International workshop on Independent Component Analysis and Blind Signal Separation.
January 11-15 Aussois, France 209-214 (1999).

ICA MIXTURE MODELS FOR UNSUPERVISED CLASSIFICATION AND AUTOMATIC CONTEXT SWITCHING

Te-Won Lee, Michael S. Lewicki and Terrence J. Sejnowski

Howard Hughes Medical Institute
Computational Neurobiology Laboratory
The Salk Institute
10010 N. Torrey Pines Road
La Jolla, California 92037, USA
{tewon,lewicki,terry}@salk.edu

ABSTRACT

We present an unsupervised classification algorithm based on an ICA mixture model. A mixture model is a model in which the observed data can be categorized into several mutually exclusive data classes. In an ICA mixture model, it is assumed that the data in each class are generated by a linear mixture of independent sources. The algorithm finds the independent sources and the mixing matrix for each class and also computes the class membership probability of for each data point. This approach extends the Gaussian mixture model so that the clusters can have non-Gaussian structure. Performance on a standard classification problem, the Iris flower data set, demonstrates that the new algorithm can improve classification accurately over standard Gaussian mixture models. We also show that the algorithm can be applied to blind source separation in nonstationary environments. The method can switch automatically between learned mixing matrices in different environments.

1. INTRODUCTION

Recently, Blind Source Separation (BSS) by Independent Component Analysis (ICA) has received attention because of its potential signal processing applications such as speech enhancement systems, telecommunications and medical signal processing. ICA is a technique for finding a linear non-orthogonal coordinate system in multivariate data. The directions of the axes of this coordinate system are determined by the data's second and higher-order statistics. The goal of the ICA is to linearly transform the data such that the transformed variables are as statistically independent from each other as possible (Bell and Sejnowski, 1995; Cardoso and Laheld, 1996; Lee et al., 1999a).

ICA generalizes the technique of principal component analysis (PCA) and, like PCA, has proven a useful tool for finding structure in data. ICA has also been successfully applied to processing real world data, including separating mixed speech signals (Lee et al., 1997) and removing artifacts from EEG recordings (Jung et al., 1998).

One limitation of ICA is the assumption that the sources are independent. Here we present an approach for relaxing this assumption using mixture models. In a mixture model, the observed data can be categorized into several mutually exclusive classes (Duda and Hart, 1973). When the class variables are modeled as multivariate Gaussian densities, the mixture model is called a Gaussian mixture model. We generalize the Gaussian mixture model by modeling each class with a mixture of independent components (ICA mixture model). This allows modeling of clusters with non-Gaussian (e.g., platykurtic or leptokurtic) structure. An algorithm for learning the parameters is derived using the expectation maximization (EM) algorithm. We demonstrate that this approach shows improved performance in data classification problems. In addition, we apply the algorithm to BSS by learning mixing matrices in different environments. This presents a method for addressing the problem of nonstationarity.

2. THE ICA MIXTURE MODEL

We assume that the data were generated by a mixture density (Duda and Hart, 1973):

$$p(\mathbf{x}|\Theta) = \sum_{k=1}^{K} p(\mathbf{x}|C_k, \theta_k) p(C_k), \qquad (1)$$

where $\Theta = (\theta_1, \cdots, \theta_K)$ are the unknown parameters for each $p(\mathbf{x}|C_k, \theta_k)$, called the component densities.

We further assume that the number of classes, K, and the a priori probability, $p(C_k)$, for each class are known. In the case of a Gaussian mixture model, $p(\mathbf{x}|C_k, \theta_k) \propto N(\mu_k, \Sigma_k)$. Here we assume that the form of the component densities is non-Gaussian and the data within each class are described by an ICA model.

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{s}_k + \mathbf{b}_k, \tag{2}$$

where $\mathbf{A}_k$ is a $N \times M$ scalar matrix (called the basis or mixing matrix) and $\mathbf{b}_k$ is the bias vector for class k. The vector $\mathbf{s}_k$ is called the source vector (these are also the coefficients for each basis vector). It is assumed that the individual sources s_i within each class are mutually independent across a data ensemble. For simplicity, we consider the case where $\mathbf{A}_k$ is full rank, i.e. the number of sources (M) is equal to the number of mixtures (N).

Figure 1 shows a simple example of a dataset that can be described by ICA mixture model. Each class was generated with eq.2 using different $\mathbf{A}$ and $\mathbf{b}$. Class (o) was generated by two uniform distributed sources, whereas class (+) was generated by two Laplacian distributed sources ($p(s) \propto \exp(-|s|)$).

To model the unlabeled data points, the task is to determine the parameters for each class, $\mathbf{A}_k, \mathbf{b}_k$ and the probability of each class $p(C_k|\mathbf{x}, \theta_{1:K})$ for each data point. A learning algorithm can be derived by an expectation maximization approach (Ghahramani, 1994) and implemented in the following steps:

1. Compute the log-likelihood of the data for each class:

$$\log p(\mathbf{x}|C_k, \theta_k) = \log p(\mathbf{s}_k) - \log(\det|\mathbf{A}_k|), \tag{3}$$

where $\theta_k = \{\mathbf{A}_k, \mathbf{b}_k, \mathbf{s}_k\}$.

2. Compute the probability for each class given the data vector $\mathbf{x}$

$$p(C_k|\mathbf{x}, \theta_{1:K}) = \frac{p(\mathbf{x}|\theta_k, C_k)p(C_k)}{\sum_k p(\mathbf{x}|\theta_k, C_k)p(C_k)}. \tag{4}$$

3. Adapt the basis functions $\mathbf{A}$ and the bias terms $\mathbf{b}$ for each class. The basis functions are adapted using gradient ascent

$$\begin{aligned} \Delta\mathbf{A}_k &\propto \frac{\partial}{\partial \mathbf{A}_k} \log p(\mathbf{x}|\theta_{1:K}) \\ &= p(C_k|\mathbf{x}, \theta_{1:K}) \frac{\partial}{\partial \mathbf{A}_k} \log p(\mathbf{x}|C_k, \theta_k) \end{aligned} \tag{5}$$

Note that this simply weights any standard ICA algorithm gradient by $p(C_k|\mathbf{x}, \theta_{1:K})$. The gradient can also be summed over multiple data points.

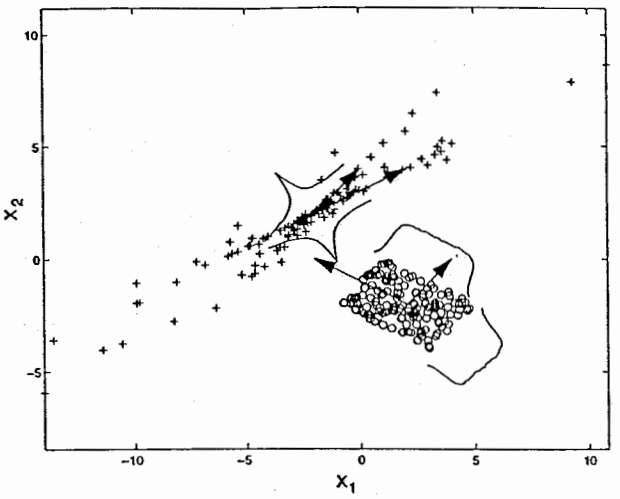

Figure 1: A simple example for classifying an ICA mixture model. There are two classes (+) and (o); each class was generated by two independent variables, two bias terms and two basis vectors. Class (o) was generated by two uniform distributed sources as indicated next to the data class. Class (+) was generated by two Laplacian distributed sources with a sharp peak at the bias and heavy tails. The inset graphs show the distributions of the source variables, $\mathbf{s}_{i,k}$, for each basis vector.

The bias term is updated according to

$$\mathbf{b}_k = \frac{\sum_t \mathbf{x}_t p(C_k|\mathbf{x}_t, \theta_{1:K})}{\sum_t p(C_k|\mathbf{x}_t, \theta_{1:K})}, \tag{6}$$

where t is the data index ($t = 1, \ldots, T$).

The three steps in the learning algorithm perform gradient ascent on the total likelihood of the data

$$p(\mathbf{x}|\theta_{1:K}) = \sum_{k=1}^{K} p(\mathbf{x}|\theta_k, C_k)p(C_k). \tag{7}$$

The extended infomax ICA learning rule is able to blindly separate mixed sources with sub- and super-Gaussian distributions. This is achieved by using a simple type of learning rule first derived by Girolami (1998). The learning rule in Lee et al. (1999b) uses the stability analysis of Cardoso and Laheld (1996) to switch between sub- and super-Gaussian regimes. The learning rule expressed in terms of $\mathbf{W} = \mathbf{A}^{-1}$, called the filter matrix is:

$$\Delta\mathbf{W} \propto \left[\mathbf{I} - \mathbf{K}\tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T\right]\mathbf{W}, \tag{8}$$

where k_i are elements of the N-dimensional diagonal matrix $\mathbf{K}$ and $\mathbf{u} = \mathbf{W}\mathbf{x}$. The unmixed sources $\mathbf{u}$ are the source estimate $\mathbf{s}$ (Bell and Sejnowski, 1995; Lee et al., 1999a). The k_i's are (Lee et al., 1999b)

$$k_i = \text{sign}\left(E[\text{sech}^2 u_i]E[u_i^2] - E[u_i \tanh u_i]\right). \tag{9}$$

The source distribution is super-Gaussian when $k_i = 1$ and sub-Gaussian when $k_i = -1$.

For the log-likelihood estimation in eq.3 the term $\log p(\mathbf{s})$ can be approximated as follows

$$\begin{aligned} \log p(\mathbf{s}) &\propto -\sum_n |s_n| && \text{super-G.} \\ \log p(\mathbf{s}) &\propto -\sum_n \log\cosh s_n - \frac{s_n^2}{2} && \text{sub-G.} \end{aligned} \tag{10}$$

Super-Gaussian densities, are approximated by a Laplacian density model; Sub-Gaussian densities are approximated by a bimodal density (Girolami, 1998). Although the source density approximation is crude it has been demonstrated that simple density models are sufficient for standard ICA problems (Lee et al., 1999b).

3. UNSUPERVISED CLASSIFICATION

To demonstrate the learning algorithm, we generated random data drawn from different classes and used the proposed method to learn the parameters and to classify the data. Figure 2 shows an example of four classes in a two-dimensional data space. Each class was generated from eq.2 using random choices for the class parameters. The task for the algorithm was to learn the four mixing matrices and bias vectors given only the unlabeled two dimensional data set. The parameters were randomly initialized. The algorithm converged in 300 iterations through the data. The arrows in figure 2 indicate the basis vectors $\mathbf{A}_k$ and the bias terms $\mathbf{b}_k$ were learned correctly for each class. Testing was accomplished by processing each instance with the learned parameters $\mathbf{A}_k$ and $\mathbf{b}_k$. The probability of the class $p(C_k|\mathbf{x}, \theta_k)$ was computed and the corresponding instance label was compared to the highest class probability. For this example, in which the classes had several overlapping areas, the classification error on the whole data set was 7.5%. The Gaussian mixture model used in AutoClass (Stutz and Cheeseman, 1994) gave an error of 8.5%. For the k-means (Euclidean distance measure) clustering algorithm, the error was 11.3%.

3.1. Iris Data Classification

To compare the proposed method to other classification algorithms, the method was applied to the classification of real data from the machine learning benchmark (Merz and Murphy, 1998). As an example, we show the classification of the well known iris flower data set. The data set (Fisher, 1936) contains 3 classes, 4 numeric attributes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two, but the other two are not linearly separable from each other. Note that from the viewpoint of the algorithm, all the data is unlabeled and learning is unsupervised. The algorithm converged after one hundred passes through the data. The classification error on this data set was 2% whereas the error using AutoClass was 3.3%. We also performed a k-means clustering which gave an error rate of 4.7%.

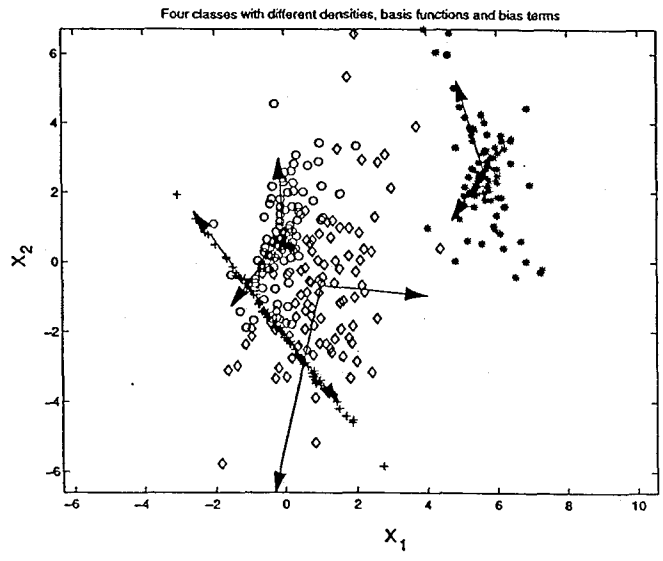

Figure 2: An example of classification of a mixture of independent components. There are 4 different classes, each generated by two independent variables and bias terms. The algorithm is able to find the independent directions (basis vectors) and bias terms for each class.

4. CONTEXT SWITCHING BETWEEN CLASSES

The ICA mixture model can be used to identify different contexts in data and classify them accordingly. Imagine the following situation: There are two people talking to each other while they are listening to music in the background. Two microphones are placed somewhere in the room that record the conversation. The conversation is in an alternating manner so that person #1 talks while person #2 listens, then person #1 listens to person #2 and so on. The mixing matrix

changes as a function of the location of the speaker. In this case, the voice of person #1 gets mixed with the background music signal with A_1 while the voice of person #2 get mixed with the music signal with A_2. Figure 3 shows the two observed channels x_1 and x_2. Each channel contains the voices of person #1 and #2 and the music signal.

The algorithm was trained on 11 seconds sampled at 8 kHz to learn two classes of ICA representations. The two basis vectors A_1 and A_2 were randomly initialized. For each gradient in eq.5 a stepsize was computed as a function of the amplitude of the basis vectors and the number of iterations.

The time course of the unmixed signals using the ICA mixture model is shown in figure 4. The top plot shows the two speech signals with correct markers indicating which speaker was talking. The bottom plot shows the time course of the background music signal.

Figure 5 (Top) shows the class conditional probability, $p(C_2|\mathbf{x},\theta_2) = 1 - p(C_1|\mathbf{x},\theta_1)$, for each sample (data vector) in the series. Note that a single sample typically does not contain enough information to unambiguously assign class membership. The intermediate values for the class probability represent uncertainty about the class membership. A threshold at $p(C_2|\mathbf{x},\theta_2) = 0.5$ can be used to determine the class membership. Using this threshold for single samples in figure 5 (Top) gave an error rate of 27.4%. This can be rectified using the a priori knowledge that a given context persists over many samples. This information could be incorporated into a more complex temporal model for $p(C_k)$, but here we use the crude but simple procedure of computing the class membership probability for an n-sample block. This value is plotted for a block size of 100 samples in figure 5 (Middle). The value provides a much more accurate estimate of class membership (6.5% error). The error rate dropped to zero when the block size was increased to 2000 samples (figure 5 (Bottom)) the correct class probabilities were recovered and matched those in figure 4 (Top).

The Signal to Noise Ratio (SNR) for the experiment with a block size of 100 samples was 20.8 dB and 21.8 dB using the context switching ICA mixture model. The standard ICA algorithm is able to learn only one unmixing matrix and the SNR using infomax (Bell and Sejnowski, 1995) was 8.3 dB and 6.5 dB respectively.

5. DISCUSSION

This new method is similar to other approaches such as the mixture density networks by Bishop (1994) in which a neural network was used to find arbitrary density functions. This algorithm reduces to the Gaussian mixture model when the source priors are Gaussian. Purely Gaussian structure, however, is rare in real data sets. Here we have used priors of the form of super-Gaussian and sub-Gaussian densities. But these could be extended as proposed by Attias (1999). The proposed model was used for learning a complete set of basis functions without additive noise. However, the method can be extended to take into account additive Gaussian noise and an overcomplete set of basis vectors (Lewicki and Sejnowski, 1998). A completely different approach to data modeling is the multidimensional ICA algorithm by Cardoso (1998) which is based on a geometric parameterization of the ICA matrices.

We have performed several experiments on benchmark data sets for classification problems. The results were comparable or improved over those obtained by AutoClass (Stutz and Cheeseman, 1994) which uses a Gaussian mixture model.

Another application of the proposed method is the automatic detection of sleep stages by observing EEG signals. The method can identify these stages due to the changing source priors and their mixing.

In Lee et al. (1999c), we show experiments with natural images using the ICA mixture model (Lewicki and Olshausen, 1998). The algorithm can be used to find efficient representations of image patterns such as text in newspapers and natural scenes. Preliminary results show that the algorithm is able to find classes so that one class encodes the natural images and the other class specializes on encoding the text segments.

6. CONCLUSIONS

The new algorithm for unsupervised classification presented here is based on a maximum likelihood mixture model using independent component analysis to model the structure of the classes. We demonstrated on simulated and real world data that the algorithm gives highly competitive classification results. We also show that the algorithm can be applied to blind source separation in nonstationary environments. The method can switch automatically between learned mixing matrices in different environments. We believe that this method provides greater flexibility in modeling structure in high-dimensional data and has many potential applications.

References

Attias, H. (1999). Blind separation of noisy mixtures: An em algorithm for independent factor analysis. *Neural Computation*, in press.

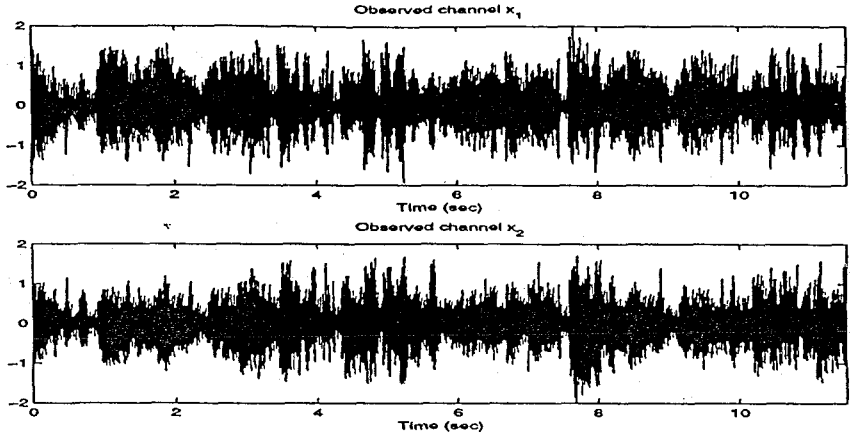

Figure 3: The two observed channels x_1 and x_2 sampled at 8 kHz. Each channel contains the voices of person #1 and #2 and the music signal.

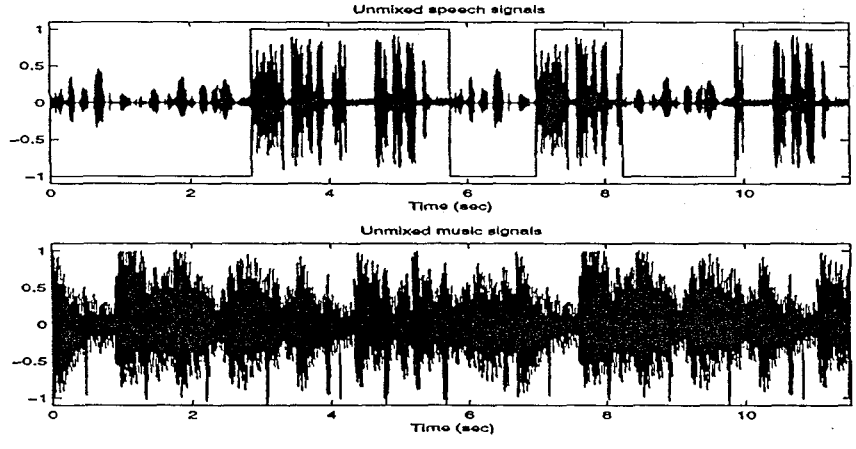

Figure 4: The time course of the unmixed signals using the mixture model and a block size of 2000 samples for estimating the class probability. (Top) The two speech signals with correct markers indicating which speaker was talking. (Bottom) The time course of the background music signal.

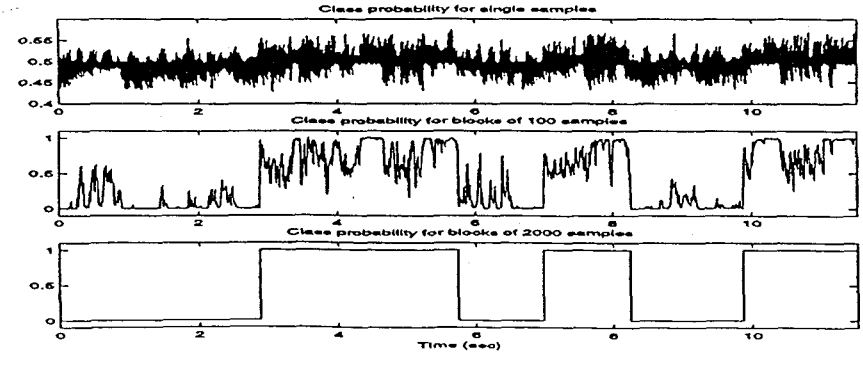

Figure 5: The class conditional probability $p(C_2|\mathbf{x}, \theta_2)$. (Top) Class probability for single samples. (Middle) Class probability for blocks of 100 samples. (Bottom) Class probability for blocks of 2000 samples.

Bell, A. J. and Sejnowski, T. J. (1995). An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7:1129–1159.

Bishop, C. (1994). Mixture density networks. *Technical Report*, NCRG/4288.

Cardoso, J.-F. (1998). Multidimensional independent component analysis. In *Proc. ICASSP'98*, volume 4, pages 1941–1944, Seattle.

Cardoso, J.-F. and Laheld, B. (1996). Equivariant adaptive source separation. *IEEE Trans. on S.P.*, 45(2):434–444.

Duda, R. and Hart, P. (1973). *Pattern classification and scene analysis.* Wiley, New York.

Fisher, R. (1936). *The use of multiple measurements in taxonomic problem.* Annual Eugenics, 7, Part II, 179-188.

Ghahramani, Z. (1994). Solving inverse problems using an em approach to density estimation. *Proceedings of the 1993 Connectionist Models Summer School*, pages 316–323.

Girolami, M. (1998). An alternative perspective on adaptive independent component analysis algorithms. *Neural Computation*, 10(8):2103–2114.

Jung, T.-P., Humphries, C., Lee, T.-W., Makeig, S., McKeown, M., Iragui, V., and Sejnowski, T. J. (1998). Extended ica removes artifacts from electroencephalographic recordings. *Advances in Neural Information Processing Systems 10*, pages 894–900.

Lee, T.-W., Bell, A. J., and Lambert, R. (1997). Blind separation of convolved and delayed sources. In *Advances in Neural Information Processing Systems 9*, pages 758–764. MIT Press.

Lee, T.-W., Girolami, M., Bell, A. J., and Sejnowski, T. J. (1999a). A unifying framework for independent component analysis. *International Journal on Mathematical and Computer Models*, in press.

Lee, T.-W., Girolami, M., and Sejnowski, T. J. (1999b). Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources. *Neural Computation*, in press.

Lee, T.-W., Lewicki, M. S., and Sejnowski, T. J. (1999c). Unsupervised classification with non-gaussian mixture models using ica. In *Advances in Neural Information Processing Systems 11*, volume in press. MIT Press.

Lewicki, M. and Olshausen, B. (1998). Inferring sparse, overcomplete image codes using an efficient coding framework. *J. Opt.Soc., A: Optics, Image Science and Vision*, submitted.

Lewicki, M. and Sejnowski, T. J. (1998). Learning non-linear overcomplete represenations for efficient coding. In *Advances in Neural Information Processing Systems 10*, pages 815–821.

Merz, C. and Murphy, P. (1998). UCI repository of machine learning databases.

Stutz, J. and Cheeseman, P. (1994). Autoclass - a bayesian approach to classification. *Maximum Entropy and Bayesian Methods*, Kluwer Academic Publishers.