ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 333-338

Independent Component Analysis for Mixture Densities

Francesco Palmieri, Alessandra Budillon,
and Davide Mattera

Dip. di Ing. Elettronica e delle Telecomunicazioni
Universita' di Napoli "Federico II" , Italy
email: {palmieri,budillon,mattera}@diesun.die.unina.it

Abstract. Independent component analysis (ICA), formulated as a density estimation problem, is extended to a mixture density model. A number of ICA blocks, associated to implicit equivalent classes, are updated in turn on the basis of the estimated density they represent. The approach is equivalent to the EM algorithm and allows an easy non linear extension of all the current ICA algorithms. We also show a preliminary test on bi-dimensional synthetic data drawn from a mixture model.

1. Introduction

The ICA technique aims to extract linear projections as statistically independent as possible. It has been proved to be a very powerful approach to the blind source separation problem, some of the most relevant references are [1], [2], [3], [4]. The separation of linearly mixed independent components, when their marginal probability density functions are known, can be formulated as a maximum likelihood estimation problem, or as a search for the linear projections that match at best the known separable density function, or as the result of an infomax criterion [2], [3]. Therefore, even when we have no knowledge of the underlying generative mechanism of our data, we can formulate the independent component analysis as a search for the linear projections that match at best a *desired* separable density function. The limitations of such a density search are clearly imposed by the linear structure of the transform, specially in dealing with natural data for which a generative model may not be known. In [8] we have generalized the idea of independent component analysis formulating it as a multidimensional density estimation (*Generalized Independent Component Analysis*, GICA). In this paper we assume a mixture density model for the data and we use a non linear transformation to achieve the desired separable density conditionally to the mixture component. The assumption on the data model is convenient because it allows an easy extension of the standard ICA approach. A set of linear transformations are adapted on line and selected on the data according to a maximum a posteriori probability criterion. The idea of modeling data as coming from a mixture density has become quite standard in the literature with the EM algorithm being a very efficient and simple method to estimate the free parameters. The extension of the ICA technique to a multi-class scenario has been recently proposed in [6] and [7], where various strategies for selecting the cluster within which performing the ICA search, are suggested. In this paper we focus on the fact that the paradigm can be seen as a density estimation and the learning algorithm appears equivalent to the EM algorithm [3] since it estimates the mixture parameters using a maximum likelihood criterion. The key idea of our algorithm is that *the posterior probabilities are estimated using the same implicit density information that the ICA blocks represent.* The multiple ICA structure resembles a competitive network and we believe it may bear strong resemblance to biological self-organizing neural networks. Once the neural network weights have been learnt, the sigmoidal outputs can be easily checked for uniformity and the reversed structure excited by random data can become a generative model of our observation space. We show some preliminary simulation results obtained on bi-dimensional synthetic data drawn from a mixture model.

2. The standard ICA framework

Consider an N-dimensional input random vector $\mathbf{x} \in \mathcal{R}^N$ and an affine transformation

$$\mathbf{y} = W\mathbf{x} + \mathbf{w}_o, \tag{1}$$

with the output vector $\mathbf{y} \in \mathcal{R}^M$. Suppose that the outputs $\mathbf{y}$ are fed into a set of sigmoidal functions $\{\phi_i(\xi), i = 1, ..., M\}$ obtained from a set of M *desired* cumulative distribution functions $\{F_{Di}(\xi), i = 1, ..., M\}$ as

$$\phi_i(\xi) = \alpha_i F_{Di}(\xi) + \beta_i, \tag{2}$$

with $\alpha_i > 0 \ \forall \ i = 1, ..., M$. Clearly $\beta_i < \phi_i(\xi) < \alpha_i + \beta_i$ and $\phi_i'(\xi) \geq 0 \ \forall \ \xi$. Define $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_M)^T$ and $z_i = \phi_i(y_i)$. We search for the projections $\{y_i\}$ that maximize the differential entropy $h(\mathbf{z})$:

$$h(\mathbf{z}) = -D_{KL}(f_{\mathbf{z}}; U(\boldsymbol{\beta}, \boldsymbol{\alpha} + \boldsymbol{\beta})) + \sum_{i=1}^{M} \log \alpha_i, \tag{3}$$

where $U(\boldsymbol{\beta}, \boldsymbol{\alpha} + \boldsymbol{\beta})$ denotes the separable uniform density $\prod_{i=1}^{M} U(\beta_i, \alpha_i + \beta_i)$ and $D_{KL}(p; q)$ is the Kullback-Leibler (KL) divergence between the densities p and q. We note that the search for the projections $\{y_i\}$ that maximize $h(\mathbf{z})$ is equivalent to looking for the projections that make the outputs $\{z_i\}$ as uniform and as independent as possible, all at the same time. If we assume that the transformation $\Phi = (\phi_1, ..., \phi_M)^T$ from $\mathbf{y}$ to $\mathbf{z}$ is invertible, i.e. we rule out degenerate density functions, we have that $f_{\mathbf{z}}(\boldsymbol{\eta}) = f_{\mathbf{y}}(\Phi^{-1}(\boldsymbol{\eta}))/|J(\Phi^{-1}(\boldsymbol{\eta}))|$, where $J = diag(\frac{\partial \phi_1}{\partial y_1}, ..., \frac{\partial \phi_M}{\partial y_M})$ is the Jacobian matrix whose determinant is

$|J| = \prod_{i=1}^{M} \alpha_i f_{Di}(y_i)$. The notation f_{Di} denotes the probability density function corresponding to F_{Di}. Therefore,

$$\begin{aligned} h(\mathbf{z}) &= E_{\mathbf{z}}[\log|J(\Phi^{-1}(\mathbf{z}))|] - E_{\mathbf{z}}[\log f_{\mathbf{y}}(\Phi^{-1}(\mathbf{z}))] \\ &= \sum_{i=1}^{M} \log \alpha_i + E_{\mathbf{y}}[\log \prod_{i=1}^{M} f_{Di}(y_i)] - E_{\mathbf{y}}[\log f_{\mathbf{y}}(\mathbf{y})] \\ &= -D_{KL}(f_{\mathbf{y}}; \prod_{i=1}^{M} f_{Di}(y_i)) + \sum_{i=1}^{M} \log \alpha_i. \end{aligned} \tag{4}$$

The search for the parameters $(\mathbf{W}, \mathbf{w}_o)$ *that maximizes* $h(\mathbf{z})$ *is equivalent to the search for an output* $\mathbf{y}$ *that has a joint density which is as close as possible to the separable desired density* $\prod_{i=1}^{M} f_{Di}(y_i)$ in the KL sense. We take this criterion as the standard ICA objective. Note how the scale and the location of the sigmoidal functions is irrelevant since the maximization affects only the first term in $h(\mathbf{z})$. Also, if the sigmoidal functions had already been chosen, maximization of $h(\mathbf{z})$ implicitly searches for the densities $\{f_{Di} = \frac{\dot{\phi}_i}{\alpha_i}\}$. The idea of maximizing the entropy of the outputs of sigmoidal function has been proposed for the logistic function by Bell and Sejnowsky in [2]. In the following we will focus on the standard case $M = N$. To derive the learning algorithm we should define the score function :

$$\begin{aligned} \mathbf{\Psi}^T &= (\psi_1, \ldots, \psi_M) \\ &= \left(-\frac{\ddot{\phi}_1(y_1)}{\dot{\phi}_1(y_1)}, \ldots, -\frac{\ddot{\phi}_M(y_M)}{\dot{\phi}_M(y_M)}\right) \\ &= \left(-\frac{\dot{f}_{D_1}(y_1)}{f_{D_1}(y_1)}, \ldots, -\frac{\dot{f}_{D_M}(y_M)}{f_{D_M}(y_M)}\right). \end{aligned} \tag{5}$$

Typical *score-functions*, or *influence functions* $\Psi_i = -\frac{f'_{D_i}}{f_{Di}}$ are the ones related to the Gaussian density $\mathcal{N}(\mu_i, \sigma_i^2)$ with $\Psi_i(\xi) = \frac{1}{\sigma_i^2}(\xi - \mu_i)$, the logistic density $f_{D_i}(\xi) = \dfrac{e^{-(\xi-\mu_i)}}{(1+e^{-(\xi-\mu_i)})^2}$ with $\Psi_i(\xi) = tanh\frac{\xi-\mu_i}{2}$, the Laplacian density $f_{D_i}(\xi) = \dfrac{1}{2\gamma_i} e^{-\frac{|\xi-\mu_i|}{\gamma_i}}$ with $\Psi_i(\xi) = \frac{1}{\gamma_i} sgn(\xi - \mu_i)$.

The free parameters of the problem are then $\mathbf{W}$ and $\mathbf{w}_o$ and since $h(\mathbf{y}) = h(\mathbf{x}) + log|\mathbf{W}|$, we can easily compute the gradients obtaining an ICA gradient ascent algorithm of the type

$$\left\{ \begin{aligned} \Delta\mathbf{W} &= \mu_1(\mathbf{W}^{-T} - \mathbf{\Psi}(\mathbf{y})\mathbf{x}^T) \\ \Delta\mathbf{w}_o &= -\mu_2 \mathbf{\Psi}(\mathbf{y}). \end{aligned} \right. \tag{6}$$

3. ICA for mixtures

Assume that the random vector $\mathbf{x} \in \mathcal{R}^N$ is distributed according to a parametric mixture density model $f_{\mathbf{x}}(\mathbf{x}, \mathbf{\Theta}) = \sum_{i=1}^{C} \pi_i f_{\mathbf{x}}(\mathbf{x}|i, \boldsymbol{\theta}_i)$, where C, the number of classes, and $\{\pi\}_{i=1,\ldots,C}$ are supposed to be known and $\Theta = (\theta_1, \ldots, \theta_C)$ are the mixture parameters. We want to learn from examples of $\mathbf{x}$ an invertible nonlinear function $\mathbf{z} = \boldsymbol{g}(\mathbf{x})$ that maps $\mathbf{x}$ into a random vector $\mathbf{z}$ whose components are as independent and as uniform as possible. If we knew which class $\mathbf{x}$ belongs to, we can apply the standard ICA paradigm and search, within each class, for a linear transformation such that $\mathbf{z}$ exhibits a separable uniform density. In the previous section we note that this is equivalent to searching for a transformation such that the linear projections are as independent and as close as possible to a set of desired marginals. As depicted in Fig. 1, consider a one-layer neural network with C blocks performing an affine transformation with parameters $(\mathbf{W}_i, \mathbf{w}_{oi})$ and a sigmoidal function $\boldsymbol{\Phi}_i = (\phi_{i1}, \ldots, \phi_{iN})^T$ obtained from a set of *desired* cumulative distribution functions $(F_{D_{i1}}, \ldots, F_{D_{iN}})$ in the following way

$$\phi_{ij}(\xi) = \alpha_{ij} F_{D_{ij}}(\xi) + \beta_{ij}, \tag{7}$$

with $\alpha_{ij} > 0$ $\forall i = 1, \ldots, C$ and $\forall j = 1, \ldots, N$. The vector $\mathbf{b} = (b_1, \ldots, b_C)^T$ is binary, only one component is high and it is the one correspondent to the class associated to the input data. So that each unit i is active at a time and it depends on the input data. This reminds a competitive network of the type *winner-takes-all.* We indicate with $\mathbf{y}_i = (y_{i1}, \ldots, y_{iN})^T$ and $\mathbf{z}_i = (z_{i1}, \ldots, z_{iN})^T$, respectively the inputs and the outputs to the sigmoids. Once the sigmoidal functions have been chosen, the ICA criterion on each block search for the separable densities $f_{D_i}(\mathbf{y}_i) = \prod_{j=1}^N f_{D_{ij}}(y_{ij}) = \prod_{j=1}^N \frac{\dot{\phi}_{ij}(y_{ij})}{\alpha_i j}$ with $i = 1, \ldots, C$. The i^{th} class membership function can be derived for self consistency as

$$p(i|\mathbf{x}) = \frac{\pi_i f_{\mathbf{y}_i}(\mathbf{W}_i \mathbf{x} + \mathbf{w}oi)\,|\mathbf{W}_i|}{\sum_{j=1}^C \pi_j f_{\mathbf{y}_j}(\mathbf{W}_j \mathbf{x} + \mathbf{w}oj)\,|\mathbf{W}_j|}. \tag{8}$$

If the independence had been accomplished one could directly compute

$$p(i|\mathbf{x}) = \frac{\pi_i \prod_{j=1}^N \frac{\dot{\phi}_{ij}(y_{ij})}{\alpha_{ij}}\,|\mathbf{W}_i|}{\sum_{j=1}^C \pi_j \prod_{l=1}^N \frac{\dot{\phi}_{jl}(y_{jl})}{\alpha_{jl}}\,|\mathbf{W}_j|}. \tag{9}$$

Then the k^{th} class is chosen if the associated posterior probability is maximum, and $b_j = \delta_{kj}$ $\forall j = 1, \ldots, C$. We have also to define a score function for each block $\boldsymbol{\Psi}_i(\mathbf{y}_i)^T = \left(-\frac{\dot{f}_{D_{i1}}(y_{i1})}{f_{D_{i1}}(y_{i1})}, \ldots, -\frac{\dot{f}_{D_{iN}}(y_{iN})}{f_{D_{iN}}(y_{iN})}\right)$ with $i = 1, \ldots, C$.

Summary of the stochastic algorithm:

(a) Initialize $\{\mathbf{W}_i(0), w_{oi}(0)\}_{i=1}^C$ to random values.
(b) Present an input $\mathbf{x}(n)$ and forward propagate it (with fixed weights), computing $\dot{\phi}_{ij}(\mathbf{y}_{ij}(n))$ and $\psi_{ij}(\mathbf{y}_{ij}(n))$ $\forall i = 1, \ldots, C$ and $\forall j = 1, \ldots, N$.
(c) Compute $p(i|\mathbf{x})$ from (9) $\forall i$ and select the winner unit k :
$\mathbf{b} = (0, .., 0, \underbrace{1}_{k^{th}}, 0, .., 0)$.

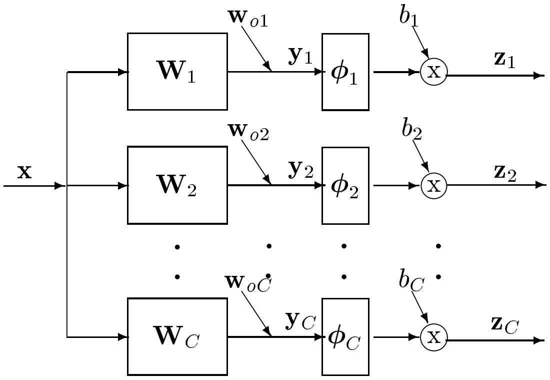

Figure 1: The network structure for ICA for mixtures

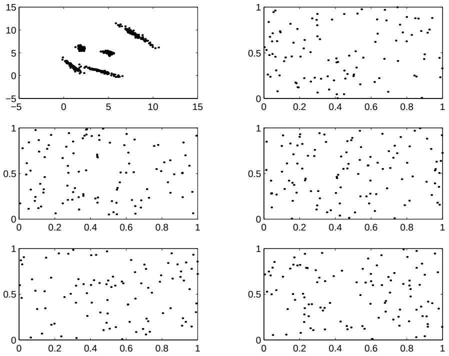

Figure 2: Input examples for a five-term logistic mixture and relative outputs

(c) Update only the parameters of the k^{th} unit according to the rule :

$$\begin{cases} \mathbf{W}_k(n+1) &= \mathbf{W}_k(n) + \mu_1(\mathbf{W}_k(n)^{-T} - \mathbf{\Psi}_k(\mathbf{y}_k(n))\mathbf{x}(n)^T) \\ \mathbf{w}_{ok}(n+1) &= \mathbf{w}_{ok}(n) - \mu_2\mathbf{\Psi}_k(\mathbf{y}_k(n)) \end{cases} \tag{10}$$

(d) Go back to step (b).

4. Simulations

We report a set of simulations obtained on a two-dimensional input. The inputs are 500 examples drawn out of a five-term logistic mixture. Figure 2 shows the input examples and the five outputs. Note how almost exact uniform distribution is achieved at the outputs. Figure 3 shows the results of the generative process, with the reversed structure fed by uniform random data. The obtained synthetic distribution is shown in (b) and it matches closely the original input distribution shown in (a).

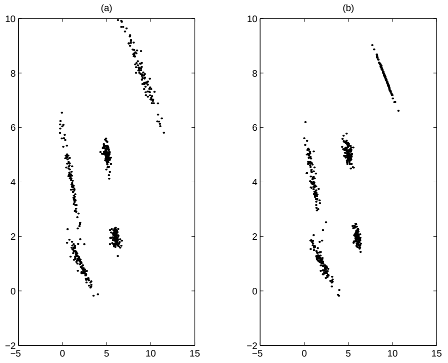

Figure 3: Input examples (a) and results of the generative model (b)

References

[1] A. Amari, A. Chichocki and H. H. Yang (1996). "A New Learning Algorithm for Blind Signal Separation," *Proc. of Neural Information Processing Systems Conf.*, NIPS 8, Ed. D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, MIT Press, 1996, pp. 757-763.

[2] A. J. Bell and T. J. Sejnowski (1995). "A Non-Linear Information Maximisation Algorithm that Performs Blind Signal Separation," *Proc. of Neural Information Processing Systems Conf.*, NIPS 7, Ed. G. Tesauro, D. S. Touretzky and T. K. Leen, MIT Press, 1995, pp. 467-474.

[3] J. F. Cardoso (1997). "Infomax and Maximum Likelihood for Blind Source Separation," *IEEE Signal Processing Letters*, Vol. 4, 1997, pp. 109-111.

[4] P. Comon (1994). "Independent Component Analysis, A New Concept?", *Signal Processing*, Vol. 36, 1994, pp. 287-314.

[5] A. Dempster, N. Laird and D. Rubin (1977). "Maximul likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, Vol. B39, pp. 1-38.

[6] J. Karhunen, and S. Malaroiu (1999). "Local Independent Component Analysis Using Clustering". *International Workshop on Independent Component Analysis* (ICA'99)), Aussoi, France, Jan. 99,in press.

[7] T-W. Lee, M.S. Lewicki and T.S. Sejnowski (1999) "ICA Mixture Models For Unsupervised Classification And Automatic Context Switching" ,*International Workshop on Independent Component Analysis* (ICA'99), Aussois, France, Jan. 99, in press.

[8] F. Palmieri, D. Mattera and A. Budillon (1999). "Multilayer Independent Component Analysis ". *International Workshop on Independent Component Analysis* (ICA'99)), Aussoi, France, Jan. 99,in press.