Local Linear Independent Component Analysis Based on Clustering

Juha Karhunen, Simona Mălăroiu, and Mika Ilmoniemi

Helsinki University of Technology, Neural Networks Research Centre
P.O.Box 5400, FIN-02015 HUT, Espoo, FINLAND
email: `Juha.Karhunen@hut.fi, Simona.Malaroiu@hut.fi, milmonie@cc.hut.fi`
URL: `http://www.cis.hut.fi/projects/ica`
Fax: +358-9-451 3277

November 1, 2000

Abstract

In standard Independent Component Analysis (ICA), a linear data model is used for a global description of the data. Even though linear ICA yields meaningful results in many cases, it can provide a crude approximation only for general nonlinear data distributions. In this paper a new structure is proposed, where local ICA models are used in connection with a suitable grouping algorithm clustering the data. The clustering part is responsible for an overall coarse nonlinear representation of the data, while linear ICA models of each cluster are used for describing local features of the data. The goal is to represent the data better than in linear ICA while avoiding computational difficulties related with nonlinear ICA.

Several data grouping methods are considered, including standard K-means clustering, self-organizing maps, and neural gas. Connections to existing methods are discussed, and experimental results are given for artificial data and natural images. Furthermore, a general theoretical framework encompassing a large number of methods for representing data is introduced. These range from global, dense representation methods to local, very sparse coding methods. The proposed local ICA methods lie between these two extremes.

Keywords: Independent component analysis, local methods, blind source separation, unsupervised learning, neural networks.

1 Introduction

Independent Component Analysis (ICA) [1, 2, 3] is a statistical technique which in many cases characterizes the data in a natural way. ICA and the related blind source separation (BSS) problem [4, 2, 3, 1] are currently studied extensively both in unsupervised neural learning and statistical signal processing.

The basic data model used in ICA is as follows. Denote by $\mathbf{x}(t) = [x_1(t), \ldots, x_n(t)]^T$ the n-dimensional t:th data vector. The vectors $\mathbf{x}(t)$ have a common unknown zero-mean non-Gaussian statistical distribution. In ICA, we try to fit to the data $\mathbf{x}(t)$ the expansion

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) = \sum_{j=1}^{m} s_j(t)\mathbf{a}_j. \tag{1}$$

Here the vector $\mathbf{s}(t) = [s_1(t), \ldots, s_m(t)]^T$ contains the m independent components (or source signals) $s_j(t)$ for the data vector $\mathbf{x}(t)$. $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]$ is a constant full-rank $n \times m$ matrix, called the mixing matrix. The vectors $\mathbf{a}_j$, $j = 1, \ldots, m$, are the basis vectors of ICA; see [1, 2]. In (1), the number of independent components m is usually assumed to be at most equal to the number of mixtures n, and often $m = n$.

The independent components are found by determining an $m \times n$ inverse mapping (separating matrix) $\mathbf{B}$ so that the m-vector

$$\mathbf{y}(t) = \mathbf{B}\mathbf{x}(t) \tag{2}$$

becomes an estimate $\mathbf{y}(t) = \hat{\mathbf{s}}(t)$ of the independent component vector $\mathbf{s}(t)$. To this end, many algorithms have been proposed; see for example [1, 2, 4] for further information and reviews. The estimate of the mixing matrix $\mathbf{A}$ is generally obtained as the pseudoinverse of $\mathbf{B}$.

In ICA, the basis vectors $\mathbf{a}_j$ are generally not mutually orthogonal. This can be compared to standard Principal Component Analysis (PCA), where the form of the expansion (1) is the same but the basis vectors $\mathbf{a}_j$ must be mutually orthogonal and the coefficients $s_i(t)$ have maximal variances. The orthogonality requirement of PCA is often an unnatural technical constraint, whereas the independence assumption in ICA is roughly valid [1]. On the other hand, PCA is easier to compute because it requires second-order statistics only.

In spite of its usefulness, the basic ICA model (1) is often too simple for describing well real-world data. Aiming at a general method for describing the input data rather than BSS applications, we point out two specific weaknesses of the model (1). First, the standard ICA model (1) is *linear*, while generally we need a suitable *nonlinear* model for representing the data adequately. Second, the ICA model (1) tries to describe all the data using the same *global* features (ICA basis vectors) $\mathbf{a}_j$, $j = 1, \ldots, m$. Generally, a "rich" natural data set has often statistically varying characteristics in different parts of its overall distribution. This calls for using *local* features for efficient representation of qualitatively different domains of the data. In this paper we develop a fairly simple data model that tries to overcome these drawbacks, and is computationally not too demanding.

The rest of the paper is organized as follows. In the next section, we briefly consider nonlinear ICA, difficulties associated with it, and its connection to local ICA. Then we introduce a general theoretical framework for representing data in a linear basis, which includes a large number of methods that are at first sight quite different. After this, we present our local ICA method, discuss various choices of parameters and grouping methods, and connections with other related approaches. Finally, we show experimental results both for both artificial and real-world image data, and discuss connections of the proposed approach with related other methods.

2 Nonlinear ICA

A natural extension of the linear ICA (or BSS) model (1) is to assume that the components of the data vectors $\mathbf{x}(t)$ depend nonlinearly on some statistically independent components (source signals) $\mathbf{s}(t)$. Thus the instantaneous mixtures

$$\mathbf{x}(t) = F(\mathbf{s}(t)) \tag{3}$$

of the sources $\mathbf{s}(t) = [s_1(t), \dots, s_m(t)]^T$ are observed, where $F : R^m \rightarrow R^m$ is an unknown nonlinear mixing function. For simplicity, we assume that the number of independent components m is equal to the number of mixtures n. The nonlinear ICA problem now consists of finding an inverse mapping $G : R^m \rightarrow R^m$ which gives estimates of the independent components as

$$\mathbf{y}(t) = G(\mathbf{x}(t)). \tag{4}$$

This important problem has drawn increasingly attention lately, see for example [1, 5, 6, 7, 8, 9, 10, 11, 2] and the references therein. However, nonlinear ICA is often difficult to apply for reasons discussed below.

First, solution of the nonlinear ICA problem is usually highly non-unique [8, 1, 10]. For getting more unique solutions, various constraints have been introduced, but it is not yet clear which constraints are most meaningful in different situations. If some inverse mapping (4) is found such that the components of the vector $\mathbf{y}(t)$ are mutually independent, these components may have nothing to do with the original source signals.

Lin [12] has recently derived some interesting theoretical results on ICA. Let the matrices $\mathbf{H}_s$ and $\mathbf{H}_x$ denote the Hessians of the logarithmic probability densities $\log p_s(\mathbf{s})$ and $\log p_x(\mathbf{x})$ of the source vector $\mathbf{s}$ and data (mixture) vector $\mathbf{x}$, respectively. Then for the basic linear ICA model (1) it holds that

$$\mathbf{H}_s = \mathbf{A}^T \mathbf{H}_x \mathbf{A}. \tag{5}$$

where $\mathbf{A}$ is the mixing matrix. If the components of $\mathbf{s}$ are truly independent, $\mathbf{H}_s$ should be a diagonal matrix. Due to the symmetry of the Hessian matrices $\mathbf{H}_s$ and $\mathbf{H}_x$, (5) imposes $n(n-1)/2$ constraints for the elements of the $n \times n$ matrix $\mathbf{A}$. Thus a constant mixing matrix $\mathbf{A}$ can be solved by estimating $\mathbf{H}_x$ at two different points, and assuming some values for the diagonal elements of $\mathbf{H}_s$ [12].

If the nonlinear mapping (3) is twice differentiable, we can approximate it locally at any point by the linear mixing model (1). There $\mathbf{A}$ is defined by the first order term $\partial F(\mathbf{s})/\partial \mathbf{s}$ of the Taylor series expansion of $F(\mathbf{s})$ at the desired point. But now $\mathbf{A}$ generally changes from point to point, so that the constraint conditions (5) still leave $n(n-1)/2$ degrees of freedom for determining the mixing matrix $\mathbf{A}$ (omitting the diagonal elements).

Another difficulty in the general nonlinear ICA methods proposed thus far is that they tend to be computationally rather demanding. Moreover, the computational load usually increases very rapidly with the dimensionality of the problem, preventing in practice the application of nonlinear ICA methods to high-dimensional data sets.

3 A framework for representing data in a linear basis

Let $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m]$ denote an $n \times m$ basis or weight matrix. Its column vectors $\mathbf{w}_j$ are generally assumed to be some non-parallel n-dimensional vectors. We can now form an approximation $\hat{\mathbf{x}}(t)$ to the data vector $\mathbf{x}(t)$ as follows:

$$\hat{\mathbf{x}}(t) = \sum_{j=1}^{m} g_j(\mathbf{x}(t), \mathbf{W})\mathbf{w}_j = \mathbf{W}\mathbf{g}(\mathbf{x}(t), \mathbf{W}) \tag{6}$$

Here $g_j(\mathbf{x}(t), \mathbf{W})$ is the scalar coefficient corresponding to the basis vector $\mathbf{w}_j$, and

$$\mathbf{g}(\mathbf{x}(t), \mathbf{W}) = [g_1(\mathbf{x}(t), \mathbf{W}), g_2(\mathbf{x}(t), \mathbf{W}), \ldots, g_m(\mathbf{x}(t), \mathbf{W})]^T \tag{7}$$

is the column vector consisting of these m coefficients. Generally, the coefficients g_j of the expansion (6) may depend either linearly or nonlinearly on the basis vectors $\mathbf{w}_1, \ldots, \mathbf{w}_m$ and the data vector $\mathbf{x}(t)$. However, the model (6) always represents the data vector $\mathbf{x}(t)$ in a linear basis. A wide variety of well-known data representation methods can be expressed in the formalism (6), depending on the choice of the basis vectors $\mathbf{w}_1, \ldots, \mathbf{w}_m$ and the coefficients $g_j(\mathbf{x}(t), \mathbf{W})$.

Assume first that the coefficients in (6) are real-valued linear functions of the data vector $\mathbf{x}(t)$ and the weight matrix $\mathbf{W}$. Then it is possible to represent $\mathbf{x}(t)$ in the basis $\mathbf{w}_1, \ldots, \mathbf{w}_m$ without any approximation error, provided that $m = n$ and the basis is linearly independent. Such methods include all orthogonal basis methods, for which $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ and $g_j(\mathbf{x}(t), \mathbf{W}) = \mathbf{w}_j^T\mathbf{x}(t)$. An example of this type of representation is standard principal component analysis (PCA) [14], which is optimal with respect to the mean-square representation error $\mathrm{E}\{\| \mathbf{x}(t) - \hat{\mathbf{x}}(t) \|^2\}$.

If a nonorthogonal basis is used, (6) can be written in the form

$$\hat{\mathbf{x}}(t) = \mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{x}(t) \tag{8}$$

and the coefficient vector of the expansion (6) becomes $\mathbf{g}(\mathbf{x}(t), \mathbf{W}) = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{x}(t)$. Linear ICA is a method which uses a non-orthogonal basis [13, 1]. Irrespective of the orthogonality of the basis, the approximation $\hat{\mathbf{x}}(t)$ is obtained by using less than n basis vectors or coefficients in the expansion (6) when the coefficients g_j are linear real-valued functions of the data vectors $\mathbf{x}(t)$.

On the other hand, if the coefficients $g_j(\mathbf{x}(t), \mathbf{W})$ depend nonlinearly on the data vector $\mathbf{x}(t)$ or are quantized, some approximation error is usually present even though the number m of the terms in the expansion (6) is higher than the dimensionality n of the data and basis vectors ($m \geq n$). A nonlinear extension of standard PCA where $\mathbf{g}(\mathbf{x}(t), \mathbf{W}) = \mathbf{g}(\mathbf{W}^T\mathbf{x}(t))$ [13] belongs to this class. This method is not as efficient as standard PCA in approximating the input vectors $\mathbf{x}(t)$, but it implicitly takes into account higher-order statistics in computations, and can be used for blind separation after prewhitening [13, 3, 1].

Also the well-known self-organizing map (SOM) [14, 15, 16] can be expressed in terms of the model (6). In SOM, only one of the coefficients $g_j(\mathbf{x}(t), \mathbf{W})$ is nonzero and actually equal to unity. This coefficient corresponds to the weight (basis) vector $\mathbf{w}_l$ which is closest to the data vector $\mathbf{x}(t)$ in Euclidean norm. By learning the weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_m$ so that they provide a nearly optimal vector quantization [14], SOM forms a nonlinear data compressing mapping which has proved out to be very useful in visualizing and characterizing the essential features of the underlying data.

In the general framework (6), PCA and ICA are qualitatively quite different from SOM. In PCA and ICA, a *global, dense* code is used for representing the data vectors $\mathbf{x}(t)$. This means that every basis vector $\mathbf{w}_j$, $j = 1, \ldots, m$ is used for representing all the data vectors, and their real-valued coefficients are typically nonzero. On the other hand, in SOM only one basis vector is used for representing each data vector, and the coefficients of the expansion (6) are binary, either 0 or 1. This kind of extremely *sparse, local* and quantized representation can anyway provide a fairly good approximation $\hat{\mathbf{x}}(t)$ to the data vector $\mathbf{x}(t)$ provided that the number m of basis vectors $\mathbf{w}_j$ is sufficiently large, usually $m \gg n$. This corresponds to using a large overdetermined basis in representing the data.

In standard PCA, an overdetermined basis where the number m of basis vectors is larger than the dimensionality n of the data is impossible due to the orthogonality of the basis. In ICA, an overdetermined basis can be used [17, 1, 2]. The idea behind this is that for an overdetermined basis, less coefficients and basis vectors are needed for achieving a prescribed accuracy, leading to a sparser representation. In such a representation, most of the coefficients in the expansion (6) are close to zero for a given data vector $\mathbf{x}(t)$, and the few clearly non-zero coefficients vary for each data vector.

Between these two extremes, PCA and SOM, lies a large domain of proposed and potential local modeling methods. In local methods, a moderate number of basis vectors are used to represent the data. These basis vectors are generally different in qualitatively different regions of the distribution of the data. This kind of local models have recently become popular. The goal in applying them is to combine the benefits of global dense data representation methods and very sparse, local vector quantization type methods.

4 Local Independent Component Analysis

4.1 The proposed local ICA method

We propose the use of local linear ICA models in order to represent nonlinearly distributed data better than using a single global linear ICA model. Our local ICA method consists of the following stages.

1. Group or cluster the data into K more homogenous groups (clusters). Denote the set of data vectors $\{\mathbf{x}_i^{(k)}\}$ belonging to the kth cluster by S_k.
2. Compute the mean vector $\mathbf{m}_k$ of the vectors belonging to each cluster S_k, and subtract it from them.
3. Estimate a linear ICA basis $\mathbf{a}_j^{(k)}$, $j = 1, \ldots, N_k$, of dimension N_k for each zero-mean data set S_k using a suitable ICA algorithm.

Thus the total representation of the data is defined by the K clusters, their mean vectors $\mathbf{m}_k$, $k = 1, \ldots, K$, and the associated K ICA bases $\mathbf{a}_j^{(k)}$, $j = 1, \ldots, N_k$, $k = 1, \ldots, K$. Each of these local ICA bases describes local features of the data. For getting reliable estimates of the basis vectors, the number of data vectors falling into each of the K clusters should be large enough.

Lin's results [12] discussed in the section on nonlinear ICA earlier show that local linear ICA expansions are not unique without extra contraints. Furthermore, the degree of non-uniqueness is essentially similar as for nonlinear ICA. In our case, the local ICA expansions are defined by the sets S_k of data vectors belonging to each cluster. It is clear that changes in the clusters affect their respective local ICA expansions, too.

Selected reasonable clustering or grouping method will be discussed in the next subsections. There are also other choices which affect the results. In this paper, we have chosen the number K of clusters or groups somewhat heuristically by inspecting the quality of the obtained results. The number N_k of the basis vectors in the local ICA expansion of each cluster is selected usually so that the mean-square error is reduced by a certain amount (typically by at least 80%). Still another choice is the algorithm for estimating the independent components. If the ICA model

(1) does not hold for the data sets S_k exactly, various ICA algorithms may yield different results. We have used the symmetric fixed-point algorithm [19, 1]. It uses prewhitening, converges fast, and yields accurate results both for super-Gaussian and sub-Gaussian independent components simultaneously [20, 1].

4.2 K-means clustering

In principle, one can use any suitable clustering method of the many possible alternatives [21]. We have used the well-known K-means algorithm, which has the following basic form [21, 14]:

1. Begin with arbitrary assignment of data vectors to K clusters or begin with arbitrary set of K cluster centers and assign data vectors to nearest centers.
2. Compute the sample mean $\mathbf{m}_k$ of the data vectors $\{\mathbf{x}_i^{(k)}\}$ belonging to each cluster S_k.
3. Reassign each data vector to the cluster with the nearest mean (measured using Euclidean distance).
4. If the clustering of all data vectors has not changed, stop; else go to step 2.

The K-means method is simple and intuitively reasonable, and it has the important feature that it tries to minimize the criterion

$$J_{MSE} = \sum_{k=1}^{K} \sum_{\mathbf{x}_i^{(k)} \in S_k} \| \mathbf{x}_i^{(k)} - \mathbf{m}_k \|^2 \tag{9}$$

The minimum is achieved when the data vectors are divided into K clusters so that the overall mean-square error estimated from the data vectors is the smallest possible. Instead of the data vectors $\mathbf{x}$ themselves, one can use for determining the clusters transformed vectors $\mathbf{f}(\mathbf{x})$ describing meaningful features in the application at hand. The K-means method can be refined in several ways if desired; see [21].

Xu has derived interesting theoretical results [22] that give a lot of insight in what K-means clustering is actually doing. He has shown that K-means clustering is an optimal clustering algorithm for the data model where the probability density of the data is a mixture of Gaussians. All the Gaussians in this mixture have the same probability (mixing coefficient), and the Gaussians are spherical, with the same covariance matrix $\sigma^2\mathbf{I}$, where σ^2 is the common variance.

This derivation yields as a by-product a simple criterion for determining the optimal number of clusters in the K-means algorithm, given by [22]

$$I(K) = d\ln(\rho) + \ln(K), \quad \text{where } \rho^2 = \frac{J_{MSE}}{dN^2} \tag{10}$$

Here N is the total number of the data vectors $\mathbf{x}$, and d is their dimensionality. The optimal number K^* of clusters in the K-means algorithm is the integer value of K that provides the minimum of the criterion $I(K)$.

These results at first seem to limit the general applicability of the K-means algorithm severely. However, it should be recalled that by using a suitable mixture of Gaussians it is possible to model any sufficiently smooth probability density adequately [24]. The limitation of using spherical Gaussians can be compensated by using more of them, with their centers placed suitably. Xu has generalized his results for a mixture of elliptical Gaussians having arbitrary positive definite, symmetric covariance matrices. However, the treatment of a mixture of Gaussians becomes much more complicated in that case [24, 22].

Because ICA tries to take into account non-Gaussianity in the data, it is worthwhile to study modifications of the K-means clustering algorithm that hopefully yield more non-Gaussian clusters. A simple way to do this is to replace the standard Euclidean distance by a suitable non-Euclidean distance in the criterion (9). The generalized criterion is given by

$$J_{GEN} = \sum_{k=1}^{K} \sum_{\mathbf{x}_i^{(k)} \in S_k} d(\mathbf{x}_i^{(k)}, \mathbf{m}_k) \tag{11}$$

where $d(\mathbf{x}_i^{(k)}, \mathbf{m}_k)$ is a scalar distance measure between the vectors $\mathbf{x}_i^{(k)}$ and $\mathbf{m}_k$.

Non-Euclidean distances can be obtained from the well-known general Minkowski distance or norm, which is defined for two d-dimensional vectors $\mathbf{x} = [x_1, x_2, \ldots, x_d]^T$ and $\mathbf{y} = [y_1, y_2, \ldots, y_d]^T$ by

$$d_p(\mathbf{x}, \mathbf{y}) = \left[\sum_{j=1}^{d} | x_j - y_j |^p \right]^{\frac{1}{p}} \tag{12}$$

The choice $p = 2$ yields the standard Euclidean distance, $p = 1$ the city-block distance (L_1 distance)

$$d_1(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{d} | x_j - y_j | \tag{13}$$

and $p = \infty$ leads to the chessboard distance (L_∞ distance)

$$d_\infty(\mathbf{x}, \mathbf{y}) = \max\{| x_j - y_j |\}, \; j = 1, 2, \ldots, d \tag{14}$$

The region for which the distance $d_p(\mathbf{x}, \mathbf{y}) \le c$, where c is a constant, is circular for the Euclidean distance, diamond-shaped for the city-block distance, and square for the chessboard distance. Hence these distance measures tend to provide clusters having the respective shapes when used in the K-means clustering algorithm.

These distance measures have a connection with ICA, too. Consider the generalized Gaussian or exponential power family of probability densities. For a single random variable x, it is given by [23]

$$f(x) = C \exp\left(-\frac{|x - m|^p}{p\mathrm{E}\{|x - m|^p\}} \right) \tag{15}$$

The positive real-valued exponent p determines the type of distribution, and C is a scaling constant which normalizes the distribution to unit area (see [23]). If the parameter $p = 2$, the usual Gaussian density is obtained. The choice $p = 1$ yields the Laplacian density, and $p \to \infty$ the uniform density. The parameter values $p < 2$ in (15) give rise to super-Gaussian densities, and $p > 2$ to sub-Gaussian ones.

Clearly, the exponent p in the generalized density (15) has a similar role as in the Minkowski distance (12). This can be seen from the numerator of the exponent in (15), even though normalization takes place in a different manner as in the Minkowski distance (12). This suggests that if the data consists of sub-Gaussian densities, we should use the chessboard distance in K-means clustering, and the city-block distance for super-Gaussian densities. The experimental results in the next section show that this somewhat intuitive argument is indeed valid.

4.3 Neural grouping methods

The K-means clustering algorithm is related with the self-organizing map (SOM) [14, 15], in particular with the batch version of SOM. A major difference between these two methods is that in SOM a neighborhood is used in forming the clusters. Usually the neighborhood is decreased during the learning process. See for example [14, 15, 16] for a detailed description of the SOM learning algorithm. If learning in SOM is continued sufficiently long so that the neighborhood eventually approaches to zero, batch SOM provides in practice quite similar results as K-means clustering. Another major difference between SOM and K-means clustering is that in SOM the input data is usually mapped onto a two-dimensional lattice. The adjacent neurons (units) in the lattice, corresponding to the clusters in the K-means algorithm, describe closely related parts of the data compared with neurons farther away from each other [14, 15, 16].

Usually SOM is used so that the number of nodes (here clusters) is fairly large (typically tens or hundreds), but each node is represented by one weight vector only. Ritter [16] has justified that SOM then provides a discrete approximation to the so-called principal manifold [18], which is a lower dimensional nonlinear surface fitting the data best in the mean-square error sense. The cluster mean vectors obtained using K-means clustering also form a crude approximation to the principal manifold due to the minimization of the estimated mean-square error (9). Our method differs from the standard SOM mainly in that the number of clusters is typically fairly small, and that each cluster is represented by its mean vector and the associated local ICA subspace instead of a single weight vector.

A variant of SOM is so-called "neural gas" method [25], which is an extension of the standard K-means clustering procedure. It also takes into account a neighborhood ranking of the reference (weight) vectors. The neural gas method is more suitable for clustering than standard SOM, which is intended mainly to visualization and nonlinear mapping of the input data. For a detailed description of the neural gas method, see [25].

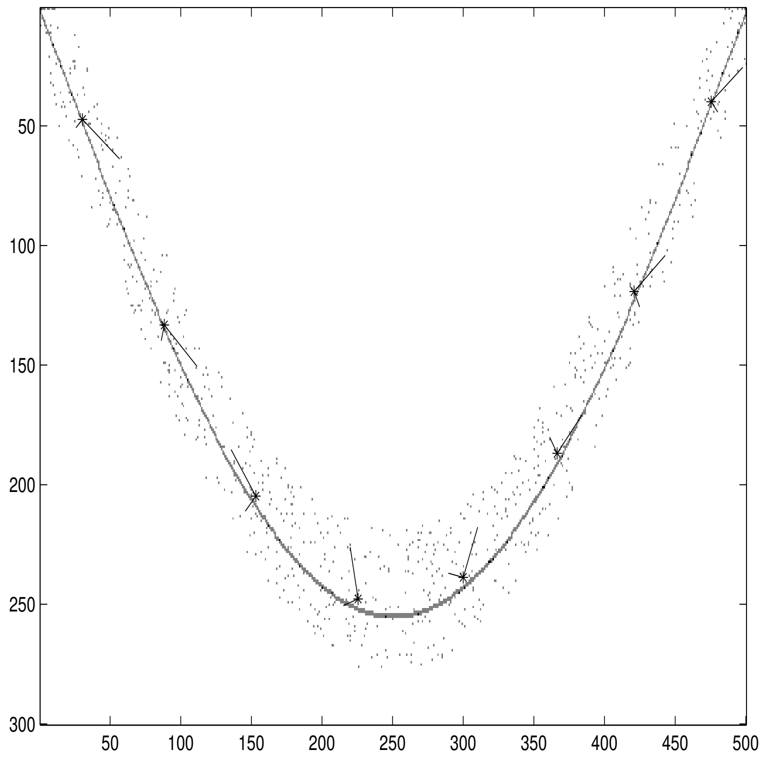

Figure 1: Eight cluster centers and the corresponding local ICA bases provided by the K-means algorithm for parabolically distributed data.

5 Experimental results

5.1 Simple experiments

We first present some simple experiments illustrating how the proposed local ICA method performs. In these experiments, the data vectors were two-dimensional only, allowing an easy visual inspection of the results.

In the first case, the data consisted of points which were uniformly distributed around a parabola as shown in Figure 1. The number of clusters was chosen to be $K = 8$, and standard K-means clustering was used for learning the clusters. The two local ICA basis vectors found by the symmetric fixed-point algorithm are shown for each cluster, attached to the mean vector of the respective cluster. These basis vectors are at least for some of the clusters clearly different from the respective local PCA basis vectors, which are constrained to be mutually orthogonal. One can also note that the data vectors belonging to the two lowest clusters are still mildly nonlinearly distributed.

Figure 2 depicts the results of the SOM algorithm for the same data. Due to the neighborhood effect, the cluster centers given by SOM are clearly above the parabola especially close to its

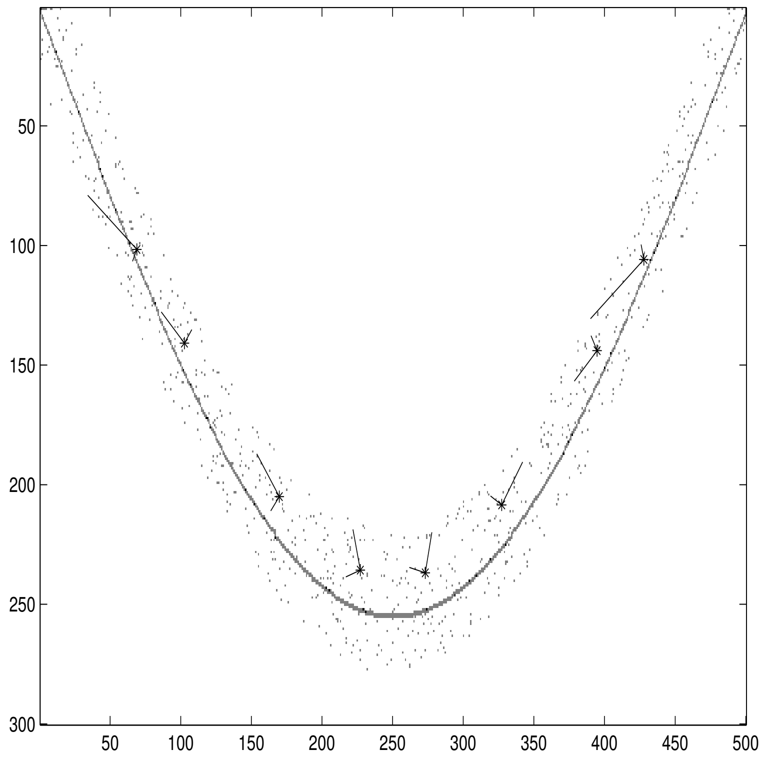

Figure 2: Eight cluster centers and the corresponding local ICA bases provided by the self-organizing map for parabolically distributed data.

bottom. Another observation compared with Figure 1 is that SOM tends to draw the cluster centers closer to each other than the K-means algorithm. When the size of the neighborhood in SOM was reduced, the clusters and their local ICA bases approached those of the K-means algorithm, and coincided for zero neighborhood. The results given by the neural gas method are quite similar to those provided by the K-means clustering, and therefore they are not shown here.

In another experiment, the data consisted of two distinct distributions: a uniform sub-Gaussian one (in the upper right part of Figure 3), and a Laplacian super-Gaussian distribution (in the lower left part). The neural gas method combined with the the fixed-point algorithm performs well, finding the local ICA bases shown in Figure 3. It works fairly well also in more difficult cases where the two distributions are closer to each other. Also in these experiments K-means clustering performed on an average equally well as the neural gas method, while the quality of the results given by the SOM clustering was poorer than for these two methods.

The Euclidean distance was compared with the city-block distance (13) and the chessboard distance (14) using artificially generated two-dimensional data. The data consisted of either four uniform distributions or four Laplacian distributions forming four clusters. The center points of

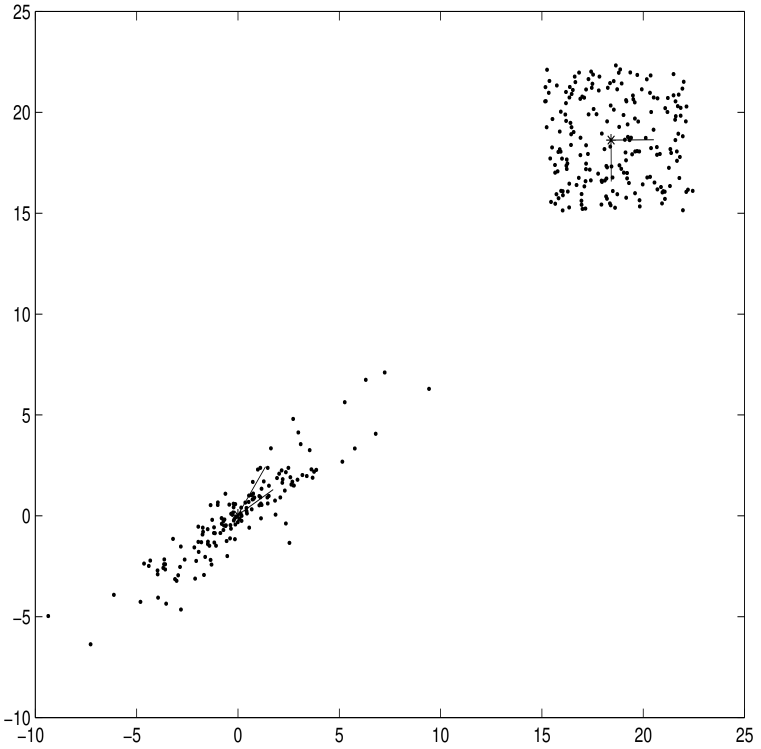

Figure 3: Local ICA basis vectors provided by neural gas grouping method in a test example.

these distributions were chosen randomly, so that the clusters (disributions) were possibly overlapping. The generated data were clustered using different distances in the K-means algorithm. After this, the local ICA basis vectors of each cluster were computed and compared with the respective theoretically correct basis vectors. This was done by computing the squared error $\| \mathbf{a}_j^{(k)} - \hat{\mathbf{a}}_j^{(k)} \|^2$ between each basis vector $\mathbf{a}_j^{(k)}$, $j = 1, \ldots, N_k$, of each cluster $k = 1, \ldots, K$ and the respective estimated basis vector $\hat{\mathbf{a}}_j^{(k)}$, and taking an average over all the basis vectors and clusters. Generally, this provides the error measure

$$E_{BV} = \frac{1}{K} \sum_{k=1}^{K} \sum_{j=1}^{N_k} \| \mathbf{a}_j^{(k)} - \hat{\mathbf{a}}_j^{(k)} \|^2 \tag{16}$$

In the simulations described here, $K = 4$ and $N_i = 2$ for all clusters.

Also the average error E_{CC} of the cluster center (mean) vectors were computed from the formula

$$E_{CC} = \frac{1}{K} \sum_{k=1}^{K} \| \mathbf{m}_k - \hat{\mathbf{m}}_k \| \tag{17}$$

where $\mathbf{m}_k$ is the theoretical mean vector of cluster k and $\hat{\mathbf{m}}_k$ is the mean vector of the respective cluster estimated using the K-means algorithm.

Because in the K-means algorithm the initial cluster centers are chosen randomly, the final clusters can be different for different trials. Therefore clustering and estimation of the local

	$\mu_{E_{BV}}$	$\sigma_{E_{BV}}$	$\mu_{E_{CC}}$	$\sigma_{E_{CC}}$	$\min_{E_{BV}}(\%)$	$\min_{E_{CC}}(\%)$
d_∞	0.0108	0.0305	0.0171	0.0147	70.7	66.8
d_1	0.0774	0.1019	0.1051	0.1545	5.1	6.3
d_2	0.0186	0.0390	0.0275	0.0173	24.2	26.9

Table 1: Average errors given by different distance measures for uniform distributed clusters in K-means clustering.

	$\mu_{E_{BV}}$	$\sigma_{E_{BV}}$	$\mu_{E_{CC}}$	$\sigma_{E_{CC}}$	$min_{E_{BV}}(\%)$	$min_{E_{CC}}(\%)$
d_∞	0.1458	0.1012	0.2542	0.4891	18.9	32.0
d_1	0.1099	0.0971	0.2227	0.4940	50.3	59.8
d_2	0.1295	0.0982	0.2887	0.5773	30.8	8.2

Table 2: Average errors given by different distance measures for Laplacian distributed clusters in K-means clustering.

ICA basis vectors were performed for 1000 different realizations of the data set. For uniformly distributed clusters, the chessboard distance (14) gave the best local ICA basis vectors as can be seen from Table 1. Also the error E_{CC} of cluster centers is the smallest for the d_∞-distance. The tables give both the means $\mu_{E_{BV}}$, $\mu_{E_{CC}}$ and the variances $\sigma_{E_{BV}}$, $\sigma_{E_{CC}}$ of the error criteria E_{BV} and E_{CC}, respectively. The last two columns $\min_{E_{BV}}$ and $\min_{E_{CC}}$ show the percentages of the cases in which the error given by each distance was the smallest. Clearly, d_1-distance is the worst in clustering uniformly distributed data.

On the other hand, Table 2 shows that d_1-distance performs the best in clustering Laplacian data. The differences in errors are smaller than for uniform distributed data. Table 2 also shows that the standard d_2-norm performs fairly well in estimating the directions of the local ICA basis vectors but poorer in estimating the centers of the clusters. The standard deviations $\sigma_{E_{CC}}$ of the estimated cluster centers are fairly high for all the distance measures because the K-means algorithm does not always yield good results. For example two cluster centers can be very near each other.

We tested also Xu's criterion (10) for determining the optimal number of clusters in K-means algorithm. It turned out that it performs very well in predicting the number of clusters when the data is Gaussian as assumed in the derivation of the formula (10). We tested this criterion using several real-world data sets, too. Regrettably, it turned out that the criterion does not work for these data sets. There is no global minimum or minimum at all, because the value of the criterion (10) tends to decrease gradually with increasing number of clusters, making it impossible to predict the optimal number of clusters.

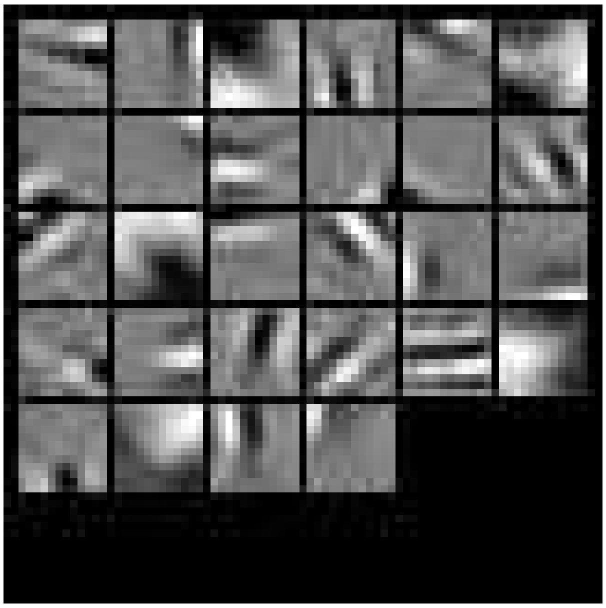

Figure 4: Global ICA basis vectors for image data.

5.2 Natural image data

We have made more extensive experiments with image data, consisting of 15 natural images describing different scenes, plants and animals. In these experiments a set of typically 100000 possibly overlapping 12×12 subimages was randomly extracted from the original images. The subimages were then represented as 144-component data (mixture) vectors $\mathbf{x}_k$. They were preprocessed by subtracting from each $\mathbf{x}_k$ the mean of its components, and then dividing $\mathbf{x}_k$ by its norm for reducing the variance.

In the first experiments, we used the data vectors $\mathbf{x}_k$ as such in clustering. Later on we found that the proposed local ICA method yields more meaningful results if the data vectors are clustered according to their mean frequency in the K-means algorithm. This was accomplished by applying discrete cosine transform to the data vectors before clustering them. However, this operation was used only in finding the clusters. After the clusters were found, the local ICA bases were computed for the data vectors belonging to each cluster without application of discrete cosine transform. In these experiments, the standard Euclidean norm was used in clustering, because some of the independent components of natural images preprocessed as described above are sub-Gaussian while most of them are super-Gaussian [26].

Figure 4 shows the 28 first basis vectors, represented again as 12×12 subimages, found by fitting the global linear ICA model (1) to the data without any clustering. These subimages correspond to various types of features, such as smooth changes in subimages, edges in different directions, and local information in the corners of the images.

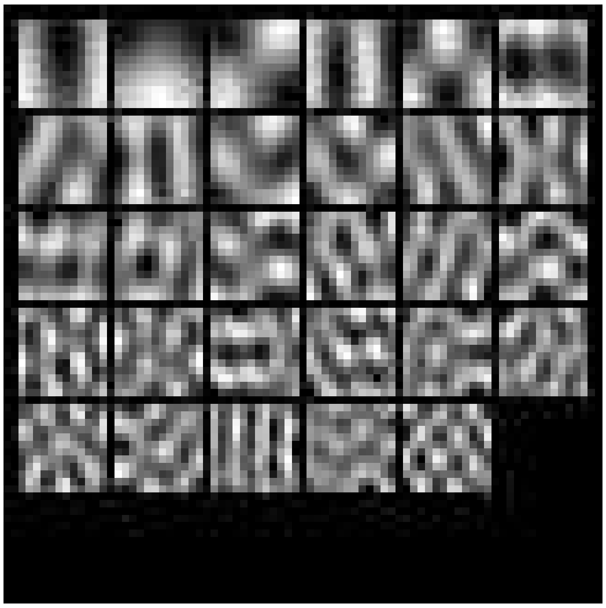

Figure 5: Local PCA basis vectors for the first cluster of the image data.

Figure 5 depicts the 29 basis vectors (subimages) obtained by computing the standard PCA expansion for the data vectors belonging to the first cluster of the image data. These basis images show very clearly typical features extracted by standard PCA. They are sensitive to spatial frequencies, starting from low frequencies and then proceeding to higher ones for which the basis images resemble more or less a checkerboard. The local PCA expansions for the other 3 clusters formed using the K-means algorithm were almost similar and are therefore not shown here. We tried also global PCA expansion, which provided qualitatively very similar features to those in Figure 5, sensitive to spatial frequencies.

Finally, Figures 6–9 show the basis images of the local ICA expansions computed for the data vectors belonging to each of the four clusters found by the K-means algorithm. The number of ICA basis vectors is different for each cluster because we determined it by requiring that 80% of the total energy is included. In all the experiments, the symmetric fixed-point algorithm described in [19] was used for computing the needed ICA expansions.

Clearly, local ICA basis vectors yield filters that are sensitive to edges and lines of varying orientations, as well as wavelet type filters which are sensitive to local variations in the images. More specifically, one can observe that the basis images in Figure 6 have much more variation in horizontal than in vertical direction. Many of these local ICA basis images correspond to filters sensitive to horizontal edges and lines. The basis images of Figure 8 are largely qualitatively similar as in Fig. 6 but this time sensitive to vertical variations. Most of the 9 subimages in Figure 7 represent smooth variations, while the basis images in Figure 9 are sensitive to local wavelet type features with a few exceptions. These results can be understood by recalling that

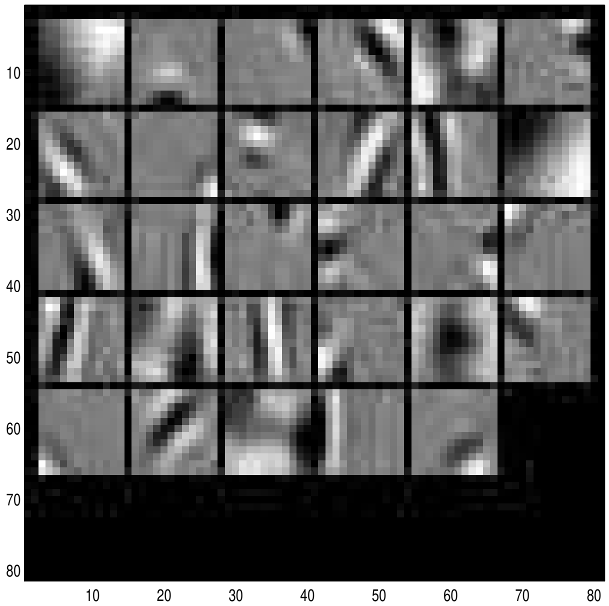

Figure 6: Local ICA basis vectors for the first cluster of the image data.

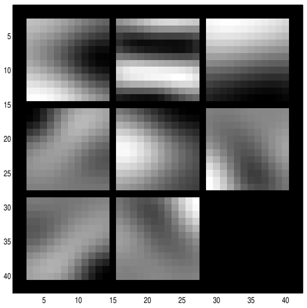

Figure 7: Local ICA basis for the second cluster.

clustering was based on frequency information provided by the discrete cosine transformation.

We applied also SOM and the neural gas method to the same natural image data. The local ICA basis images found by these methods were visually less meaningful than the local ICA basis images shown above, found using standard K-means clustering.

6 Related work

The local ICA method described above has connections with several other approaches. We first discuss related local models, and then briefly other approaches used for extracting features from natural images.

Kambhatla and Leen [27] have developed a local PCA method, which applies vector quan-

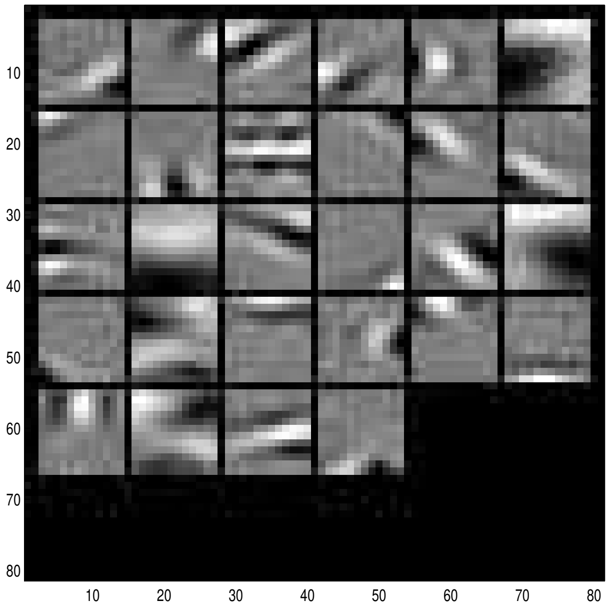

Figure 8: Local ICA basis for third cluster.

tization instead of SOM or K-means clustering to dividing the data into disjoint regions. Then for the data vectors falling into each region a local PCA expansion is computed. In a similar manner as in our method, vector quantization tries to capture the overall nonlinear structure in the data coarsely, while the local PCA expansions of each region describe local variations. This method and a related one introduced by Joutsensalo [28] are useful mainly in data compression due to the properties of PCA.

Another approach to local PCA has been adopted in an interesting paper by Hinton, Dayan, and Revow [29]. They use a mixture of linear PCA models, and fit the mixture to the data by using an EM type maximum likelihood algorithm. This reduces to the k-means algorithm in a limiting case. Alternatively, one can use a mixture of linear factor analysis models. Recently, these ideas have been in a straightforward manner extended to a mixture of linear ICA models in [2, 30, 31].

Kohonen [15] has introduced so-called ASSOM method as an extension of SOM for unsupervised extraction of invariant local features from the input data. This is done by associating a subspace instead of a single weight vector to each node of the self-organizing map. The basis vectors of the subspace are then used as the invariant features for the data vectors classified to that subspace. The results are interesting and meaningful for real-world speech and image data [15]. Oja and Valkealahti [32] have applied ICA to form the subspaces in ASSOM. This also leads to a local ICA method which, however, differs considerably from ours. In particular, they use a classification based criterion function for clustering, and apply it to extraction of ICA type features from textures.

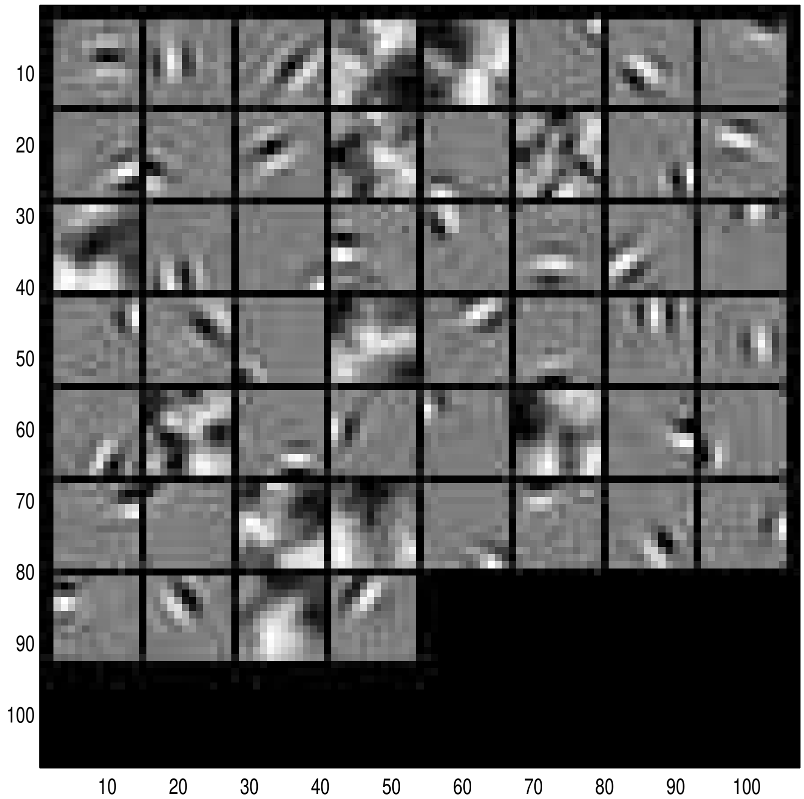

Figure 9: Local ICA basis for the fourth cluster.

Recently, ICA and various its extensions have been applied to extracting features from natural images in several papers. Bell and Sejnowski were the first who used standard linear ICA for this task in [33]. Many of the features found by them resemble those found using local ICA, but there are more local checkerboard type features. Overcomplete bases [17, 1] have been used for image feature extraction in [34, 1]. The results resemble very much those obtained using our local ICA method. Other extensions of the standard ICA model which have been used for image feature extraction are independent subspace analysis [35, 1], topographic ICA [38, 1], and sparse priors imposed on the mixing matrix [37, 1]. These methods seem to yield somewhat different results which the interested reader can compare with ours.

Image features obtained using ICA are also related to features provided by recent constructions in harmonic analysis such as ridgelets and curvelets [36]. A general difficulty in comparing image features given by various methods is that it is very difficult if impossible to compare the results quantitatively in an appropriate way. That is, there is no good numerical performance index available for assessing the quality of the extracted image features. Moreover, the usefulness of extracted features depends on the specific application.

7 Concluding remarks

In this paper, we have introduced a method where local linear ICA models are used for describing suitably clustered data. The clustering part is responsible for an overall coarse nonlinear representation of the data, while linear ICA models of each cluster are used for describing local features. The goal is to represent the data better than in linear ICA while avoiding computational difficulties related with nonlinear ICA. Experimental results are presented both for artificial data and natural images, and connections with related existing methods are discussed.

There exist several possibilities for extending the ideas described in this work. For example, a smooth transition from one local ICA expansion to another neighboring one could be achieved by using partly overlapping local ICA models. Generally, the proposed method in its present form seems to work well if the data is more or less clustered. But if no clear clusters can be found, the method does not necessarily perform any better than standard linear ICA.

Acknowledgments

The authors thank Petteri Pajunen, Harri Lappalainen, and Kimmo Valkealahti for helpful comments.

References

[1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis.* New York, USA: Wiley, to appear in 2001.

[2] T.-W. Lee, *Independent Component Analysis - Theory and Applications.* Dordrecht, The Netherlands: Kluwer, 1998.

[3] M. Girolami, *Self-Organising Neural Networks - Independent Component Analysis and Blind Source Separation.* London, United Kingdom: Springer, 1999.

[4] S. Haykin (Ed.), *Unsupervised Adaptive Filtering, Vol. I: Blind Source Separation.* York, USA: Wiley, 2000.

[5] L. Parra, G. Deco, and S. Miesbach, "Statistical independence and novelty detection with information-preserving nonlinear maps," *Neural Computation*, vol. 8, pp. 260-269, 1996.

[6] P. Pajunen and J. Karhunen, "A maximum likelihood approach to nonlinear blind separation," in *Proc. 1997 Int. Conf. on Artificial Neural Networks*, Lausanne, Switzerland, October 1997, pp. 541-546.

[7] J. Lin, D. Grier, and J. Cowan, "Faithful representation of separable distributions," *Neural Computation*, vol. 9, 1997, pp. 1305-1320.

[8] A. Hyvärinen and P. Pajunen, "Nonlinear independent component analysis: existence and uniqueness results", *Neural Networks*, vol. 12, no. 2, 1999, pp. 209-219.

[9] H. Yang, S. Amari, and A. Cichocki, "Information-theoretic approach to blind separation of sources in non-linear mixture," *Signal Processing*, vol. 64, 1998, pp. 291-300.

[10] A. Taleb and C. Jutten, "Source separation in post-nonlinear mixtures," *IEEE Trans. on Signal Processing*, vol. 47, no. 10. October 1999, pp. 2807-2820.

[11] A. Cichocki, L. Zhang, S. Choi, and S. Amari, "Nonlinear dynamic independent component analysis using state-space and neural network models," in *Proc. First Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, Aussois, France, January 1999, pp. 99-104.

[12] J. Lin, "Factorizing probability density functions: generalizing ICA," in *Proc. First Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, Aussois, France, January 1999, pp. 313-318.

[13] J. Karhunen, E. Oja, L. Wang, R. Vigario, and J. Joutsensalo, "A class of neural networks for independent component analysis," *IEEE Trans. on Neural Networks*, vol. 8, May 1997, pp. 486-504.

[14] S. Haykin, *Neural Networks - A Comprehensive Foundation, 2nd ed.* Englewood Cliffs, NJ: Prentice-Hall, 1998.

[15] T. Kohonen, *Self-Organizing Maps.* New York, Berlin: Springer-Verlag, 1995.

[16] H. Ritter, T. Martinetz, and K. Schulten, *Neural Computation and Self-Organizing Maps.* Reading, MA: Addison-Wesley, 1992.

[17] M. Lewicki and T. Sejnowski, "Learning overcomplete representations," *Neural Computation*, vol. 12, 2000, pp. 337-365.

[18] T. Hastie and W. Stuetzle, "Principal curves," *Journal of the American Statistical Association.* vol. 84, 1989, pp. 502-516.

[19] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Trans. on Neural Networks*, vol. 10, no. 3, May 1999, pp. 626-634.

[20] X. Giannakopoulos, J. Karhunen, and E. Oja, "An experimental comparison of neural algorithms for independent component analysis and blind separation," *Int. J. of Neural Systems*, vol. 9, no. 2, April 1999, pp. 99-114.

[21] A. Jain and R. Dubes, *Algorithms for Clustering Data.* Englewood Cliffs, NJ: Prentice-Hall, 1988

[22] L. Xu, "Bayesian Ying-Yang machine, clustering and number of clusters," *Pattern Recognition Letters*, vol. 18, 1997, pp. 1167-1178.

[23] G. Box and G. Tiao, *Bayesian Inference in Statistical Analysis.* Reading, Mass.: Addison-Wesley, 1973.

[24] C. Bishop, *Neural Networks for Pattern Recognition.* Oxford: Clarendon Press, 1995.

[25] T. Martinetz, S. Berkovich, and K. Schulten, "Neural-gas network for vector quantization and its application to time-series prediction," *IEEE Trans. on Neural Networks*, vol. 4, July 1993, pp. 558-569.

[26] J. Hurri, A. Hyvärinen, and E. Oja, "Wavelets and natural image statistics," in *Proc. 10th Scandinavian Conf. on Image Analysis*, Lappeenranta, Finland, June 1997.

[27] N. Kambhatla and T. Leen, "Dimension reduction by local principal component analysis," *Neural Computation*, vol. 9, 1997, pp. 1493-1516.

[28] J. Joutsensalo, "Nonlinear dimension reduction by combining competitive and distributed learning," in *Proc. 1995 Int. Conf. on Artificial Neural Networks*, Paris, France, October 1995, pp. 395-400.

[29] G. Hinton, P. Dayan, and M. Revow, "Modeling the manifolds of images of handwritten digits," *IEEE Trans. on Neural Networks*, vol. 8, January 1997, pp. 65-74.

[30] T.-W. Lee, M. Lewicki, and T. Sejnowski, "ICA mixture models for unsupervised classification and automatic context switching," in *Proc. First Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, Aussois, France, January 1999, pp. 209-214.

[31] T.-W. Lee and M. Lewicki, "The generalized Gaussian mixture model using ICA," in *Proc. 2nd Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, Espoo, Finland, June 2000, pp. 239-244.

[32] E. Oja and K. Valkealahti, "Local independent component analysis by the self-organizing map," in *Proc. 1997 Int. Conf. on Artificial Neural Networks*, Lausanne, Switzerland, October 1997, pp. 553-558.

[33] A. Bell and T. Sejnowski, "The "independent components" of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, 1997, pp. 3327-3338.

[34] M. Lewicki and B. Olshausen, "Inferring sparse, overcomplete image codes using an efficient coding framework," in M. Jordan, M. Kearns, and S. Solla (Eds.), *Advances in Neural Information Processing Systems 10*, MIT Press, 1998, pp. 815-821.

[35] A. Hyvärinen and P. Hoyer, "Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces," *Neural Computation*, vol. 12, 2000, pp. 1705-1720.

[36] D. Donoho, "Nature vs. math: interpreting independent component analysis in light of computational harmonic analysis," in *Proc. 2nd Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, Espoo, Finland, June 2000, pp. 459-470.

[37] A. Hyvärinen and R. Karthikesh, "Sparse priors on the mixing matrix in independent component analysis," in *Proc. 2nd Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, Espoo, Finland, June 2000, pp. 477-482.

[38] A. Hyvärinen, P. Hoyer, and M. Inki , "Topographic independent component analysis: visualizing the dependence structure," in *Proc. 2nd Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, Espoo, Finland, June 2000, pp. 591-596.