CONTOUR LEVEL ESTIMATION FROM GAUSSIAN MIXTURE MODELS APPLIED TO NONLINEAR BSS

Jânio Canuto, Jugurta Montalvão

Universidade Federal de Sergipe (UFS)
Núcleo de Engenharia Elétrica
São Cristóvão, Brazil, CEP. 49100-000.

ABSTRACT

Probability density function estimation from limited data sets is a classical problem in pattern recognition. In this paper we propose a reformulation of the well-known nonparametric Parzen method as a parametrically regularized Gaussian Mixture Model, from which we can easily estimate density contour level. As an application illustration to the proposed contour level estimator, we also address the Blind Source Separation problem through the analysis of contour level distortions in joint probability density functions. Finally, we use the proposed estimator to undo a nonlinear mixture of two images.

Index Terms— Parzen method, GMM, PDF Contour Level Estimation, ICA, BSS.

1. INTRODUCTION

Probability Density Function (PDF) estimation from limited data sets is a classical problem in pattern recognition, for which many approximated solutions are presented in literature [1]. In this work, we focus on PDF approximations provided by Normal Mixtures, or Gaussian Mixture Models (GMM). In these models, "good" mixture parameters (i.e. Gaussian kernel means and covariance matrices, as well as the mixture weight of each kernel) can be found in many ways, such as through the well-known Expectation-Maximization (EM) algorithm [2].

Although the EM is not the fastest algorithm for mixture optimization [3], it is usually simpler to apply, which can partially explain its widespread popularity in many application fields. However, it presents some drawbacks [4], some of them linked to likelihood computation in high-dimensional problems, which can be true in some low-dimensional problems as well [5]. In order to cope with these drawbacks, model regularization is a common solution. Indeed, model regularization imposes constraints on the Gaussian mixture composition, which increases generalization [6].

Regularization strategies can be roughly split into four categories, namely:

(I) One general approach to regularization is based on the addition of a regularization term to the unconstrained criterion function, which expresses constraints or desirable properties of solutions.

(II) For models obtained via clustering-like algorithms (including the EM, which can be loosely seen as a soft clustering algorithm [4, 7]), a straightforward regularization approach is that of averaging estimates from many independent initializations.

(III) For Mixture Models, regularization can be easily obtained by imposing constraints on the mixture component parameters (e.g. by imposing constraints or lower limits on the covariance matrix of Gaussian kernels in GMM).

(IV) Conexionist models (e.g. artificial neural networks) can also be regularized, or partially regularized by pruning [8], though it is not always explicitly referred to as a regularization procedure.

On the other hand, the nonparametric Parzen method [1, 4] can loosely be regarded as a mixture model based method with strongly-constrained mixture components (category III). The Parzen approach gives an instant PDF approximation (no iterations) and, in spite of its simplicity, it is known that, under some constrains on its window width parameter, the convergence of the estimated PDF with the actual one is guaranteed, when the number of samples tends to infinity[1, 9]. In other words, many small *isotropic* (radial basis) Gaussian kernels, with identical dispersion, can virtually approximate any PDF "shape". This corresponds to a trade from kernel complexity (elliptical kernels, for instance, typically obtained via the EM approach) to kernel number.

Although EM and Parzen approaches come from different paradigms – namely, parametric and nonparametric PDF estimation, respectively – they share a striking structural similarity, whenever the Parzen method is based on Gaussian kernels. In both cases, the actual PDF is approximated by a Mixture of Gaussians. Therefore, hereafter we will refer to estimates from both approaches as Gaussian Mixture Models (GMM).

Thanks to CNPq agency for funding.

In this work, we take advantage of the Parzen model simplicity to develop a new PDF contour level estimator. The whole estimation process includes Gaussian kernel optimization through likelihood validation, and a deterministic annealing [10] like iterative algorithm which provides a gradual improvement of the contour level estimate.

Finally, as a straightforward application to the proposed estimator, we address the blind separation of two independent signal mixtures from a very simple perspective: the geometric distortion of contour level in joint PDF.

In Section 2, the PDF estimation problem from a finite data set is addressed, whereas in Section 3 we reformulate the Parzen method as a parametrically regularized GMM . In Section 4 a new PDF contour level estimation approach from the Parzen model is proposed. Finally, in Section 5 an application of the proposed PDF contour level estimator is illustrated. This application is based on nonlinear Independent Component Analysis (ICA), from a geometric point of view.

2. PDF MODELING WITH GAUSSIAN MIXTURES

Given a data set of cardinality N, $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where $\mathbf{x}_i$ is a real-valued column vector of length D, we assume that these vectors are independent samples drawn from an unknown multivariate probability density function $p(\mathbf{x})$.

We further assume that there is a suitable parametric approximation for $p(\mathbf{x})$, given by a mixture of multivariate Gaussian functions, i.e.:

$$p(\mathbf{x}) \approx \hat{p}(\mathbf{x}|\Theta) = \sum_{i=1}^{M} \alpha_i g(\mathbf{x}|\mathbf{c}_i, \mathbf{R}_i) \tag{1}$$

where $\mathbf{\Theta} = [\alpha_1, \ldots, \alpha_M, \mathbf{c}_1, \ldots, \mathbf{c}_M, \mathbf{R}_1, \ldots, \mathbf{R}_M]$ stands for the mixture parameter vector, and

$$g(\mathbf{x}|\mathbf{c}_i, \mathbf{R}_i) = \frac{1}{(2\pi)^{D/2}|\mathbf{R}_i|^{1/2}} \exp\left[-0.5(\mathbf{x}-\mathbf{c}_i)^t \mathbf{R}_i^{-1}(\mathbf{x}-\mathbf{c}_i)\right] \tag{2}$$

corresponds to the i-th Gaussian kernel of the mixture, with mean vector and covariance matrix given by $\mathbf{c}_i$ and $\mathbf{R}_i$, respectively. We further impose that $0 \leq \alpha_i \leq 1$ and $\sum_{i=1}^{M} \alpha_i = 1$.

Accordingly, denoting the likelihood of $\mathcal{X}$ by $l(\mathbf{\Theta}) = p(\mathcal{X}|\mathbf{\Theta})$, likelihood adjustment of a Gaussian mixture model to a given PDF can be summarized as finding the optimal parameter vector, $\mathbf{\Theta}_o$, that maximizes the log-likelihood $\log(l(\mathbf{\Theta}))$, i.e.:

$$\mathbf{\Theta}_o = \arg\max_{\mathbf{\Theta}} (\log l(\mathbf{\Theta}))$$

If we do not impose any restriction on $\mathbf{\Theta}$, finding $\mathbf{\Theta}_o$ turns out to be a non-trivial optimization problem, to which the Expectation-Maximization algorithm is typically applied [2].

It is well known that Gaussian mixture density estimates are particularly problematic in high-dimensional spaces with relatively few training data sets [11], or even in some low-dimensional problems [5]. This drawback can be tackled with regularization strategies. Indeed, one particularly interesting regularization category is based on structural restrictions, because it can simplify learning algorithms as well. Accordingly, in Section 3, we reformulate the Parzen method as a parametrically regularized GMM.

3. PARZEN METHOD FROM A PARAMETRIC PERSPECTIVE

Now, let us reformulate the GMM optimization problem under very strong constraints on the parameter vector. First, we constrain the placement of Gaussian kernel centers to M randomly chosen samples from $\mathcal{X}$.

For this purpose, we randomly split the data set $\mathcal{X}$ into two disjoint subsets: the *prototyping subset*, $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M\}$ of size M, and the *optimization subset*, $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{N-M}\}$, consisting of the remaining $N - M$ samples.

Moreover, we want to use identical isotropic Gaussian kernels with uniform weights in the mixture. Therefore, we further impose that $\mathbf{R}_i = \sigma^2\mathbf{I}$ and $\alpha_i = 1/M$, respectively, for $i = 1, \ldots, M$. It clearly corresponds to the following restrictions on the parameter vector:

$$\mathbf{\Theta} = [\alpha_i = 1/M, \mathbf{c}_i = \mathbf{p}_i, \mathbf{R}_i = \sigma^2\mathbf{I}] \tag{3}$$

where $i = 1, \ldots, M$.

These restrictions lead to a Gaussian Mixture Model equivalent to that obtained by the nonparametric Parzen method, where each Gaussian kernel center, $\mathbf{c}_i$, is a sample from $\mathcal{P}$. Applying these restrictions to Equation 1 yields:

$$\hat{p}(\mathbf{x}|\sigma) = (1/M) \sum_{i=1}^{M} g(\mathbf{x}|\mathbf{p}_i, \sigma^2\mathbf{I}) \tag{4}$$

in which the the only free parameter is σ (see Equation 3). This is a single scalar parameter, and optimizing $\mathbf{\Theta}$ through likelihood maximization, in this case, is equivalent to optimizing σ, which can be done in a rather straightforward manner, by a simple exhaustive one-dimensional search, through a grid of values empirically set, according to the following algorithm:

Algorithm for σ optimization

1. **Rough variance estimation:** for each sample from $\mathcal{P}$, $\mathbf{p}_i$, the two nearest neighbors are found, $\mathbf{p}_j$ and $\mathbf{p}_k$, so that a rough i-th variance estimate is provided by $\sigma_i^2 = (\|\mathbf{x}_i - \mathbf{x}_j\|^2 + \|\mathbf{x}_i - \mathbf{x}_k\|^2)/2$. Note that $M \geq 3$.

2. **Setting the 1D likelihood optimization search grid:** the median value from all rough variance estimates is

taken, i.e.

$$\sigma_m^2 = \text{median}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_M^2)$$

from which, we set $\sigma_{min} = \sqrt{\sigma_m^2}/10$, $\sigma_{max} = 10\sqrt{\sigma_m^2}$ and $\Delta_\sigma = \sqrt{\sigma_m^2}/20$.

3. **Prototyping:** from the prototyping subset, a Gaussian Mixture Model is obtained, where each Gaussian kernel center, $\mathbf{c}_i$, is a sample from $\mathcal{P}$, according to Equation 4.

4. **Optimization:** Since the log-likelihood depends on the scalar parameter σ, according to:

$$\log(l(\sigma)) = \sum_{j=1}^{N-M} \log \hat{p}(\mathbf{v}_j|\sigma) \qquad (5)$$

where $\mathbf{v}_j$ is the j-th (independent) sample drawn from the optimization subset, we simply proceed with an exhaustive 1D search for the standard deviation σ_o that maximizes $\log(l(\sigma))$. This search is done through a finite set of values for σ, corresponding to a regular grid of real values from σ_{min} to σ_{max}, with grid interval Δ_σ.

For the Parzen method, the choice of the so called *window width*, σ_o, plays a pivotal role. Many methods for this purpose are available in the Literature. The algorithm proposed here uses the very same working principles as in the cross-validation method [12, 11, 4], but in a simpler way. Therefore, we should refer to this as a 'simple validation' method.

To provide an illustration, we consider the L-shaped joint probability density function, $p(x_1, x_2)$, shown in Figure 1.

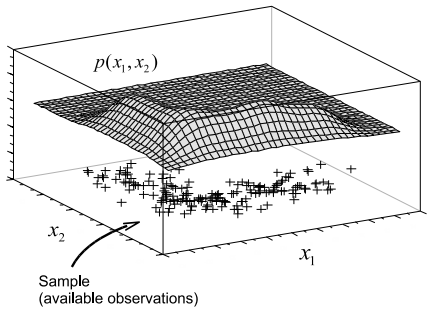

Fig. 1. An L-shaped joint probability density function

The crosses under the 3D surface represent 168 data samples, from which we estimate $p(x_1, x_2)$ from $M = 20$ samples randomly taken to be set as kernel centers, whereas the remaining 148 samples are used to optimize σ. On the left side of Figure 2, Gaussian kernels are represented by circles. On the right side, there is an illustration of the σ optimization through an exhaustive 1D search.

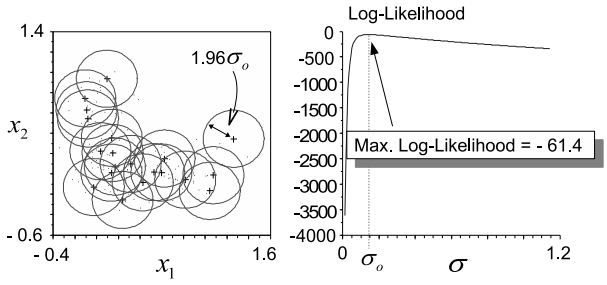

Fig. 2. Isotropic Gaussian kernels (left) and σ optimization (right)

4. CONTOUR LEVEL ESTIMATION FROM THE PARZEN MODEL

A potentially useful concern in probabilistic modeling of data sources is the estimation of contour levels of PDFs, mainly that contour level bounding the 95% confidence region. On the other hand, the simplicity of the described Parzen model, thanks to the identical radial dispersion of each Gaussian kernel, allows for a very straightforward approach to PDF contour level estimation. In this section, we briefly present this approach.

Now, let σ be a control parameter, and $\Gamma_{95}(\sigma)$ the contour level of $\hat{p}(\mathbf{x})$ (see Equation 4) inside which the density integral equals 0.95. It is easy to see that $\Gamma_{95}(\sigma)$ tends to be a circle (respectively a sphere or a hyper-sphere, for $D = 3$ or $D > 3$) of radius $r = 1.96\sigma$ whenever σ tends to infinity.

In other words, given a high enough σ, we may assume that almost all data points from $\mathcal{V}$ lie inside $\Gamma_{95}(\sigma)$, which, in turn, is almost a circle. This assumption is illustrated in Figure 3 for a $D = 2$: compare the contour points represented by '*' (a circle of radius $r = 1.96\sigma$) to the most external solid line which represents the estimated $\Gamma_{95}(\sigma)$, for a high enough σ.

Accordingly, in order to gradually approximate $\Gamma_{95}(\sigma_o)$ (our goal contour level, not necessarily a circle), we first accept a naive first guess of $\Gamma_{95}(\sigma)$ — where $\sigma = N_\sigma \sigma_o$ and N_σ is a big enough Real scale factor — given by a simple sphere of radius $1.96 N_\sigma \sigma_o$. Furthermore, continuous $\Gamma_{95}(\sigma)$ is represented by a set of points equally spaced, i.e. a circular grid of points on $\Gamma_{95}(\sigma)$. Afterwards, the value of σ is gradually reduced to σ_o, while each point on it is gradually adapted to minimize the following cost function:

$$J(\mathbf{x}|\sigma, \mathcal{P}) = (\hat{p}(\mathbf{x}|\sigma, \mathcal{P} - g(\mathbf{r}_{1.96}|\mathbf{0}, 1))^2 \qquad (6)$$

where $||\mathbf{r}_{1.96}|| = 1.96$ and $\mathbf{0}$ is a null vector. In 2D, it yields the following stochastic iteration rule:

$$\mathbf{x}_{new} \leftarrow \mathbf{x}_{old} - (\hat{p}(\mathbf{x}_{old}) - 0.023)\nabla\hat{p}(\mathbf{x}_{old}) \qquad (7)$$

where $0.023 \approx g(\mathbf{r}_{1.96}|\mathbf{0}, 1)$ for a 2D multivariate Gaussian, and $\hat{p}(\mathbf{x}_{old})$ stands for $\hat{p}(\mathbf{x}_{old}|\sigma, \mathcal{P})$.

We highlight that $\nabla\hat{p}(\mathbf{x}_{old})$ is here a simple weighted vector sum, thanks to the symmetrical and identical dispersion of Gaussian kernels in the constrained PDF Parzen model.

Figure 3 illustrates the step-by-step contour level estimation of $\Gamma_{95}(\sigma_o)$, in a joint PDF of two dependent variables, resulting from the nonlinear mixture of two independent images (see scatter plots in Figure 4). Further detail on the nonlinear mixture is provided in Section 5.

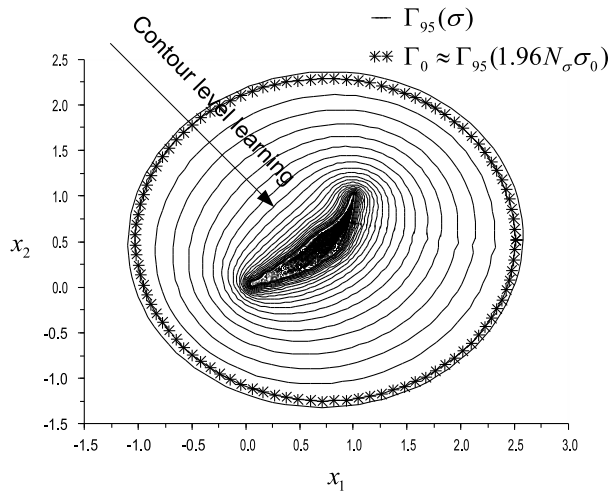

Fig. 3. Step-by-step contour level estimation

5. AN APPLICATION CASE STUDY: NONLINEAR ICA

To provide a straightforward application of our PDF contour level estimation approach, we address the problem of blind separation of independent signals from a mixture. It is well-known that linear mixtures may be separated through a de-mixing matrix, whose blind adaptation is guided by entropy based cost functions [8, 13, 14, 15].

By contrast, nonlinear mixtures demand much harder de-mixing approaches. Fortunately, for mixtures of 2 independent signals, the simple visualization of scatter plots may be a helpful tool, which in turn is closely related to joint PDF analysis of resulting variables.

For instance, if samples from two independent signals, say $s_1(n)$ and $s_2(n)$, $n = 0, 1, 2 \ldots$, follow two known laws, $p_1(s_1)$ and $p_2(s_2)$, respectively, then their joint PDF, $p(s_1, s_2)$, equals $p_1(s_1) \times p_2(s_2)$. Consequently, if the interval of values of s_i ($i = 1, 2$) corresponding to the highest values of p_i, over which the integral of p_i equals 0.95, is a continuous interval, then the corresponding $\mathcal{R}^2$ region over which the integral of $p(s_1, s_2)$ equals 0.95 is a closed contour, namely $\Gamma_{95}(\sigma_o)$. Indeed, we limit our approach to the case where $\Gamma_{95}(\sigma_o)$ is expected to be an uninterrupted single contour.

Specifically, when $p_1(s_1)$ and $p_2(s_2)$ are flat (uniform) distributions, $\Gamma_{95}(\sigma_o)$ is expected to be a square. On the other hand, linear memoryless mixtures of independent variables cause liner distortions of such contours, whereas, nonlinear mixtures yield nonlinear distortions. This is illustrated in Figures 4 and 5, with flat densities, and corresponding square contours.

From this point of view, any memoryless mixing of independent signals can be associated to a map $f : \mathcal{R}^2 \rightarrow \mathcal{W}^2$, where the expected contour, $\Gamma_{95}(\sigma_o)$, is mapped onto a new closed contour, in $\mathcal{W}^2$. Consequently, whenever an inverse map does exist, the de-mixing transformation is given by $g : \mathcal{W}^2 \rightarrow \mathcal{R}^2$.

Note that if we have at least a rough approximation to $p_1(s_1)$ and $p_2(s_2)$ (which is a common assumption in ICA), we are able to easily infer an equally rough sketch for $\Gamma_{95}(\sigma_o)$, if the sources are independent. For instance, independent images with flat PDFs produce square contours, whereas independent speech signals with picked densities around zero produce cross-like ('+') contours $\Gamma_{95}(\sigma_o)$. Let Γ_I stand for this rough sketch, supposedly available whenever the signal source laws are known (or guessed).

Then, one possible way to find g from samples is to look for a space transformation that maps back the distorted contour (from the mixture joint PDF) into the rough sketched contour Γ_I (from the joint PDF of the independent sources). Clearly, this approach has its application limited to smooth maps (i.e. linear mixtures or soft nonlinear mixtures), where space distortions inside the contour level are well represented by deformations along its border. The whole de-mixing process can be summarized as follows:

a) From $\mathcal{X}$ (samples from the mixed sources), a PDF model is estimated according to Section 3.

b) The contour level $\Gamma_{95}(\sigma_o)$ of the mixture is gradually estimated, from a first radial contour guess, as illustrated in Figure 3.

c) The contour curvature of $\Gamma_{95}(\sigma_o)$ is computed and compared to the *expected* curvature of Γ_I. A dynamic warping algorithm [1] is then applied to find the "best" correspondence between points from the two contours (see illustration in Figure 5).

d) The corresponding points from $\Gamma_{95}(\sigma_o)$ and Γ_I are given as input and target, respectively, to a adjustable parametric nonlinear mapper. For simplicity, we choose a classic Multilayer Perceptron (MLP) Neural Network (NN) to be this nonlinear mapper [8]. Furthermore, in order to limit g to a soft nonlinear mapping, we limited the number of hidden neurons to 2 and, in our experiments, we trained the NN, through the backpropagation algorithm, with very low initial weights. This was done to induce the learning of quasi-linear mappings (i.e. smooth mappings).

Finally, the trained NN — trained to map back $\Gamma_{95}(\sigma_o)$

to Γ_I — is expected to perform the inverse of the nonlinear mixture f. Therefore, from the PDF contour level analysis, a de-mixing device candidate is provided, and "good" estimates to independent signals, s_1 and s_2, correspond to the output of the NN — not necessarily in the same order — when the mixed signals, x_1 and x_2, are given as inputs.

Figure 6 illustrates this process with two 256x256 graylevel images as independent signals. In this illustration, signals correspond to pixel graylevels, from 0 to 1, whereas the nonlinear mixture is given by:

$$x_1(n) = \frac{\tanh(1.2s_1(n) + 0.8s_2(n)) - 0.1001}{0.8640}$$

$$x_2(n) = \frac{(5s_1(n) + 7s_2(n)) - 0.0477}{11.4276}$$

where $n = 1, 2, \ldots, 65536$.

To reduce the computational burden, the whole set of 2D points from the mixture was randomly subsampled by 10. This is to say that $\mathcal{X}$ corresponds to a set of only 6,554 samples.

Afterwards, according to Section 3, $\mathcal{X}$ was split into two disjoint subsets, $\mathcal{P}$ and $\mathcal{V}$, both with 3,277 samples.

To provide some visual comparison, Figures 6 and 7 present, respectively, the "de-mixed" images from the proposed approach and the well-known Fast ICA algorithm[13].

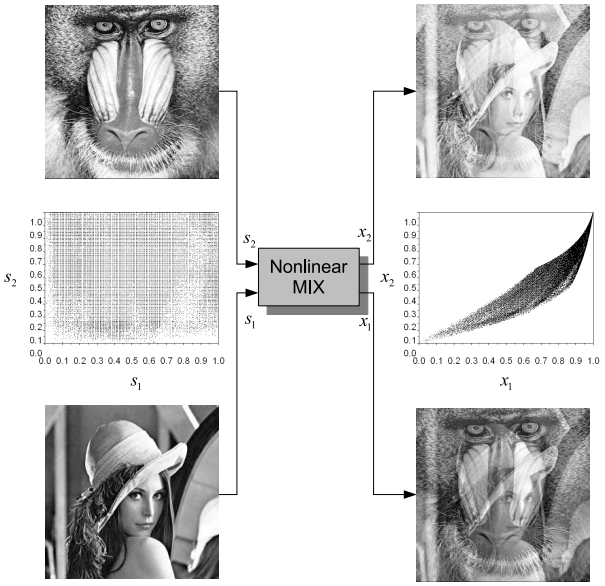

Fig. 4. Nonlinear mixture of two independent digital images

6. CONCLUSIONS

A new approach to PDF contour level estimation was presented, along with an illustrative example of how it can be applied to the Blind Source Separation of nonlinear mixtures.

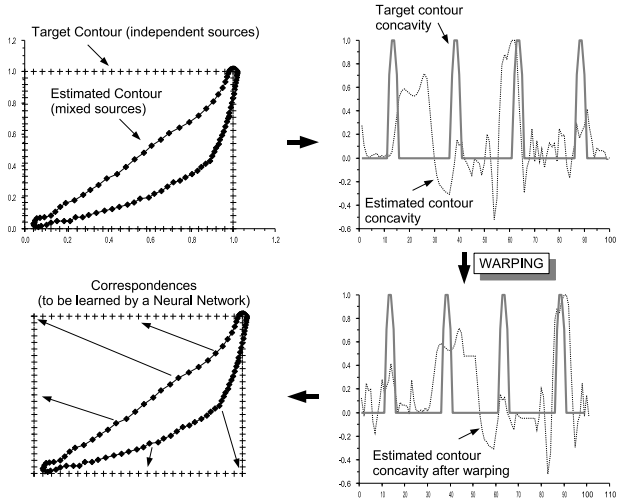

Fig. 5. Dynamic warping between contour levels

Unfortunately, the source separation approach seems to be limited to mixtures of only two independent sources, otherwise warping algorithms for nonlinear alignment of surfaces (or even hyper-surfaces) must be considered. On the other hand, the PDF contour level estimation approach, illustrated here through the gradual shrinking of a circle, in 2D, is not limited to the 2D case. Indeed, it can be easily extended to higher dimensions, which will be addressed in the sequel of this preliminary work.

Another important point to be addressed in the future is that the success of the proposed contour level estimator strongly depends on a very questionable assumption, according to which the contour level is almost a circle (a sphere or even a hypersphere) for a given "high enough" σ. Obviously, such an approximation depends on the actual data dispersion, and a numerical test must take place here. Indeed, a simple test would be the comparison between the first spherical/ circular contour guess and the first contour estimation (found after algorithm convergence under the highest σ value). If σ is really high enough, we should obtain a small distortion, otherwise, the initial σ must be increased until this test holds.

This paper presents preliminary results from a recently started research study. Accordingly, it presents more illustrations than solid theoretical results. Nonetheless, it points to some interesting potential ways for developing new strategies for nonlinear BSS, for instance. It is worth nothing that, in the presented case study, with just two images, in spite of its simplicity, it is clear that de-mixing can only be achieved by a nonlinear mapping. Furthermore, through visual inspection of the resulting images, we claim that the proposed approach did the job quite satisfactorily, even though it does not use any information-based cost-function, as is usual in Independent Component Analysis.

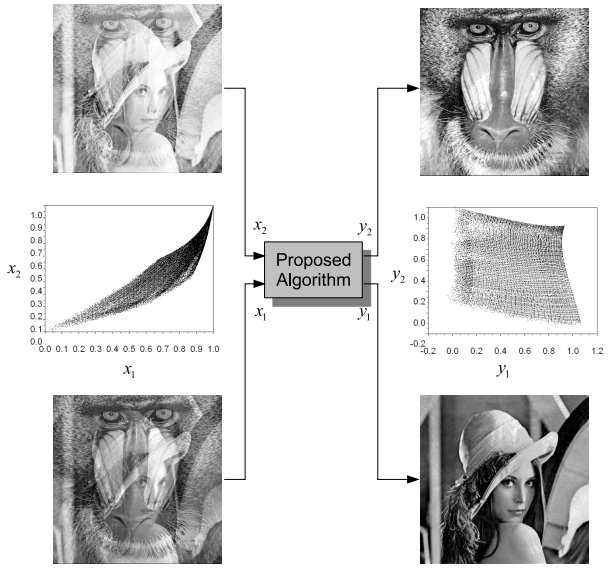

Fig. 6. Mixture de-mixing with the Multilayer Perceptron trained to map PDF contours

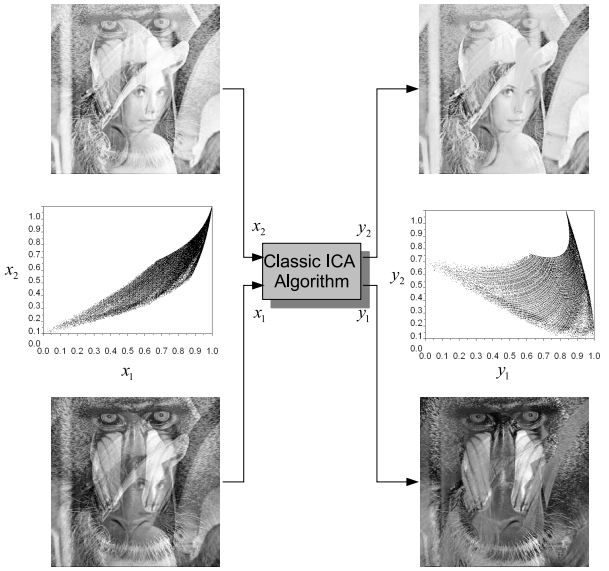

Fig. 7. Mixture processing with the classic Fast ICA algorithm

7. REFERENCES

[1] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley-Interscience, New York, 1973.

[2] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data using the em algorithm," *J Royal Stat Soc.*, vol. 39, pp. 1–38, June 1977.

[3] D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*, John Wiley & Sons, 1985.

[4] A. Webb, *Statistical Pattern Recognition*, Wiley, 2 edition, 2002.

[5] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.

[6] J. Larsen, *Design of Neural Network Filters*, Ph.D. Thesis, 2 edition, Jan. 1996.

[7] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 1999.

[8] S. Haykin, *Neural Networks, A Comprehensive Foundation*, Prentice-Hall, Englewood Cliffs, USA, 2 edition edition, 1999.

[9] A. Artes-Rodriguei J. M. Leiva-Murillo, "A fixed-point algorithm for finding the optimal covariance matrix in kernel density modeling," *ICASSP 2006 Proceedings*, vol. 5, pp. V–V, 2006.

[10] K. Rose, "Deterministic annealing for clustering compression, classification, regression and related optimization problems," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2210–2239, December 1998.

[11] D. Ormoneit and V. Tresp, *Improved Gaussian Mixture Density Estimates Using Bayesian Penalty Terms and Network Averaging, in Neural Information Processing Systems*, vol. 8, The MIT Press, 1996.

[12] R. P. W. Duin, "On the choice of smoothing parameters for parzen estimators of probability density functions," *IEEE Trans. on Computers*, vol. 25, pp. 11751179, 1976.

[13] A. Hyvrinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, pp. 411430, 2000.

[14] J. F. Cardoso, "Blind signal separation: statistical principles," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 20092025, October 2000.

[15] S. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separation," *Advances in Neural Information Processing Systems*, vol. 8, pp. 757763, 1996.