathbf{y})\} \end{aligned} \tag{5}$$

The derivative of R_1 w.r.t. $\mathbf{A}^*$ is $\frac{\partial R_1}{\partial \mathbf{A}^*} = -2E\{(\mathbf{x}-\mathbf{A}^*\mathbf{y})\mathbf{y}^T\}$. Setting the derivative to $\mathbf{0}$ gives $\mathbf{A}^*$: $E\{(\mathbf{x}-\mathbf{A}^*\tilde{\mathbf{y}})\tilde{\mathbf{y}}^T\} = \mathbf{0} \Leftrightarrow \mathbf{A}^* = E\{\mathbf{x}\mathbf{y}^T\}[E\{\mathbf{y}\mathbf{y}^T\}]^{-1}$. We can see that due to the adoption of MSE, $\mathbf{A}^*$ can be obtained in closed form. This will greatly simplify the derivation of learning rules.

We then have $R_1 = \mathrm{Tr}\big(E[(\mathbf{x}-\mathbf{A}^*\mathbf{y})(\mathbf{x}-\mathbf{A}^*\mathbf{y})^T]\big) = -\mathrm{Tr}\big(E[\mathbf{x}\mathbf{y}^T]\{E[\mathbf{y}\mathbf{y}^T]\}^{-1} \cdot E[\mathbf{y}\mathbf{x}^T]\big) + \text{const}$. Since in the learning process, y_i are approximately independent from each other, they are approximately uncorrelated. We can also normalize the variance of y_i after each iteration. Consequently $E[\mathbf{y}\mathbf{y}^T] = \mathbf{I}$. Let $\mathbf{L} = E[\mathbf{x}\mathbf{z}^T]$. we have $E[\mathbf{x}\mathbf{y}^T] = \mathbf{L}\mathbf{W}^T$. Thus $R_1 = -\mathrm{Tr}(\mathbf{L}\mathbf{W}^T\mathbf{W}\mathbf{L}^T) + \text{const}$. This gives

$$\frac{\partial R_1}{\partial \mathbf{W}} = -2\mathbf{W}\mathbf{L}^T\mathbf{L} \tag{6}$$

It was suggested to initialize λ_1 in Eq. 4 with a large value at the beginning of training and decreasing it to a small constant during the learning process [12]. A large value for λ at the beginning helps to reduce the possibility of getting into unwanted solutions or local optima. As training goes on, the influence of the regularization term is reduced, and $\mathcal{G}$ gains more freedom. In addition, initializing $\mathcal{G}$ to an almost identity mapping would also be useful. This can be achieved by simply initializing $\mathbf{W}$ with $\mathbf{W} = E[\mathbf{x}\mathbf{z}^T]\{E[\mathbf{z}\mathbf{z}^T]\}^{-1}$.

The MND principle can be incorporated in many nonlinear ICA/BSS methods to avoid unwanted solutions, under the condition that the nonlinearity in the mixing procedure is not too strong. As an example, for kTDSEP [5], MND provides a way to select a subset of output components corresponding to the original sources [12].

3.2 Smoothness: Local Minimum Nonlinear Distortion

Both MND and smoothness are used for regularization in our nonlinear ICA method. In fact, the smoothness regularizer exploiting second-order derivatives is related to MND. Particularly, enforcing *local* closeness to linear of the transformation at every point will lead to such a smoothness regularizer [12].

For a one-dimensional sufficiently smooth function $g(\mathbf{x})$, its second-order Taylor expansion in the vicinity of $\mathbf{x}$ is $g(\mathbf{x}+\boldsymbol{\varepsilon}) \approx g(\mathbf{x})+\left(\frac{\partial g}{\partial \mathbf{x}}\right)^T \cdot \boldsymbol{\varepsilon}+\frac{1}{2}\boldsymbol{\varepsilon}^T \mathbf{H}_{\mathbf{x}} \boldsymbol{\varepsilon}$. Here $\boldsymbol{\varepsilon}$ is a small variation of $\mathbf{x}$ and $\mathbf{H}_{\mathbf{x}}$ denotes the Hessian matrix of g. Let $\bigtriangledown_{ij} = \frac{\partial^2 g}{\partial x_i \partial x_j}$. It can be shown [12] that if we use the first-order Taylor expansion of g at $\mathbf{x}$ to approximate $g(\mathbf{x}+\boldsymbol{\varepsilon})$, the square error is

$$\left|\left| g(\mathbf{x}+\boldsymbol{\varepsilon}) - g(\mathbf{x}) - \left(\frac{\partial g}{\partial \mathbf{x}}\right)^T \cdot \boldsymbol{\varepsilon} \right|\right|^2 \approx \frac{1}{4}\left|\left|\boldsymbol{\varepsilon}^T \mathbf{H}_{\mathbf{x}} \boldsymbol{\varepsilon}\right|\right|^2 = \frac{1}{4}\Big(\sum_{i,j=1}^{n} \bigtriangledown_{ij} \varepsilon_i \varepsilon_j \Big)^2$$

$$\leq \frac{1}{4}\Big(\sum_{i=1}^{n} \bigtriangledown_{ii}^2 + 2 \sum_{i,j=1, i\neq j}^{n} \bigtriangledown_{ij}^2 \Big)\Big(\sum_{i=1}^{n} \varepsilon_i^4 + 2 \sum_{i,j=1, i\neq j}^{n} \varepsilon_i^2 \varepsilon_j^2 \Big) = \frac{1}{4}||\boldsymbol{\varepsilon}||^4 \cdot \sum_{i,j=1}^{n} \bigtriangledown_{ij}^2$$

The above inequality holds due to the Cauchy's inequality. We can see that in order to make g locally close to linear at every point in the domain of $\mathbf{x}$, we just need minimize $\int_{D_{\mathbf{x}}} \sum_{i,j=1}^{n} \bigtriangledown_{ij}^2 d\mathbf{x}$. When the mapping is vector-valued, we need apply this regularizer to each output of the mapping. R_2 in Eq. 4 can then be constructed as $R_2 = \int_{D_{\mathbf{x}}} \sum_{i,j=1}^{n} P_{ij} d\mathbf{x}$, where $P_{ij} \triangleq \sum_{l=1}^{n} \left(\frac{\partial^2 y_l}{\partial x_i \partial x_j}\right)^2$. The derivation of $\frac{\partial R_2}{\partial \mathbf{W}}$ is straightforward. In the result, $\frac{\partial^2 z_p}{\partial x_i \partial x_j}$ is involved. It can be computed and saved in the first step of kernel-based nonlinear ICA.

4 Experiments

According to the experimental results in [1] and our experience, mixtures of subgaussian sources are more difficult to be separated well, than those of supergaussian sources. So for saving space, here we just present some experimental results on separating two subgaussian sources. The sources are a sawtooth signal (s_1) and an amplitude-modulated waveform (s_2), with 1000 samples. x_i are generated in the same form as the example in Sec. 4 of [5], i.e. $\mathbf{x} = \mathbf{B}\mathbf{s} + \mathbf{c}s_1 s_2$, but here $\mathbf{c} = (-0.15, 0.5)^T$. The waveforms and scatterplots of s_i and x_i are shown in Fig. 1, from which we can see that the nonlinear effect is significant.

The regularization parameter for enforcing smoothness is $\lambda_2 = 0.2$, and that for enforcing MND, λ_1, decays from 0.3 to 0.01 during the learning process.

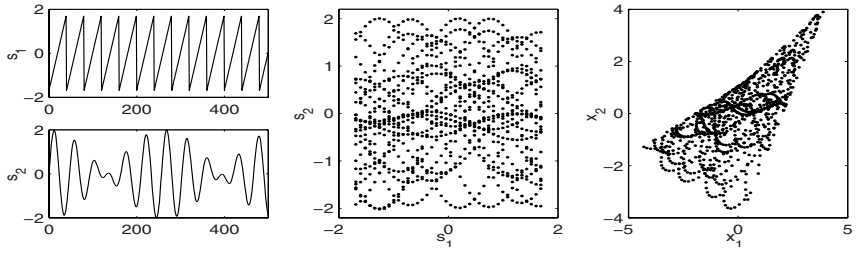
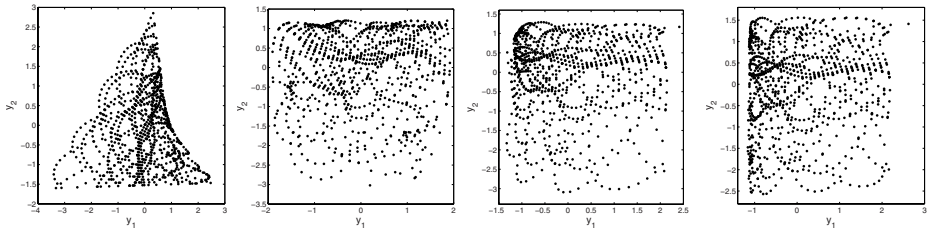


Fig. 1. Source and their nonlinear mixtures. Left: waveforms of sources. Middle: scatterplot of sources. Right: scatterplot of mixtures.

We chose the polynomial kernel of degree 4, i.e. $k(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^T\mathbf{b} + 1)^4$, and found $d = 14$. Here we compare the separation results of four methods/schemes, which are linear ICA (FastICA is adopted), kernel-based nonlinear (kNICA) without explicit regularization, kNICA with only the smoothness regularization, and kNICA with both smoothness and MND regularization. Table 1 shows the SNR of the recovered signals. Numbers in parentheses are the SNR values after trivial indeterminacies are removed.[4] Fig. 2 shows the scatterplots of y_i obtained by various schemes. In this experiment, clearly kNICA with the smoothness and MND regularization gives the best separation result.

Table 1. SNR of the separation results on various methods (schemes)

Channel	FastICA	kNICA (no regu.)	kNICA (smooth)	kNICA (smooth & MND)
No. 1	3.72 (4.59)	9.25(9.69)	11.1(14.4)	**12.1 (16.5)**
No. 2	5.76 (6.04)	6.07(8.19)	8.9(12.7)	**15.4 (25.1)**



(a) (b) (c) (d)

Fig. 2. Scatterplot of y_i obtained by various methods/schemes. (a) FastICA. (b) kNICA without explicit regularization. (c) kNICA with the smoothness regularizer. (d) kNICA with the smoothness and MND regularization.

[4] We applied a 1-8-1 MLP, denoted by $\mathcal{T}$, to y_i to minimize the square error between s_i and $\mathcal{T}(y_i)$. In this way trivial indeterminacies are removed.

5 Conclusion

We have proposed to solve the nonlinear ICA problem using kernels. In the first step of the method, the data are mapped to high-dimensional feature space and the effective subspace is extracted. Thanks to the kernel trick, in the second step, we need to solve a linear problem. The algorithm in the second step was derived, in a form similar to standard ICA. In order to achieve nonlinear BSS, we incorporated the minimal nonlinear distortion principle and the smoothness regularizer for regularization of the proposed nonlinear ICA method. MND helps to overcome the ill-posedness of nonlinear ICA, under the condition that the nonlinearity in the mixing procedure is not very strong. This condition usually holds for practical problems.

References

1. Almeida, L.B.: MISEP - linear and nonlinear ICA based on mutual information. Journal of Machine Learning Research 4, 1297–1318 (2003)
2. Amari, S., Cichocki, A., Yang, H.H.: A new learning algorithm for blind signal separation. In: Advances in Neural Information Processing Systems (1996)
3. Bach, F.R., Jordan, M.I.: Beyond independent components: trees and clusters. Journal of Machine Learning Research 4, 1205–1233 (2003)
4. Cardoso, J.F., Souloumiac, A.: Blind beamforming for non-Gaussian signals. IEE Proceeding-F 140(6), 362–370 (1993)
5. Harmeling, S., Ziehe, A., Kawanabe, M., Müller, K.R.: Kernel-based nonlinear blind source separation. Neural Computation 15, 1089–1124 (2003)
6. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. IEEE Transactions on Neural Networks 10(3), 626–634 (1999)
7. Hyvärinen, A.: The fixed-point algorithm and maximum likelihood estimation for independent component analysis. Neural Processing Letters 10(1), 1–5 (1999)
8. Hyvärinen, A., Pajunen, P.: Nonlinear independent component analysis: Existence and uniqueness results. Neural Networks 12(3), 429–439 (1999)
9. Jutten, C., Karhunen, J.: Advances in nonlinear blind source separation. In: Proc. ICA2003, pp. 245–256 (2003) Invited paper in the special session on nonlinear ICA and BSS
10. Martinez, D., Bray, A.: Nonlinear blind source separation using kernels. IEEE Transaction on Neural Network 14(1), 228–235 (2003)
11. Schölkopf, B., Smola, A., Muller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299–1319 (1998)
12. Zhang, K., Chan, L.: Nonlinear independent component analysis with minimum nonlinear distortion. In: ICML 2007, Corvallis, OR, US, pp. 1127–1134 (2007)
13. Ziehe, A., Müller, K.R.: TDSEP − an efficient algorithm for blind separation using time structure. In: Proc. ICANN98, Skövde, Sweden, pp. 675–680 (1998)