Fuzzy skeletons and statistical learning theory

Pando Georgiev
Computer Science Department
University of Cincinnati
Cincinnati, OH 45221
e-mail: pgeorgie@ececs.uc.edu

Anca Ralescu
Computer Science Department
University of Cincinnati
Cincinnati, OH 45221
e-mail: aralescu@ececs.uc.edu

Abstract

We define a fuzzy subspace skeleton of data points and propose an algorithm for finding it. Such a skeleton has direct applications in statistical learning theory. We propose a new type of classifiers: fuzzy skeleton classifiers, which might be a better alternative to Support Vector Machines in some cases. Another application is presented to the unsupervised learning - Blind Signal Separation, based on mild sparsity assumptions. Our methods are illustrated by examples. Potential application include problems from bioinformatics as separation of protein spectra, gene expressions, etc., as well as any problems requiring signal separation in which Independent Component Analysis doesn't work, or gives unsatisfactory results.

1 Introduction

The notion of skeleton of data arises in a natural way when we want to approximate a data set by a simple one which could be described easily. The simplest example of skeleton is presented by Principle Component Analysis, which finds a subspace with appropriate dimension, which fits the data in the best way. In this paper we extend this idea to the case when the data vectors are approximated by elements in a union of finitely many affine subspaces, called *skeleton* of the given data set. More precisely, we call it *fuzzy skeleton* because of the specific algorithm which we use - fuzzy subspace clustering on subspaces. In such a way we extend the idea from [2], where the skeleton is a union of hyperplanes. Something more, we extend this idea to *nonlinear fuzzy skeletons* when we work in Reproducing Kernel Hilbert Spaces. The idea of fuzzy skeletons gives a new type of classifiers - we called them fuzzy skeleton classifiers.

Another application of the fuzzy skeleton gives a new approach to the Blind Signal Separation. The goal of the Blind Signal Separation (BSS) is to recover the underlying source signals of some given set of observations $\mathbf{X}$ obtained by a linear mixture of the sources:

$$\mathbf{X} = \mathbf{A}\mathbf{S}, \tag{1}$$

where the matrices $\mathbf{A}$ and $\mathbf{S}$ with dimensions $m \times n$ and $n \times N$ respectively (often called mixing matrix or *dictionary* and source matrix) are unknown ($m \leqslant n < N$).

BSS has potential applications in many different fields such as medical and biological data analysis, communications, audio and image processing, etc. In order to decompose the data set, different assumptions on the sources have to be made. The most common assumption nowadays is statistical independence of the sources, which leads to the field of *Independent Component Analysis* (ICA), see for instance [5], [11] and references therein. ICA is very successful in the linear *complete* case, when as many signals as underlying sources are observed, and the mixing matrix is non-singular. In [7] it is shown that the mixing matrix and the sources are identifiable except for permutation and scaling. In the *overcomplete* or *underdetermined* case, less observations than sources are given. It can be seen that

L. Magdalena, M. Ojeda-Aciego, J.L. Verdegay (eds): Proceedings of IPMU'08, pp. 818–825
Torremolinos (Málaga), June 22–27, 2008

still the mixing matrix can be recovered [8], but source identifiability does not hold. In order to approximatively detect the sources, additional requirements have to be made, usually sparsity of the sources. We refer to [12, 17, 18, 19] and reference therein for some recent papers on sparsity and underdetermined ICA ($m < n$).

In [9] only the case $r = 1$ was considered (each column of the source matrix $\mathbf{S}$ contains at most $m-1$ non-zero elements). In this paper we consider the case $r \geqslant 2$ (each column of $\mathbf{S}$ contains at most $m-r$ non-zero elements) and develop a fuzzy algorithm for clustering over subspaces, which is essential for identification of the mixing matrix A.

2 Skeletons of data sets

The solution $\{(\mathbf{n}_i^0, b_i^0)\}_{i=1}^k$ of the minimization problem:

$$\begin{aligned} &\text{minimize} \sum_{j=1}^{N} \min_{1\leqslant i\leqslant k} |\mathbf{n}_i^T\mathbf{x}_j - b_i|^l \\ &\text{subject to} \quad \|\mathbf{n}_i\| = 1, b_i \in \mathbb{R}, i = 1, ..., k, \end{aligned}$$

defines $k^{(l)}$-*skeleton* of $\mathbf{X}$, introduced for $l = 1$ in [15], and for $l = 2$ in [2]. It consists of a union of k hyper-planes

$$H_i = \{x \in \mathbb{R}^m : \mathbf{n}_i^T\mathbf{x} = b_i\}, i = 1, ..., k, \quad (2)$$

such that the sum of minimum distances raised to power l, from every point $\mathbf{x}_j$ to them is minimal.

The notion "subspace skeleton" [10] is defined by the solution of the following minimization problem

$$\begin{aligned} &\text{minimize} \sum_{j=1}^{N} \min_{1\leqslant i\leqslant k} \sum_{s=1}^{r_i} |\mathbf{n}_{i,s}^T\mathbf{x}_j - b_{i,s}|^l \\ &\text{subject to} \ \|\mathbf{n}_{i,s}\| = 1, i = 1, ..., k, s = 1, ..., r_i, \\ &\qquad \mathbf{n}_{i,p}^T\mathbf{n}_{i,q} = 0, p \neq q, b_{i,s} \in \mathbb{R}. \end{aligned}$$

It consists of a union of k affine subspaces $H_i = \{x \in \mathbb{R}^m : \mathbf{n}_{i,s}^T\mathbf{x} = b_{i,s}, s = 1, ..., r_i\}, i = 1, ..., k$, with codimension r_i such that the sum of minimum distances raised to power l, from every point $\mathbf{x}_j$ to them is minimal.

The solution $\mathbf{V} = \{\mathbf{v}_{ij}\}$ of the following minimization problem:

$$\begin{aligned} &\text{minimize} \\ &f(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{k}\sum_{j=1}^{N}\sum_{s=1}^{r_i} u_{ij}^p (\mathbf{v}_{i,s}^T\mathbf{x}_j)^2 \\ &\text{subject to} \\ &\qquad \|\mathbf{v}_{i,s}\| = 1, i = 1, ..., k, s = 1, ..., r_i, \\ &\qquad \mathbf{v}_{i,p}^T\mathbf{v}_{i,q} = 0, p \neq q, \\ &\qquad \sum_{i=1}^{k} u_{ij} = 1, j = 1, ..., N \\ &\qquad u_{ij} \geqslant 0, i = 1, ..., k, j = 1, ..., N \end{aligned}$$

defines a *fuzzy subspace skeleton* of the data points $\mathbf{X}$. It consists of a union of k subspaces $H_i = \{x \in \mathbb{R}^m : \mathbf{n}_{i,s}^T\mathbf{x} = 0, s = 1, ..., r_i\}, i = 1, ..., k$, with codimension r_i. It is clear that affine subspaces in $\mathbb{R}^m$ (like (2)) can be found by working in $\mathbb{R}^{m+1}$ and finding the usual subspaces by considering the data points $(\mathbf{x}_j, 1), j = 1, ..., N$; then the vectors $\mathbf{n}_i$ are replaced by $(\mathbf{n}_i, -b_i)$.

Determining of $\{r_i\}$ can be performed as in PCA – by the number of the significant eigenvalues of corresponding matrices.

The both skeletons (subspace skeleton and fuzzy subspace skeleton) coincide, if the data points belong to the union of small number of subspaces. This fact is used below in the BSS examples.

It is clear that if the matrix $\mathbf{S}$ in (ref1) is sparse in sense that each column of $\mathbf{S}$ has at most $m - r_i$ nonzero elements for some $i \in \{1, ..., k\}$, then the data vectors (columns of $\mathbf{X}$) lie on an union of $\binom{n}{r_i}$ r_i-codimensional subspaces - this is the main idea from [9], for the case $r_i = 1$. So, finding any of these skeletons will alow us to identify these subspaces and subsequently, the mixing matrix.

3 Fuzzy subspace clustering algorithm

In order to find the fuzzy subspace skeleton of the data set $\mathbf{X}$, we will apply iteratively the following two steps, like in the classical fuzzy c-means clustering algorithm [3], [4]:

Step 1): For given $\mathbf{U} = \{u_{ij}\}_{i=1,j=1}^{k,N}$, minimize $f(\mathbf{U}, .)$. This problem can be converted to

k eigen-value problems as follows. Denote by $\mathbf{Y}^{(i)}$ the matrix with elements $\mathbf{Y}^{(i)}_{rj} = u_{ij}^{p/2} x_{rj}$. Then the minimum of $f(\mathbf{U}, .)$ is attained at the union of the first r_i eigen-vectors of the matrices $\mathbf{Y}^{(i)}(\mathbf{Y}^{(i)})^T, i = 1, ..., k$. Justification of this assertion is as follows: the function $f(\mathbf{U}, \mathbf{V})$ transforms in

$$f(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{k} \mathrm{trace} \mathbf{V}_i^T \mathbf{Y}^{(i)} (\mathbf{Y}^{(i)})^T \mathbf{V}_i, \quad (3)$$

where $\mathbf{V}_i, i = 1, ..., k$ is the matrix with columns $\mathbf{v}_{i,s}, s = 1, ..., r_i$. We minimize (1) under orthogonality constrains $\mathbf{V}_i^T \mathbf{V}_i = \mathbf{I}_{r_i}$ ($\mathbf{I}_{r_i}$ is the $r_i \times r_i$ identity matrix). For such problems we apply Theorem 11.12.13 of [13] (which is a consequence Theorem 11.11.10 in [13] (the Poincare separation theorem)) stating that the minimum of $trace \mathbf{V}_i^T \mathbf{Y}^{(i)} (\mathbf{Y}^{(i)})^T \mathbf{V}_i$ under orthogonality constraints is achieved when the columns of $\mathbf{V}_i$ consist of the first r_i eigenvectors corresponding to the first r_i minimal eigenvalues of $\mathbf{Y}^{(i)}$.

Step 2): For given $\mathbf{V} = \{\mathbf{v}_{ij}\}$, minimize $f(., \mathbf{V})$. Applying Kuhn-Taker optimality conditions, we find that $\mathbf{U}$ must satisfy the following conditions (similar to those in the classical fuzzy clustering algorithm):

$$u_{ij} = \frac{d_{ij}^{\frac{1}{1-p}}}{\sum_{r=1}^{k} d_{rj}^{\frac{1}{1-p}}}, \quad (4)$$

where $d_{ij} = \sum_{s=1}^{r_i} (\mathbf{v}_{i,s}^T \mathbf{x}_j)^2$.

We will not consider the convergence properties of this fuzzy subspace clustering algorithm. For the classical fuzzy c-means clustering algorithm such properties are considered in [3], [4], for instance. In the examples considered below, the convergence occurs quite fast, after few random initializations.

4 Reproducing Kernel Hilbert Spaces

Recall one of the possible constructions of Reproducing Kernel Hilbert Spaces (RKHS) (see [1], [6] for more details). Let $\mathbf{X} \subset \mathbb{R}^m$ and $K : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$ be a continuous and symmetric function such that $\{K(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j=1}^n$ is positive definite for every $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbf{X}$ and every n, i.e. $\sum_{i,j=1}^n c_i c_j K(x_i, x_j) \geqslant 0$ for any $n \in N$ and any choice of $x_i \in X$ and $c_i \in \mathbb{R}$ $(i = 1, ..., n)$. Note $K(x, x) \geqslant 0$ for all x. The mapping $\mathbf{K}$ is called *positive definite kernel*.

Let H_0 be the linear span of the functions $\{K(\mathbf{x}, .), \mathbf{x} \in \mathbf{X}\}$:

$$H_0 = \Big\{ f : \mathbf{X} \to \mathbb{R} : f(.) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, .), \; \mathbf{x}_i \in \mathbf{X}, \alpha_i \in \mathbb{R}, n \in \mathbb{N} \Big\}. \quad (5)$$

Define an inner product in H_0 by

$$\langle f, g \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n'} \alpha_i \beta_j K(\mathbf{x}_i, \mathbf{x}'_j) \quad (6)$$

where $f, g \in H_0$, $f(.) = \sum_{i=1}^n \alpha_i K(\mathbf{x}_i, .), g(.) = \sum_{i=1}^{n'} \beta_i K(\mathbf{x}'_i, .)$.

The *reproducing property*

$$f(\mathbf{x}) = \langle f, K(\mathbf{x}, .) \rangle \quad \forall \mathbf{x} \in \mathbf{X}$$

follows from (6).

The induced norm $\|.\|_K$ in H_0 is $\|f\|_K = \sqrt{\langle f, f \rangle}$. It is indeed a norm, since if $\|f\| = 0$, then

$$f(\mathbf{x}) = \langle f, K(\mathbf{x}, .) \rangle \leqslant \|f\| \|K(\mathbf{x}, .)\| = 0 \quad \forall \mathbf{x}.$$

The completion of $(H_0, \|.\|_K)$ is called *Reproducing Kernel Hilbert Space*.

The so called *kernel trick*

$$K(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle \text{ for every } \mathbf{x}, \mathbf{y} \in \mathbf{X}$$

where $\Phi(\mathbf{x}) = K(\mathbf{x}, .)$ is called *feature map*, follows again from (6).

We give two typical examples of positive definite kernels. The first is the Gaussian kernel $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ defined by $K(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2 / c^2)$ $(c > 0)$.

The second is the polynomial kernel of degree p, $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ defined by $K_p(\mathbf{x}, \mathbf{x}') = (\langle \mathbf{x}, \mathbf{x}' \rangle + 1)^p$. In the case when for $m = 2, p = 3$, we have (for $\mathbf{x} = (x, y), \mathbf{x}' = (x', y')$),

$$\Phi(\mathbf{x}) = \Big(x^3, y^3, \sqrt{3}x^2 y, \sqrt{3}xy^2, \sqrt{3}x^2, \sqrt{3}y^2, \sqrt{3}xy, \sqrt{3}x, \sqrt{3}y, 1 \Big) \quad (7)$$

$$\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle = K\Big((x, y), (x', y') \Big).$$

5 Nonlinear fuzzy skeletons

The notion of fuzzy skeleton developed in previous sections can be extended easily to the notion "nonlinear fuzzy skeleton" as follows: in previous definitions we change $\mathbf{x}$ with $\Phi(\mathbf{x})$ and work in a Reproducing Kernel Hilbert space H defined by a kernel K such that $\Phi(\mathbf{x}) = K(\mathbf{x}, .)$. For instance, the notion of hyperplane skeleton of a data set $\{\mathbf{x}_j\}_{j=1}^N$ is extended to nonlinear hyperplane skeleton by the following definition:

$$NS(\mathbf{X}) = \Big\{\mathbf{x} \in \mathbb{R}^m : \langle \Phi(\mathbf{x}), \mathbf{h}_i \rangle = 0 \text{ for some } i \in \{1, ..., k\}\Big\}, \quad (8)$$

where $\{\mathbf{h}_i\}_{i=1}^k \subset H$ the is solution of the following minimization problem:

$$\text{minimize} \sum_{j=1}^{N} \min_{1 \leqslant i \leqslant k} \langle \Phi(\mathbf{x}_j), \mathbf{h}_i \rangle^2 \quad (9)$$

$$\text{subject to} \quad \|\mathbf{h}_i\| = 1, i = 1, ..., k. \quad (10)$$

Clustering on subspaces in RKHS. We have two algorithms for clustering on subspaces in RKHS:

– a *Primal Kernel Subspace Fuzzy Clustering Algorithm*, when the feature map Φ is known and the feature space (RKHS) is finite dimensional; The algorithm is performed with changing $\mathbf{x}$ with $\Phi(\mathbf{x})$.

– a *Dual Kernel Fuzzy Subspace Clustering Algorithm* - when the kernel K is known only. It is similar to the Kernel PCA [16] (not considered here).

The primal algorithm terminates in finitely many steps to a solution which is locally optimal, i.e. it finds k clusters $\mathbf{X}_i, i = 1, ..., k$ such that $\min_{1 \leqslant l \leqslant k} \langle \Phi(\mathbf{x}_j), \mathbf{h}_l \rangle = \langle \Phi(\mathbf{x}_j), \mathbf{h}_i \rangle$ if $\mathbf{x}_j \in \mathbf{X}_i$, where $\{\mathbf{h}_i\}_{i=1}^k$ is a local solution of (9), (10). For small values of k (less than 7), this algorithm terminates usually in clusters which are globally optimal. A big challenge is to design a global optimization algorithm for subspace (and kernel subspace) clustering.

Examples. Fig. 1 shows artificially created data which are fitted by two ellipses. We used the primal kernel subspace clustering algorithm with polynomial kernel of degree 2. Fig. 2 shows another set of artificially created data which are fitted by three level sets of polynomials of degree three. We used again the primal kernel subspace clustering algorithm with polynomial kernel of degree 3.

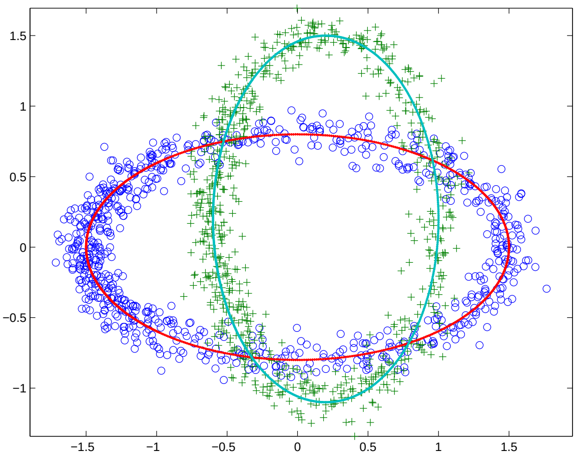

Figure 1: Nonlinear skeleton of degree 2

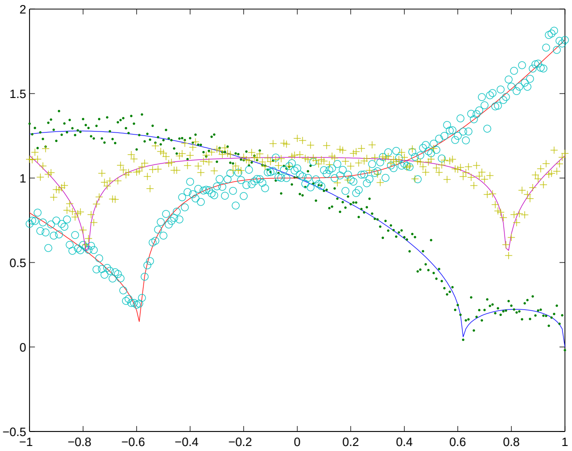

Figure 2: Nonlinear skeleton of degree 3

6 Skeleton classifiers.

Here we present the idea that binary classification tasks can be performed by finding nonlinear skeletons of the training points in the both classes. Fig. 3 shows a possible separation of the two classes belonging to two lines (forming a cross) by standard support vector machine (SVM) classifier. It is clear that this classifier cannot classify correctly new points belonging to these lines and converging to their intersecting point (not shown).

Now we propose that the skeletons of the two classes can serve as a classifier (called skeleton classifier) in sense that a new point belongs to a class A if it is at a nearest distance to the skeleton of the class A. Then it is clear that the new points from the above example that cannot be classified correctly by SVM (converging to the intersection of the two skeletons, which are two lines) can be classified perfectly by the proposed skeleton classifier. In this example the skeleton is affine hyperplane skeleton (union of hyperplanes). We have to note that a similar (but more complicated) idea was proposed in [14].

Fig. 4 shows a nonlinear perturbation of the previous data set (nonlinear cross, two level sets of polynomials of degree 2). After mapping of the points in both classes by the feature map $\Phi : (x, y) \mapsto (x^2, y^2, \sqrt{2}xy, \sqrt{2}x, \sqrt{2}y, 1)$, the data set belong to two hyperplanes in $\mathbb{R}^6$ (which are also the image of the skeletons of the two classes after applying the feature map Φ). The same reasoning like in previous example shoes that next points from both classes (belonging near to the corresponding skeletons and converging to the intersection of nonlinear skeletons) cannot be classified by SVM classifier, while can be classified perfectly by the proposed skeleton classifier.

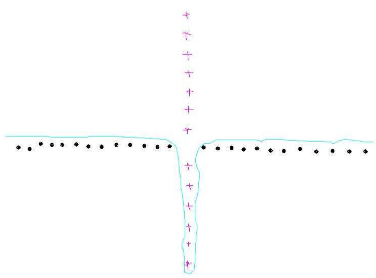

Figure 3: Binary classification by classical SVM classifier. New data points from both classes, converging to the intersection of their skeletons (which in this case are lines) cannot be classified correctly, while the proposed skeleton classifier classifies them perfectly.

Several issues concerning skeleton classifiers remain to be investigated in future research: statistical properties, stability, applications to real data, etc.

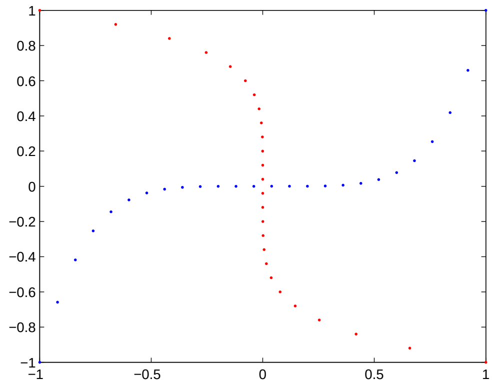

Figure 4: Nonlinear cross. The skeletons of the two classes are level sets of polynomials of two variables of degree 2. They are hyperplanes in $\mathbb{R}^6$ after applying the feature map $\Phi : (x, y) \mapsto (x^2, y^2, \sqrt{2}xy, \sqrt{2}x, \sqrt{2}y, 1)$. The same reasoning like in previous example shows the advantage of the proposed skeleton classifier.

7 Applications to Blind Signal Separation

Estimating the mixing matrix - extension of the algorithm in [9]

1) Cluster the columns $\{\mathbf{X}(:, j) : j \in \mathcal{N}_1\}$ in k groups $\mathcal{H}_i, i = 1, ..., k$ such that the span of the elements of each group $\mathcal{H}_i$ produces r_i-codimensional subspace and these r_i-codimensional subspaces are different.

2) Calculate any basis of the orthogonal complement of each of these r_i-codimensional subspaces.

3) Find all possible groups such that each of them is composed of the elements of at least m bases in 2), and the vectors in each group lie on a hyperplane. The number of these hyperplanes gives the number of sources n. The normal vectors to these hyperplanes are estimations of the columns of the mixing matrix $\mathbf{A}$ (up to permutation and scaling).

In practical realization all operations in the above algorithm are performed up to some precision $\varepsilon > 0$.

The estimation of the mixing matrix by the above algorithm is unique up to permutation and scaling.

Sufficient condition for this uniqueness is the assumption that each column of $\mathbf{S}$ contains at most $m - r_i$ non-zero elements for some i, plus some algebraic conditions on $\mathbf{S}$ assuring that there are enough samples (from the columns of $\mathbf{X}$) in each possible subspaces formed by all subsets of the columns of $\mathbf{A}$ containing $n - m + r_i$ elements.

8 Identification of sources

The sources are uniquely identifiable generically, i.e. up to a set with a measure zero, if they compose a matrix which is sparse in the above mentioned sense, and the mixing matrix is known. This statement, as well as the algorithm below for such identification, can be justified similarly to those in [9] for the case $r_i = 1$.

Source recovery algorithm

1. Repeat for $j = 1$ to N:

2.1. Identify the subspace $\mathcal{H}_i$ containing $\mathbf{x}_j := \mathbf{X}(:, j)$, or, in practical situation with presence of noise, identify $\mathcal{H}_i$ to which the distance from $\mathbf{x}_j$ is minimal and project $\mathbf{x}_j$ onto $\mathcal{H}_i$ to $\tilde{\mathbf{x}}_j$;

2.2. if $\mathcal{H}_i$ is produced by the linear hull of column vectors $\mathbf{a}_{p_1}, ..., \mathbf{a}_{p_{m-r_i}}$, then find coefficients $\mathbf{L}_{j,l}$ such that $\tilde{\mathbf{x}}_j = \sum_{l=1}^{m-r_i} \mathbf{L}_{j,l}\mathbf{a}_{p_l}$. These coefficients are uniquely determined generically (i.e. up to measure zero) for $\tilde{\mathbf{x}}_j$ (see [9], Theorem 3).

2.3. Construct the solution $\mathbf{s}_j = \mathbf{S}(:, j)$: it contains $\mathbf{L}_{j,l}$ in the place p_l for $l = 1, ..., m - r_i$, the other its components are zero.

9 Computer simulation example

Example 1. We created artificially four source signals, sparse of level 2, i.e. each column of the source matrix contains at least 2 zeros (shown in Figure 5). They are mixed with a square normalized matrix $\mathbf{H}1$ (each column of it has norm one):

$$\mathbf{H}1 = \begin{pmatrix} -0.5701 & -0.4607 & .1841 & -0.9526 \\ -0.5198 & -0.6480 & -0.7710 & 0.1488 \\ -0.3628 & 0.4671 & -0.1173 & -0.2526 \\ 0.5225 & -0.3869 & 0.5983 & -0.0807 \end{pmatrix}.$$

The mixed signals are shown in Figure 6. We apply our fuzzy subspace clustering algorithm in order to identify the 2-fuzzy 2-subspace skeleton of the data points, and after that apply our matrix identification algorithm. We obtain an estimation $\mathbf{W}1$ of the mixing matrix (after normalization of each column):

$$\mathbf{W}1 = \begin{pmatrix} 0.5702 & 0.4605 & 0.9527 & 0.1838 \\ 0.5198 & 0.6479 & -0.1487 & -0.7709 \\ 0.3629 & -0.4672 & 0.2525 & -0.1173 \\ -0.5225 & 0.3871 & 0.0805 & 0.5984 \end{pmatrix}.$$

which is very near to $\mathbf{H}1$ (up to permutation and sign).

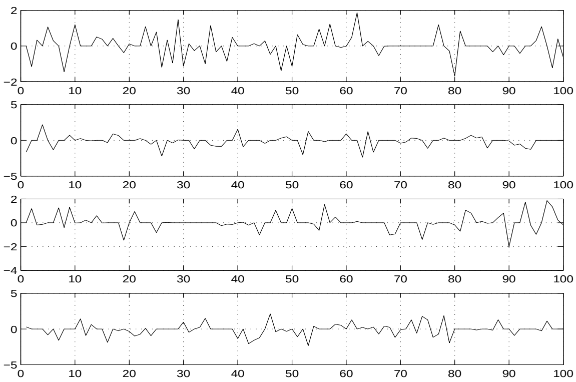

Figure 5: Example 1: original source signals.

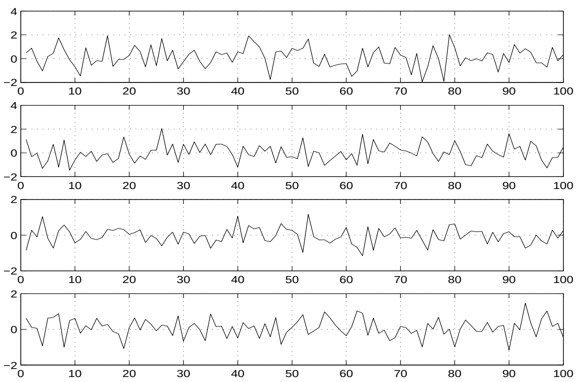

Figure 6: Example 1: mixed signals.

Example 2. The next example shows four dependent source signals, sparse of level 2 (shown in Figure 7). They are mixed with a square normalized matrix $\mathbf{H}2$ (each column of it has norm one):

$$\mathbf{H}2 = \begin{pmatrix} 0.8650 & -0.4616 & 0.3813 & 0.3467 \\ 0.1296 & 0.0685 & -0.0587 & 0.2996 \\ 0.1697 & 0.0214 & -0.5688 & 0.1711 \\ 0.4541 & -0.8842 & 0.7264 & -0.8722 \end{pmatrix}.$$

The mixed signals are shown in Figure 8. We identify the 2-fuzzy 2-subspace skeleton of the data points by our fuzzy subspace clustering algorithm and after that apply the matrix identification algorithm. The estimated mixing matrix (after normalization of each column) is:

$$\mathbf{W}2 = \begin{pmatrix} 0.3813 & 0.4616 & 0.8650 & 0.3463 \\ -0.0587 & -0.0685 & 0.1296 & 0.2995 \\ -0.5687 & -0.0216 & 0.1697 & 0.1713 \\ 0.7264 & 0.8842 & 0.4541 & -0.8724 \end{pmatrix}.$$

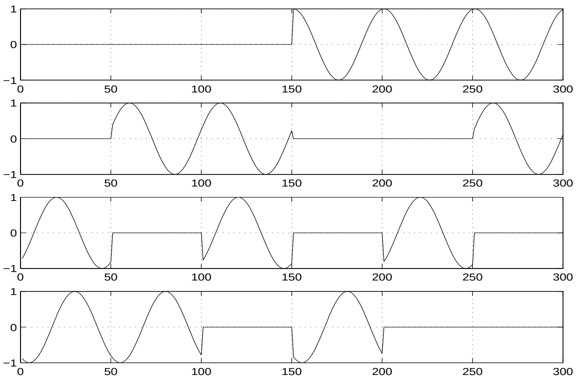

Figure 7: Example 2: original source signals.

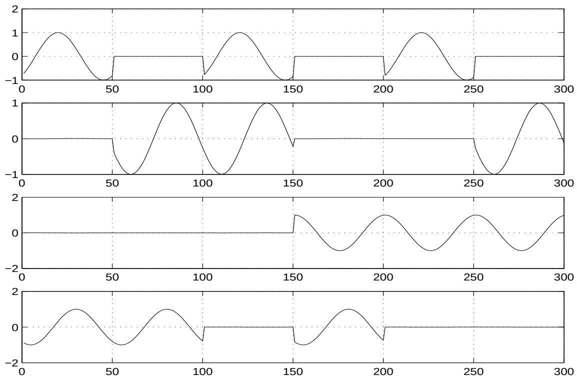

Figure 9: Example 2: separated signals.

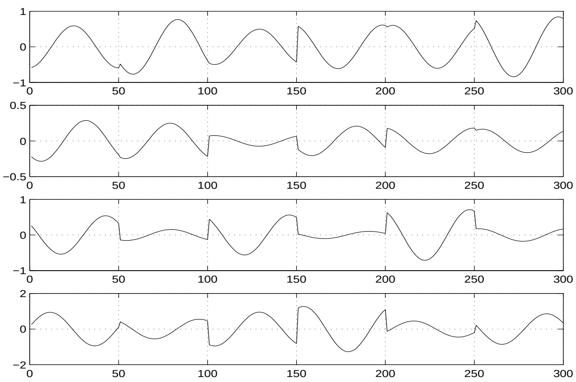

Figure 8: Example 2: mixed signals.

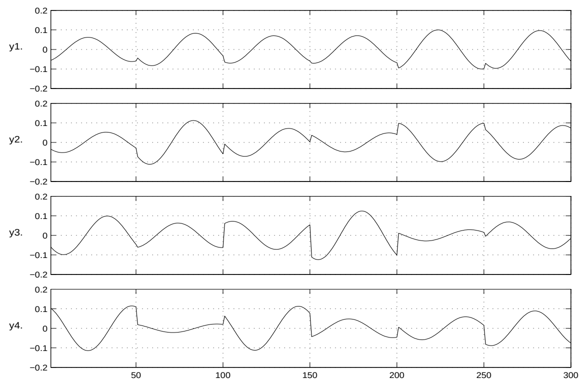

Figure 10: Example 2: results obtained by applying SOBI.

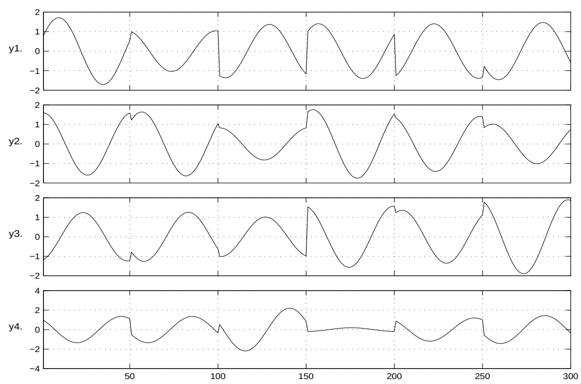

Figure 11: Example 2: results obtained by applying FAST ICA with hyperbolic tangent as activation function.

The separated sources are shown in Fig. 5.

In the next figures we show the results of applying on the last example the SOBI algorithm (Figure 10) and Fast ICA algorithm (Figure 11).

10 Conclusion

We develop the idea of nonlinear fuzzy skeleton and present a fuzzy clustering algorithms for finding such a skeleton. We present applications in statistical learning theory. First application is a new classifier - a fuzzy skeleton classifier, which classifies the points in accordance with their nearness to the fuzzy skeleton of a corresponding class. Presented examples suggest that the proposed fuzzy skeleton classifier could be better compared with the classical SVM classifier. The second application is to the unsupervised learning. Finding the affine skeletons of data sets has a direct application to Blind Signal Separation problems for de-mixing of unknown mixture of source signals under mild sparsity assumptions. We present identifiability conditions for sparse BSS problems, allowing less hyperplanes in the data points for full recovery of the original sources and the mixing matrix. The ideas are illustrated with examples.

References

[1] N. Aronszajn, "Theory of reproducing kernels", Transactions of the Amer. Math. Soc.68 (1950), 337–404.

[2] P.S. Bradley and O. L. Mangasarian, "k-Plane Clustering", *J. Global optim.*, 16, (2000), no.1, 23-32.

[3] J.C. Bezdek, "A convergence theorem for the fuzzy ISODATA clustering algorithm", Fuzzy models for pattern recognition: methods that search for structure in data, J. Bezdek, S. Pal (Eds.), IEEE Press, 1992.

[4] J.C. Bezdek, R.J. Hathaway, M.J. Sabin and W.T. Tucker, "Convergence theory for fuzzy c-means: counterexamples and repairs", Fuzzy models for pattern recognition: methods that search for structure in data, J. Bezdek, S. Pal (Eds.), IEEE Press, 1992.

[5] A. Cichocki and S. Amari. *Adaptive Blind Signal and Image Processing*. John Wiley, Chichester, 2002.

[6] F. Cucker, S. Smale, "On the mathematical foundation of learning", Bulletin (New Series) of the Americal Mathematical Society Vol. 39, Number 1, pp 1–49.

[7] P. Comon. *Independent component analysis - a new concept?* Signal Processing, 36: 287314, 1994.

[8] J. Eriksson and V. Koivunen. *Identifiability and separability of linear ica models revisited.* In Proc. of ICA 2003, pages 2327, 2003.

[9] P. Georgiev, F. Theis and A. Cichocki, "Sparse Component Analysis and Blind Source Separation of Underdetermined Mixtures" *IEEE Transactions of Neural Networks*, Vol. 16, No. 4, July 2005, 992 – 996.

[10] Pando Georgiev, Anca Ralescu and Dan Ralescu, *Fuzzy Subspace Clustering Algorithm and Applications to Blind Signal Separation*, in 2006 IEEE World Congress on Computational Intelligence, Proc. FUZZ-IEEE, Vancouver, July, 16-21, 2006.

[11] A. Hyvärinen, J. Karhunen and E. Oja, *Independent Component Analysis*, John Wiley & Sons, 2001.

[12] T.-W. Lee, M.S. Lewicki, M. Girolami, T.J. Sejnowski, "Blind sourse separation of more sources than mixtures using overcomplete representaitons", *IEEE Signal Process. Lett.*, Vol. 6, no. 4, pp. 87–90, 1999.

[13] Jan R. Magnus, Heinz Neudecker, "Matrix Differential Calculus with Applications in Statistics and Econometrics", Second Edition, J. Wiley & Sons, 1999.

[14] Olvi L. Mangasarian and Edward W. Wild, "Multisurface Proximal Support Vector Machine Classification via Generalized Eigenvalues", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, NO. 12, December 2005, 1–6.

[15] A. M. Rubinov and J. Ugon, "Skeletons of finite sets of points", Research working paper 03/06, 2003, School of Information Technology & Mathematical Sciences, Univ. of Ballarat.

[16] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 10:12991319, 1998.

[17] F.J. Theis, E.W. Lang, and C.G. Puntonet, A geometric algorithm for overcomplete linear ICA. *Neurocomputing*, in print, 2003.

[18] K. Waheed, F. Salem, "Algebraic Overcomplete Independent Component Analysis", in *Proc. Int. Conf. ICA2003*, Nara, Japan, pp. 1077–1082.

[19] M. Zibulevsky, and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary", *Neural Comput.*, Vol. 13, no. 4, pp. 863–882, 2001.