On Comparison of Clustering Techniques for Histogram PDF Estimation

E. Aitnouri, S. Wang, and D. Ziou

Department of Mathematics and Computer Science, Faculty of Sciences, University of Sherbrooke, Sherbrooke, Qc, Canada J1K 2R1
e-mail: {aitnouri, wang, ziou}@dmi.usherb.ca

Abstract—This paper discusses the problem of finding the number of component clusters in gray-level image histograms. These histograms are often modeled using a standard mixture of univariate normal densities. The problem, however, is that the number of components in the mixture is an unknown variable that must be estimated, together with the means and the variances. Computing the number of components in a mixture usually requires “unsupervised learning”. This problem is denoted as “cluster validation” in the cluster analysis literature. The aim is to identify sub-populations believed to be present in a population. A wide variety of methods have been proposed for this purpose. In this paper, we intend to compare two methods, each belonging to a typical approach. The first, somewhat classical method, is based on criterion optimization. We are particularly interested in the Akaike’s information criterion. The second method is based on a direct approach that makes use of a cluster’s geometric properties. In this paper, we develop an algorithm to generate non-overlapped test vectors, allowing the generation of a large set of verified vectors that can be used to perform objective evaluation and comparison.

1. INTRODUCTION

Cluster analysis began to be widely used in many scientific fields in the 1960s. Although it has been proposed earlier in the century [1, 2], it was only in the early 1960s that cluster analysis began to attract a great deal of attention, with the publication of Sokal and Sneath’s [3] revolutionary book on numerical taxonomy, and the development of high speed computers, which made implementation of clustering procedures possible. Since then, cluster analysis has appeared in the literature of many fields, including not only engineering domains but also biology, zoology, psychology, etc. More than 100 different schemes for cluster analysis have been proposed. A review of these methods can be found in [4–9].

Due to the large number of methods, potential users of cluster analysis face several problems in choosing the right clustering algorithms for their applications. Indeed, different methods are likely to produce different results when applied to the same problem. Thus, great effort has been devoted to the evaluation, and hence, to the comparison, of clustering methods [5, 9–13]. Different parametric comparison schemes and similarity measures have been proposed to determine which algorithm is best suited to a specific application. For numerical comparison, the model commonly used to generate test data has been the mixture model. Under this model, the task of cluster analysis is defined as the classification of a mixture of populations into its components, according to the condition that the number of underlying populations and their parameters are unknown.

The use of mixture models to evaluate clustering methods has resulted in much interesting work. Indeed, different schemes for generating artificial mixtures of test data have been proposed [9, 14–16]. However, the contribution of mixture models to cluster analysis is not limited only to test purposes; they have become a target for the design of clustering algorithms. This is due to the capacity of mixture models, particularly mixtures of normal densities, to approximate a wide class of continuous probability densities. Estimation of the parameters in a mixture model can be accomplished using the Expectation Maximization (EM) algorithm [17], which maximizes the likelihood function of the observed data. This has made the use of mixture models possible in applications. However, the use of mixture models encounters a number of problems, among them the choice of the number of components in the mixture. This problem is known as cluster validation [18].

Determining the number of components in a mixture of distributions is an important but difficult problem, since separation between clusters is generally fuzzy. Many researchers have tried to find mathematically rigorous solutions to the cluster validation problem. For this purpose, clustering algorithms are divided into two broad categories: (1) direct approaches, and (2) verificative approaches. Examples of direct approaches are those algorithms that explicitly exploit certain geometric properties by using graphical tools or visual inspection. These algorithms are often based on ad hoc principles.
and their performance depends on the application data in hand. We can cite the estimation of the number of components via the inflexion points of the histogram [20, 21]; see also [12, 22] for other proposed methods. A direct method that has been recently proposed—called EFC (Elimination of False Clusters) [23, 24]—estimates the number of modes versus components in gray-level image histograms. Verificative approaches, on the other hand, are general algorithms which do not depend on the application data. A range of criteria for selecting the number of components has been proposed. These include Akaike’s information criterion (AIC) [25], the partition coefficient (PC) [26], the ICOMP [27], the minimum description length (MDL) [28] and the minimum message length (MML) [29].

It is easy to understand that methods which take explicit account of the characteristics of the observed data have the potential to perform very well since they are direct algorithms. These methods are attractive because they are generally simple and fast. In the case of our research, we are interested in image histograms. Indeed, 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 suppose that the histogram is a mixture of univariate normal densities and to estimate the parameters of each component of the mixture. Since the number of components is unknown, cluster validation techniques must be used to estimate it. In this paper, we propose to compare the EFC algorithm and an AIC-based algorithm. As pointed out in the previous paragraph, the two algorithms belong to different categories. The goal of this comparison is twofold: to evaluate the performance of the EFC algorithm using a well-known and frequently used method; and to propose a new method for generating test data from a mixture of normal densities. In fact, the EFC algorithm has given promising results [23]. Nevertheless, though it may be faster than methods based on the AIC, a fair comparison of these methods has not yet been done. For this purpose, in this paper, we develop a new method for generating mixtures of normal densities. The generated data, called non-overlapping vectors are very useful for comparing and evaluating clustering algorithms on the task of estimating the number of components in gray-level image histograms. The data used verify the definition of the term “non-overlapping” proposed in this paper. This means that the components are sufficiently separated so that the clustering algorithm has a chance to distinguish between subpopulations. This is achieved by using a novel algorithm which controls the degree of overlap between two adjacent components. We will use vectors of this type to compare the EFC algorithm with an AIC-based algorithm. We have chosen for this purpose the basic AIC as first defined by Akaike in [31].

The rest of 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 $X = x_1, x_2, \ldots, x_N$, 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 $x \in \mathbb{R}^N$:

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

where $f_j(\cdot)$ is the $j$th normal distribution with parameter $\theta_j = (\mu_j, \sigma_j)$ representing respectively the mean and the width (standard deviation) of the $j$th component; $\kappa_j$ ($j = 1, \ldots, M$) are the mixing parameters, with the restriction that $\kappa_j > 0$ for $j = 1, \ldots, M$ and $\sum_{j=1}^{M} \kappa_j = 1$; and $\Gamma_j = (\theta_j, \kappa_j)$ ($j = 1, \ldots, M$) 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 $\mu_j$ and $\sigma_j$ 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 $\Gamma$ 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:

$$
    E = -\ln \{ L(X, \Gamma) \} = -\left\{ \sum_{n=1}^{N} \ln (p(x_n, \Gamma)) \right\}
    = -\sum_{n=1}^{N} \left[ \sum_{j=1}^{M} \kappa_j f_j(x, \theta_j) \right].
$$

Here, the log-likelihood function is considered an error, and its minimization relative to $\Gamma$ 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 accru-
of such histograms. Figure 1 shows a block diagram of the model, which consists of two major steps.

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 $y_j$ ($j = 1, \ldots, K$) 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, $\beta$ and $\gamma$. $\beta$ 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, $\beta$ can be as large as 0.975 and as small as 0.5. The parameter $\gamma$ 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 $||\mu_i - y_j||$, is greater than $\gamma$, then $y_j$ is rejected, where $\mu_i$, $j = 1, \ldots, M$, are the real centers of modes. In real applications, $\mu_i$ are unknown. Thanks to the fact that a true center divides a

### 2.1. Cluster Analysis and Initial Estimation of $\Gamma$

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 $\Gamma$ 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.

Concerning 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, $h(x)$, $x \in G_{l_N}$, of the gray-level frequencies of the image, where $G_{l_N} = \{0, 1, \ldots, N - 1\}$ corresponds to the gray levels of the image. When a given image contains objects/regions having quite different gray-level values, different modes appear on 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
mode into two symmetric parts, an equivalent test can be performed without knowing the position of the true center. Reasonable values for $\gamma$ 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 [25–29]. 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 fit 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 [42] trade-off in curve-fitting problems. 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 $K_{\text{max}}$ models, where $K_{\text{max}}$ is given by the user. The choice of the best model thus becomes a model selection problem, since we find ourselves faced with competing $K_{\text{max}}$ 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 $C(K)$ ($K = 1, \ldots, K_{\text{max}}$) is given by:

$$C(K) = -aL_K + g(K),$$

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:

$$\hat{K} = \arg\min_{K} C(K).$$

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

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

The general algorithm can be given as follows:

- Choose $K_{\text{max}}$.
- for $K = 1, \ldots, K_{\text{max}}$
  - Estimate $\Gamma$ using k-means with $K$ initial clusters.
  - Compute the maximum likelihood for the estimated $\Gamma$.
  - 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, 44–46]. In our work, we use the classical AIC proposed by Akaike and given by:

$$AIC(K) = -2L_K + 2N_p,$$

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 $K_{\text{max}} = 7$.

Figure 3 shows the resulting AIC plotted against the number of clusters ($K = 1, \ldots, K_{\text{max}}$), computed from (3). The maximum likelihood technique has been used to estimate $\hat{\Gamma}_j$ ($j = 1, \ldots, K$). We can see clearly from Fig. 3 that the minimum AIC is for $K = 3$, the exact number of components in the mixture.

Figure 4 shows the plot of both the original histogram and the resulting histogram at the 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 $K_{\text{max}} = 7$. 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 $K_{\text{max}} = 7$. 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 $\beta$ and $\gamma$ were set as in the experiments performed in [23], namely $\beta = 0.97$ and $\gamma$ chosen from the $\gamma$ table.

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 algorithm, 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 \( K = K_{\text{max}} = 7 \) 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 \( K_{\text{max}} \) times, followed by the maximum likelihood to compute \( \text{AIC}(K) \) (\( K = 1, \ldots, K_{\text{max}} \)). In contrast, the EFC algorithm requires one k-means run, followed by the maximum likelihood. Thus, the EFC technique is roughly \( K_{\text{max}} \) 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 permits 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. 6a, b is the value of the second width \( \sigma_2 \). 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,

![Fig. 5. The reconstructed PDF using the resulting EFC parameters, MSE = 0.0041.](image-url)
Control the overlapping rate if we want to perform a refined evaluation of algorithms.

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 $\sigma$. It is proven that a false edge appears if and only if $2\sigma < \mu_d$, where $\mu_d$ 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 $\sigma$, $OLR < 1$ 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 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.**

1. Generate $M$, the number of components in the mixture.
2. For $i = 1, \ldots, M$, randomly generate $\mu_i$ and $\sigma_i$, such that $\mu_i < \mu_{i+1}$, $\sigma_i > 0$, $\sum_{i=1}^{M} \sigma_i = 1$.
3. Randomly generate $\sigma_1$ such that $\sigma_1 < \mu_1 - \mu_2$.
4. For $i = 2, \ldots, M$, compute $\sigma_i$ 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 ($\mu_i, i = 1, \ldots, M$) and sort them so that $\mu_i < \mu_{i+1}$. This will facilitate the execution of the algorithm. However, $\sigma_i$ should be smaller than $\mu_2 - \mu_1$; otherwise, there will be no solution for $\sigma_i$ to satisfy the non-overlapping condition. Thus, the problem is to find an upper bound for $\sigma_{i+1}$ given $\sigma_i$, in order to ensure that $OLR < 1$.

In order to develop a general algorithm, let us consider the overlap of two adjacent components $G_1(x)$ and $G_2(x)$ with $\Gamma_1 \neq \Gamma_2$. Let us denote by $S_0 = \kappa \Gamma (\mu - \sigma)$ the height at which the width $\sigma$ is located for a given component $\Gamma(x)$. For the general case of the overlapping process for two adjacent components $G_1(x)$ and $G_2(x)$,
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 \( S_{\sigma} \), for a pair of adjacent components. Here, \( S_{\sigma} \) 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 Gaussian components with different heights, OLR < 1 iff \( \hat{\sigma}_h > \sigma_i \) where \( \hat{\sigma}_h \) is the apparent width of the higher component, \( \sigma_i \) is the width of the lower component, and \( \mu_e \) 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 \( \hat{\sigma}_h = \sigma = \sigma_i \). 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 \( \Gamma_1 \) and \( \Gamma_2 \). 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 \( \sigma_2 \).

For the two cases we have:

\[
\begin{aligned}
A_{\sigma} \ln(B_{\sigma_2}) + \sigma_2 < \mu_d & \quad \text{(case 1)} \\
C_{\sigma_2} \ln(D_{\sigma_2} + \sigma_1 < \mu_d & \quad \text{(case 2)},
\end{aligned}
\]

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 \( \sigma_2 \) 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.

3.2.2. **Approximation of the Gaussian.** Consider a family of lines \( \Delta_i = \{ \Delta_{i1}, \Delta_{i2}, \Delta_{ip}, \ldots, \Delta_{ip} \} \), where \( \Delta_{i1} \) (i = 1, ..., \( p \)) is a tangent line to the point \( x = \mu - i\sigma, p(x_{\sigma}) \). If we approximate a Gaussian by the series of lines \( \Delta_i \) (i = 1, ..., \( p \)), we obtain a piece-wise linear approximation, denoted by \( \hat{g_i}(x) \) and given by

\[
\hat{g_i}(x) = i \frac{S_{\sigma}}{\sigma} e^{\frac{1}{2} \sigma^2} \left[ x - \mu + \frac{(i + 1)^2}{i} \sigma \right] \quad \text{for} \quad x \in [\mu - f(i - 1)\sigma, \mu - f(i)\sigma],
\]

where \( f(i) = \frac{(i + 1)^2}{i} \) and \( f(i - 1) = \frac{(i + 1)^2}{i} \). 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:

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

It is proven in [48] that \( E_{app}(p) \) does not depend on the width \( \sigma \). 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 \( \Delta_i \), (i = 1, ..., \( p \)) and we will have (10) as solutions to the condition of Hypothesis 1:

\[
0 < \sigma_2 \leq \frac{(2\sigma_1 - \mu_d) + \sqrt{\delta}}{2} \quad \text{(case 1)},
\]
Table 2. Comparison between EFC and AIC applied to the non-noisy sets. NCE = Number of clusters estimated

<table>
<thead>
<tr>
<th>Noise width $\sigma_n = 0$</th>
<th>CVS1</th>
<th>CVS2</th>
<th>CVS3</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCE</td>
<td>AIC</td>
<td>EFC</td>
<td>AIC</td>
</tr>
<tr>
<td>1</td>
<td>54</td>
<td>56</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>42</td>
<td>23</td>
<td>47</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Noise width $\sigma_n = 2$</th>
<th>CVS1</th>
<th>CVS2</th>
<th>CVS3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster</td>
<td>AIC</td>
<td>EFC</td>
<td>AIC</td>
</tr>
<tr>
<td>1</td>
<td>54</td>
<td>46</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>38</td>
<td>31</td>
<td>45</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>15</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

where $\delta$ is the discriminant of the quadratic form given by

$$T_1(\sigma_2) = \sigma_2^2 + (2\sigma_1 - \mu_d)\sigma_2 - \frac{k_2}{k_1}\sigma_1^2; \quad \text{and}$$

$$0 < \sigma_2 \leq \frac{-2\sigma_1 + \sqrt{\delta}}{2} \quad \text{(case 2)},$$

where $\delta$ is the discriminant of the quadratic form given by

$$T_2(\sigma_2) = \sigma_2^2 - \frac{k_2}{k_1}2\sigma_1\sigma_2 + \frac{k_2}{k_1}\mu_d\sigma_1 - k_2\sigma_1^2.$$

All deduction details are available in [48]. The conditions in (10) and (11) express the bound within which $\sigma_2$ should be picked in order that the overlap of the two adjacent components satisfies $OLR < 1$. In each step $i$ of the algorithm $i = 1, \ldots, M - 1$, (10) and (11) are used to generate $\sigma_{i+1}$. By choosing $\sigma_{i+1}$ 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 an 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 into two parts. First, we 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 CVSSs 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 noised 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 differ-
ences 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 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, non-noisy 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 $K_{\text{max}}$ 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.

<table>
<thead>
<tr>
<th>Noise width $\sigma_n = 0$</th>
<th>Noise width $\sigma_n = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVS$_1$</td>
<td>CVS$_2$</td>
</tr>
<tr>
<td>AIC</td>
<td>0.022</td>
</tr>
<tr>
<td>EFC</td>
<td>0.008</td>
</tr>
<tr>
<td>CVS$_1$</td>
<td>0.021</td>
</tr>
<tr>
<td>CVS$_2$</td>
<td>0.019</td>
</tr>
<tr>
<td>CVS$_3$</td>
<td>0.008</td>
</tr>
</tbody>
</table>

Table 4. Average MSE for each set

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 “These measures and others appearing in the literature tend to increase (or decrease) in the absence of cluster structure, so that the best solution may be indicated by an elbow or knee in a plot of the values versus the number of components. Keep in mind, however, that the occurrence of an 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

In this paper, we have described a comparison between the EFC algorithm and the AIC algorithm. The two algorithms belong to different clustering algorithm categories, i.e., direct and verificative methods. The latter category provides practical algorithms that can be used for all types of application data. However, when a priori knowledge about certain characteristics of the application data is available, direct methods can be designed in order to exploit this knowledge. For the case of gray-level image histograms, the EFC clearly performs better than the 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 $K_{\text{max}}$ times ($K_{\text{max}} = 7$ in our experiments) faster than the AIC. Note that a more extensive comparison can be conducted, including other clustering algorithms.

The comparison described in this paper 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 one-dimensional space.

REFERENCES


Elmehdi Aitnouri. Received his Electrical Engineering degree from Algiers University of Technology in 1992. Then, he joined the CDTA Research Center of Algiers, and worked until 1997 in the design of DSP ASICs architecture. Since 1997, he is pursuing his study toward a PhD in Computer Science at the University of Sherbrooke. His research interests include image processing and statistical analysis of clustering algorithms.

Shengrui Wang. Full professor at the Department of Mathematics and Computer Science of the University of Sherbrooke. He received his PhD in Computer Science from the Institut National Polytechnique de Grenoble in 1989. His research interests include neural networks, image processing and understanding, image databases, remote sensing, and expert systems.

Djemel Ziou. Received the BEng degree in Computer Science from the University of Annaba (Algeria) in 1984, and Phd degree in Computer Science from the Institut National Polytechnique de Lorraine (INPL), France, in 1991. From 1987 to 1993, he served as lecturer in several universities in France. During the same period, he was a researcher in the Centre de Recherche en Informatique de Nancy (CRIN) and in the Institut National de Recherche en Informatique et Automatique (INRIA) in France. Presently, he is an Associate Professor at the University of Sherbrooke in Canada. His research interests include image processing, computer vision, and pattern recognition.