Computer Science Journal of Moldova, vol.**18**, no.3(54), 2010

Performance evaluation of clustering techniques for image segmentation

Elmehdi Aitnouri Mohammed Ouali

Abstract

In this paper, we tackle the performance evaluation of two clustering algorithms: EFC and AIC-based. Both algorithms face the cluster validation problem, in which they need to estimate the number of components. While EFC algorithm is a direct method, the AIC-based is a verificative one. For a fair quantitative evaluation, comparisons are conducted on numerical data and image histograms data are used. We also propose to use artificial data satisfying the overlapping rate between adjacent components. The artificial data is modeled as a mixture of univariate normal densities as they are able to approximate a wide class of continuous densities.

Keywords: Performance evaluation, probability density function, clustering algorithm, unsupervised learning, univariate normal mixtures, gray-level histogram.

1 Introduction

Cluster analysis appeared ever since the works in [1, 2]. It began to attract a great deal of attention with the publication of Sokal and Sneath's revolutionary book on numerical taxonomy [3] and the development of high speed computers. More than a hundred different schemes of cluster analysis have been proposed [4-9] which makes it difficult for users to choose the right clustering algorithm for their application. Some authors have partially addressed this problem by evaluating and comparing several clustering techniques [5, 9-13].

For numerical comparison, the common model used to generate test data is the mixture model. Several schemes for generating artificial mixtures of test data have been proposed [9, 14-16]. The task of cluster analysis is cast as the classification of a mixture of populations into its components provided that the number of populations and their parameters are unknown. The contribution of mixture models is not limited to test data generation. They are now used in the design of clustering algorithms. This is due to the ability of mixture models, and particularly mixture of normal densities, to approximate a wide class of continuous probability densities. Parameters estimation in a mixture model is accomplished using the Expectation Maximization algorithm (EM) [17], which maximizes the likelihood function of the observed data.

Mixture models face the difficult problem of determining the number of components in the mixture. This is known as the cluster verification [18]. Because the mixture components are not always well separated, the determination of their number is a difficult task. There are two categories of clustering algorithms: direct and verificative methods. Direct methods are those which exploit geometric properties or other *ad hoc* principles such as inflexion points in histograms [20, 21]; see also [12, 22] for other methods. Their performances depend on the application. Recently, another direct method has been proposed: elimination of false clusters (EFC) [23]. Methods which take explicitly account of the nature of the observed data, perform very well, compared to verificative methods. They are generally simple and fast. Verificative methods on the other hand do not depend on the application. They are general algorithms such as Akaike's information criterion (AIC) [25, 31], the partition coefficient (PC) [26], the ICOMP [27], the minimum description length (MDL) [28] and the minimum message length (MML) [29]. Many applications, such as image segmentation and image retrieval [30], require collecting information about the overall distribution of the pixels in the image. The common approach is to consider the histogram as a mixture of univariate normal densities and to estimate the parameters of each component of the mixture. Cluster validation techniques must be used to estimate the number of components in the

mixture.

In this paper, we propose to benchmark EFC and AIC-based algorithms. The goal of this comparison is twofold: the evaluation of the performances of EFC against AIC-based algorithm on the same benchmarking data since a fair a comparison has never been done before; and the development of a new method for generating test data from a mixture of normal densities. The artificial data shows the property of components separation as they are generated according to components overlapping rate. The overlapping rate allows the control of the degree of overlap between two adjacent components. This results in mixtures with components sufficiently separated so that the clustering algorithm has a chance to distinguish between sub-populations. The paper is organized as follows: standard clustering algorithms and mixture models are reviewed in Section 2. Section 3 deals with comparison purposes and experimental results. Finally, the discussion is presented in Section 4.

2 Unsupervised learning and standard mixtures

Consider that the data can be represented by N random vectors denoted by $X = x_1, x_2, \cdots, x_N$, and assume that it arises from a mixture of M normal densities. The distribution of the data can be approached by a PDF which takes the following form, for any $x \in X$:

$$p(x, \Gamma) = \sum_{j=1}^{M} \kappa_j f_j(x, \Theta_j) \tag{1}$$

where $f_j(.)$ is the j^{th} normal distribution with parameter $\Theta_j = (\mu_j, \sigma_j)$ representing respectively the mean and the standard deviation of the j^{th} component; κ_j are the mixing parameters, with the restrictions that $\kappa_j > 0$ for $j = 1, \cdots, M$ and $\sum_{j=1}^{M} \kappa_j = 1$, and $\Gamma_j = (\Theta_j, \kappa_j)$ totally describe $p(x, \Gamma)$. In the following, we use $\Gamma(x, \Gamma_j)$ to describe the j^{th} component of the mixture. Note that μ_j and σ_j are scalar since we are

dealing with one-dimensional distributions. Such a representation of X is called a mixture representation [20].

Mixture models have shown better data classification capacities than many conventional neural network models such as layered networks trained with the Back-Propagation algorithm. They have been used as basic configurations for radial functions in Radial Basis Networks (RBF) [32]. Various procedures have been developed for determining the parameters of a mixture of normal densities, often based on the maximum likelihood technique, leading to the EM algorithm [17] and stochastic sequential estimation [33]. The technique used to maximize the likelihood function relies on the choice of Γ most likely to give rise to the observed data. For analytical convenience, it is equivalent to minimize the log-likelihood function, which, for the given X yields:

$$E = -\log\{L(X,\Gamma)\} = -\left\{\sum_{n=1}^{N}\log(p(x_n,\Gamma))\right\} \tag{2}$$

$$= -\sum_{n=1}^{N}\log\left\{\sum_{j=1}^{M}\kappa_j f_j(x,\Theta_j)\right\}$$

Here, the log-likelihood function is considered as an error, and its minimization with respect to Γ leads to an estimate, denoted by $\hat{\Gamma} = (\hat{\Theta}, \hat{\kappa})$. A review of the maximum-likelihood technique in the context of mixture models is given in [34]. However, due to the structural complexity of mixture models, most of the maximum likelihood procedures are numerical and iterative, resulting in only locally optimal estimates. The accuracy of the final estimate $\hat{\Gamma}$ depends heavily on the initial value of Γ. This is essentially the reason why clustering algorithms are usually used to produce initial estimates of Γ.

2.1 Cluster analysis and initial estimation of Γ

Clustering algorithms basically perform an unsupervised learning that groups the input data into different categories, from which the initial values of means and widths can be calculated. A common algorithm

widely used in the literature to initialize Γ is the k-means algorithm [35, 36], an effective and popular algorithm developed within the pattern recognition community. Improved schemes for the k-means algorithm using fuzzy concepts (fuzzy c-means) are available in the literature [37]. However, clustering algorithms do not provide an estimation of M, the number of clusters in the data set, but reorganize data into M clusters, with M given by the user. Thus, a key problem, known as cluster validation [18] concerns estimation of M. Most of the previously proposed solutions to the cluster validation problem can be classified into two broad categories [12, 19, 38]: direct approaches and verificative methods. Regarding direct approaches, different methods have been proposed in the literature. However, each method is generally applicable only to a specific type of application data. In our study, we are interested in estimating the number of components in gray-level image histograms. Thus, we developed a direct algorithm, denoted by EFC (Elimination of False Clusters), to solve the cluster validation problem [23, 24]. In the next section, we review the basic principle of the EFC.

2.1.1 EFC algorithm

A gray-level image histogram can be represented by a function, $h(x)$, $x \in Gl_N$, of the gray-level frequencies of the image, where $Gl_N = \{0, 1, \cdots, N-1\}$ corresponds to the set of gray levels of the image. When a given image contains objects/regions having quite different gray-level values, different modes appear in the histogram of the image. This type of histogram is called multi-modal. However, when objects/regions in the image have close gray-level averages, they may overlap to give a single mode. Our hypothesis is that each mode corresponds to a normal distribution. This is acceptable in a large number of practical applications [39]. The EFC algorithm has been developed especially to estimate the number of modes of such histograms. Figure 1 shows a block diagram of the model, which consists of two major steps.

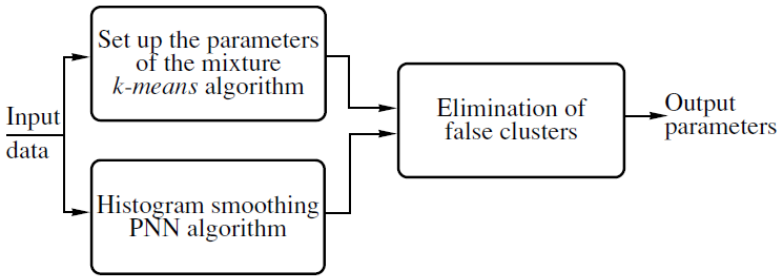

Fig. 1. A block diagram of the EFC algorithm.

In the first step, initial estimation of the mixture parameters is done using the k-means algorithm. In order to approximate each mode by at least one Gaussian, the k-means algorithm is applied with a number of clusters K greater than M, the number of modes in the image histogram. The next step mainly concerns the EFC procedure for suppressing false clusters that may result from the k-means algorithm. Basically, it takes advantage of the Gaussian PDF. Before proceeding with the elimination, a smoothing operation is performed on the histogram using a PNN (Probabilistic Neural Network or, equivalently, Parzen Window) [40]. While this operation is not essential in all cases, it greatly increases the robustness of our model to noise (especially when applied to radar images). Finding the optimal smoothing parameter for the PNN is another interesting problem that we have studied [41].

To choose the best number of clusters to use in the k-means algorithm, we have experimentally studied the accuracy of cluster centers, denoted by y_j, $j = 1, \cdots, K$ estimated by the k-means algorithm, as the function of K, the number of initial clusters. The experiment involves computing the average distance between any true center and the closest center estimated over a set of artificial histograms. In other words, we wanted to get a statistical assessment of the quality of the k-means algorithm in terms of the (average) precision with which the true centers of histogram are estimated. For this purpose, an error function was proposed to measure the quality of the set of clusters computed by the k-means algorithm. The experiment yielded a very interesting relationship between the number of initial clusters K, the true number

of modes M, and the precision of the approximation. From the statistical point of view, the k-means algorithm accurately finds all the true centers if K is chosen as at least $M + 4$. This in itself is an important discovery concerning the k-means algorithm. We further note that the choices $M+2$ and $M+3$ are also good candidates for M. Consequently, for real images it is not necessary to impose a very strict condition on the accuracy with which M is estimated.

Since the k-means algorithm is applied with an initial number of clusters greater than the number of modes, there are false clusters that must be eliminated. The EFC is used to eliminate them. The proposed EFC procedure depends on two parameters, β and γ. β is related to the relative level of the histogram at which the symmetry is measured, because it specifies the percentage of histogram height $h(y_j)$ for any cluster center y_j. In practice, β can be as large as 0.975 and as small as 0.5. The parameter γ is used as a threshold for the acceptable deviation between the true center and the closest center for the clusters computed by the k-means algorithm. If the deviation, written as $||\mu_j - y_j||$, is greater than γ, then y_j is rejected, where μ_j, $j = 1, \cdots, M$, are the real centers of modes. In real applications, μ_j are unknown. Thanks to the fact that a true center divides a mode into two symmetric parts, an equivalent test can be performed without knowing the position of the true center. Reasonable values for γ have been computed experimentally. We have measured the deviation between the true center and the closest center found by the k-means algorithm, for each combination of M and K.

Some performances of the EFC procedure are listed in [23]. The results are encouraging; nevertheless, they should be compared with those of other cluster validation methods. For this purpose, in the next section, we will present the general structure of the second broad category of algorithms, known as verificative methods.

2.2 Verificative Methods

While some practical problems can be solved by using direct approaches, they do not provide a general solution. To find generally

applicable solutions to cluster validation, many researchers have tried to formulate the problem as a multiple-decision problem [25, 26, 27, 28, 29]. The philosophy of these techniques is quite simple. Instead of asking which hypothesis is acting (how many classes are really there), we ask which model, parameterized by K, the number of clusters, best fits the data. However, as the number of clusters increases, the estimated PDF fits the empirical density increasingly tightly, at the expense of its generalization capacity. This problem can be recognized as the bias and variance trade-off in curve-fitting problems [42]. Thus, one should always try to find the minimum number of components describing a PDF, without overfitting the empirical data. In the case of mixture models, we can generate K_{max} models, where K_{max} is given by the user. The choice of the best model thus becomes a model selection problem, since we find ourselves faced with competing K_{max} models. To choose among them, researchers have developed selection criteria. The criteria are composed of two parts. The first part concerns the adequacy, usually evaluated using the maximum likelihood. The second part deals with penalization, which employs an expression that essentially depends on the number of components, K. A general expression for a criterion $C(K)$, where $K = 1, \cdots, K_{max}$ is given by:

$$C(K) = -aL_K + g(K) \tag{3}$$

where a is a scalar, L_K is the logarithm of the likelihood of the model of order K, and $g(.)$ is an increasing function depending on K. The best model is the one which minimizes $C(K)$, namely:

$$\hat{K} = \arg_{min_K} C(K) \tag{4}$$

2.2.1 Implementation

The principle is that for each value of K, $K = 1, \cdots, K_{max}$, Γ is initialized using the k-means algorithm with K initial clusters. Then, a maximum-likelihood solution is obtained using the estimated Γ and the value of K. There are no formal rules for choosing the value of K_{max}. However, several heuristics have been proposed:

- The number of observations N must be greater than the total number of parameters [43].
- Suggested choices for K_{max} include $K_{max} = \sqrt{\frac{N}{2}}$ and $K_{max} = \left(\frac{N}{\log N}\right)^{\frac{1}{3}}$ [27].

The general algorithm can be given as follows:

Algorithm 1.

Input: K_{max}
Output: $\hat{K}$
For $K = 1, \cdots, K_{max}$*:*
  Estimate Γ *using k-means with* K *initial clusters.*
  Compute the maximum likelihood for the estimated Γ*.*
  Compute $C(K)$*.*
  Choose $\hat{K}$*, such that* $\hat{K} = \arg_{min_K} C(K)$*.*

2.2.2 Akaike's information criterion (AIC)

In this paper, we are interested in the AIC. This technique was originally proposed by Akaike [31]. However, different schemes based on the AIC have been developed and used in different applications [19, 44,45,46]. In our work, we use the classical AIC proposed by Akaike and given by:

$$AIC(K) = -2L_K + 2N_p \tag{5}$$

where K is the number of components considered, L_K is the logarithm of the likelihood at the maximum likelihood solution $\hat{\Gamma}$, and N_p is the number of parameters estimated. We select K which leads to the minimum value of $AIC(K)$.

A brief comparison of some criteria for comparing models is presented in [38]. The AIC figures among these algorithms. The AIC performed better than both the ICOMP and the PC, and quite similarly to the MDL, but relatively poorly compared to the MML. Nevertheless, the AIC is the most popular criterion used in the literature. We

have chosen the classical AIC method since it does not depend on the size of the data. This is suitable for image histograms since a normalized histogram represents only the distribution of data, without any information of its size.

3 Comparison between the EFC and the AIC

It is unfortunately not possible to use results obtained respectively from [23] (EFC evaluation) and [38] (AIC evaluation) to compare the EFC and the AIC, for the simple reason that the test data used in the two experiments are different. Furthermore, neither method has been evaluated on an adequately verified data set. In order to perform a fair comparison, we must apply both algorithms to the same test data, which we will describe in the present section.

3.1 Comparison based on one vector

We used an artificial histogram composed of 3 modes as illustrated in Fig.2. To the smoothed histogram we added a Gaussian white noise of width $\sigma_n = 3$. Both algorithms were applied with $K_{max} = 7$. Figure 3 shows the resulting AIC plotted against the number of clusters $K = 1, \cdots, K_{max}$, computed from (3). The maximum likelihood technique has been used to estimate $\hat{\Gamma}_j$, $j = 1, \cdots, K$. We can see clearly from Fig. 3 that the minimum AIC is for $K = 3$, the exact number of components in the mixture.

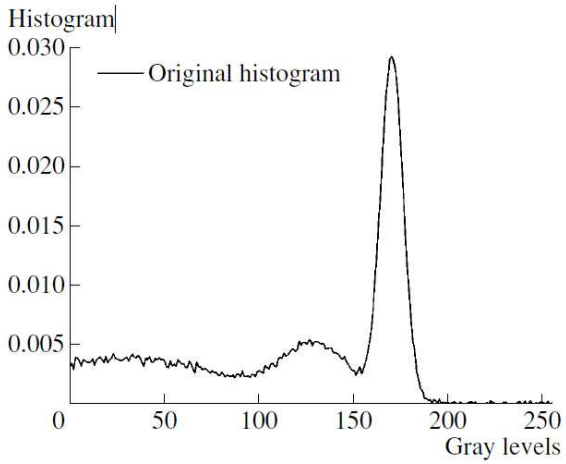

Fig. 2. Original histogram with Γ = ((28.79, 53.24, 0.44), (129.92, 17.6, 0.17), (170.1, 5.92, 0.39)).

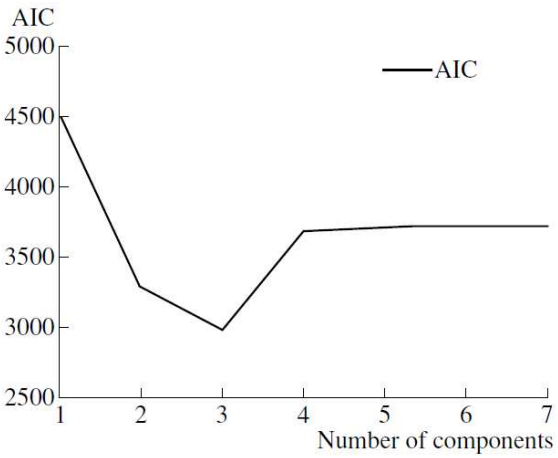

Fig. 3. Values of the AIC versus number of components.

Figure 4 shows the plot of both the original histogram and the

resulting histogram at the $\hat{\Gamma}_3$ solution, given by

$$\hat{\Gamma}_3 = ((36.00, 21.63, 0.26)(141.39, 21.24, 0.25)(168.53, 8.12, 0.49))$$

The estimation using the AIC is good: although there is a relative shift in the means for the first two components of the estimated histogram, the result is still acceptable. For the case of the EFC, Table 1 summarizes the parameters estimated using the k-means algorithm with $K_{max} = 7$. Table 1 is divided into two parts. The first multi-column, denoted by "Before EFC" presents the parameters of each component resulting from applying the k-means algorithm with $K_{max} = 7$. The second part of Table 1, the multi-column "after EFC ML", presents the resulting estimated parameters of each component after applying the EFC procedure and the maximum-likelihood algorithm respectively. In this part, dashes represent components eliminated by the EFC procedure. The values of both β and γ were set as in the experiments performed in [23], namely $\beta = 0.97$ and γ chosen from the γ table.

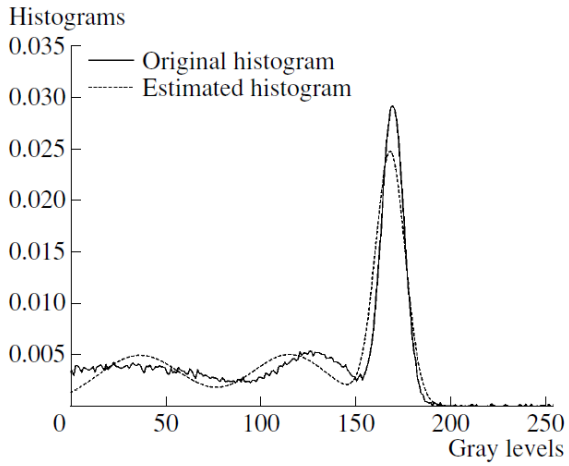

Fig. 4. Reconstructed PDF using the resulting AIC parameters, MSE = 0.0085.

Table 1. Results for the EFC for the artificial histogram, $K_{max} = 7$ and $\beta = 0.97$.

	Before EFC			After EFC & ML		
cluster	means	width	MP	means	width	MP
1	15.33	9.26	0.11	29.5	51.24	0.43
2	44.77	10.08	0.11	–	–	–
3	76.86	11.16	0.09	–	–	–
4	112.32	9.51	0.10	–	–	–
5	136.85	8.72	0.11	131.56	15.84	0.18
6	169.69	6.38	0.45	169.32	6.09	0.39
7	208.14	26.65	0.03	–	–	–

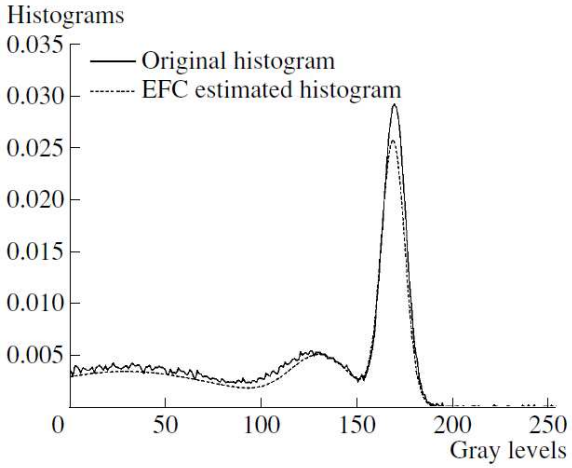

Fig. 5. The reconstructed PDF using the resulting EFC parameters, MSE = 0.0041.

From Table 1, we can see that the EFC procedure has eliminated four spurious clusters. Only clusters corresponding to the true components of the mixture have been kept. Thus, the EFC also finds the exact number of components. Classification using the k-means algo-

rithm, initialized using the values of the remaining cluster means (see Table 1), permits the redistribution of points belonging to eliminated clusters. Finally, we apply the maximum-likelihood technique. Figure 5 shows the plot of both the original and the resulting EFC mixtures for the parameters given in Table 1.

From Figs. 4 and 5, we can see that while both algorithms estimate the exact number of components, the EFC procedure is more accurate than the AIC (see MSE in Figs. 4 and 5). This result is in perfect agreement with the k-means experiment performed in [23]. Indeed, for the case of the AIC, the parameters used to initialize the likelihood were obtained using the k-means algorithm. For the AIC [3], the k-means algorithm was performed with $K = 3$ initial clusters. This is the exact number of components of the test histogram. In this case, the estimated centers of the components are less accurate than those obtained with a large value of K. This does not help the maximum likelihood to result in accurate final estimates. In contrast, for the case of the EFC, the initialization of each component's mean is obtained using the k-means algorithm with $K = K_{max} = 7$ clusters. Therefore, the centers of the real components are very well estimated. This helps the maximum-likelihood procedure to converge, which explains why the parameters estimated using the EFC were clearly more accurate than those estimated by the AIC. We can also evaluate the two algorithms in terms of complexity. For example, we need to perform the k-means algorithm K_{max} times, followed by the maximum likelihood to compute $AIC(K)$, $K = 1, \cdots, K_{max}$. In contrast, the EFC algorithm requires one k-means run, followed by the maximum likelihood. Thus, the EFC technique is roughly K_{max} times faster than the AIC. Although the EFC compares favorably to the AIC in the example described above, it is necessary to examine the effectiveness of the EFC in more general situations. For this purpose, we need to apply both algorithms to a large set of test data. In this way, we can obtain a statistical assessment of the general performance trend. Statistical comparison between algorithms is possible, since mixture models have been used as general test data for clustering algorithms. Indeed, a number of different schemes for generating artificial mixtures have been proposed. Blashfield [9]

and Edelbrock [14] used unconstrained multivariate normal mixtures with fairly complex covariance structures. Milligan and Issac proved in [8] that the generation processes used in [9] and [14] lead to data with overlapping clusters. Other researchers such as Kuiper and Fisher [15], and Bayne et al. [11] have used multivariate normal clusters with simple covariance structure and have directly manipulated variables such as the separation between clusters. In 1985, Milligan and Cooper [12] examined the ability of about 30 different clustering algorithms to determine the number of clusters in a data set. For this purpose, Milligan has developed an algorithm for generating test data [16]. The algorithm is described in nine steps. The main assumption, however, is that the generated data do not overlap in the first dimensional space. The verification of this assumption was mainly done using non-automatic techniques such as visual inspection. All of these generation methods try to define ad hoc criteria in order to handle overlapping components. However, there is no formal definition of the concept of overlap. This raises questions concerning the effectiveness of the generation schemes. In this paper, we introduce a new algorithm for generating mixtures of univariate normal densities. First, we give an analytic definition of the concept of overlap. We have chosen relative component overlap instead of total component overlap in order to preserve the appearance of each component in the mixture. This definition allows us to control the overlapping process, thus offering the possibility of generating a large number of such mixtures with known degrees of overlap. The generation of such examples, denoted by non-overlapped vectors, is described in the next section.

3.2 Generation of non-overlapped vectors

When we want to generate a large set of mixture data, the problem is how to ensure that modes are not totally overlapped. As an example of what we mean by overlapped, we generate a three-component mixture, but due to component overlap, the mixture results in only two components. The example in Fig. 6 illustrates this phenomenon. The mixture in Fig. 6a is actually composed of three components, despite

the fact that only two components are visible. The vector in Fig. 6a is called an overlapped vector; the vector in Fig. 6b on the other hand is not. Note that the difference between the parameters of Fig. 6.a, b is the value of the second width σ_2. We can perform algorithm comparison using overlapped vectors, as in the example of Fig. 6a. Each algorithm will most likely estimate the same number of components. The process is indeed still fair. However, such vectors are not valid for use in evaluating algorithms, due to the degree of component overlap. To avoid such situations, it is necessary to control the overlapping process.

Definition 1. *We define the overlapping rate, denoted by OLR, as,*

$$OLR = \frac{\min(p(x))(\mu_i \leq x \leq \mu_{i+1})}{\min(p(\mu_i), p(\mu_{i+1}))}. \tag{6}$$

In the above formulae, $p(x)$ is the mixture function and $\mu_i < \mu i + 1$. Figure 7 illustrates two different overlappings for a mixture of two Gaussians. $OLR \rightarrow 0$ when the Gaussians are almost totally separated, since $\min(p(x)) \rightarrow 0$, $(\mu_i \leq x \leq \mu_{i+1})$ as in Fig. 7a. $OLR \rightarrow 1$ when $\min(p(x)) \rightarrow \min(p(\mu_i))$ as in Fig. 7b. Thus, OLR specifies the degree of overlap between two adjacent components of a mixture. Our goal here is to develop an automatic algorithm for generating mixture components that are not completely overlapped; in other words $OLR < 1$ should be true for all pairs of adjacent components. Such an algorithm allows us to generate a large set of non-overlapped vectors. Furthermore, we could also control the overlapping rate if we want to perform a refined evaluation of algorithms.

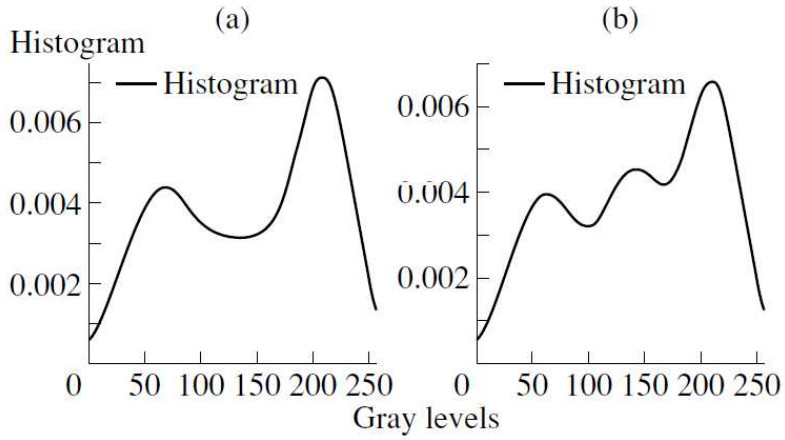

Fig. 6. Artificial histograms generated from a true mixture of two Gaussians. (a) $\Gamma_1 = (60, 28, 0.3)$, $\Gamma_2 = (140, 32, 0.3)$, $\Gamma_3 = (210, 25, 0.4)$. (b) $\Gamma_1 = (60, 31, 0.3)$, $\Gamma_2 = (140, 22, 0.3)$, $\Gamma_3 = (210, 25, 0.4)$.

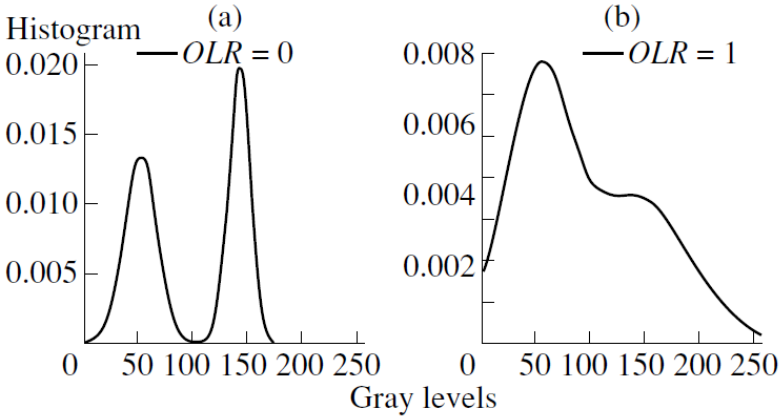

Fig. 7. OLR principle. (a) $OLR \approx 0$, (b) $OLR \approx 1$.

From (6), a general condition for $OLR < 1$ can be obtained using the derivatives of $p(x)$. This idea has been applied to the solution of the edge-detection problem using the Laplacian of Gaussian edge detector [47]. Interestingly, there is an astonishing similarity between the case of overlapping components and the case of a double-step edge. The edge-detection problem involves the appearance of a false edge located between the two true edges when they are smoothed with a Gaussian. In [47], the problem is treated as a mixture of two Gaussians with equal width σ. It is proven that a false edge appears if and only if $2\sigma < \mu_d$, where μ_d is the distance between the means of the two Gaussians. Viewed from the perspective of our case, the false edge corresponds to $\min(p(x))$ defined in (6). Thus, we have:

Corollary 1 - *If we have a mixture of two Gaussians with the same width* σ, $OLR < 1$ *iff* $2\sigma < \mu_d$.

Corollary 1 is a direct consequence of results obtained in [47]. Unfortunately, it is not valid for the case where $\sigma_1 \neq \sigma_2$, and the mathematical approach used in [47] cannot be extended to this general case. As it can be seen in what follows, developing a general condition for $OLR < 1$ is much more difficult. The algorithm will be iterative, dealing with two adjacent components at each iteration. When the width of the first component is fixed, the condition $OLR < 1$ will depend on the width of the next component. Here is a sketch of the algorithm:

3.2.1 Algorithm for generating non-overlapped vectors

Algorithm 2.

Generate M*, the number of components in the mixture.*
For $i = 1, \cdots, M$*:*
    Randomly generate μ_i *and* κ_i *such that* $\mu_i < \mu_{i+1}$, $\kappa_i > 0$, *and* $\sum_{i=1}^{M} \kappa_i = 1$.
    Randomly generate σ_1 *such that* $\sigma_1 < \mu_1 - \mu_2$.
For $i = 2, \cdots, M$*:*
    Compute σ_i *such that* $OLR < 1$.

The number of components M can be set by the user. It is also

convenient, without loss of generality, to generate M means μ_i and sort them so that $\mu_i < \mu_{i+1}$. This will facilitate the execution of the algorithm. However, σ_1 should be smaller than $\mu_2 - \mu_1$; otherwise, there will be no solution for σ_2 to satisfy the non-overlapping condition. Thus, the problem is to find an upper bound for σ_{i+1} given σ_i, in order to ensure that $OLR < 1$.

In order to develop a general algorithm, let us consider the overlap of two adjacent components $\Gamma_1(x)$ and $\Gamma_2(x)$ with $\Gamma_1 \neq \Gamma_2$. Let us denote by $S_\sigma = \kappa\Gamma(\mu - \sigma)$ the height at which the width σ is located for a given component $\Gamma(x)$. For the general case of the overlapping process for two adjacent components $\Gamma_1(x)$ and $\Gamma_2(x)$, we have $S_{\sigma_1} \neq S_{\sigma_2}$ since the two components do not necessarily have the same height.

Definition 2. *We define a notion of apparent width, denoted by $\hat{\sigma}$, as the deviation of the higher component from its center at height S_{σ_l}, for a pair of adjacent components. Here, S_{σ_l} is the height at which the width of the lower component is located.*

We distinguish two cases: $\Gamma_1(\mu_1) > \Gamma_2(\mu_2)$, denoted by case 1, and $\Gamma_1(\mu_1) < \Gamma_2(\mu_2)$, denoted by case 2. Figure 8 illustrates the principle of apparent width for the two cases. Using Definition 2, a generalization of Corollary 1 can be stated as a hypothesis:

Hypothesis 1 - *If we have a mixture of two Gaussians with different heights, $OLR < 1$ iff $\hat{\sigma}_h + \sigma_l < \mu_d$.*

$\hat{\sigma}_h$ is the apparent width of the higher component, σ_l is the width of the lower component, and μ_d is the distance between the two means. It is self-evident that if the components have the same widths and the same heights, Hypothesis 1 collapses into Corollary 1, since $\hat{\sigma}_h = \sigma_l = \sigma$. However, it is very difficult to prove Hypothesis 1. For each of the above cases, we can compute $\hat{\sigma}_h$ as a function of Γ_1 and Γ_2. Then, by introducing the expression of $\hat{\sigma}_h$ in Hypothesis 1, we can solve the resulting relation in order to obtain the bound on σ_2. For the two cases we have:

$$\begin{cases} A\sqrt{\ln(B\sigma_2)} + \sigma_2 < \mu_d & \text{case 1} \\ C\sigma_2\sqrt{\ln\frac{D}{\sigma_2}} + \sigma_1 < \mu_d & \text{case 2} \end{cases} \tag{7}$$

where A, B, C and D are known values. For deduction details, see [48]. By solving (7), we can obtain a condition on the upper bound of σ_2 as a function of the remaining known parameters. However, (7) is non-linear, which will introduce difficulties for obtaining general solutions. The non-linearity of (7) arises from the non-linearity of the Gaussian expression. Thus, it is necessary to simplify (7) by approximating the Gaussian expression.

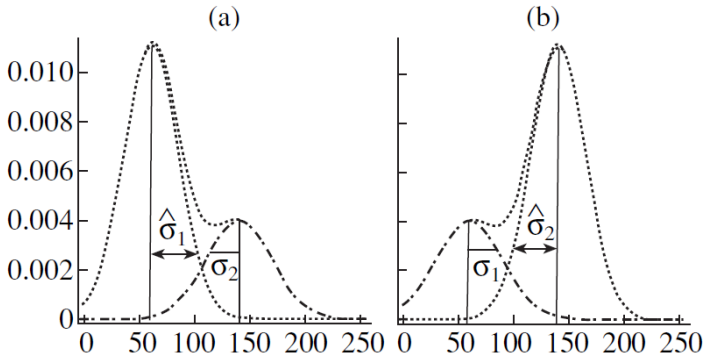

Fig. 8. Apparent width. (a) when $\Gamma_1(\mu_1) > \Gamma_2(\mu_2)$, (b) when $\Gamma_1(\mu_1) < \Gamma_2(\mu_2)$.

3.2.2 Approximation of the Gaussian

Consider a family of lines $\Delta_l = \Delta_1, \Delta_2, \Delta_3, \cdots, \Delta_p$, where Δ_i, $i = 1, \cdots, p$, is a tangent line to the point $(x_{i\sigma} = \mu - i\sigma, p(x_{i\sigma}))$. If we approximate a Gaussian by the series of lines Δ_i, we obtain a piece-wise linear approximation, denoted by $\hat{g}_l(x)$ and given by

$$\begin{cases} \hat{g}_l(x) = i\frac{S}{\sigma}e^{-\frac{1}{2}i^2}\left(x - \mu + \frac{i^2+1}{i}\sigma\right) \\ x \in [\mu - f(i-1)\sigma, \mu - f(i)\sigma] \end{cases} \tag{8}$$

where

$$f(i) = \frac{(i^2+1)e^{-\frac{i^2}{2}} - ((i+1)^2+1)e^{-\frac{(i+1)^2}{2}}}{ie^{-\frac{i^2}{2}} - (i+1)e^{-\frac{(i+1)^2}{2}}}$$

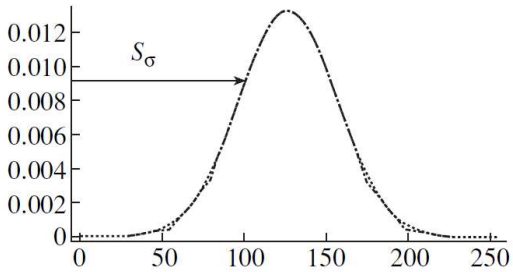

Fig. 9. Approximation of a Gaussian with $p = 3$ and MSE = $7.48742e^{-06}$.

Due to the symmetry of the Gaussian, we have developed only the approximation of the left part. Figure 9 shows the result of this approximation for $p = 3$. The approximation error, denoted by E_{app}, is given by:

$$E_{app}(p) = \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{p} e^{-\frac{i^2}{2}} \times [f(i-1) - f(i)] \left[\frac{f(i) - f(i-1)}{2} - \frac{i^2+1}{i} \right] \tag{9}$$

It is proven in [48] that $E_{app}(p)$ does not depend on the width σ. However, $E_{app}(p)$ decreases as p increases and is almost constant when $p \geq 3$. Note that this approximation has been developed especially for this work. With this new approximation of the Gaussian, the computation of the apparent width $\hat{\sigma}$ can be done on the tangent lines Δ_i , $i = 1, \cdots, p$, and we will have (10) as solutions to the condition of Hypothesis 1:

$$0 < \sigma_2 \leq \frac{-(2\sigma_1 - \mu_d) + \sqrt{\delta}}{2} \text{ (case 1)} \tag{10}$$

where δ is the discriminant of the quadratic form given by

$$\begin{gathered} T_1(\sigma_2) = \sigma_2^2 + (2\sigma_1 - \mu_d)\sigma_2 - \tfrac{\kappa_2}{\kappa_1}\sigma_1^2 \text{ and} \\ 0 < \sigma_2 \leq \frac{\frac{\kappa_2}{\kappa_1} 2\sigma_1 - \sqrt{\delta}}{2} \text{ case 2} \end{gathered} \tag{11}$$

where δ is the discriminant of the quadratic form given by

$$T_2(\sigma_2) = \sigma_2^2 - \frac{\kappa_2}{\kappa_1} 2\sigma_1\sigma_2 + \frac{\kappa_2}{\kappa_1}\mu_d\sigma_1 - \kappa_2\sigma_1^2.$$

All deduction details are available in [48]. The conditions in (10) and (11) express the bound within which σ_2 should be picked in order that the overlap of the two adjacent components satisfies $OLR < 1$. In each step i of the algorithm $i = 1, \cdots, M-1$, (10) and (11) are used to generate σ_{i+1}. By choosing σ_{i+1} as suggested in (10) and (11), we ensure control of all the parameters of the vector. For implementation purposes, especially when we want to generate a large set of non-overlapped vectors, we define other parameter measures that are grouped in a characteristic vector, denoted by CVS (Characteristic Vector of the Set). These measures are:

(a) the number of vectors forming the set;

(b) the maximum number of components in the set;

(c) the minimum number of components in the set;

(d) the minimum distance between the means of components;

(e) the minimum width of Gaussian white noise added to vectors of the set; and finally,

(f) the maximum width of Gaussian white noise added to vectors of the set.

The example used here to compare the EFC with the AIC has a CVS = (1, 3, 3, 12, 3, 3).

3.3 Comparison based on a set of vectors

In this section, we intend to compute a kind of average comparison between the EFC and the AIC. For this purpose, we will use a set of non-overlapped vectors. The evaluation proposed here is divided in two parts. First, we will compute the ability of each algorithm to

estimate the exact number of components. This type of evaluation is used in [12, 23, 38]. The results will be presented in tables showing all statistics. Secondly, we will use the parameters of each component and reconstruct the PDF in order to compute the measure of adequacy of the estimated vectors.

Both experiments use three different sets of 1000 vectors each, generated by the algorithm described in the previous section. The CVSs of the three sets are given respectively by: (1) CVS_1 = (1000, 1, 1, 12, 0, 0), a set containing only vectors of one component, (2) CVS_2 =(1000, 2, 2, 12, 0, 0), a set containing only vectors of two components and (3) CVS_3 = (1000, 3, 3, 12, 0, 0), a set containing only vectors of three components. Moreover, in order to evaluate the robustness of both algorithms against noise, we have used the same generated sets and added a Gaussian white noise of width $\sigma_n = 2$ to form new noisy sets. Both algorithms were applied with $K_{max} = 7$.

The results of the first experiment are presented in Tables 2 and 3 (non-noisy and noisy sets, respectively). Each of these tables is divided into three parts. Each part, identified by its CVS (row 2), presents the application of both algorithms to a given set. The first column gives the different possibilities for the number of components each algorithm can estimate. The cells of the table report the percentage of cases in which the two algorithms estimate the number of components given in column 1. As an example, when the AIC is applied to the non-noisy set with CVS_1, it estimates two components in 42% of cases.

From the results reported in Table 2 (non-noisy sets), the AIC and EFC performances are quite similar. Nevertheless, as the number of components of the set increases, the EFC performs relatively better than the AIC (15% better for the set with CVS_3). For the noisy sets reported in Table 3, we see that the AIC is more robust than the EFC. Indeed, there are no great differences between the AIC results shown in Tables 2 and 3. The EFC is less robust since its performances were degraded by about 12% for all three sets. Note that the smoothing operation using the PNN was not performed in these experiments, in order to evaluate the robustness of the EFC against noise.

The second experiment computes the adequacy degree to which the

Table 2. Comparison between EFC and AIC applied to the noiseless sets. NCE is the estimated number of clusters.

	Noise standard deviation $\sigma_n = 0$					
	CVS_1		CVS_2		CVS_3	
NCE	AIC	EFC	AIC	EFC	AIC	EFC
1	54	56	15	8	0	0
2	42	23	47	52	13	6
3	2	12	21	16	51	66
4	2	5	10	10	28	15
5	0	3	6	12	6	10
6	0	0	1	2	2	2
7	0	0	0	0	0	0

Table 3. Comparison between EFC and AIC applied to noisy samples.

	Noise standard deviation $\sigma_n = 2$					
	CVS_1		CVS_2		CVS_3	
cluster	AIC	EFC	AIC	EFC	AIC	EFC
1	54	46	12	5	0	0
2	38	31	45	43	6	3
3	5	15	25	31	49	54
4	2	4	12	11	25	26
5	1	2	2	6	13	10
6	0	1	1	3	5	5
7	0	0	1	1	2	2

Table 4. Average MSE for each set.

	$\sigma_n = 0$			$\sigma_n = 2$		
	CVS_1	CVS_2	CVS_3	CVS_1	CVS_2	CVS_3
AIC	0.022	0.047	0.248	0.021	0.43	0.0122
EFC	0.008	0.038	0.072	0.019	0.042	0.081

two algorithms fit, measured by the mean square error (MSE). When an algorithm estimates a given number of components K, the maximum likelihood estimates for K components are used to reconstruct the estimated PDF. Thus, a MSE is computed between the original and the estimated vectors. This experiment is applied to both sets, noiseless and noisy. Table 4 shows the average MSEs resulting from the application of the two algorithms to the different sets. We can see from Table 4 that in all cases, the EFC has better overall adequacy than the AIC. The average MSE, however, does not provide specific information regarding the behavior of the adequacy as a function of the estimated number of components. Such information would help to perform an objective comparison.

To this end, we compared the adequacy of the two algorithms using only vectors resulting in the estimation of the same number of components. In other words, instead of using, for example, all the vectors of CVS_1 to compute the MSE, we divide them into groups. Each group contains only vectors resulting in the estimation of the same number of components. Thus, we will have K_{max} groups. We then compute an average MSE for each group. Note that if a group contains only a few vectors, the average MSE can be biased. On the other hand, the average MSE is representative when a group contains a large number of vectors.

Figure 10 shows the plots of adequacy for EFC, for CVS_1 in (a), CVS_2 in (b) and CVS_3 in (c). The adequacy behaves similarly for the two algorithms. Indeed, adequacy for vectors resulting in correct estimation of the number of components M is relatively good compared

to those resulting in a close estimation of M. However, the results totally deteriorate for a significant underestimation of M (see Fig. 10c with CVS_3 for vectors resulting in an estimation of $M = 1$). Finally, the adequacy is better when we overestimate M (see Fig. 10a with CVS_1 for vectors resulting in an estimation of $M = 6$). When no vector results in an estimation of a given number of components, the corresponding error is set to $MSE = 1$, which explains the behavior of the curves for $K = 7$. This overall behavior of the adequacy is also observed for the noisy sets in Fig. 11.

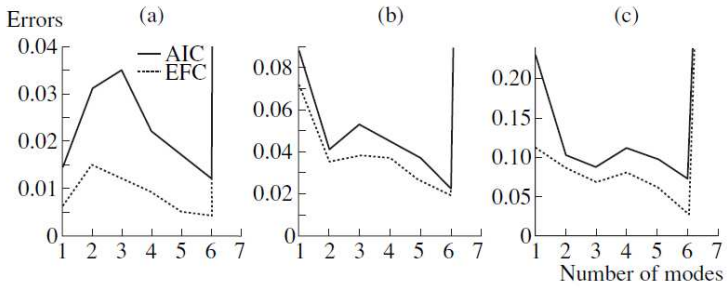

Fig. 10. Adequacy of the AIC and the EFC applied to non-noisy sets: (a) CVS_1, (b) CVS_2, and (c) CVS_3.

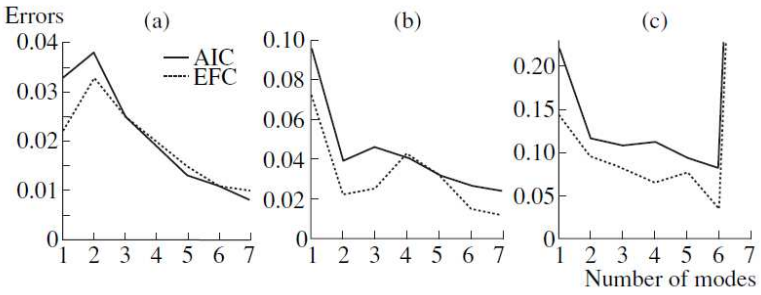

Fig. 11. Adequacy of the AIC and the EFC applied to noisy sets: (a) CVS_1, (b) CVS_2 and (c) CVS_3.

The behavior of clustering algorithms observed above is not unique to our experiments. Indeed, Windham and Cutler [49] observed it,

and developed an algorithm, called MIREV in order to use it. They argued that the best solution may be indicated by an elbow or a knee in a plot of the values versus the number of components. However, the occurrence of the elbow or knee may not necessarily mean that a solution is particularly good, but simply that its neighbors are relatively bad. Note that the experiments performed by Windham and Cutler were done using bivariate normal densities of three components each.

4 Conclusion

The EFC and AIC algorithms are two clustering algorithms pertaining to two different categories: direct and verificative methods. The category of verificative methods provides algorithms than can be used for all types of application data. However, when a prior knowledge about certain characteristics of the application data is available, direct methods can be designed in order to exploit this knowledge. In the case of gray-level image histograms, the EFC clearly outperforms AIC. Indeed, the effectiveness of the EFC has been shown in this paper in terms of its ability to estimate the exact number of modes in a histogram and the adequacy resulting from PDF estimation using the estimated parameters. It is also straightforward to verify that the EFC is roughly K_{max} times ($K_{max} = 7$ in our experiments) faster than the AIC. Note that a more extensive comparison can be conducted, including other clustering algorithms. In this paper, we have conducted a comparison between the EFC algorithm and the AIC algorithm. The comparison was designed to use a novel algorithm for generating mixture test data with non-overlapped components. This algorithm makes it possible to perform statistical tests and evaluations, since it can handle a large number of test data. Moreover, its flexibility allows the design of more detailed and appropriate statistical tests, such as algorithm's robustness in relation to component overlap. This type of test provides information about algorithm limitations. The formal definition of component overlap introduced in this paper can be used to design multivariate mixtures of test data. Indeed, one can keep a Milligan [16] generation framework, while using our algorithm to satisfy the

non-overlap condition necessary in 1D.

References

[1] Tryon, R.C., Cluster Analysis, volume of Ann Arbor, Mich. Edwards Brothers, MA, 1939.

[2] Zubin, J.A., *A Technique for Measuring Like-mindedness*, Abnormal and Social Psychology, 1938, vol. 33.

[3] Sokal, R.R. and Sneath, P.H.A., Principles of Numerical Taxonomy, San Francisco: Freeman, 1963.

[4] Anderberg, M.R., Cluster Analysis for Applications, New York: Academic Press, 1973.

[5] Bailey, K.D., Cluster Analysis, Heise, D., Ed. of Sociological Methodology edition, San Francisco: Jossey- Bass, 1974.

[6] Cormack, R.M., *A Review of Classification*, J. of the Royal Statisticians, Series A, 1971, vol. 134(3).

[7] Everitt, B.S., Cluster Analysis, London: Halstead Press, 1974.

[8] Milligan, G.W., *An Examination of the Effect of Six Types of Errors Perturbation on Fifteen Clustering Algorithms*, Psychometrica, 1980, vol. 45(3), pp. 325–342.

[9] Blashfield, R.K., *Mixture Model Test of Cluster Analysis: Accuracy of Four Agglomerative Hierarchical Methods*, Psychological Bulletin, 1976, vol. 83(3), pp. 377–388.

[10] Wolfe, J.H., *Pattern Clustering by Multivariate Mixture Analysis*, Multivariate Behavioral Analysis, 1970, vol. 5(4), pp. 329–349.

[11] Bayne, C.K., Beauchamp, J.J., Begovich, C.L., and Kane, V.E., *Monte Carlo Comparisons of Selected Clustering Procedures*, Pattern Recognition, 1980, vol. 12(2), pp. 51–62.

[12] Milligan, G.W. and Cooper, M.C., *An Examination of Procedures for Determining the Number of Clusters in a Data Set*, Psychometrica, 1985, vol. 50(2), pp. 159–179.

[13] Dudes, R. and Jain, A.K., *Clustering Techniques: the User's Dilemma*, Pattern Recognition, 1976, vol. 8(4), p. 247.

[14] Edelbrock, C., *Comparing the Accuracy of Hierarchical Grouping Techniques: the Problem of Classifying Everybody*, Multivariate Behavioral Research, 1979, vol. 14, pp. 367–384.

[15] Kuiper, F.K. and Fisher, L., *A Monte Carlo Comparison of Six Clustering Procedures*, Biometrica, 1975, vol. 31(1), pp. 86–101.

[16] Milligan, G.W., *An Algorithm for Generating Artificial Test Clusters*, Psychometrica, 1985, vol. 50(1), pp. 123– 127.

[17] Dempster, A.P., *Maximum Likelihood from Incomplete Data*, Via the EM Algorithm, J. of the Royal Statistical Society, 1977, vol. B 39(1), pp. 1–38.

[18] Jain, A.K. and Dubes, R.C., *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[19] Zhang, J. and Modestino, J.M., *A Model-fitting Approach to Cluster Validation with Application to Stochastic Model-based Image Segmentation*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 1990, vol. 12(10), pp. 1009–1017.

[20] McLachlan, G.J. and Basford, K.E., Mixture Models, New York: Marcel Deker, 1988.

[21] Aitnouri, E. and Ouali, M., *Multithreshold-based SAR image segmentation to targets and shadows detection*, Journal of Applied Remote Sensing (SPIE-JARS), Submitted in October 2010.

[22] Dudes, R. and Jain, A.K., *Validity Studies in Clustering Methodologies*, Pattern Recognition, 1979, vol. 11(4), pp. 235–254.

[23] Aitnouri, E.M., Wang, S., Ziou, D., Vaillancourt, J., and Gagnon, L., *Estimation of a Multi-modal Histogram's PDF using a Mixture Model*, Neural, Parallel & Scientific Computation, 1999, vol. 7(1), pp. 103–118.

[24] Aitnouri, E.M., Wang, S., Ziou, D., Vaillancourt, J., and Gagnon, L., *An Algorithm for Determination of the Number of Modes in Image Histograms*, Vision Interface V'99, Trois-Rivières, 1999, pp. 368–374.

[25] Akaike, H., *Information and an Extension of the Maximum Likelihood Principle*, 2nd Int. Symp. on Information Theory, Budapest, 1973, pp. 267–281.

[26] Bozdek, J.C., Pattern Recognition with Fuzzy Objective Function, New York: Plenum, 1981.

[27] Bozdogan, H., *Mixture-model Cluster Analysis Using Model Selection Criteria and a New Informational Measure of Complexity*. In Bozdogan, H. et al., Eds., The first US/Japan Conf. on the Frontiers of Statistical Modeling: An Informational Approach, Kluwer Academic Publishers, 1994, pp. 69–113.

[28] Rissanen, J., *Modeling by Shortest Data Description*, Automatica, 1978, vol. 14(3), pp. 465–471.

[29] Wallace, C.S. and Boulton, D.M., *An Information Measure for Classification*, Computer Journal, 1968, vol. 11(2), pp. 185–194.

[30] Kelly, P.M., Cannon, T.M., and Hush, D.R., *Query by Image Example: The CANDID Approach*, SPIE, 1995, vol. 242, pp. 238–248.

[31] Akaike, H., On Entropy Maximization Principle, in Applications of Statistics, Krishnaiah, P.R., Ed., North Holland Publishing Company, 1977, pp. 27–41.

[32] Powell, M.J.D., *Radial Basis Functions for Multivariate Interpolation: a Review, Algorithms for Approximation*, 1987, no. 1, pp. 143–167.

[33] Traven, H.G.C., *A Neural Network Approach to Statistical Pattern Classification by Semiparametric Estimation of Probability Density Functions*, IEEE Transactions on Neural Networks, 1991, vol. 2(3), pp. 366–377.

[34] Redner, R.A. and Walker, H.F., *Mixture Densities, Maximum Likelihood and the EM Algorithm*, SIAM Review, 1984, vol. 26(2), pp. 195–239.

[35] Fukunaga, K., Introduction to Statistical Pattern Recognition, New York: Academic Press, 1972.

[36] Tou, J.T. and Gonzalez, R.C., Pattern Recognition Principles, Addison-Wesley, Reading, MA, 1974.

[37] Bezdek, J.C., Pattern Recognition with Fuzzy Objective Function Algorithms, New York: Plenum, 1981.

[38] Oliver, J.J., Baxter, R.A., and Wallace, C.S., *Unsupervised Learning Using MML*. 13th Conf. on Machine Learning, Bari, Italy, 1996, pp. 364–372.

[39] Lawrence, D.B. and Gene Hwang, J.T., *How to Approximate a Histogram by a Normal Density*, The American Statistician, 1993, vol. 47(4).

[40] Parzen, E., *On the Estimation of a Probability Density Function and Mode*, Ann. Math. Stat., 1962, vol. 33, pp. 1065–1076.

[41] Jiang, Q., Aitnouri, E.M., Wang, S., and Ziou, D., *Ship Detection Using PNN Model*. In CD-ROM of ADRO'98, Montreal, 1998.

[42] Bishop, C.M., Neural Networks for Pattern Recognition, Oxford: Clarendon Press, Oxford Univ. Press, 1995.

[43] Sardo, L. and Kittler, J., *Minimum Complexity PDF Estimation for Correlated Data*. Int. Conf. on Pattern Recognition'96, Madisson, 1996, pp. 750–754.

[44] Sclove, S.L., *Application of the Conditional Populationmixture Model to Image Segmentation*, IEEE Tran. Patter. Anal. Machine Intell., 1983, vol. PAMI-5(4), pp. 428–433.

[45] Schwarz, G., *Estimating the Dimension of a Model*, The Annals of Statistics, 1978, vol. 6(3), pp. 661–664.

[46] Oliver, C., Jouzel, F., and Elmatouat, A., *Choice of the Number of Component Clusters in Mixture Models by Information Criteria*, Vision Interface VI'99, Trois- Rivieres, 1999, pp. 74–81.

[47] Tabbone, S., *Détection Multi-échelles de Contours Sous-Pixel et de Jonctions*, PhD thesis, Institut National Polytechnique de Lorraine, France, 1994.

[48] Aitnouri, E.M., Dubeau, F., Wang, S., and Ziou, D., *Generating Test Vectors for Clustering Algorithms*, Research Report 236, Dept. Math. and Comp. Scien., University of Sherbrooke, Sherbrooke, September 1999.

[49] Windham, M.P. and Cutler, A., *Information Ratios for Validating Mixture Analyses*, American Statistical Association, Theory and Methods, 1992, vol. 87(420), pp. 1188–1192.

Mohammed Ouali and Elmehdi Aitnouri, Received September 17, 2010

Mohammed Ouali
BAE Systems Canada, R&D Department
7600 Dr. Frederick Phillips Blvd, Montreal, Qc, Canada K2C 3M5
Phone: (514) 789-3000 (ext 443)
E–mail: mohammed.ouali@usherbrooke.ca

Elmehdi Aitnouri
DZScience-Sherbrooke
#1227-2820, Judge-Morin, Sherbrooke, Qc, Canada, J1E 2R6
Phone: (819) 329-4537 ext. 081
E–mail: elmehdi.aitnouri@usherbrooke.ca