Pattern Recognition 43 (2010) 782–795

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.de/locate/pr

# Normality-based validation for crisp clustering

Luis F. Lago-Fernández [a,*], Fernando Corbacho [b]

[a] Departamento de Ingeniería Informática, Escuela Politécnica Superior, Universidad Autónoma de Madrid, 28049 Madrid, Spain
[b] Cognodata Consulting, Calle Caracas 23, 28010 Madrid, Spain

ARTICLE INFO

*Article history:*
Received 23 February 2009
Received in revised form
28 July 2009
Accepted 22 September 2009

*Keywords:*
Crisp clustering
Cluster validation
Negentropy

ABSTRACT

We introduce a new validity index for crisp clustering that is based on the average normality of the clusters. Unlike methods based on inter-cluster and intra-cluster distances, this index emphasizes the cluster shape by using a high order characterization of its probability distribution. The normality of a cluster is characterized by its negentropy, a standard measure of the distance to normality which evaluates the difference between the cluster's entropy and the entropy of a normal distribution with the same covariance matrix. The definition of the negentropy involves the distribution's differential entropy. However, we show that it is possible to avoid its explicit computation by considering only negentropy increments with respect to the initial data distribution, where all the points are assumed to belong to the same cluster. The resulting *negentropy increment* validity index only requires the computation of covariance matrices. We have applied the new index to an extensive set of artificial and real problems where it provides, in general, better results than other indices, both with respect to the prediction of the correct number of clusters and to the similarity among the real clusters and those inferred.



## 1. Introduction

Cluster analysis, also known as unsupervised classification or exploratory data analysis, pursues the automatic partition of a data set into a finite number of natural structures, or clusters [1–3]. The goal of any clustering algorithm is to divide the data into different groups or categories, generally searching for homogeneity within each cluster and heterogeneity among different clusters, according to some similarity measure. That is, elements inside a cluster must be similar, while elements belonging to different clusters must not. Clustering algorithms are usually divided into crisp and fuzzy methods. In crisp clustering, each data point is uniquely assigned to a single cluster. On the contrary, fuzzy clustering allows each point to belong to any of the clusters with a certain degree of membership.

A common problem to both approaches is the lack of a general framework to measure the validity of the outcomes of a particular clustering method. Note that in cluster analysis the data have no labels which can guide the algorithms or inform about the reliability of the final results and, in general, different algorithms provide quite different results when applied to the same data set. Even worse, the same method may provide different data partitions depending on the initialization conditions, the data presentation order or the parameter values. Determining whether a certain partition is better or worse than another is not an easy task, and so the development of techniques that allow to assign a validity measure to the outcomes of clustering algorithms, known as cluster validation [4], has become a central issue in the field. The objective of cluster validation is to provide a quality measure, or validity index, that allows to evaluate the results obtained by a clustering algorithm. There is a large literature that deals with cluster validation from different approaches [5–12], both for crisp and fuzzy clustering.

In this paper we deal with cluster validation for crisp clustering. In this context, validity indices are generally based on some measure that relates the cluster diameters to the inter-cluster distances [13–16]. The data set is assumed to be well-partitioned if the former are small compared to the latter. This kind of criteria can give a general impression of the separation among clusters, but they ignore much of the information regarding how the data are distributed. Implicitly, these distance based criteria assume that the clusters are (hyper-)spheres, and so they can lead to error when the data distributions are very elongated (see Fig. 1). A few recent works take into account the cluster shape [17–20], in general searching for normally distributed clusters. Most of these works assume a Gaussian mixture model to describe the data. Validation, and in particular inference of the number of clusters in Gaussian mixtures has been the subject of much research [21–26]. Nevertheless, Gaussian mixtures assume a probabilistic (fuzzy) model for the data, and so these approaches cannot be directly applied to the validation of a

* Corresponding author. Tel.: +34 91 497 22 11; fax: +34 91 497 22 35.
*E-mail addresses:* luis.lago@uam.es (L.F. Lago-Fernández), fernando.corbacho@cognodata.com (F. Corbacho).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.09.018

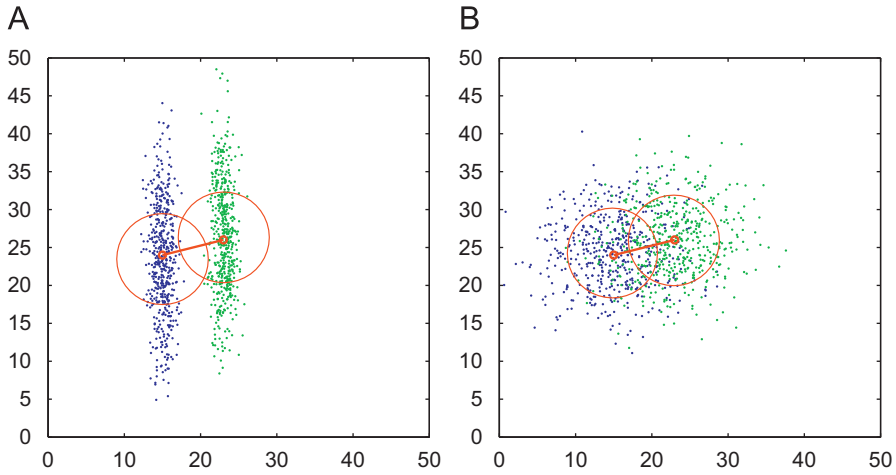


**Fig. 1.** Distance-based validity indices fail when the clusters are very elongated. (A) In spite of being clearly separable, the two clusters are merged if one considers their diameters relative to the inter-center distance. (B) Spherical equivalents of the clusters in panel A, showing the presumed overlap derived from the sphericity assumption.

crisp partition. Finally, some recent techniques based on stability criteria measure the reproducibility of clustering solutions on a second sample [27–29]. They have been applied to cluster validation mainly for gene expression data sets.

We believe that, even in the case of crisp clustering, the normal distribution is optimal as a cluster's shape. First, human vision tends to associate single clusters to Gaussian structures. When the data are described by no more than three attributes, no artificial clustering algorithm appears to perform better than visual inspection. This human ability is exploited in projection pursuit techniques [30–32], whose aim is to find "interesting" projections of the data onto a low dimensional space, such that a human analyst can visually determine the data intrinsic structure. In this context, the normal distribution is generally considered to be the least interesting, because it provides no information about possible hidden sub-structures (note that multimodal distributions showing some clustering structure are far from normality). Second, from an information theoretic point of view, the normal distribution is the one with largest entropy for a given covariance matrix [33], and so the least structured (the most uncertain) one. This means that a normally distributed cluster cannot be expected to contain additional sub-structures. Both points of view emphasize the relation between a cluster and a normally distributed set.

We propose a new validity index for crisp clustering that is based on the average normality of the clusters. In this case the main difficulty is the evaluation of the normality of a distribution in a multi-dimensional space. Many tests for multivariate normality have been proposed in the literature [34–37]. They are mainly based on the multivariate generalization of skewness and kurtosis [38,39], on the empirical characteristic function [40,41], or on estimations of the sample entropy [42,43]. The sample entropy has been widely used to measure normality in the context of projection pursuit and independent component analysis [31,32,44]. It is known that, among all the distributions with the same covariance matrix, the Gaussian is the one with the largest entropy [33]. This fact is used to test the normality of any given distribution by comparing its differential entropy to that of a normal distribution with the same covariance matrix. The difference between both quantities, known as the negentropy, is an accepted measure of distance to normality [44,45]. The negentropy of a probability distribution is always positive, and vanishes if and only if the distribution is Gaussian.

So we hereby use the negentropy to measure the normality of the clusters. Given two partitions of a data set, we will prefer the one whose clusters have lower negentropy on average. The negentropy is difficult to estimate, as it involves the computation of the differential entropy. Some approximations have been suggested, which include the use of cumulant based approximations [32], the maximum entropy principle [44], or the Edgeworth expansion [46]. Here we avoid the computation of differential entropies by considering only measures of negentropy relative to the initial distribution. We show that, by subtracting the negentropy of the initial distribution from the average negentropy of a given partition, we obtain a normality index that has all the advantages of the negentropy but avoids the explicit computation of differential entropies. We call this index the *negentropy increment* associated to the partition.

To test the negentropy increment as a cluster validity index, we have used it as the fitness function of a genetic algorithm that searches the partitions space. We have applied this method to an extensive set of synthetic clustering problems, as well as to data sets from public databases, comparing our results with those obtained by other crisp validity indices in the literature. For most of the problems considered the negentropy increment outperforms the other indices, both with respect to the prediction of the number of clusters and to the similarity among the real clusters and those inferred.

## 2. The negentropy increment as a measure of cluster validity

Our goal is to find a cluster validity index that is based on the average normality of the clusters. We consider that a normally distributed cluster is optimal, in the sense that it seems unnatural to our visual perception to perform additional partitions on it. From a more theoretical point of view, a normally distributed cluster is the most uncertain, or the least structured, one for a given covariance matrix, which suggests that no additional structures can be found on it. So we will focus on finding data partitions for which the resulting clusters are as normal as possible. Note that all the partitions considered throughout this paper are crisp partitions, that is, each data point can belong only to one of the partition regions. The normality based validation of a crisp partition summarizes into the following assumption:

**Assumption 1.** Let A and B be two partitions of a given data set. The partition A is better than the partition B if and only if the average normality of the clusters in A is higher than the average normality of the clusters in B.

The negentropy of a probability distribution is a well accepted measure of distance to normality [44,45]. For a random variable $\mathbf{x}$ with probability density function $f(\mathbf{x})$, the negentropy is given by

$$J(\mathbf{x}) = \hat{H}(\mathbf{x}) - H(\mathbf{x}) \tag{1}$$

where $\hat{H}(\mathbf{x})$ is the differential entropy of a normal distribution with the same covariance matrix as $\mathbf{x}$, and $H(\mathbf{x})$ is the differential entropy of $\mathbf{x}$:

$$H(\mathbf{x}) = -\int f(\mathbf{x}) \log f(\mathbf{x})\, d\mathbf{x} \tag{2}$$

The negentropy of a probability distribution is always equal to or greater than 0, with equality holding if and only if the distribution is Gaussian. The lower $J(\mathbf{x})$, the more Gaussian the distribution is. So our previous assumption can be rewritten as:

**Assumption 2.** Let A and B be two partitions of a given data set. The partition A is better than the partition B if and only if the average negentropy of all the clusters in A is lower than the average negentropy of all the clusters in B.

We will use this assumption to derive our cluster validity measure as follows. Imagine that some clustering algorithm provides a crisp partition of the space into a set of $k$ nonoverlapping regions $\{\Omega_1, \Omega_2, \ldots, \Omega_k\}$ (see Fig. 2). We will use the region $\Omega_0$ to refer to the original space with no partitions. We can use the average negentropy over all the regions, which we call $\bar{J}(\mathbf{x})$, as the validity index for the partition:

$$\bar{J}(\mathbf{x}) = \sum_{i=1}^{k} p_i J_i(\mathbf{x}) \tag{3}$$

where $p_i$ is the a priori probability of $\mathbf{x}$ falling into the region $\Omega_i$, and $J_i(\mathbf{x})$ is the negentropy of $\mathbf{x}$ in the region $\Omega_i$. The expression in (3) is the formalization of Assumption 2. The lower $\bar{J}(\mathbf{x})$, the better (more Gaussian on average) the partition is. We can add any constant to $\bar{J}(\mathbf{x})$ without affecting this result so, instead of (3), we will consider the index:

$$\Delta J = \bar{J}(\mathbf{x}) - J_0(\mathbf{x}) \tag{4}$$

where $J_0(\mathbf{x})$ is the negentropy of $\mathbf{x}$ if one single region $\Omega_0$ is considered, which is a constant for the problem. We call this index the *negentropy increment* after the partition. It measures the change in negentropy when we make a partition on the data set. If $\Delta J < 0$ then we are gaining normality (losing negentropy) with the partition, while if $\Delta J > 0$ we are losing normality (gaining negentropy). Given two different partitions of a data set, we will select the one for which $\Delta J$ is lower.

As an example, let us consider the data shown in Fig. 3. Panel A shows a data set consisting of 250 normally distributed points in two dimensions. We have performed different partitions of this data set by using vertical separators, at positions $x_c$, that divide the plane into the two nonoverlapping regions $\{x < x_c\}$ and $\{x \geq x_c\}$. For each of these partitions we have computed the negentropy increment $\Delta J$, which is plotted in panel B versus $x_c$. Note that $\Delta J$ is positive or zero for all the considered partitions, which indicates that none of them contributes any gain in normality. So, according to the negentropy increment criterion, it is better to consider one single cluster for this data set. In panel C we show a second data set that consists of two different groups of 250 normally distributed points each. We have performed the same kind of partitions by vertical separators, and computed the negentropy increment as before. The resulting plot of $\Delta J$ vs $x_c$ is shown in panel D. Now the negentropy increment takes negative values for some partitions, which are then preferred, in terms of normality, to the initial situation with no partition. Note that the minimum value of $\Delta J$ (maximum gain in normality) corresponds to partitions that separate between the two original clusters.

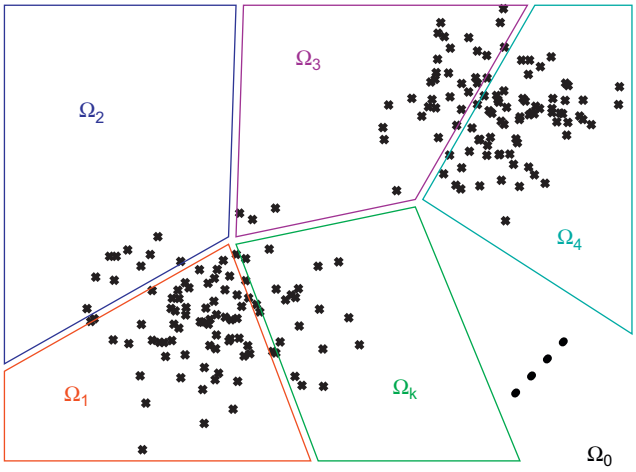


**Fig. 2.** Partition of the data set into $k$ nonoverlapping regions.

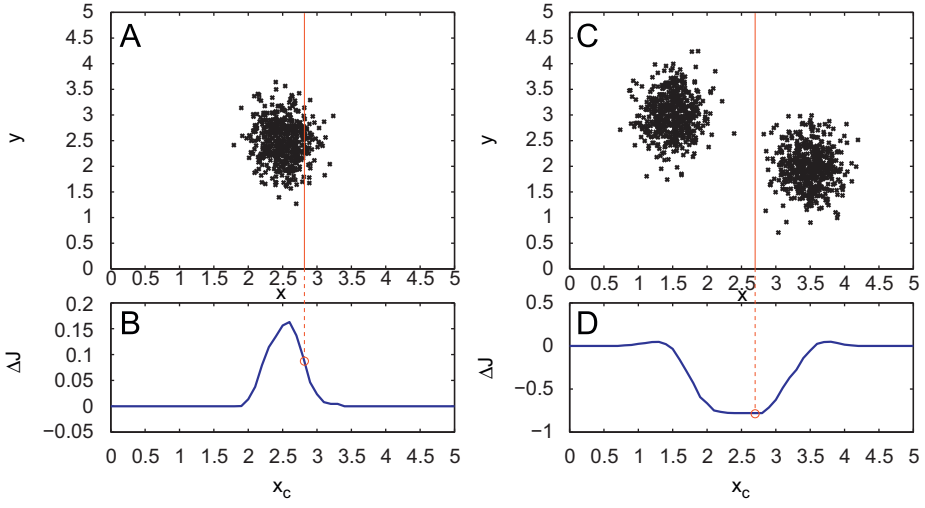


**Fig. 3.** Illustration of the negentropy increment criterion for cluster validation. Panels A and C show two different data sets in two dimensions. For each data set, different partitions are performed by vertical separators that divide the plane into the two nonoverlapping regions $\{x < x_c\}$ and $\{x \geq x_c\}$. Panels B and D show the plots of $\Delta J$ versus $x_c$ for the data sets in A and C, respectively. The dotted lines show the correspondence between an example partition in A (C) and the value of $\Delta J$ plotted in B (D).

The results obtained using the negentropy validity index are in accordance with the observation of one single cluster in Fig. 3 A and two different clusters in Fig. 3 C. The measure appears as a good candidate index for cluster validation. There is only a technical issue left, related to the calculation of the differential entropy, which we tackle in the next section.

## 3. Computation of the negentropy increment

The computation of the negentropy in (1) requires the evaluation of the differential entropy $H(\mathbf{x})$. The precise evaluation of $H(\mathbf{x})$ is in general a difficult task, as it needs the estimation of probability distributions. As soon as the dimension increases, the estimation of a probability distribution becomes impossible unless we have an infinite amount of data (curse of dimensionality). To overcome this problem some approximations to the negentropy have been proposed [32,44,46] in the context of projection pursuit and independent component analysis.

Here we follow a different approach, which avoids the explicit computation of the differential entropy by using a validity measure that discounts the negentropy of the original data set from the average negentropy of the partition being evaluated ($\Delta J$ index in (4)). When we expand the expression for $\Delta J$ all the differential entropies, except those for normal distributions, cancel out. For a crisp partition, the negentropy increment can then be expressed as

$$\Delta J = \sum_{i=1}^{k} p_i \hat{H}_i(\mathbf{x}) - \hat{H}_0(\mathbf{x}) - \sum_{i=1}^{k} p_i \log p_i \tag{5}$$

where $\hat{H}_i$ is the entropy of a normal distribution with the same covariance matrix as $\mathbf{x}$ in the region $\Omega_i$. A full derivation of this expression can be found in Appendix A. Note that it is not an approximation, but the exact expression for the negentropy increment of a crisp partition.

As shown in (5), the negentropy increment is a contribution of three terms. First, the average differential entropy of $\mathbf{x}$ over all the regions generated by the partition, assuming normality. Second, the negative of the differential entropy of $\mathbf{x}$ considering one single region $\Omega_0$, and also assuming normality. And third, the discrete entropy that is introduced as a consequence of the partition. It can be shown that, under the assumption of normality for $\mathbf{x}$, the previous expression is equivalent to the overall increment in entropy after the partition. So, the validity condition $\Delta J < 0$ favors partitions which decrease the overall entropy of the system, thus introducing some kind of order. This is in apparent contradiction with the physical intuition that the entropy can never decrease. However, under the normality assumption, there is no contradiction: the normal distribution that explains the data in the original space can have larger entropy than the set of normal distributions that explain the data in each of the regions after performing the partition.

The entropy of a normal distribution has a closed expression in terms of the covariance matrix. This allows to rewrite (5) as

$$\Delta J = \frac{1}{2}\sum_{i=1}^{k} p_i \log|\Sigma_i| - \frac{1}{2}\log|\Sigma_0| - \sum_{i=1}^{k} p_i \log p_i \tag{6}$$

where $\Sigma_i$ is the covariance matrix of $\mathbf{x}$ in the region $\Omega_i$. Note that to evaluate this final expression we only need to compute the determinants of the covariance matrices for each region. Additionally, the prior probabilities $p_i$ are approximated by the fraction of points that fall into each region. This index can be applied as a general tool to validate the outcome of any crisp clustering algorithm, and also to compare solutions provided by different algorithms for a single problem. The rest of the paper is dedicated to show the performance of $\Delta J$, in comparison with other cluster validity measures, on a variety of test data sets. A brief analysis of the behavior of the new index in cases of noise and reduced number of data points is provided in Appendix B.

## 4. Clustering algorithm

We want to test the $\Delta J$ cluster validity measure against other crisp validity indices that are frequently used in the literature. In particular, we consider the Davies–Bouldin (DB) index [14], the Dunn index [13,15], the PBM index [16] and the SIL index [47]. The DB and the Dunn indices were found among the best in a study that compared 23 validity indices on 12 data sets that consisted of bivariate Gaussian mixtures [48]. On the other hand, the PBM index was shown to outperform the other two in [16]. The SIL index is qualitatively different from the others and is included here as an additional reference. We use a genetic algorithm to search for the partition $\Gamma$ that optimizes a particular validity index $I(\Gamma)$ for the problem being addressed. The partitions we consider consist of convex nonoverlapping regions delimited by linear separators, such as those sketched in Fig. 2. The outline of the genetic algorithm follows (details on the implementation, including a comparison of the execution times for the different indices, can be found in Appendix C).

Let us consider a clustering problem in $d$ dimensions, and a partition of the parameter space into $k$ nonoverlapping regions, $\Gamma = \{\Omega_1, \Omega_2, \ldots, \Omega_k\}$. We will consider only partitions that can be expressed as a $d$-dimensional Voronoi diagram around $k$ centers. That is, any partition is fully characterized by the set of centers $\{\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_k}\}$, and the region $\Omega_i$ consists of all the points that are closer to $\mathbf{p_i}$ than to any other center. In the genetic algorithm implementation we codify each region center as a set of $10 \times d$ bits (10 bits per coordinate), and so the full partition can be coded as a binary string of length $10 \times d \times k$. We use the PGAPack genetic algorithm library [49] with the default mutation and crossover operators. In all the trials performed the population size is set to 500 individuals, each one representing a different partition, which are randomly initialized. The evaluation of any partition $\Gamma$ is done by using the validity index $I(\Gamma)$ as fitness function. The algorithm is run for 250 iterations, and the best partition at the end is used as the solution for a particular run. In general (unless otherwise specified) we make 20 different runs for each $k$, and select the solution that provides the best index value.

The following validity indices are considered:

### 4.1. Davies–Bouldin (DB) index

The Davies–Bouldin index measures the relation between within-cluster scatter and inter-cluster separation [14]. Let $k$ be the number of clusters, $|C_i|$ the number of samples in cluster $C_i$ and $\mathbf{p_i}$ the center of cluster $C_i$. The scatter is defined, for each cluster, as

$$S_i = \frac{1}{|C_i|}\sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{p_i}\| \tag{7}$$

It represents the average Euclidean distance to the cluster center. For each cluster, a measure of the overlap with other clusters is also defined as

$$R_i = \max_{j \neq i} \frac{S_i + S_j}{d_{ij}} \tag{8}$$

where $d_{ij}$ is the Euclidean distance between the cluster centers, $d_{ij} = \|\mathbf{p_i} - \mathbf{p_j}\|$. The Davies–Bouldin index is defined in terms of $R_i$ as

$$DB = \frac{1}{k}\sum_{i=1}^{k} R_i \tag{9}$$

The best partition is the one that minimizes DB. Note that, as far as we use Euclidean distances, this index is assuming that all the clusters are spherical. This assumption may lead to poor results when the clusters are very elongated (recall Fig. 1).

### 4.2. Dunn index

The Dunn index is defined as a ratio between minimum inter-cluster distance and maximum cluster diameter [15]:

$$V = \frac{\min_{i \neq j} \delta(C_i, C_j)}{\max_i \Delta_i} \tag{10}$$

where $\delta(C_i, C_j)$ is the distance between clusters $C_i$ and $C_j$ and $\Delta_i$ is the diameter of cluster $C_i$. There are a variety of Dunn's indices depending on how these quantities are defined [13]. We consider here the $V_{33}$ index, where the cluster diameter is defined as

$$\Delta_i = \frac{2}{|C_i|} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{p_i}\| \tag{11}$$

and the inter-cluster distance is defined as

$$\delta(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} \|\mathbf{x} - \mathbf{y}\| \tag{12}$$

The best partition is the one that maximizes the index. Note that, as before, the use of Euclidean distances implies the same assumption about sphericity.

### 4.3. PBM index

The Pakhira–Bandyopadhyay–Maulik index [16] is constructed to ensure the formation of a small number of compact clusters with a large separation between at least two of them. It is defined as

$$PBM = \left( \frac{1}{k} \cdot \frac{E_0}{E} \cdot D \right)^2 \tag{13}$$

The variable $E$ measures the total within-cluster scatter:

$$E = \sum_{i=1}^{k} |C_i| S_i \tag{14}$$

where $S_i$ is the scatter for cluster $C_i$ as defined in (7). The variable $E_0$ is the total scatter considering all the samples belonging to one single cluster:

$$E_0 = \sum_{\mathbf{x}} \|\mathbf{x} - \mathbf{p}\| \tag{15}$$

where $\mathbf{p}$ is the average of all $\mathbf{x}$. Finally, $D$ is the maximum distance between cluster centers:

$$D = \max_{i \neq j} \|\mathbf{p_i} - \mathbf{p_j}\| \tag{16}$$

This index is maximized in order to find the best partition. An important difference with the previous indices is that the PBM index uses the maximum inter-cluster separation. Once a partition into compact and well separated clusters has been found, $D$ remains almost constant with further partitioning while the quotient $E_0/E$ can increase. This makes necessary the introduction of the factor $1/k$ to compensate for this growth in the index. Overall, the three factors compete with each other critically. The PBM index also assumes that the clusters are spherical.

### 4.4. SIL index

The silhouette index [47] is based on the concept of silhouette width, which measures the confidence on the membership of each single data point with respect to its cluster. The silhouette width for the data point $\mathbf{x}$ is defined as

$$s(\mathbf{x}) = \frac{b(\mathbf{x}) - a(\mathbf{x})}{\max(a(\mathbf{x}), b(\mathbf{x}))} \tag{17}$$

where $a(\mathbf{x})$ is the average distance between $\mathbf{x}$ and the rest of points within its cluster, and $b(\mathbf{x})$ is the minimum (across clusters) average distance between $\mathbf{x}$ and all the points belonging to any of the other clusters. The silhouette width is close to 1 when the point is well clustered, it is about 0 when the point lies in between two clusters, and it is almost $-1$ when the point is assigned to a wrong cluster.

By averaging the silhouette width over the whole data set, we obtain the SIL validity index as

$$SIL = \frac{1}{n_x} \sum_{\mathbf{x}} s(\mathbf{x}) \tag{18}$$

where $n_x$ is the number of points in the data set. The best clustering partition is selected by maximizing the SIL index.

### 4.5. Negentropy index

The last validity index we consider is the negentropy increment $\Delta J$, as defined in (6).

## 5. Test data sets

We have performed different experiments on both synthetic and real data sets. The synthetic examples are composed of randomly generated Gaussian clusters in two and three dimensions. The real data correspond to three well known machine learning problems from the UCI database [50]. In the following we include a brief description of each problem.

### 5.1. Gaussians 2D

As a first test we consider a set of 500 randomly generated problems in two dimensions. In every problem the number of clusters, $n$, is between 1 and 5, each cluster consisting of 200 points randomly extracted from a normal distribution whose parameters are also randomly selected. The set of points $(x, y)$ belonging to a particular cluster are generated as follows. First, two random numbers, $\mu_x$ and $\mu_y$, are extracted from a uniform distribution in the interval $(0, 10)$, and other two, $\sigma_x$ and $\sigma_y$, are extracted from a uniform distribution in the interval $(0, 1)$. Then $x$ is extracted from the normal distribution $N(\mu_x, \sigma_x)$ and $y$ is extracted from $N(\mu_y, \sigma_y)$. Finally a rotation by an angle $\theta$, with center at $(\mu_x, \mu_y)$, is applied to the points $(x, y)$. The angle $\theta$ is randomly selected from a uniform distribution in $(0, 2\pi)$. There are 100 data sets for each value of $n$. Fig. 4 shows some of the data sets for $n = 3$, 4 and 5. Note that, as the number of clusters increases, the overlap among them is higher in general. This fact makes the problems more difficult for higher $n$.

### 5.2. Gaussians 3D

The second group of artificial data sets is an extension of the previous one, which increases the difficulty of the problems in the following terms. First, the dimension is increased to three. Second, the data sets can be made of up to eight clusters. And third, the number of points per cluster is reduced to 100. The reduction in the number of points, together with the dimension increase, makes the estimation of normality a more difficult task. On the other hand, by increasing the number of clusters we increase the overlap among them, which

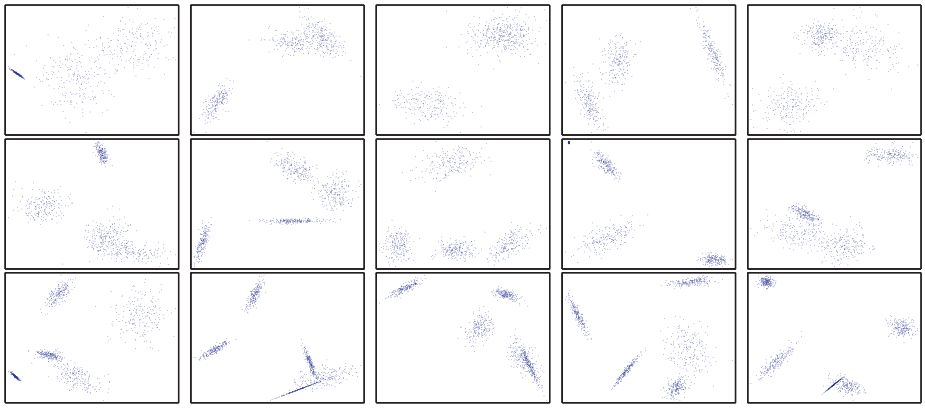

**Fig. 4.** Some examples of the 2D data sets generated to test the cluster validity measure. The top row shows some sets with tree clusters ($n=3$). The middle row shows sets consisting of four clusters ($n=4$). The bottom row shows data sets with five clusters ($n=5$). The clusters have been randomly generated and present different shapes, sizes and orientations. Note that in some cases two or more clusters can overlap, which increases the difficulty of the problem.

also increments the problem complexity. The number of clusters, $n$, in each of the data sets is between 2 and 8, and each cluster contains 100 points. As before, the coordinates $x$, $y$ and $z$ of the points belonging to a cluster are extracted from a normal distribution whose parameters are randomly chosen in the following ranges: $\mu \in (0, 10)$, $\sigma \in (0.5, 1)$. To provide an arbitrary orientation for the clusters, a random rotation is applied to each of them, as for the 2D case. We have generated 20 different sets for each $n$, which makes 140 data sets in total.

### 5.3. UCI database problems

The three real data sets we consider are the Iris data set [51], the Wisconsin Breast Cancer data set [52], and the Wine data set [53]. Although they are essentially supervised classification problems, we will use them here in an unsupervised manner (no information about the classes is available to the clustering algorithm). However, in order to test the quality of our results on these problems, the real classes will be considered as the best possible clustering partition.

#### 5.3.1. Iris

The Iris data set consists of 150 points in a 4-dimensional attribute space. The four attributes are real-valued, and represent the petal and sepal lengths and widths of Iris plants belonging to three different species. There are 50 instances of each class. It is known that one of the classes is linearly separable from the other two, which are not linearly separable from each other.

#### 5.3.2. Wisconsin Breast Cancer

This data set consists of 699 samples of cytological analysis of breast tumors belonging to two classes, benign or malignant. Each sample is characterized by a set of nine integer attributes that describe different cell properties. We consider here only the 683 patterns without missing values. Of these, 444 are of class benign and 239 are of class malignant. We have reduced the dimension of the problem by using only the first four principal components, which account for the 85.27% of the total variance.

#### 5.3.3. Wine

The last data set we consider contains data from chemical analyses of wines from three different classes. There are 178 samples characterized by 13 continuous attributes that represent the quantities of different constituents found in the wines. There are 59 samples of the first class, 71 samples of the second class, and 48 samples of the third class. We have performed a PCA transformation in order to reduce the dimension to the first six principal components, which account for the 85.10% of the total variance.

## 6. Results

We have run several trials of the clustering algorithm for each of the problems and each of the validity indices. The results are evaluated using two different error measures. First, we compare the actual number of clusters in the problem with the number of regions in the best partition provided by the algorithm. For the problems that consist of a collection of data sets (Gaussian data in 2D and 3D), we also compute the percentage of sets for which the algorithm obtains the correct number of regions. A good result in this sense does, however, not guarantee a good correspondence between the partition regions and the real clusters. So we use as a second error measure the discrepancy between the real clusters and the predicted regions, given by the entropy distance [54]:

$$D_H = H(c|r) + H(r|c) \tag{19}$$

This distance is computed as a sum of two entropies. The first one measures the uncertainty in the cluster given the region. The second one measures the uncertainty in the region given the cluster. Both of them are sources of error and should be minimized in any good partition. Note that, when there is a perfect correspondence between clusters and regions, the entropy distance $D_H$ is 0.

The steps we follow and the kind of tests we perform are essentially the same for all the problems considered. So we will provide a full description only for the Gaussian data in 2D, and we will summarize the results for the rest of data sets. We describe in detail the analysis for the $\Delta J$ validity index. Equivalent analyses are performed for the rest of indices.

### 6.1. Gaussians 2D

For each of the 500 data sets, we have performed 20 different runs of the algorithm for each value of $k$, ranging between 1 and 9 regions. In Fig. 5A we show the results for a particular data set with four clusters. The graphic plots the value of $\Delta J$ versus the number of regions $k$. Every point in the plot represents the best (minimum) value obtained over 20 runs of the genetic algorithm. The optimal number of regions is selected as that which minimizes $\Delta J$. For this particular example, $k_{opt} = k_{min} = 4$ is obtained. In Fig. 5 B we show the corresponding partition. If we perform the same analysis for all the data sets, selecting for each

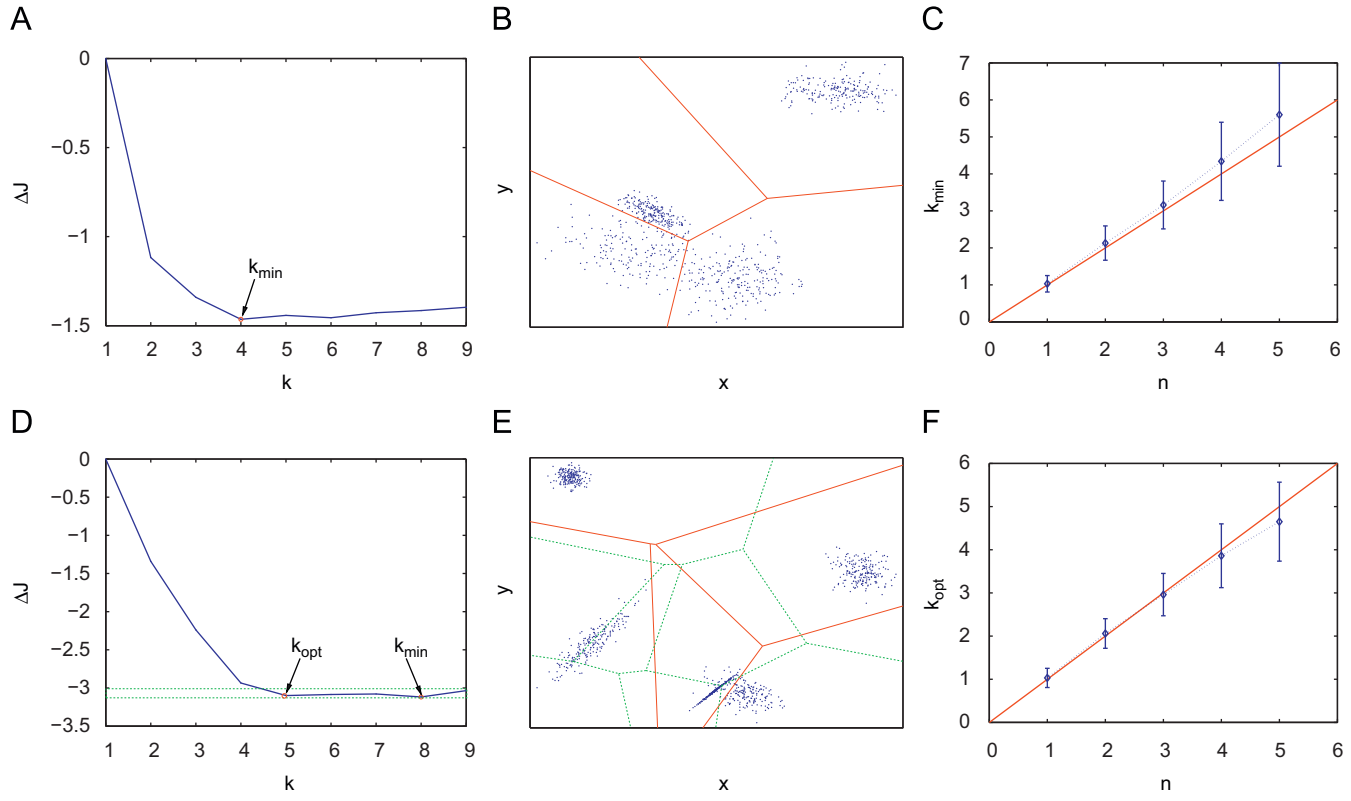


**Fig. 5.** Results obtained with the $\Delta J$ index for two different problems with $n=4$ and n=5 from the Gaussians 2D set. (A) $\Delta J$ versus $k$ for the problem with $n=4$. Each point shows the minimum $\Delta J$ obtained in 20 runs of the genetic algorithm. The point marked as $k_{min}$ represents the number of regions that provides the lowest $\Delta J$. (B) Partition of the data set into $k_{min}=4$ regions. In spite of the partial overlap among three of the clusters, the partition is able to separate all the four clusters in the data set. (C) Average $k_{min}$ versus number of clusters $n$. Each point has been calculated as the average over the 100 data sets with the same number of clusters. (D) $\Delta J$ versus $k$ for the problem with $n=5$. Each point shows the minimum $\Delta J$ obtained in 20 runs of the genetic algorithm. The point marked as $k_{min}$ represents the number of regions that provides the lowest $\Delta J$. The point marked as $k_{opt}$ represents the minimum number of regions that provides a $\Delta J$ within a 95% of its minimum. (E) Partitions of the data set into $k_{min}=8$ regions (dotted line) and into $k_{opt}=5$ regions (solid line). (F) Average $k_{opt}$ versus number of clusters $n$. Each point has been calculated as the average over the 100 data sets with the same number of clusters.

one the partition with minimum $\Delta J$, we can compute on average the predicted number of regions, $k_{min}$, given the real number of clusters, $n$. This is shown in Fig. 5 C, where each point has been calculated as the average over the 100 data sets with the same $n$. Note that the number of regions slightly overestimates the number of clusters, more clearly as this number increases.

Fig. 5 D provides a hint to understand why this overestimation is produced and how to avoid it. It plots $\Delta J$ versus $k$ for a different data set, for which the algorithm fails to find the correct number of regions. The data set presents five clusters, but the algorithm detects eight regions. Note, however, that the graphic presents an elbow at $k$ between 4 and 5, which could be used to predict the correct number of regions with more accuracy. By choosing $k$ close to this elbow, we can get almost the same $\Delta J$ with a simpler partition. We do this by selecting $k_{opt}$ as the minimum number of regions for which $\Delta J$ lies within a 95% of the absolute minimum. In Fig. 5 D, the point marked as $k_{min}$ corresponds to the minimum of $\Delta J$, which gives rise to eight regions. The point marked as $k_{opt}$ corresponds to the simplest partition with $\Delta J$ within the 95% of the minimum, which gives rise to five regions. In Fig. 5 E we show this partition (solid line), which detects the clusters in the data set quite accurately. For the sake of comparison, the figure also shows the partition into $k_{min}=8$ regions (dotted line). In Fig. 5 F we plot again the average number of regions versus the number of clusters, with the previous correction. Now the prediction is more accurate, and the number of clusters is no longer overestimated. In fact there is a slight underestimation as $n$ grows, but this can be expected since the presence of more clusters implies more overlap. For the rest of the paper we will use this method to select the optimal number of regions, $k_{opt}$. In Fig. 6 we show the selected partitions for all the data sets shown in Fig. 4. Note that the algorithm performs quite well even in cases where two clusters are partially overlapping.

The same kind of analysis has been performed for the rest of validity indices. In Fig. 7 we show the number of regions, $k_{opt}$, versus the actual number of clusters, $n$, for each of them. The following conclusions can be extracted from this figure:

- Our index is the only one which provides the correct partition for data sets with one single cluster. All the other methods tend to generate two or more regions, inventing artificial clusters.
- For data sets with two clusters all the methods, except the PBM, produce a satisfactory solution. This could be expected, as there is in general enough space not to have much overlap (so the problems are quite easy to solve). However, the PBM index fails, tending to produce on average more than two regions. The reason may be that for such a small number of clusters the factor $1/k$ cannot compensate the increase in the index produced by making an additional partition.
- For data sets with three, four and five clusters all the methods tend to underestimate the number of clusters. This is due to a higher overlap, which increases with the number of clusters. The indices PBM and $\Delta J$ behave similarly and beat the others for these problems.
- Our index is the only one that shows a good performance for the full range of $n$.

To quantify these observations we have computed, for each index and each number of clusters $n$, the percentage $R$ of data

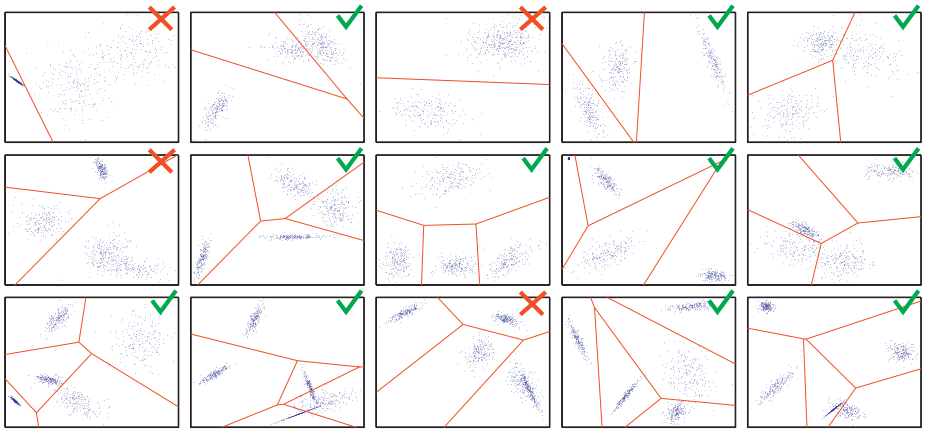

**Fig. 6.** Selected partitions, according to $\Delta J$, for all the data sets shown in Fig. 4. Partitions marked with a tick are correct. Those marked with a cross are wrong. The partitions identify the correct clusters in 11 of the 15 sets.

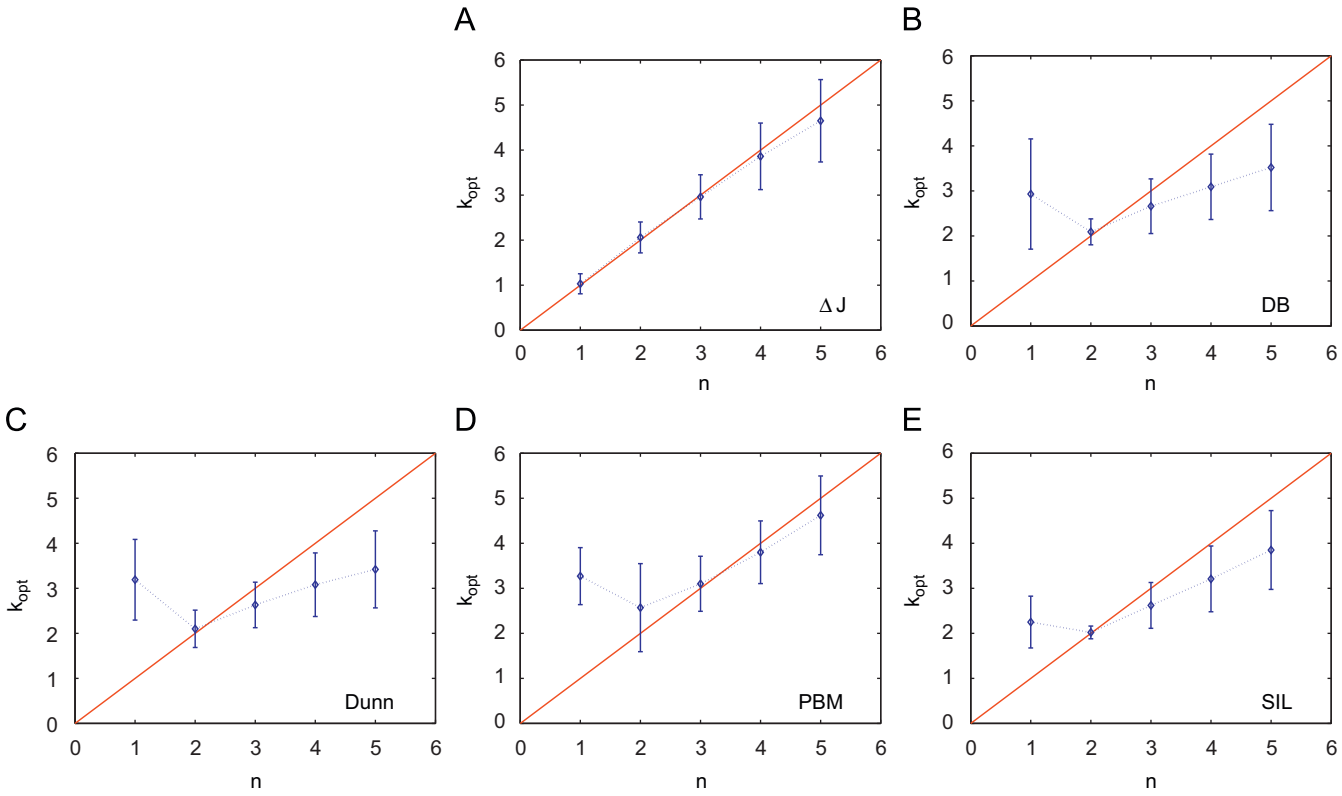


**Fig. 7.** Average number of regions versus number of clusters in the data set for all the validity indices considered. (A) $\Delta J$ index. (B) DB index. (C) Dunn index. (D) PBM index. (E) SIL index.

sets for which the algorithm predicts the correct number of clusters ($k_{opt} = n$). The results are summarized in Table 1 (left side). Our validity index provides the best results for all the cases except $n = 2$, for which the SIL index obtains the highest score (98%). Note, however, that in this case the $\Delta J$, DB and Dunn indices also reach values over 90%. The high accuracy of the indices for $n = 2$ was previously observed in Fig. 7, and is presumably due to the low overlap. Also note that the $\Delta J$ index finds the correct solution for 98% of the problems with one single cluster, while the other indices always produce partitions into two or more regions.

Finally, in order to measure the discrepancy between the real clusters and the predicted regions, we have computed the entropy distance $D_H$ (19) between regions and clusters. In Fig. 8A we plot the average $D_H$ versus the number of clusters $n$ for each of the validity indices. Each point in the plot is an average over 100 data sets. The $\Delta J$ index shows the lowest values for the full range of $n$. Note that this is true even for the case ($n = 2$) where other indices provided a higher accuracy predicting the correct number of clusters. We may conclude that the proposed cluster validity index not only predicts the number of clusters with high accuracy, but also the regions it gives rise to are closer to the real clusters than those generated by any other of the considered indices.

### 6.2. Gaussians 3D

For this set of problems we have performed 10 runs of the genetic algorithm for each data set, each validity index

**Table 1**
Percentage of data sets for which the number of clusters is correctly predicted by the algorithm, using the different validity indices.

| $n$ | Gaussians 2D (%) | | | | | Gaussians 3D (%) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\Delta J$ | DB | Dunn | PBM | SIL | $\Delta J$ | DB | Dunn | PBM |
| 1 | **98** | – | – | – | – | – | – | – | – |
| 2 | 91 | 91 | 93 | 66 | **98** | **100** | **100** | **100** | 95 |
| 3 | **87** | 60 | 61 | 74 | 60 | **95** | 75 | 80 | 90 |
| 4 | **66** | 31 | 29 | 57 | 35 | **70** | 20 | 25 | 60 |
| 5 | **47** | 14 | 12 | 45 | 30 | 35 | 15 | 20 | **55** |
| 6 | – | – | – | – | – | 10 | 20 | 20 | **25** |
| 7 | – | – | – | – | – | 10 | 5 | 10 | **20** |
| 8 | – | – | – | – | – | **15** | 5 | 5 | **15** |

Left side: Gaussian data in 2D, each value is calculated using 100 data sets. Right side: Gaussian data in 3D, each value is calculated using 20 data sets. Values in bold are the best results for a given $n$.

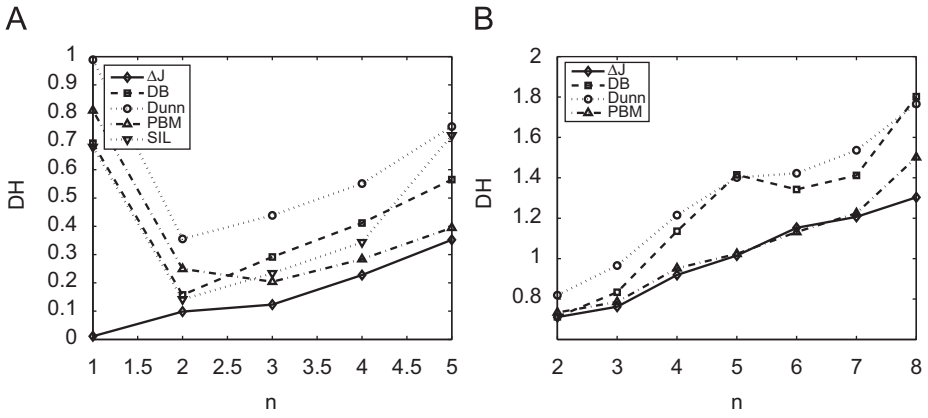


**Fig. 8.** Entropy distance between clusters and partitions, $D_H$, versus number of clusters, obtained with the validity indices considered in the paper. (A) Gaussian data in 2D. All the points are averages over 100 data sets. (B) Gaussian data in 3D. All the points are averages over 20 data sets.

(except the SIL index[1]), and each $k$. We consider $k$ values ranging from 1 to 9. The procedure for selecting the number of regions $k_{opt}$ and evaluating the different indices was described before for the 2D case. The results are summarized in Table 1 (right side) and Fig. 8 B. We have not evaluated the indices on single cluster problems ($n=1$) because, as shown for the 2D case, only the $\Delta J$ index can deal with them. In Table 1 (right side) we show the values of $R$ obtained with the four considered indices for each of the different problems. The $\Delta J$ index shows the best performance for data sets with up to four clusters, and the PBM index is the best for problems with more than four clusters. If we look at the similarity between clusters and partitions, however, our validity index slightly outperforms the PBM. Fig. 8 B plots $D_H$ versus $n$ for the 4 validity indices. We observe that the $\Delta J$ and the PBM indices behave similarly, displaying the lowest values for all $n$. As before, our validity index provides the best results for most of the data sets, both with respect to the number of clusters and to the similarity between clusters and regions. However, in this case the results provided by the PBM index are comparable, and in some cases even better. It seems that the dimension increase and the reduction in the number of points per cluster affect the $\Delta J$ index more dramatically than the others.

[1] The SIL index has been excluded from this analysis because its execution time was excessive (see Appendix C). The results obtained for the 2D case and for the UCI database problems seem sufficient to provide a fair comparison to the other indices.

**Table 2**
Entropy distance $D_H$ between clusters and partitions, and number of predicted regions $k_{opt}$, provided by the different validity indices for the Iris, Cancer and Wine problems.

| Problem | $\Delta J$ | DB | Dunn | PBM | SIL |
|---|---|---|---|---|---|
| Iris | $\mathbf{D_H = 0.19}$ | $D_H = 0.46$ | $D_H = 0.76$ | $D_H = 0.52$ | $D_H = 1.10$ |
| 3 classes | $\mathbf{k_{opt} = 3}$ | $k_{opt} = 2$ | $k_{opt} = 2$ | $\mathbf{k_{opt} = 3}$ | $k_{opt} = 2$ |
| Cancer | $\mathbf{D_H = 0.39}$ | $D_H = 0.78$ | $D_H = 0.74$ | $\mathbf{D_H = 0.39}$ | $D_H = 0.78$ |
| 2 classes | $\mathbf{k_{opt} = 2}$ | $k_{opt} = 3$ | $\mathbf{k_{opt} = 2}$ | $\mathbf{k_{opt} = 2}$ | $\mathbf{k_{opt} = 2}$ |
| Wine | $\mathbf{D_H = 0.56}$ | $D_H = 0.67$ | $D_H = 0.63$ | $D_H = 0.62$ | $D_H = 1.24$ |
| 3 classes | $\mathbf{k_{opt} = 3}$ | $\mathbf{k_{opt} = 3}$ | $\mathbf{k_{opt} = 3}$ | $k_{opt} = 2$ | $\mathbf{k_{opt} = 3}$ |

A $D_H$ value shown in bold is the minimum across indices for a given problem. Values of $k_{opt}$ in bold are matches with the actual number of clusters in the problem.

### 6.3. UCI database problems

Finally we present the results for the Iris, the Wisconsin Breast Cancer and the Wine problems. For all of them we consider partitions with $k$ ranging from 1 to 9, and run the genetic algorithm 20 times for each $k$ and each validity index. The selection of the optimal partition is done as before. The selected partitions are evaluated by comparing with the real classes in terms of number of regions and entropy distance. Table 2 summarizes the results. Note that the $\Delta J$ index is the only one which predicts the correct number of regions for the three problems. Additionally, it provides the lowest $D_H$ in all the cases, which implies a higher correlation between the real classes and the predicted regions.

To add some visual intuition to the results regarding $D_H$, we show in Figs. 9–11 the class distributions for each region in the optimal partition obtained with each of the indices. For the Iris problem (Fig. 9) only the $\Delta J$ and the PBM indices predict the correct number of regions ($k_{opt}=3$). Both of them find one region that completely corresponds to one of the classes (the one which is linearly separable from the other two). However, the other two regions are better related to the real classes for the $\Delta J$ index.

For the Wisconsin Breast Cancer problem (Fig. 10), all the indices except the DB find the correct number of regions ($k_{opt}=2$). In all the cases the two regions mix samples from both classes (note that the problem is not linearly separable), but the partitions derived from $\Delta J$ and PBM appear to be better in terms of $D_H$.

Finally, we show in Fig. 11 the class distributions for the Wine problem. Now they are the $\Delta J$, DB, Dunn and SIL indices which predict the correct number of regions ($k_{opt}=3$). The partition derived from $\Delta J$ is the only one for which all the regions mix samples belonging to two different classes only, and this gives the highest correlation with the real classes according to $D_H$.

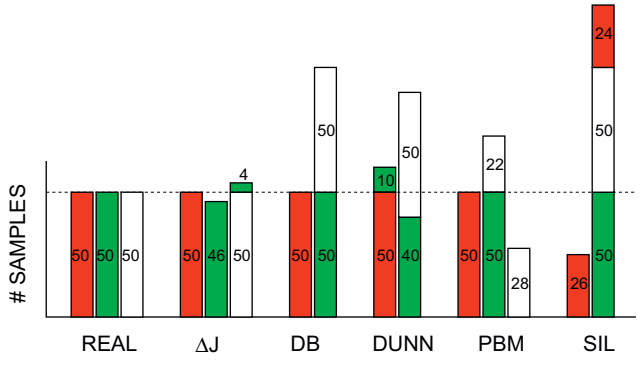


**Fig. 9.** Iris data set. Class distributions for the problem (marked as REAL), and for each region in the final partition obtained with each of the considered validity indices. The $\Delta J$ and PBM indices find the correct number of regions ($k_{opt}=3$). However, the regions seem to be more correlated with the real classes for the $\Delta J$ index. The DB, Dunn and SIL indices find partitions of only two regions.

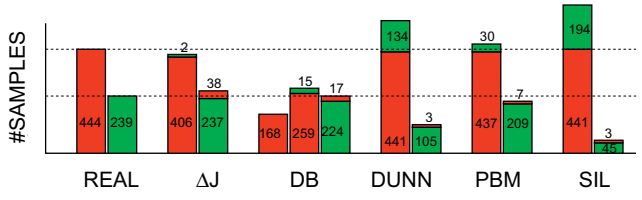


**Fig. 10.** Cancer data set. Class distributions for the problem (marked as REAL), and for each region in the final partition obtained with each of the considered validity indices. The $\Delta J$, Dunn, PBM and SIL indices find the correct number of regions ($k_{opt}=2$). The regions seem to be more correlated with the real classes for the $\Delta J$ and the PBM indices. The DB index finds a partition into three regions.

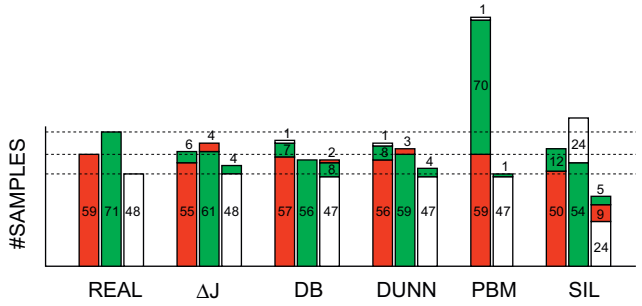


**Fig. 11.** Wine data set. Class distributions for the problem (marked as REAL), and for each region in the final partition obtained with each of the considered validity indices. The $\Delta J$, DB, Dunn and SIL indices find the correct number of regions ($k_{opt}=3$). The PBM index finds a partition with only two regions.

## 7. Discussion

In this paper we have introduced a new crisp cluster validity index that is based on the average normality of the clusters. The normality of a cluster is measured by means of its negentropy, i.e., the difference between the cluster's entropy and the entropy of a normal distribution with the same covariance matrix. To avoid the explicit calculation of the differential entropy, we subtract the negentropy of the original data distribution, where all the points are assumed to belong to the same cluster. We show that, for crisp partitions (no overlap among clusters), the final form of the validity index only requires the computation of the determinants of the covariance matrices and the prior probabilities for each partition region. Application of the index to an extensive set of artificial and real problems shows that it provides in general better results than other frequently used crisp cluster validity measures, both with respect to the prediction of the number of clusters and to the similarity among the real clusters and the partition regions. In the artificial problems, when the number of clusters increases and the available space is kept constant, the performance of all the indices decreases. This is due mainly to a higher overlap, and not just because of the larger number of clusters. The $\Delta J$ index seems to be less sensitive than the others. In particular, the results regarding the entropy distance between the real and the predicted clusters show that the clusters assessed by the new index are more closely related to the real ones than those obtained by any of the other indices.

The idea of using normality as a measure of a cluster's quality underlies the Gaussian mixture model, where many studies address the problem of cluster validation, in particular the assessment of the number of clusters [21–26]. In general these approaches use validity indices that combine the log-likelihood with some measure that penalizes the model complexity. Unfortunately, due to their probabilistic nature, these validity indices cannot be easily applied to crisp partitions.

The use of the negentropy for cluster validation has been explored in some recent works. For example, Geva et al. [18] used a cluster validity index based on the negentropy calculated along single dimensions; and Ciaramella et al. [19] calculate the negentropy along Fisher's projection to determine whether two clusters must be merged. A similar normality measure, based on multivariate skewness and kurtosis, has been used by Song and Wang [20] to discover cluster pairs that can be combined into a less complex normal cluster. Note that all these works avoid the direct estimation of normality via the negentropy, either by reducing the problem to one single dimension or by using less precise estimators such as the skewness and the kurtosis. Our contribution tackles the problem directly by considering the difference in negentropy between two different partitions. As we have shown, the terms which involve differential entropies cancel out and only covariance matrices need to be computed.

One frequent deficiency of crisp validity indices is that they are not applicable to single clusters. As they are usually based on inter-cluster distances, they need the presence of at least two clusters to be evaluated. The negentropy increment index does not present this drawback. By discounting the negentropy of the initial data distribution (all the points belong to the same cluster), we are setting the zero of the measure at the single cluster solution. Then only partitions with a negative value of the index are preferred to the trivial single cluster case.

Although here we have restricted our analysis to crisp clustering, it is possible in principle to extend the results to fuzzy

clustering. In this case, a new term measuring the uncertainty in the cluster given the data appears in the expression of $\Delta J$. Work in progress deals with the extension of the negentropy increment validity index in this direction, as well as with the sensitivity analysis on how the performance of the index degrades with the variation of parameters such as the number of clusters, the overlap among them, or the dimension. The analysis of the behavior of the new index in problems where more than one clustering partition is compatible with the data will also be addressed in future research.

## Acknowledgments

This work has been partially supported with funds from MEC BFU2006-07902/BFI, CAM S-SEM-0255-2006 and CAM/UAM CCG08-UAM/TIC-4428. The authors thank Manuel Sánchez-Montañés for insightful comments and discussions.

## Appendix A. Derivation of the expression for $\Delta J$

Let us consider the random variable $\mathbf{x}$ with pdf $f(\mathbf{x})$ in the space $\Omega_0$, and a crisp partition of $\Omega_0$ into $k$ nonoverlapping regions $\{\Omega_1, \Omega_2, \ldots, \Omega_k\}$ (Fig. 2). The differential entropy of $\mathbf{x}$ in $\Omega_0$ is

$$H_0(\mathbf{x}) = -\int f(\mathbf{x}) \log f(\mathbf{x})\, d\mathbf{x} \tag{20}$$

The differential entropy of $\mathbf{x}$ in the region $\Omega_i$, $i \neq 0$, is

$$H_i(\mathbf{x}) = -\frac{1}{p_i}\int_{\Omega_i} f(\mathbf{x}) \log \frac{f(\mathbf{x})}{p_i}\, d\mathbf{x} \tag{21}$$

where $p_i$ is the normalization constant:

$$p_i = \int_{\Omega_i} f(\mathbf{x})\, d\mathbf{x} \tag{22}$$

The negentropy of $\mathbf{x}$ in $\Omega_0$ is

$$J_0(\mathbf{x}) = \hat{H}_0(\mathbf{x}) + \int f(\mathbf{x}) \log f(\mathbf{x})\, d\mathbf{x} \tag{23}$$

And the negentropy of $\mathbf{x}$ restricted to the region $\Omega_i$, $i \neq 0$, is

$$J_i(\mathbf{x}) = \hat{H}_i(\mathbf{x}) + \frac{1}{p_i}\int_{\Omega_i} f(\mathbf{x}) \log \frac{f(\mathbf{x})}{p_i}\, d\mathbf{x} \tag{24}$$

where $\hat{H}_i(\mathbf{x})$, $i = 0, 1, \ldots, k$, is the differential entropy of a normal distribution with the same covariance matrix as $\mathbf{x}$ in the region $\Omega_i$. We can rearrange the last expression as

$$J_i(\mathbf{x}) = \hat{H}_i(\mathbf{x}) + \frac{1}{p_i}\int_{\Omega_i} f(\mathbf{x}) \log f(\mathbf{x})\, d\mathbf{x} - \log p_i \tag{25}$$

If we compute the negentropy increment $\Delta J$ as defined in (4), the integrals in (23) and (25) cancel out, and we obtain

$$\Delta J = \sum_{i=1}^{k} p_i J_i(\mathbf{x}) - J_0(\mathbf{x}) = \sum_{i=1}^{k} p_i \hat{H}_i(\mathbf{x}) - \hat{H}_0(\mathbf{x}) - \sum_{i=1}^{k} p_i \log p_i \tag{26}$$

Finally, we can substitute in (26) the expression for the entropy of the normal distribution:

$$\hat{H}(\mathbf{x}) = \frac{1}{2}\log|\Sigma| + \frac{d}{2}\log 2\pi e \tag{27}$$

where $d$ is the dimension of $\mathbf{x}$ and $|\Sigma|$ is the determinant of its covariance matrix. We get:

$$\Delta J = \frac{1}{2}\sum_{i=1}^{k} p_i \log|\Sigma_i| - \frac{1}{2}\log|\Sigma_0| - \sum_{i=1}^{k} p_i \log p_i \tag{28}$$

where $\Sigma_i$, $i = 0, 1, \ldots, k$, is the covariance matrix of $\mathbf{x}$ in the region $\Omega_i$. This is the expression that appears in (6).

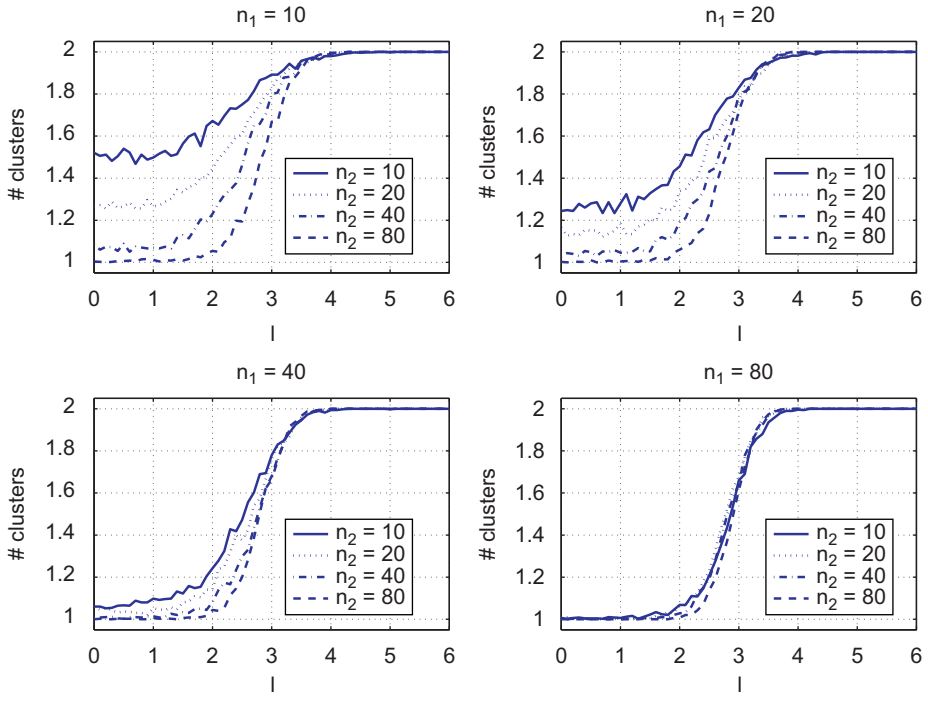


**Fig. 12.** Average number of predicted clusters versus inter-center distance $l$. Two spherical clusters with $n_1$ and $n_2$ points and centered at $(0,0)$ and $(l,0)$, respectively, are partitioned by a vertical separator at $x = l/2$, and the partition is evaluated using $\Delta J$ to determine whether one or two clusters should be considered. Each point in the plots is an average over 500 different data sets generated for a given triplet $\{n_1, n_2, l\}$.

## Appendix B. Behavior of $\Delta J$ in cases of noise and reduced number of data points

We include in this appendix a set of additional tests that were performed in order to evaluate the robustness of the new index in problems with noise and unbalanced clusters. First, to check the behavior of $\Delta J$ when the number of data points in one of the clusters is reduced, we run the following experiments.

We generate two spherical clusters in 2D. The data points in each cluster follow normal distributions with covariance matrices equal to the identity matrix. The first cluster is centered at $(0,0)$, and the second one at $(l,0)$. Both the number of points in each cluster, $n_1$ and $n_2$, and the inter-center distance, $l$, are varied across the experiments. For any data set we consider the crisp partition resulting from application of a vertical separator at $x=l/2$, and compare it in terms of $\Delta J$ with the case of no partition performed. We expect that for large $l$ the partition into two clusters is preferred, while one single cluster results for small $l$. The number of points in each cluster is selected from the set $\{10,20,40,80\}$, and the inter-center distance is varied in the range $[0,6]$. Then, for each triplet $\{n_1,n_2,l\}$ a total of 500 different data sets are generated, and the number of sets for which the $\Delta J$ index indicates one and two clusters are computed. The average number of resulting clusters is plotted versus the inter-center distance in Fig. 12. Note that, when one of the clusters has a sufficient number of points (80), reduction of the number of points in the second cluster does not noticeably affect the results. Only when the number of points in both clusters is reduced, the $\Delta J$ index tends to prefer partitions into two clusters even in cases of large overlap.

A second set of experiments was performed in order to evaluate the robustness of the new index in the presence of noise. We used the same kind of experiments as before, but with a fixed number of points in each cluster, $n_1=n_2=80$. For any given problem, we introduce noise in the following way. With probability $p$, we replace each point in the data set by a new point randomly drawn from a uniform distribution in the rectangular area given by the opposite vertices $(-2,-2)$ and $(l+2,2)$ (note that the standard deviation of the cluster distributions is 1 both in $x$ and $y$). As before, we perform the partition by a vertical separator at $x=l/2$ and compare it in terms of $\Delta J$ with the single cluster case. The results are shown in Fig. 13, where each point is an average over 500 repetitions of the experiment for a given pair $\{p,l\}$. Note that for a noise level of up to 0.25 there is no noticeable change with respect to $p=0$. When the noise level increases from this point, the algorithm starts to fail even in the limit cases of small and large $l$. Finally, when the noise level is 1, the algorithm provides always the same results regardless of $l$. The top panel in the figure shows an example of the data sets for each one of the five noise levels considered, for $l=6$. The rectangular area shown in these plots represents the domain of the uniform distribution used to generate the noise.

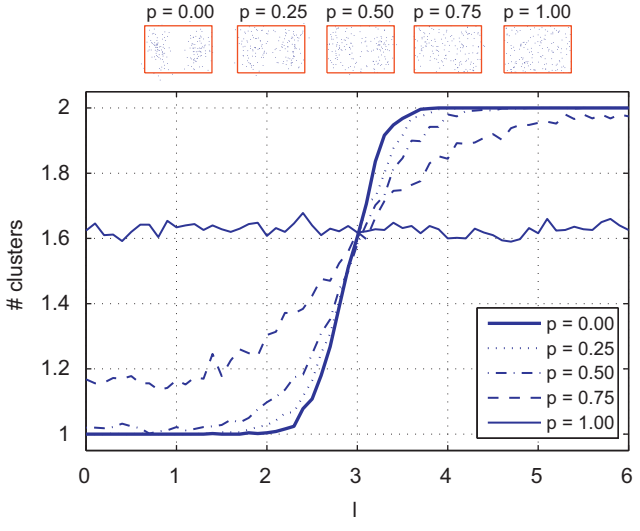


**Fig. 13.** Average number of predicted clusters versus inter-center distance $l$. Two spherical clusters with 80 points each, centered at $(0,0)$ and $(l,0)$, respectively, and subject to different noise level $p$, are partitioned by a vertical separator at $x=l/2$. The partition is evaluated using $\Delta J$ to determine whether one or two clusters should be considered. Each point in the plot is an average over 500 different data sets generated for a given pair $\{p,l\}$.

## Appendix C. Details on the genetic algorithm implementation and execution times

In this appendix we give a complete description of the genetic algorithm used to search for the clustering partitions that optimize any of the validity indices. We also include a comparison of the execution times for each index. Let $d$ be the dimension of the data space, and $k$ the number of regions or clusters in the partition. The genetic algorithm searches for partitions that can be expressed as a $d$-dimensional Voronoi diagram around $k$ centers. That is, any partition is fully characterized by the set of cluster centers $\{\mathbf{p_1},\mathbf{p_2},\ldots,\mathbf{p_k}\}$, and the region $\Omega_i$ consists of all the points that are closer to $\mathbf{p_i}$ than to any other center.

Each individual in the GA population consists of a binary string that codes the position of each of the cluster centers for a given partition. We use $b$ bits to code each coordinate of each cluster center, so the full string consists of $b\times d\times k$ bits. Fig. 14 illustrates this coding scheme for a simple case with $d=2$, $k=2$ and $b=3$. Note that the domain of each coordinate is discretized into $2^b$ bins, so that each $b$ bits of the binary string represent the bin

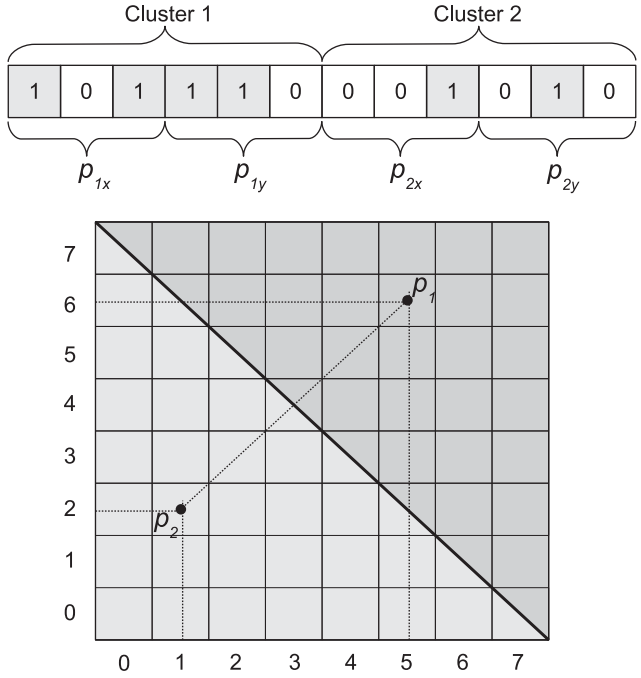


**Fig. 14.** Coding scheme used in the GA implementation for an example case with $d=2$, $k=2$ and $b=3$. Each individual in the population is a binary string of $b\times d\times k=12$ bits. The first 6 bits code the position of the first cluster center. The last 6 bits code the position of the second cluster center. Given the cluster centers $\mathbf{p_1}$ and $\mathbf{p_2}$, the region $\Omega_i$ consists of all the points that are closer to $\mathbf{p_i}$ than to the other center.

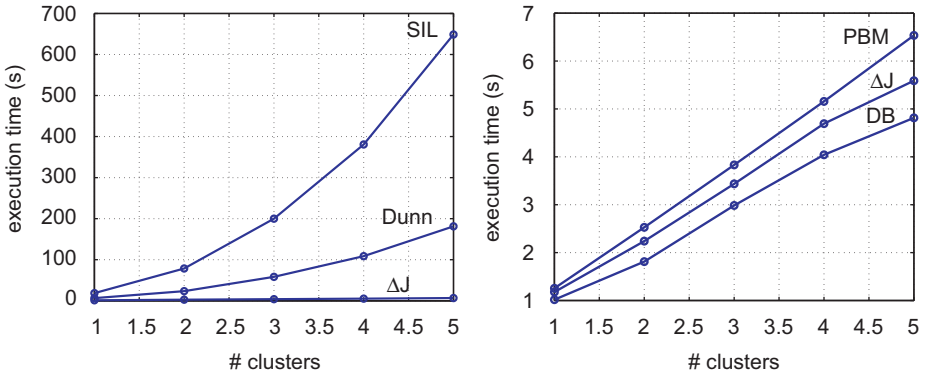


**Fig. 15.** Execution time versus problem size for the different validity indices considered in the paper.

associated to a given coordinate of a given cluster center. Cluster centers are always located in the middle of the bins. We used the value $b = 10$ in all the GA implementations of this paper. Note that this space discretization is used only to code the cluster centers, but the points in any of the data sets are generated continuously.

In all the trials performed the population size is set to 500 individuals, each one representing a different partition as shown before. The fitness function is given by one of the validity indices described in Section 4. The binary strings are randomly initialized, each bit being set to 1 with a probability of 0.5. Then the GA is run for 250 iterations, and the best partition at the end is used as the solution for a particular run. At each iteration, a new population is generated from the old one according to the following steps:

1. *Population replacement*: The 90% of the individuals with highest fitness function are copied to the new population without changes. Only 10% of the new strings are generated by recombination of the old ones.
2. *Selection*: The strings that will undergo reproduction by recombination are selected on the basis of their fitness using binary tournament selection.
3. *Crossover*: The two parents are crossed using 2-point crossover with a rate of 0.85.
4. *Mutation*: Only in the cases where crossover was not performed, each bit is inverted with a mutation rate which is the reciprocal of the string length. That is, on average only one bit per string is changed.

The scheme described above was applied to all the validity indices, changing only the fitness function. Note that our objective is not to provide a GA based clustering algorithm, but to illustrate the ability of the different indices (in particular $\Delta J$) to evaluate the quality of a clustering partition.

The execution times varied considerably depending on the validity index used as fitness function. In Fig. 15 we show, for each validity index, the total time necessary to run the 250 iterations of the GA for the *Gaussians* 2D problems in a AMD Opteron Dual Core NetPro 64 processor at 2, 6 GHz. The $x$-axis represents the number of real clusters in the data set, that is the size of the problem. The $y$-axis shows the execution time in seconds. In all the cases the GA is searching for a partition into four regions. All the points shown in the plots are averages over 2000 trials (100 different problems and 20 executions of the GA for each problem). The PBM, $\Delta J$ and DB indices display a similar behavior, with execution times that grow linearly with the problem size. The execution times for the SIL and Dunn indices grow faster than linearly. In particular, for the SIL index the time diverges very fast, which makes it inappropriate for some of the problems presented in this paper.

## References

[1] B. Everitt, S. Landau, M. Leese, Cluster Analysis, Hodder Arnold, London, 2001.
[2] A. Jain, R. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[3] R. Xu, D. Wunsch II, Survey of clustering algorithms, IEEE Trans. Neural Networks 16 (3) (2005) 645–678.
[4] A.D. Gordon, Cluster validation, in: C. Hayashi, N. Ohsumi, K. Yajima, Y. Tanaka, H.H. Bock, Y. Baba (Eds.), Data Science, Classification and Related Methods, Springer, New York, 1998, pp. 22–39.
[5] G. Celeux, G. Soromenho, An entropy criterion for assessing the number of clusters in a mixture model, J. Classification 13 (2) (1993) 195–212.
[6] Y. Ding, R.F. Harrison, Relational visual cluster validity (RVCV), Pattern Recognition Lett. 28 (15) (2007) 2071–2079.
[7] R.J. Hathaway, J.C. Bezdek, Visual cluster validity for prototype generator clustering models, Pattern Recognition Lett. 24 (9–10) (2003) 1563–1569.
[8] N.R. Pal, J. Biswas, Cluster validation using graph theoretic concepts, Pattern Recognition 30 (6) (1997) 847–857.
[9] M. Rezaee, B. Lelieveldt, J. Reiber, A new cluster validity index for the fuzzy c-mean, Pattern Recognition Lett. 19 (3–4) (1998) 237–246.
[10] H. Rhee, K. Oh, A validity measure for fuzzy clustering and its use in selecting optimal number of clusters, in: Proceedings of the 5th IEEE International Conference on Fuzzy Systems, vol. 2, 1996, pp. 1020–1025.
[11] W. Wang, Y. Zhang, On fuzzy cluster validity indices, Fuzzy Sets and Systems 158 (19) (2007) 2095–2117.
[12] X. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 13 (8) (1991) 841–847.
[13] J.C. Bezdek, R.N. Pal, Some new indexes of cluster validity, IEEE Trans. Syst. Man Cybernet. B 28 (3) (1998) 301–315.
[14] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. 1 (4) (1979) 224–227.
[15] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, J. Cybernet. 3 (3) (1973) 32–57.
[16] M.K. Pakhira, S. Bandyopadhyay, U. Maulik, Validity index for crisp and fuzzy clusters, Pattern Recognition 37 (3) (2004) 487–501.
[17] M. Bouguessa, S. Wang, H. Sun, An objective approach to cluster validation, Pattern Recognition Lett. 27 (13) (2006) 1419–1430.
[18] A.B. Geva, Y. Steinberg, S. Bruckmair, G. Nahum, A comparison of cluster validity criteria for a mixture of normal distributed data, Pattern Recognition Lett. 21 (6–7) (2000) 511–529.
[19] A. Ciaramella, G. Longo, A. Staiano, R. Tagliaferri, in: NEC: A Hierarchical Agglomerative Clustering based on Fisher and Negentropy Information, Lecture Notes in Computer Science, vol. 3931, Springer, Berlin, 2006, pp. 49–56.
[20] M. Song, H. Wang, Detecting low complexity clusters by skewness and kurtosis in data stream clustering, in: Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics, 2006.
[21] C. Biernacki, G. Celeux, G. Govaert, An improvement of the NEC criterion for assessing the number of clusters in a mixture model, Pattern Recognition Lett. 20 (3) (1999) 267–272.
[22] H. Bozdogan, Choosing the number of component clusters in the mixture-model using a new information complexity criterion of the inverse-Fisher information matrix, in: O. Opitz, B. Lausen, R. Klar (Eds.), Data Analysis and Knowledge Organization, Springer, Heidelberg, 1993, pp. 40–54.
[23] M.A.T. Figueiredo, A.K. Jain, Unsupervised learning of finite mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 381–396.

[24] C. Rasmussen, The infinite Gaussian mixture model, in: S. Solla, T. Leen, K.-R. Müller (Eds.), Advances in Neural Information Processing Systems, vol. 12, MIT Press, Cambridge, MA, 2000, pp. 554–560.
[25] R.M. Neal, Markov chain sampling methods for Dirichlet process mixture models, J. Comput. Graphical Stat. 9 (2) (2000) 249–265.
[26] S. Richardson, P. Green, On Bayesian analysis of mixtures with unknown number of components, J. R. Stat. Soc. B 59 (1997) 731–792.
[27] A. Ben-Hur, A. Elisseeff, I. Guyon, A stability based method for discovering structure in clustered data, in: R. Altman, A. Dunker, L. Hunter, T. Klein, K. Lauderdale (Eds.), Pacific Symposium on Biocomputing, vol. 7, World Scientific, Singapore, 2002, pp. 6–17.
[28] T. Lange, V. Roth, M.L. Braun, J.M. Buhmann, Stability-based validation of clustering solutions, Neural Comput. 16 (6) (2004) 1299–1323.
[29] A. Bertoni, G. Valentini, Model-order selection for bio-molecular data clustering, BMC Bioinformatics 8 (Suppl. 2) (2007) S7.
[30] J.H. Friedman, J.W. Tukey, A projection pursuit algorithm for exploratory data analysis, IEEE Trans. Comput. C-23 (1974) 881–890.
[31] P.J. Huber, Projection pursuit, Ann. Stat. 13 (2) (1985) 435–475.
[32] M.C. Jones, R. Sibson, What is projection pursuit?, J. R. Stat. Soc. A 159 (1987) 1–38.
[33] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley, New York, 1991.
[34] A.W. Bowman, P.J. Foster, Adaptive smoothing and density based test of multivariate normality, J. Am. Stat. Assoc. 88 (422) (1993) 529–537.
[35] N. Henze, T. Wagner, A new approach to the BHEP tests for multivariate normality, J. Multivariate Anal. 62 (1) (1997) 1–23.
[36] J.L. Romeu, A. Ozturk, A comparative study of goodness-of-fit tests for multivariate normality, J. Multivariate Anal. 46 (2) (1993) 309–334.
[37] G.J. Székely, M.L. Rizzo, A new test for multivariate normality, J. Multivariate Anal. 93 (1) (2005) 58–80.
[38] K.T. Fang, K.H. Yuan, P.M. Bentler, Applications of sets of points uniformly distributed on a sphere to testing multinormality and robust estimation, in: Z.P. Jiang, S.J. Yan, P. Cheng, R. Wu (Eds.), Probability and Statistics, World Scientific, Singapore, 1992, pp. 56–73.
[39] K.V. Mardia, Measures of multivariate skewness and kurtosis with applications, Biometrika 57 (3) (1970) 519–530.
[40] L. Baringhaus, N. Henze, A consistent test for multivariate normality based on the empirical characteristic function, Metrika 35 (1) (1988) 339–348.
[41] S. Csorgo, Testing for normality in arbitrary dimension, Ann. Stat. 14 (2) (1986) 708–723.
[42] O. Vasiceck, A test for normality based on sample entropy, J. R. Stat. Soc. B 38 (1) (1976) 54–59.
[43] L.X. Zhu, H.L. Wong, K.T. Fang, A test for multivariate normality based on sample entropy and projection pursuit, J. Stat. Plann. Inference 45 (3) (1995) 373–385.
[44] A. Hyvärinen, New approximations of differential entropy for independent component analysis and projection pursuit, Technical Report A47, Department of Computer Science and Engineering and Laboratory of Computer and Information Science, Helsinki University of Technology, 1997.
[45] P. Comon, Independent component analysis, a new concept?, Signal Process. 36 (3) (1994) 287–314.
[46] M.M. Van Hulle, Edgeworth approximation of multivariate differential entropy, Neural Comput. 17 (9) (2005) 1903–1910.
[47] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.
[48] J.C. Bezdek, W.Q. Li, Y. Attikiouzel, M. Windham, A geometric approach to cluster validity for normal mixtures, Soft Comput. 1 (1997) 166–179.
[49] D. Levine, PGAPack Parallel Genetic Algorithm Library 〈http://www-fp.mcs.anl.gov/CCST/research/reports_pre1998/comp_bio/stalk/pgapack.html〉.
[50] A. Asuncion, D.J. Newman, UCI Machine Learning Repository 〈http://www.ics.uci.edu/∼ mlearn/MLRepository.html〉.
[51] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugenics 7 (1936) 179–188.
[52] W.H. Mangasarian, O.L. Wolberg, Cancer diagnosis via linear programming, SIAM News 23 (5) (1990) 1–18.
[53] S. Aeberhard, D. Coomans, O. de Vel, Comparison of classifiers in high dimensional settings, Technical Report 92-02, Department of Computer Science and Department of Mathematics and Statistics, James Cook University of North Queensland, 1992.
[54] D. MacKay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, Cambridge, 2003.

**About the Author**—LUIS F. LAGO FERNÁNDEZ received the B.Sc. degree in Theoretical Physics from the Universidad Autónoma de Madrid, Spain, in 1998, and the Ph.D. degree (cum laude) in Computer Science from the same university in 2003. He is currently Professor Contratado Doctor in the Departamento de Ingeniería Informática, at the Universidad Autónoma de Madrid, Spain, and Scientific Collaborator for the data mining company Cognodata, Madrid, Spain. His research interests include machine learning, data mining and computational neuroscience.

**About the Author**—FERNANDO CORBACHO received the B.Sc. degree (magna cum laude) from the University of Minnesota, MN, in 1990, and the M.Sc. and Ph.D. degrees in Computer Science from the University of Southern California, Los Angeles, CA, in 1993 and 1997, respectively. He is currently Ad Honorem Professor in the Computer Science Department, Universidad Autónoma de Madrid, Spain and Co-founder and Chief Technology Officer of Cognodata, Madrid, Spain. Cognodata is a firm specialized in the use of data mining and artificial intelligence techniques to solve business problems specially in the area of marketing intelligence. He is engaged in the development of a theory of organization for adaptive autonomous agents. His main research interests include machine learning, schema-based learning, and the emergence of intelligence. He is a member of several computer and neuroscience associations.