Learning Hierarchies from ICA Mixtures
Addisson Salazar, Jorge Igual, Luis Vergara, Arturo Serrano

Abstract—This paper presents a novel procedure to classify data from mixtures of independent component analyzers. The procedure includes two stages: learning the parameters of the mixtures (basis vectors and bias terms) and clustering the ICA mixtures following a bottom-up agglomerative scheme to construct a hierarchy for classification. The approach for the estimation of the source probability density function is non-parametric and the minimum kullback-Leibler distance is used as a criterion for merging clusters at each level of the hierarchy. Validation of the proposed method is presented from several simulations including ICA mixtures with uniform and Laplacian source distributions and processing real data from impact-echo testing experiments.

I. INTRODUCTION

PATTERN analysis and recognition are processes familiar to many human activities. They imply to separate data into self-similar areas and analyzing each of these clusters under some application framework. Intelligent signal processing algorithms provide an important tool to support automatic pattern recognition, gain insight into problem solving, and complement expert decision making [1-4]. Usually those algorithms make assumptions about data generating process, for instance, modeling the data as a mixture of data generators or analyzers, considering that each generator produces a particular group of data. The most used data mixture model is the well known Gaussian Mixture Model that assumes each generator is a Gaussian density. The field of GMM applications is wide but many clustering problems do not fit the assumption of Gaussian density or the problem high feature dimensionality makes no suitable to use GMM [1,5,6].

Recently has emerged the mixture of Independent Component Analyzers [7,8] as a flexible model to generate arbitrary data densities using mixtures of different distributions for the components [9,10]. In order to span more applications, new versions of ICA mixtures algorithms have loosened up assumptions about component distributions and propose calculating the components in a non-parametric manner [11]. In addition, more general approaches for the procedure to find the cluster structure of ICA analyzers, relaxing the independence hypothesis of ICA in order to analyze more complex problems, and trying to derive physical interpretations, has been proposed [12-14].

The contribution of the present paper is a new procedure to classify data from mixtures of independent component analyzers. This method does not consider any assumption on the probability density function of the sources and apply a non-parametric approach to estimate the parameters of the ICA mixture (basis vectors and bias terms). A clustering algorithm is applied on the estimated parameters and the data to merge the ICA mixtures in a bottom-up agglomerative scheme. The symmetric Kullback-Leibler distance [15] is used to select the grouping of the clusters.

We validate the procedure with several simulations, varying the number and kind of distributions of the sources, the number of ICA mixtures and using different supervision ratios in the learning stage of the parameters. In addition, we demonstrate real application of the proposed method to automatic classification of specimens tested by non-destructive impact-echo testing [16]. There are few references of applying ICA in impact-echo signals [17].

The underlying problem of the experiments was to obtain automatically a suitable hierarchical classification of parallelepiped-shape specimens of aluminum alloy with different defective conditions. The pieces were one-defect and multiple-defects, and the defects consisted in holes and cracks with different spatial orientations.

The following sections contain the detailed algorithms and computations of the procedure, results, and conclusions.

II. LEARNING MIXTURES OF INDEPENDENT COMPONENT ANALYZERS

Normally in mixture of ICAs, the observation vectors \( \mathbf{x}_k \) corresponding to a given class \( C_k, (k = 1...K) \) are modeled as the result of applying a linear transformation \( \mathbf{A}_k \) to a vector \( \mathbf{s}_k \) (sources), whose elements are independent random variables, plus a bias vector \( \mathbf{b}_k \), i.e.

\[
\mathbf{x}_k = \mathbf{A}_k \mathbf{s}_k + \mathbf{b}_k \quad k = 1...K
\]  

Optimum classification of a given feature vector \( \mathbf{x} \) of unknown class, consists in selecting the class \( C_k \) having maximum conditional probability \( p(C_k / \mathbf{x}) \). Applying Bayes theorem we obtain:

\[
p(C_k / \mathbf{x}) = \frac{p(\mathbf{x} / C_k) \cdot P(C_k)}{p(\mathbf{x})} = \frac{p(\mathbf{x} / C_k) \cdot p(C_k)}{\sum_{l=1}^{K} p(\mathbf{x} / C_l) \cdot p(C_l)}
\]  

If \( \mathbf{x} \) is a feature vector corresponding to class \( C_k \), then...
\[ p(x/C_i) = \frac{\det A_i^{-1} [s_i]}{\sum_{i' \neq i} \det A_{i'}^{-1} [s_{i'}]} \]

where \( s_i = A_i^{-1} (x - b_i) \). Considering equations (3) and (2), we may write

\[ p(C_i / x) = \frac{\det A_i^{-1} [s_i]}{\sum_{i} \det A_i^{-1} [s_i]} \]  

(4)

Thus, given a feature vector \( x \), we have to compute the corresponding source vectors \( s_i, (k = 1 \ldots K) \), from \( s_i = A_i^{-1} (x - b_i) \), to finally select the class having the maximum computed value \( \det A_i^{-1} [s_i] \) (note that the denominator in (4) is not depending on \( k \), so that it does not influence the maximization of \( p(C_i / x) \)).

To make the above computation, we need knowledge of \( A_i^{-1} \), \( b_i \) (to compute \( s_i \) from \( x \)) and of the multidimensional pdf of the source vectors for every class (to compute \( p(s_i) \)).

Two assumptions will be considered to solve this problem. First, that the elements of \( s_i \) are independent random variables (ICA assumption), so that the multidimensional pdf can be factored into the corresponding marginal pdf’s of every vector element. The second assumption is that there is available a set of independent feature vectors (learning vectors) represented by matrix \( X = [x^{(1)} \ldots x^{(N)}] \). We consider a hybrid situation where some of the learning vectors have known class (supervised learning), meanwhile others are of unknown class (unsupervised learning). Actually, we are going to consider a more general case in which knowledge of \( p(C_i / x) \) could be available for some subset of \( k-n \) pairs (supervised learning) and unknown for the rest (unsupervised learning).

If we assume a non-parametric model for \( p(s_i, (k = 1 \ldots K)) \), we can estimate the source pdf’s from a set of training samples obtained from the original set \( X = [x^{(1)} \ldots x^{(N)}] \), by applying \( s_i^{(n)} = A_i^{-1} (x^{(n)} - b_i) \) \((n = 1 \ldots N, k = 1 \ldots K)\). Hence we can concentrate in the optimum estimation of \( A_i^{-1} \), \( b_i \) \((k = 1 \ldots K)\). Let us define \( W_i = A_i^{-1} \) to ease the notation.

We need a cost function definition to obtain optima estimates of the unknowns. In a probability context it is usual to maximize the log-likelihood of the data, i.e., to select those unknowns maximizing the probability of the training set conditioned to the unknowns. As we have assumed independent feature vectors in the training set, we may write the log-likelihood in the form:

\[ L(X / \Psi) = \log p(X / \Psi) = \sum_{n=1}^{N} \log p(x^{(n)} / \Psi) \]  

(5)

where \( \Psi \) is a compact notation for all the unknown parameters \( W_i, \ b_i \) for all the classes. In a mixture model, the probability of every available feature vector can be separated into the contributions due to each class.

The general algorithm (including non-parametric source pdf estimation, supervised-unsupervised learning, and possibility of selecting a particular ICA algorithm) is

0. Initialize \( i = 0 \), \( W_0(0), \ b_0(0) \).
1. Compute \( s_i^{(n)}(i) = W_i(0)(x^{(n)} - b_i(0)) \) \((k = 1 \ldots K, n = 1 \ldots N)\)
2. Directly use \( p(C_i / x^{(n)}, \Psi(i)) = p(C_i / x^{(n)}) \)
   for those \( k-n \) pairs with knowledge about \( p(C_i / x^{(n)}, \Psi) \).
   Compute \( p(C_i / x^{(n)}, \Psi(i)) = \frac{\det W_i(i)[s_i^{(i)}]}{\sum_{i} \det W_i(i)[s_i^{(i)}]} \) \((k = 1 \ldots K)\).
   for the rest of \( k-n \) pairs.
3. Use \( p(x^{(n)}) = \sum_{m=1 \ldots M} \frac{1}{1+x^{2}} |x^{(n)} - \alpha \cdot \Delta W_k(i)| k = 1 \ldots K \) to estimate the marginals \( p(s_i^{(n)}(i)) \).
4. Compute \( \Delta W_k(i) \) using
   \[ \Delta W_k(i) = \sum_{n=1}^{N} \frac{1}{h} \left[ \frac{f(s_i^{(n)}(i))}{s_i^{(n)}(i)} \right] W_n(i) \cdot p(C_i / x^{(n)}, \Psi(i)) \]
   Use \( f(s_i^{(n)}(i)) = \frac{1}{h} \sum_{n=1}^{N} e^{-\frac{1}{h} (s_i^{(n)}(i))^2} - s_i^{(n)}(i) \) to estimate \( f(s_i^{(n)}(i)) \).
   Actualize \( b_i(k + 1) = b_i(k) + \beta \cdot \Delta b_i(k) \) \((k = 1 \ldots K)\), or simply re-estimate \( b_i(k + 1) = \frac{\sum_{n=1}^{N} x^{(n)} p(C_i / x^{(n)}, \Psi(i))}{p(C_i / x^{(n)}, \Psi(i))} k = 1 \ldots K \).
5. Go back to step 1, with the new values \( W_i(k + 1), \ b_i(k + 1) \) and \( i \rightarrow i + 1 \).

III. HIERARCHICAL ICA MIXTURES

This describes the procedure to cluster the ICA mixtures obtained for the non-parametric algorithm explained in the previous section. The procedure follows a bottom-up agglomerative scheme for merging the mixtures.
The conditional probability density of \( \mathbf{x} \) for cluster \( C_h^k, k = 1, 2, ..., K - h + 1 \) in layer \( h = 1, 2, ..., K \) is \( p(x / C_h^k) \).

At the first level, \( h = 1 \), it is modelled by the \( K \)-ICA mixtures, i.e., \( p(x / C_1^k) \) is:

\[
p(x / C_h^k) = \left| \det A_i^{-1} \right| p(\mathbf{s}_x) = A_i^{-1}(\mathbf{x} - \mathbf{b}_i) \quad (6)
\]

At each consecutive level, two clusters are merged according to some minimum distance measure until we reach at level \( h = K \) only one cluster.

As distance measure we use the symmetric Kullback-Leibler divergence between the ICA mixtures. It is defined for the clusters \( u, v \) by:

\[
D_{KL}(C_h^u, C_h^v) = \int p(x / C_h^u) \log \frac{p(x / C_h^u)}{p(x / C_h^v)} dx + \int p(x / C_h^v) \log \frac{p(x / C_h^v)}{p(x / C_h^u)} dx.
\]

For layer \( j = 1 \), from (7), we can obtain (we write \( p_{s_x}(x) = p(x / C_h^k) \) and omit the superscript \( h = 1 \) for brevity):

\[
D_{KL}(C_x, C_v) = D_{KL}(p_{s_x}(x) // p_{s_v}(x)) = \int p_{s_x}(x) \log \frac{p_{s_x}(x)}{p_{s_v}(x)} dx + \int p_{s_v}(x) \log \frac{p_{s_v}(x)}{p_{s_x}(x)} dx.
\]

where, imposing the independence hypothesis and supposing that both clusters have the same number of sources \( M \) for simplicity:

\[
p_{s_x}(x) = \prod_{i=1}^M p_{s_{x_i}}(s_{x_i}) = A_i^{-1}(x - b_i), \quad s_{x_i} = A_i^{-1}(x - b_i) \quad (9)
\]

The pdf of the sources is approximated by a non-parametric kernel-based density for both clusters:

\[
p_{s_x}(s_{x_i}) = \sum_{a, h} e^{-\frac{1}{2}(\frac{s_{x_i} - s_{x_i}(a)}{h})^2}, p_{s_v}(s_{x_i}) = \sum_{a, h} e^{-\frac{1}{2}(\frac{s_{x_i} - s_{x_i}(a)}{h})^2} \quad (10)
\]

where again for simplicity we have assumed the same kernel function with the parameters \( a, h \) for all the sources and the same number of samples \( N \) for each one. Note that this corresponds to a mixture of Gaussians model where the number of Gaussians is maximum (one for every observation) and the parameters are equal. Reducing to standard mixture of Gaussians with different number of components and priors for every source does not help in order to compute the Kullback-Leibler distance because there is not analytical solution to it. Therefore, we prefer to maintain the non parametric approximation of the pdf in order to model more complex distributions than a mixture of a small finite number of Gaussians, such as three or four.

The symmetric Kullback-Leibler distance between the clusters \( u, v \) can be expressed such as:

\[
D_{KL}(p_{s_x}(x) // p_{s_v}(x)) = -H(x_a) - H(x_b) - \int p_{s_v}(x) \log p_{s_x}(x) dx - \int p_{s_x}(x) \log p_{s_v}(x) dx.
\]

where \( H(x) \) is the entropy, defined as \( H(x) = -E[\log p_x(x)] \). To obtain the distance, we have to calculate the entropy for both clusters and the cross-entropy terms \( E_x[\log p_x(x)], E_x[\log p_y(x)] \).

The entropy for the cluster \( u \) can be calculated through the entropy of the sources of that cluster considering the linear transformation of the random variables and their independence (9):

\[
H(x_a) = \sum_{i=1}^M H(s_{x_i}) + \log |\det A_i|.
\]

The entropy of the sources can not be analytically calculated. Instead, we can obtain a sample estimate of it, \( \hat{H}(s_{x_i}) \), using the training data. Denote the \( i-th \) source obtained for the cluster \( u \) by \( \{ s_{x_1}, s_{x_2}, ..., s_{x_N(Q)} \} \). The entropy can be approximated as follows:

\[
\hat{H}(s_{x_i}) = -\hat{E}[\log p_{s_{x_i}}(s_{x_i})] = -\frac{1}{Q} \sum_{n=1}^Q \log p_{s_{x_i}}(s_{x_i}(n))
\]

\[
p_{s_{x_i}}(s_{x_i}(n)) = \sum_{a, h} e^{-\frac{1}{2}(\frac{s_{x_i}(n) - s_{x_i}(a)}{h})^2}.
\]

The entropy of \( H(x_a) \) is obtained analogously:

\[
H(x_b) = \sum_{i=1}^M H(s_{x_i}) + \log |\det A_i| = \sum_{i=1}^M \hat{H}(s_{x_i}) + \log |\det A_i|.
\]

\[
\hat{H}(s_{x_i}) = -\frac{1}{Q} \sum_{n=1}^Q \log p_{s_{x_i}}(s_{x_i}(n))
\]

\[
p_{s_{x_i}}(s_{x_i}(n)) = \sum_{a, h} e^{-\frac{1}{2}(\frac{s_{x_i}(n) - s_{x_i}(a)}{h})^2}.
\]

with \( \hat{H}(s_{x_i}) \) defined analogously to (13).

Once the entropy is computed, we have to obtain the cross-entropy terms. After some operations and considering the relationships \( x = A_s s_x + b_s, x = A_s s_x + b_s \) and
\[ s_i = A_i^{-1} \{ A_i s_i + b_i - b_i \}, \]

the independence of the sources, and that the samples for clusters \( u, v \) follow the corresponding distribution \( \{ s_i (1), s_i (2), \ldots, s_i (Q) \} \), \( i = 1, \ldots, M \), \( \{ s_j (1), s_j (2), \ldots, s_j (Q) \} \), \( j = 1, \ldots, M \) the following computation is obtained:

\[
D_{ij}(p_u (x) // p_v (x)) = - \sum_{n=1}^{Q} \hat{H}(s_i (n)) - \sum_{n=1}^{Q} \hat{H}(s_j (n))
\]

\[
\hat{H}(s_i (n)) = \frac{1}{Q} \sum_{a=1}^{Q} \log p_a^\infty (s_i (n)),
\]

where \( p_a^\infty (s_i (n)) = \sum_{x=1}^{r} \frac{1}{s_i (n)} \).

\[
\hat{H}(s_j (n)) = \frac{1}{Q} \sum_{a=1}^{Q} \log p_a^\infty (s_j (n)),
\]

where \( p_a^\infty (s_j (n)) = \sum_{x=1}^{r} \frac{1}{s_j (n)} \).

As we can observe, the similarity between clusters depends not only on the similarity between the bias term, but the similarity between the distributions and the mixing matrices.

Once the distances are obtained for all the clusters, the two clusters with minimum distance are merged in level \( h = 2 \). This is repeated in every step of the hierarchy until we reach one cluster in the level \( h = K \). To merge cluster in level \( h \) we can calculate the distances from the distances of level \( h - 1 \). Suppose that from level \( h - 1 \) to \( h \) the clusters \( C_{a}^{h-1}, C_{b}^{h-1} \) are merged in cluster \( C_{a}^{h} \). Then, the density for the merged cluster at level \( h \) is:

\[
p_h (x / C_{a}^{h}) = \frac{p_{a,i} (C_{a}^{h-1}) p_{b,i} (x / C_{b}^{h-1}) + p_{b,i} (C_{b}^{h-1}) p_{a,i} (x / C_{a}^{h-1})}{p_{a,i} (C_{a}^{h-1}) + p_{b,i} (C_{b}^{h-1})}.
\]

where \( p_{a,i} (C_{a}^{h-1}), p_{b,i} (C_{b}^{h-1}) \) are the priors or proportions of the clusters \( u, v \) at level \( h - 1 \). The rest of terms are the same in the mixture model at level \( h \) that at level \( h - 1 \). The only difference from one level to the next one in the hierarchy is that there is one cluster less and the prior for the new cluster is the sum of the priors of its components and the density the weighted average of the densities that are merged to form it. Therefore, the estimation of the distance at level \( h \) can be done easily starting from the distances at level \( h - 1 \) and so on until level \( h = 1 \). Consequently, we can calculate the distances at level \( h \) from a cluster \( C_{a}^{h} \) to a merged cluster \( C_{a}^{*} \) obtained by the agglomeration of clusters \( C_{a}^{h-1}, C_{b}^{h-1} \) at level \( h - 1 \). As the distance to its components weighted by the mixing proportions:

\[
D_{a}(p_a (x / C_{a}) // p_b (x / C_{b})) = p_{a,i} (C_{a}^{h-1}) D_{a}(p_a (x / C_{a}^{h-1}) // p_{a,i} (x / C_{a}^{h-1}))

+ p_{b,i} (C_{b}^{h-1}) D_{a}(p_b (x / C_{b}^{h-1}) // p_{b,i} (x / C_{b}^{h-1}))
\]

\[
= \frac{p_{a,i} (C_{a}^{h-1}) + p_{b,i} (C_{b}^{h-1})}{p_{a,i} (C_{a}^{h-1}) + p_{b,i} (C_{b}^{h-1})}.
\]

IV. RESULTS

A. Simulations

Fig. 1 shows ten different ICA mixtures, each generated by two independent variables and bias terms. The independent distributions are Laplacians and uniforms. The figure shows an example of how first the algorithm learned the basis vectors and bias terms for each class (Fig. 1a) at level 1, and then uses these estimations in order to create a hierarchical tree of the clusters. The second plot shows the merged clusters at level 5, where 6 clusters remain. The closest clusters are merged. In the third plot there are only 2 clusters remaining. Here we can see how not only the relative bias between clusters is used to merge them, but also similar distributions are merged together. Fig. 1d shows the resulting dendrogram with the distances at which the clusters have been merged.

In order to obtain a good performance of the agglomerative clustering, the bandwidth parameter \( h \) in equation 15) has to be adjusted to avoid the distances are thresholded to one value. There is a compromise between the resolution of the distances and the range of them. Thus we chose a high bandwidth.

To determine a quality criterion that allows knowing the optimum hierarchical level, the partition and entropy coefficient were estimated [17]. Fig. 3 shows the evolution of those coefficients through the levels of clustering of Fig 1. The best number of clusters is at level 5 (6 clusters, Fig. 1b).

The partition coefficient and the partition entropy both tend towards monotone behaviour depending on the number of clusters. So as to find the "best" number of clusters one chooses the number where the entropy value lies below the rising trend and the value for the partition coefficient lies above the falling trend. On viewing the curve of all the
Fig. 1. An example of hierarchical classification with ten ICA mixtures. On the upper left it is shown the distributions and the basis vectors for each class. Fig. 1b and 1c show the construction of higher-level clusters. Fig. 1d shows the dendrogram with the KL-distances between clusters at each merge level.

connected values, this point can be identified as a kink ("elbow criterion"). The best partitioning of the clusters applies at that point with a value of \( h \) to get the highest cluster differentiation (maxima of inter-clusters mean distances) with good homogeneity within cluster members (minima of distances between cases and centroids).

In order to measure the performance of the classification as a non-parametric density estimation algorithm, we made 400 MonteCarlo simulations with toy data like the distributions showed in the example of Fig. 1. The algorithm was executed varying the number of mixtures, distributions, and supervision ratio. Fig 2 shows the classification error for simulations for different number of ICA mixtures (hierarchical levels). We can note the percentage of error in classification becomes very low with a little of supervision.

B. Experiments

The experiments to test the proposed method consisted in 1200 executions of impact-echo testing [18] on various specimens of defective parallelepiped-shape materials of Aluminium alloy series 2000. Dimensions of the pieces were 0.07x0.05x0.22 m. (width, height and length) and supported to one third and two thirds of the block length (direction \( z \)). The material was excited by an impact and its response was measured by a multichannel system of sensors (accelerometers), see Fig. 4.

ICA has been demonstrated as a suitable technique for applying in Non Destructive Testing by impact-echo [19], where the transfer functions between the impact point and the defects in the material are modelled as sources for blind source separation. In the present experiments, the objective was to obtain automatically a suitable three-level hierarchical classification of one-defect and multiple-defect pieces. The defects consisted in holes in the form of 10 mm. \( \Phi \) cylinders, and cracks in the form of 5 mm. parallelepipeds.

The highest level of the hierarchy (material condition) divides the pieces in two classes: many defects and one defect. The medium hierarchical level (kind of defect) separates the specimens in three classes: many defects, one-hole and one-crack. The lowest level of the hierarchy (defect orientation) splits the materials in five classes: many defects, one-hole in axis-x, one-hole in axis y, one crack in plane xy, and one crack in plane \( zy \).

The equipment used in measuring was composed of: an instrumented hammer 084A14 PCB, 7 mono-axial accelerometers 353B17 PCB, a data acquisition module NI 6067E, a ICP signal conditioner F482A18 and a notebook for signal processing and control, see Fig.5. The sample frequency in signal acquisition was 100,000 kHz, and observation time recorded was 50 ms.

From measured signals, the following parameters were extracted: principal frequency, principal frequency amplitude and attenuation, centroid frequency, signal power, attenuation curve initial value and signal attenuation. Those 7 parameters were calculated for the 7 channels of the accelerometers and normalized by the impact signal.
General results of the application of the hierarchical classification procedure in experiments, using 0.3 as supervision ratio, are shown in Fig. 6. High percentages of success in classification obtained by the proposed procedure are showed for the different classes at different levels of the tree in Fig. 6. The procedure was able to learn automatically the defect patterns of the materials and build a suitable hierarchical structure (dendrogram) allocating the pieces in the right place of the classification tree.

![Fig. 4. Outline of a two defect specimen inspected by impact-echo. Sensor setup is depicted.](image)

![Fig. 5. Equipment setup.](image)

**V. CONCLUSION**

The general non-parametric approach of the proposed procedure allows its application to several fields, such as image segmentation. In the illustrated application of defective material classification in non-destructive testing, the classification tree obtained using a simple operation technique as impact-echo is very useful and it could be extended to manufacturing contexts. For instance, in certain industries, such as marble plants, the knowledge of the kind of defect is critical to optimize the manufacturing process of cutting marble blocks. In addition, the supervised-unsupervised scheme of the procedure would facilitate testing and building the signal database in real applications.

![Fig. 6. Results of the experiments.](image)

**REFERENCES**


