LOCALLY LINEAR INDEPENDENT COMPONENT ANALYSIS

Juha Karhunen and Simona Mălăroiu

Helsinki University of Technology, Laboratory of Computer and Information Science
P.O.Box 5400, FIN-02015 HUT, Espoo, FINLAND
email: Juha.Karhunen@hut.fi, simona@james.hut.fi
http://www.cis.hut.fi/projects/ica

ABSTRACT

Linear Independent Component Analysis (ICA) has become an important technique in unsupervised neural learning. 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 we study techniques where local ICA models are applied to data first grouped or clustered using some suitable algorithm. The grouping part is responsible for an overall coarse nonlinear representation of the underlying data, while linear ICA models of each group (cluster) are used for describing local features of the data. The goal is to represent the data better than in linear ICA in a computationally feasible manner. We study several grouping methods, including standard K-means clustering, self-organizing maps, and neural gas. We also present a general theoretical framework encompassing a large number of methods for representing data. These range from global, dense representation methods (for example global PCA or ICA) to local, very sparse coding methods (for example self-organizing maps). The proposed local ICA methods lie between these two extremes.

1. INTRODUCTION

Independent Component Analysis (ICA) [1, 2, 3, 4, 5] is a statistical technique which in many cases characterizes the data in a natural way. ICA and the related blind source separation (BSS) problem [2, 3, 6] have grown important research and application topics 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

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) = \sum_{i=1}^{m} s_i(t)\mathbf{a}_i. \tag{1}$$

Here the vector $\mathbf{s}(t) = [s_1(t), \ldots, s_m(t)]^T$ contains the m independent components (or source signals) $s_i(t)$ 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 $\mathbf{a}_i$, $i = 1, \ldots, m$, are the basis vectors of ICA; see [1, 2, 3]. 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, 3, 4, 5, 6] for further information and reviews. The estimate of the mixing matrix $\mathbf{A}$ is obtained as the pseudoinverse of $\mathbf{B}$.

In ICA, the basis vectors $\mathbf{a}_i$ 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 $\mathbf{a}_i$ 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. 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) $\mathbf{a}_i$, $i = 1, \ldots, m$. 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.

During the last years, several attempts have been made to generalize linear ICA and BSS problems to nonlinear data models. See [2, 7, 8] for more information and references. However, nonlinear ICA and BSS methods have met only limited success due to two major reasons. First, solution of the nonlinear ICA problem is usually highly non-unique [7]. The nonlinear BSS problem is somewhat better defined, but even there the ambiguities in the found solutions are greater than in standard linear BSS. Another difficulty in the proposed general nonlinear ICA and BSS methods 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 these methods to high-dimensional data sets.

In this paper we develop a fairly simple data model that tries to overcome these drawbacks, and is computationally not too demanding.

2. 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 $\mathbf{w}_i$ are generally 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:

$$\hat{\mathbf{x}}(t) = \sum_{i=1}^{m} g_i(\mathbf{x}(t), \mathbf{W})\mathbf{w}_i = \mathbf{W}\mathbf{g}(\mathbf{x}(t), \mathbf{W}) \qquad (3)$$

Here $g_i(\mathbf{x}(t), \mathbf{W})$ is the scalar coefficient corresponding to the basis vector $\mathbf{w}_i$, and $\mathbf{g}(\mathbf{x}(t), \mathbf{W})$ is the column vector consisting of these m coefficients. Generally, the coefficients g_i of the expansion (3) 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 (3) 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 (3), depending on the choice of the basis vectors $\mathbf{w}_1, \ldots, \mathbf{w}_m$ and the coefficients $g_i(\mathbf{x}(t), \mathbf{W})$.

Assume first that the coefficients in (3) 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 $g_i(\mathbf{x}(t), \mathbf{W}) = \mathbf{w}_i^T\mathbf{x}(t)$. An example of this type of representation is standard principal component analysis (PCA) [5], 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, (3) can be written in the form $\hat{\mathbf{x}}(t) = \mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{x}(t)$, and the coefficient vector of the expansion (3) 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 nonorthogonal basis [3]. 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 (3) when the coefficients g_i are linear real-valued functions of the data vectors $\mathbf{x}(t)$.

On the other hand, if the coefficients $g_i(\mathbf{x}(t), \mathbf{W})$ depend nonlinearly on the data vector $\mathbf{x}(t)$ or are quantized, some approximation error is usually present even though $m \geq n$. A nonlinear extension of standard PCA where $\mathbf{g}(\mathbf{x}(t), \mathbf{W}) = \mathbf{g}(\mathbf{W}^T\mathbf{x}(t))$ [3] 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 BSS after prewhitening [3].

Also the well-known self-organizing map (SOM) [5, 9, 10] can be expressed in terms of the model (3). In SOM, only one of the coefficients $g_i(\mathbf{x}(t), \mathbf{W})$ is nonzero and actually equal to unity. This coefficient corresponds to the weight (basis) vector $\mathbf{w}_j$ 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 roughly speaking provide an optimal vector quantization [5], SOM is anyway able to form 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 (3), 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 all the basis vectors $\mathbf{w}_1, \ldots, \mathbf{w}_m$ are 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 (3) 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 $\mathbf{x}(t)$, if the number m of basis vectors $\mathbf{w}_i$ 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 [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. For a given data vector $\mathbf{x}(t)$, most of the coefficients in the expansion (3) are close to zero, and the few clearly nonzero coefficients vary for each data vector.

Between these two extremes, PCA and SOM, there 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, trying to combine the good properties of global dense data representation methods and very sparse, local vector quantization type methods.

3. LOCAL INDEPENDENT COMPONENT ANALYSIS

3.1. 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 domains (clusters). Denote the set of data vectors $\mathbf{x}^{(i)}$ belonging to the ith cluster by S_i.
2. Compute the mean vector $\mathbf{m}_i$ of each cluster S_i, and subtract it from the vectors of S_i.
3. Estimate a linear ICA basis $\mathbf{a}_j^{(i)}$, $j = 1, \ldots, N_i$, of dimension N_i for each zero-mean data set S_i using a suitable ICA algorithm.

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

In our earlier paper [8], we have justified based on Lin's results [11] 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_i of data vectors belonging to each cluster. It is clear that changes in the clusters affect their respective local ICA expansions, too.

Some alternative 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 on heuristic grounds, inspecting the obtained results. The number N_i of the basis vectors of ICA in each cluster has been 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_i exactly, various ICA algorithms may yield different results. We have used the symmetric fixed-point algorithm [12]. It uses prewhitening, converges fast, and yields accurate results both for super-Gaussian and sub-Gaussian independent components simultaneously.

3.2. K-means clustering

In principle, one can use any well-defined clustering method of the many possible alternatives. We have used the well-known K-means algorithm, which has the following form [5]:

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 of each cluster S_i.
3. Reassign each data vector to the cluster with the nearest mean.
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

$$J = \sum_{k=1}^{K} \sum_{\mathbf{x}^{(i)} \in S_i} \| \mathbf{x}^{(i)} - \mathbf{m}_i \|^2 . \qquad (4)$$

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 vectors $\mathbf{f}(\mathbf{x})$ describing meaningful features in the application at hand.

Xu has recently derived interesting theoretical results [13] that give a lot of insight of 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 the mixture have the same probability (mixing coefficient); and the Gaussians are spherical, with a common covariance matrix $\sigma^2\mathbf{I}$, where σ^2 is a common variance. As a byproduct, a simple criterion for determining the optimal number of clusters in the K-means algorithm is obtained [13].

These results at first seem to limit the general applicability of the K-means algorithm seriously. However, it should be recalled that by using a suitable mixture of Gaussians it is possible to model well virtually any sufficiently smooth probability density [14]. 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 is much more complicated in this case [14, 13].

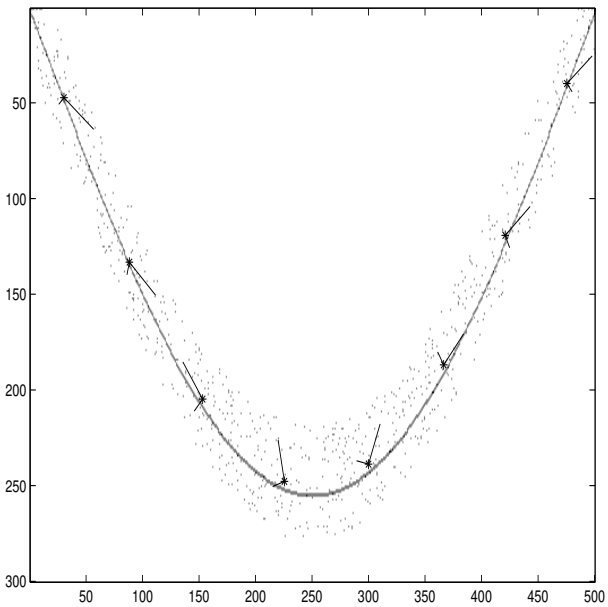

Figure 1: Eight cluster centers and the corresponding local ICA bases provided by the K-means algorithm for parabolically distributed data.

3.3. Neural grouping methods

The K-means clustering algorithm is related with the self-organizing map (SOM) [5, 9], 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 [5, 9, 10] 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 rather similar results as K-means clustering. Another major difference between SOM and K-means clustering is that in SOM the input data is mapped onto a usually two-dimensional lattice. The adjacent neurons (units) in the lattice, corresponding to clusters in the K-means algorithm, describe closely related parts of the data compared with neurons farther away from each other [5, 9, 10].

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 vector only. Ritter [10] has justified that SOM then provides a discrete approximation to the principal manifold, which is a lower dimensional surface fitting the data best in the mean-square error sense. The cluster mean vectors obtained using K-means clustering also form at least a crude approximation to the principal manifold due to the minimization of the estimated mean-square error (4). 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.

A variant of SOM is so-called "neural gas" method [15], 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 [15].

3.4. Relations to other methods

The local ICA method introduced above has connections with several other approaches.

Kambhatla and Leen [16] have developed a local PCA method where they use vector quantization (instead of SOM or K-means clustering) for dividing the data into disjoint regions. Then for each region a local PCA expansion is computed. Their method is useful mainly in data compression.

Another approach to local PCA has been taken in an interesting paper by Hinton, Dayan, and Revow [17]. They use a mixture of linear PCA models, and fit the mixture to the data by using an EM type maximum

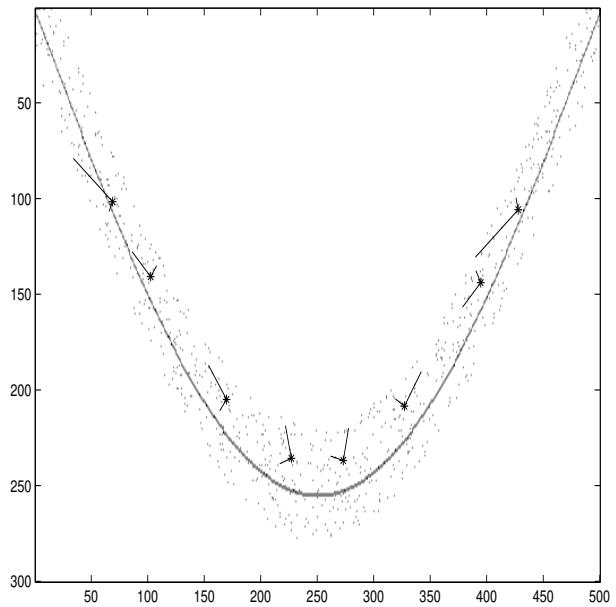

Figure 2: Eight cluster centers and the corresponding local ICA bases provided by the self-organizing map for parabolically distributed data.

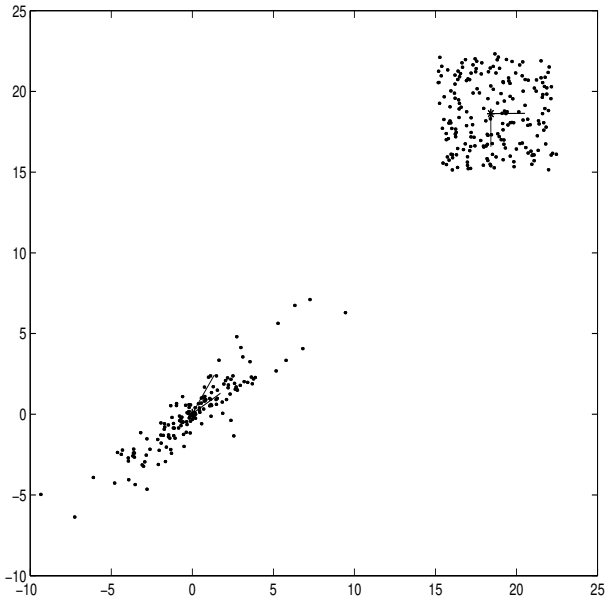

Figure 3: Local ICA basis vectors provided by neural gas grouping method in a test example.

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 extended to a mixture of linear ICA models in [2, 18], with an application to unsupervised classification.

Kohonen [9] introduced so-called ASSOM method 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. Recently, Oja and Valkealahti [19] applied ICA to form the subspaces in ASSOM. The resulting local ICA method differs considerably from ours, because a classification based criterion function is used for clustering. See also our previous paper [8] for more information on these connections.

Also Lin's considerations [11] based on local geometric information can be used to determine locally linear ICA coordinate systems.

4. EXPERIMENTAL RESULTS

We first present some simple experiments illustrating how the proposed local ICA method performs. In these experiments, the data 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 bottom. Another observation compared with Fig. 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 neural gas method provided similar results as the K-means clustering.

In another experiment, the data consisted of two distinct distributions: a uniform sub-Gaussian one (in the upper right part of Fig. 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

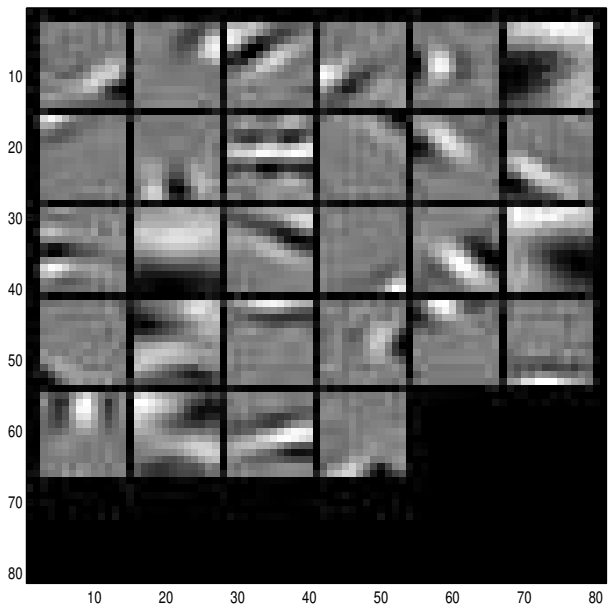

Figure 4: Local ICA basis for the third cluster of natural image data, K-means algorithm.

in Figure 3. It also works fairly well in more difficult cases where the two distributions are closer to each other.

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 required preprocessing and experimental results for the K-means clustering are described in [8]. Figure 4 show an example of the results in the case of four clusters. These ICA basis images of the third cluster are mainly sensitive to edges, lines, and other variations in the vertical direction. The ICA basis images of the other 3 clusters are qualitatively different, representing horizontal features, smooth features, and wavelet type localized features. See [8] for a more detailed description, showing results for all the four clusters and comparing them with global ICA, global PCA, and local PCA. In this experiment, SOM and the neural gas method yielded visually less meaningful ICA basis images than standard K-means clustering.

5. REFERENCES

[1] P. Comon, "Independent component analysis - a new concept?," *Signal Processing*, vol. 36, pp. 287-314, 1994.

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

[3] 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.

[4] E. Oja, J. Karhunen, A. Hyvärinen, R. Vigario, and J. Hurri, "Neural independent component analysis - approaches and applications," in S.-I. Amari and N. Kasabov (Eds.), *Brain-Like Computing and Intelligent Information Systems*, Springer, Singapore, 1997, pp. 167-188.

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

[6] S.-I. Amari and A. Cichocki, "Adaptive blind signal processing - neural network approaches," *Proc. of the IEEE*, vol. 86, October 1998, pp. 2026-2048.

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

[8] J. Karhunen and S. Malaroiu, "Local independent component analysis using clustering," in *Proc. First Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, Aussois, France, January 1999, pp. 43-48.

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

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

[11] J. Lin, "Factorizing probability density functions: generalizing ICA," in *Proc. ICA'99 Workshop*, Aussois, France, January 1999, pp. 313-318.

[12] A. Hyvärinen, "A family of fixed-point algorithms for independent component analysis," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Munich, Germany, April 1997, pp. 3917-3920.

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

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

[15] 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.

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

[17] 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.

[18] T.-W. Lee, M. Lewicki, and T. Sejnowski, "ICA mixture models for unsupervised classification and automatic context switching," in *Proc. ICA'99 Workshop*, Aussois, France, January 1999, pp. 209-214.

[19] 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.