Pattern Recognition Letters 31 (2010) 454–461

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Cluster validation using information stability measures

Damaris Pascual [a], Filiberto Pla [b,*], J. Salvador Sánchez [b]

[a] Centro de Reconocimiento de Patrones y Minería de Datos, Universidad de Oriente, Av. Patricio Lumumba s/n, Santiago de Cuba 90500, Cuba
[b] Institute of New Imaging Technologies, Departament de Llenguatges i Sistemes Informàtics, Universitat Jaume I, 12071 Castelló, Spain

ARTICLE INFO

Article history:
Available online 21 July 2009

Keywords:
Cluster validation
Stability index
Information theory

ABSTRACT

In this work, a novel technique to address the problem of cluster validation based on cluster stability properties is presented. The stability index here proposed is based on the variation on some information measures over the partitions generated by a given clustering model due to the variability in clustering solutions produced by different sample sets.

1. Introduction

Clustering algorithms are commonly used to infer properties of data sets in an unsupervised way. The goal of clustering is to divide the data into groups so that objects of the same group are more similar than objects of different groups. Given a data set, one of the problems to be solved in a clustering process is to choose the clustering model that could be more appropriate and explain better the structure of data (Ertoz et al., 2003; Pascual et al., 2006). Another important question is to assess the "natural" number of groups in a given data set, which is even more challenging when no clustering model is available.

Different cluster validity indices have been proposed in the literature to address the problem of determining the number of clusters. For example, several approaches (Dunn, 1974; Davies and Bouldin, 1979; Halkidi et al., 2000; Halkidi et al., 2001; Bolshakova and Azuaje, 2003) use compactness and separation measurements for evaluating and selecting an optimal number of clusters. They usually choose a representative point from each cluster and calculate the distance between these points. Others exploit the idea of within-cluster variability and the "elbow" phenomenon (Bouguessa et al., 2006; Bezdek and Pal, 1998; Sugar, 1998; Sugar et al., 1999). Other methods focus on the "elbow" phenomenon, proposing statistical measures, e.g. the gap statistic Tibshirani et al., (2004). Indices on fuzzy clustering can be found in the paper by Pal and Bezdek (1995). A comprehensive survey of methods for estimating the number of clusters is given in (Milligan and Cooper, 1985).

An alternative approach to asses the "natural" number of clusters is the so-called stability behaviour of the resulting clustering with respect to variations in the data sample used. The stability of the clustering solutions is then assessed by defining a variability measure of the clustering solutions. The rationale behind this approach is that the "natural" number of clusters is assumed to be the one for which the different solutions provided by the respective sample sets shows the lowest variation of the stability index used to measure the clustering solution.

Most of the works based on cluster stability try to assess this variability by measuring indices related to the ratios of objects or pair of objects that have not been included in the same partition by two different solutions of the clustering algorithm on two different samples (Ben-Hur and Guyon, 2003), or using statistical tests on these variations (Mufti et al., 2005), which usually need some parameter to be set to find the optimal number of clusters.

Other approaches on cluster stability are based on a transfer by prediction strategy (Lange et al., 2004), using the prediction made by a given classifier trained on the resulting partitions of clustering solutions from different data samples. Apart from the need of choosing a certain type of classifier, this method involves a considerable computational burden, due to the assessment of all possible permutations of label assignments of the clustering solutions. To avoid the dependency of the index proposed with respect to the number of clusters, the index is normalised with respect to the cost of a random predictor.

In this paper, we propose a clustering stability approach to determine the "natural" number of clusters in a data set, based on information measures as a cluster stability criterion, modeling the partition of a data set as a noisy communication channel (Cover and Thomas, 1991), exploiting the relationship of some information measures with a pattern recognition problem. The proposed algorithm has been inspired in the foundations of the transfer by prediction, but with a new focus trying to avoid the above mentioned drawbacks of the transfer by prediction strategy in (Lange et al., 2004) and the need of setting-up parameters. The method

* Corresponding author. Fax: +34 964 728435.
E-mail addresses: damaris@cerpamid.co.cu (D. Pascual), pla@lsi.uji.es (F. Pla), sanchez@lsi.uji.es (J.S. Sánchez).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.07.009

presented is also aimed at assessing clustering solutions from any clustering model and algorithm, and the experimental results show how this validity index behaves when using different clustering algorithms over the same data set. An open question would be the assessment of the clustering model that better fits a given data set, but this question is beyond the scope of the present work.

2. Cluster stability as an information problem

2.1. About the problem of pattern classification and channel communication

Let X be the random variable distributed as $p(x)$, representing the dataset in a d-dimensional space $(x_1,\ldots,x_d)$, and let Y be the random variable distributed as $p(y)$ representing the k class labels, $y \in \{y_1,\ldots,y_k\}$, of the objects in the database.

A classification process can be modeled as a noisy channel communication (Cover and Thomas, 1991), where the channel transition probability distribution can be represented by the class likelihoods $p(x/y)$. The pattern recognition process is then represented by a set W of m possible messages $w \in \{w_1,\ldots,w_m\}$, using a mn-code $C^{(n)} = (m,n)$ made by sequences of n label values y^n, whose code values are distributed as the class labels probability $p(y)$. When the sender sends a sequence y^n, the receiver sees on the other side of the channel the corresponding sequence x^n. The receiver then uses a decoding function $g(x^n):X^n \to W$, making a guess about the message sent $g(x^n) = w$.

In a pattern recognition problem, the decoding function can be represented by the decision rule. If we use the Bayes decision rule, then, the decoding function becomes

$$y = g(x) = \arg\max_{j=1,\ldots,k} \{p(y_j/x)\} = \arg\max_{j=1,\ldots,k} \{p(x/y_j)p(y_j)\} \tag{1}$$

On the other hand, the channel capacity of a channel represented by $p(x/y)$, is defined as the supreme of its possible achievable rates. According to the Channel Capacity Theorem (Cover and Thomas, 1991) the channel capacity is provided by

$$C = \max_{p(y)} I(X;Y) \tag{2}$$

Li (1997) showed that the mutual information $I(X;Y)$ between the data distribution $p(x)$ and the class distribution $p(y)$ in a decision problem is related to the Bayes error R of the decision problem when using the decision rule (1) as

$$\frac{1}{4(k-1)}(H(Y) - I(X;Y))^2 \leqslant R \leqslant \frac{1}{2}(H(Y) - I(X;Y)), \tag{3}$$

where $H(X)$ and $H(Y)$ are the Shannon entropies of random variables X and Y, respectively. Expression (3) provides a lower and an upper bound for the Bayes error. Therefore, if we can make an estimate of these information measures for a given pattern recognition problem, we could have an estimate of these Bayes error bounds.

2.2. Cluster stability as a variation in the channel transmission rate

The approach here proposed is based on measuring the transfer by prediction variability by means of information measures, as a way of assessing cluster stability. Variability on prediction will convey a variability of the decision error, the proposed method will estimate this variability by means of assessing the variation of some of the information measures involving expression (3) and the decision rule (1).

Let two different data samples be extracted by a statistically independent process, L_1 and L_2, drawn from the unknown true distribution $p(x)$ representing the data. Let a clustering algorithm representing an optimization rule of a clustering model. For a given number of clusters k, the clustering algorithm will provide a partition solution of data sets L_1 and L_2, represented by distributions $p_1(y)$ and $p_2(y)$, respectively.

Let us assume that, for a given number of clusters k, the data partition provided by the clustering algorithm over the true data distribution is represented by $p_k(y)$. Therefore, if we fix $H_k(Y)$ in expression (3), the variation due to two different clustering solutions in the estimation of the Bayes error bounds in expression (3), could be assessed by estimating the variation in the mutual information due to the use of two different samples in the transfer by prediction between the corresponding two different clustering solutions $p_1(y)$ and $p_2(y)$.

In order to estimate the transfer by prediction variability, let us have a look to the mutual information measure $I_k(X;Y)$ between the true data distribution X and the data partition of k clusters provided of the given clustering algorithm Y

$$\begin{aligned} I_k(X;Y) &= \int_x p(x) \sum_y p(y/x) \log \frac{p(y/x)}{p(y)}\, dx \\ &= \mathrm{E}_{p(x)}\{KL(p(y/x)\|p(y))\}. \end{aligned} \tag{4}$$

Mutual information between the data set X and the partition in k clusters provided by Y can be interpreted in terms of communication channels as the channel transmission rate (see Section 2.1). Previous expression can also be interpreted as the expected value according to $p(x)$, of the Kullback–Leibler divergence between the data partition distribution $p(y)$ generated by the clustering algorithm in k different classes, and the posterior probabilities $p(y/x)$ used in the decision rule (1).

In order to estimate I_k given the sample data sets L_1 and L_2, let us assume data set L_1 is used as an empirical estimate for the true data distribution $p(x) \approx p_1(x)$, and the data partition of the clustering algorithm on L_1 as an estimate for the data partition $p(y) \approx p_1(y)$. Therefore, the empirical data distribution of data set L_1 containing N_1 samples, can be expressed as

$$p(x) \approx p_1(x) = \frac{1}{N_1} \sum_{i=1}^{N_1} \delta(x, x_i). \tag{5}$$

Then, expression (4) becomes

$$I_k(X;Y) \approx \frac{1}{N_1} \sum_{i=1}^{N_1} \sum_y p(y/x_{1i}) \log \frac{p(y/x_{1i})}{p(y)}. \tag{6}$$

On the other hand, in order to make an estimate of the posteriors $p(y/x)$, if we have used data sample L_1 to estimate the true probability distribution and the data partition distribution, let us use data sample L_1 and the labeling made by the partition solution provided by the clustering algorithm, as the test set, and data sample L_2, as a training set. Thus, the posteriors $p(y/x)$ can be estimated using training set L_2 and test set L_1 as

$$p(y/x) = \frac{p_2(y/x)}{\sum_{j=1}^{k} p_2(y_j/x)}; \quad y = \{y_1,\ldots,y_k\} \tag{7}$$

being,

$$p_2(y/x) = \sum_{i=1}^{knn} \delta(y_1(x_i), y) \frac{1}{\varepsilon + d(x, x_i)}, \tag{8}$$

where $y_1(x_i)$ is the class label assigned to sample x_i (neighbour of x) of data set L_1, y is an arbitrary label in L_2 and knn denotes the number of neighbours of sample x used to estimate $p_2(y/x)$. Eventually, the estimate of $I_k(X;Y)$ using two independent data samples L_1 and L_2, is expressed as

$$I_k(X;Y) \approx \widehat{I}_k(X;Y) = \frac{1}{N_1} \sum_{i=1}^{N_1} \sum_{j=1}^{k} p_2(y_j/x_{1i}) \log \frac{p_2(y_j/x_{1i})}{p_1(y_j)}. \tag{9}$$

Since the estimate $\widehat{I}_k$ varies for different sample sets, it is a random variable. Thus, according to the law of large numbers, when the number of measurements tends to infinity, the expected value of this variable tends to the true value. For a finite number of measurements N, the true value of this variable concentrates around its mean with standard deviation

$$\sigma_k^2 = E(\widehat{I}_k - \mathrm{E}(\widehat{I}_k))^2. \quad (10)$$

This variance represents the variability of the mutual information between the data set X and the data partitions Y_k for k clusters generated because of the fluctuations and differences in the data sample used. Moreover, the variation in the mutual information is related to the variations in the achievable transmission rate along the communication channel $Y \rightarrow X$, that is, in the error classification rate of the pattern classification problem it represents (see Section 2.1).

Thus, the standard deviation σ_k will represent the cluster stability index of the algorithm for k clusters. This cluster stability index is an estimate of the variation of the transfer by prediction between Ncouples of clustering solutions, when partitioning every pair of independent sample sets into k clusters using a certain clustering algorithm. Therefore, for a given clustering algorithm the correct number of clusters k^* will be chosen as the number of clusters that minimizes the stability index (10), that is,

$$k^* = \min_k \sigma_k. \quad (11)$$

3. Experimental results

In order to show the performance of the cluster validity index proposed, three types of databases were used. Two types of synthetic databases, one type consists of Gaussian clusters (Fig. 1), and the other type of data consists of clusters of arbitrary shapes and sizes (Fig. 2), in two dimensions. A third group of data consists of real databases.

In (Fig. 1), the first database consists of four Gaussians (4Gauss) with little overlapping among them. The second one is formed by six Gaussians with different degree of overlapping (6Gauss), and the third database has also six clusters (6Gauss-II), but two of them are completely overlapped, with the same mean and different covariance.

The following three synthetic databases have clusters with different shapes, sizes and densities, and they are separated by zones of low density (Fig. 2). The first database consists of three concentric rings (3CRings), the second one is a pair of half-rings (2HRings), and the third one is the DS1 database used in other works (Pascual et al., 2006).

Regarding to the clustering method, three different clustering algorithms are used in the experiments: the well known K-Means, the Gaussian mixture model using EM (Expectation Maximization) and H-Density (Pascual et al., 2006). These algorithms correspond to three different clustering models, where K-Means looks for compact clusters around a mean, the Gaussian mixture model looks for Gaussian-shape clusters, and the H-Density is an agglomerative hierarchical algorithm based on data density estimates and a single link strategy adequate for cluster structures of any shape.

3.1. Stability index performance

For each database used in the experiments, the experimental set-up consists of splitting randomly the database into two equal size sample sets L_1 and L_2. For $k = 2,\ldots,10$ number of clusters, each of the clustering algorithm is run on data samples L_1 and L_2, and the cluster stability index (Eq. (11)) is assessed over 10 different realisations of L_1 and L_2. Inverting the role of L_1 and L_2 in the

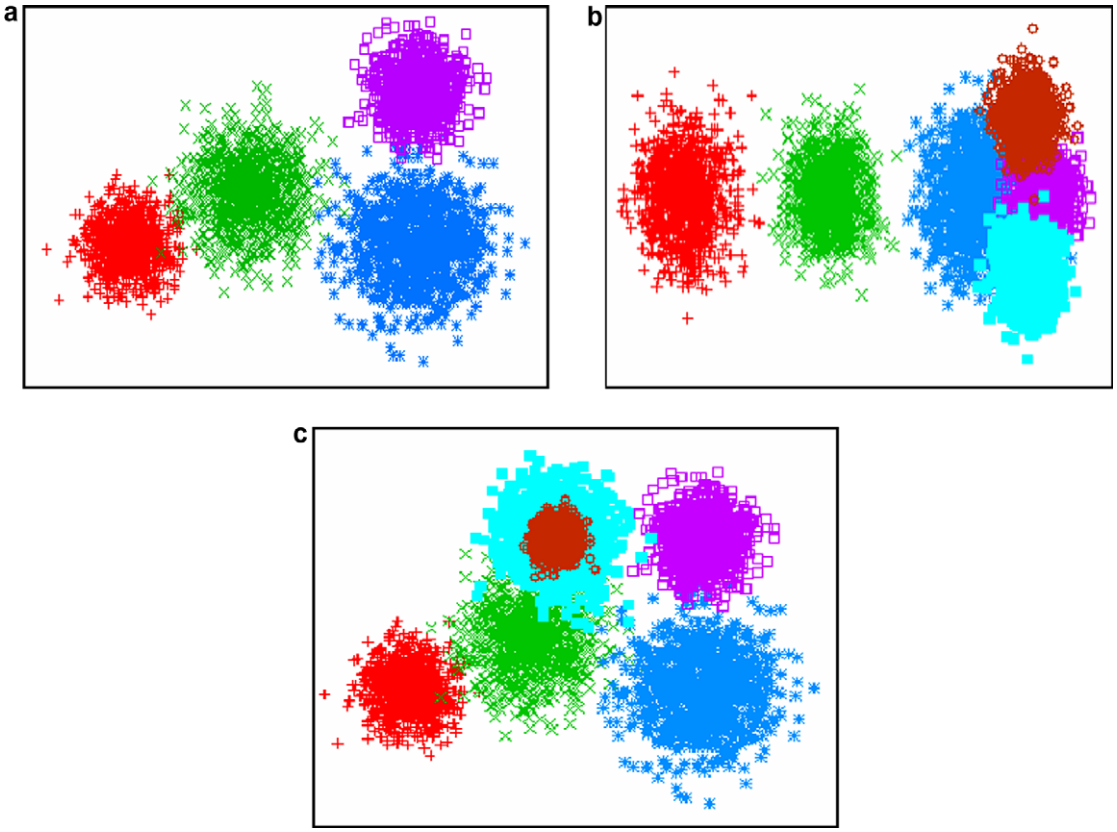

Fig. 1. First group of databases. Left to right: (a) 4 Gauss, (b) 6 Gauss and (c) 6 Gauss-II (two concentric).

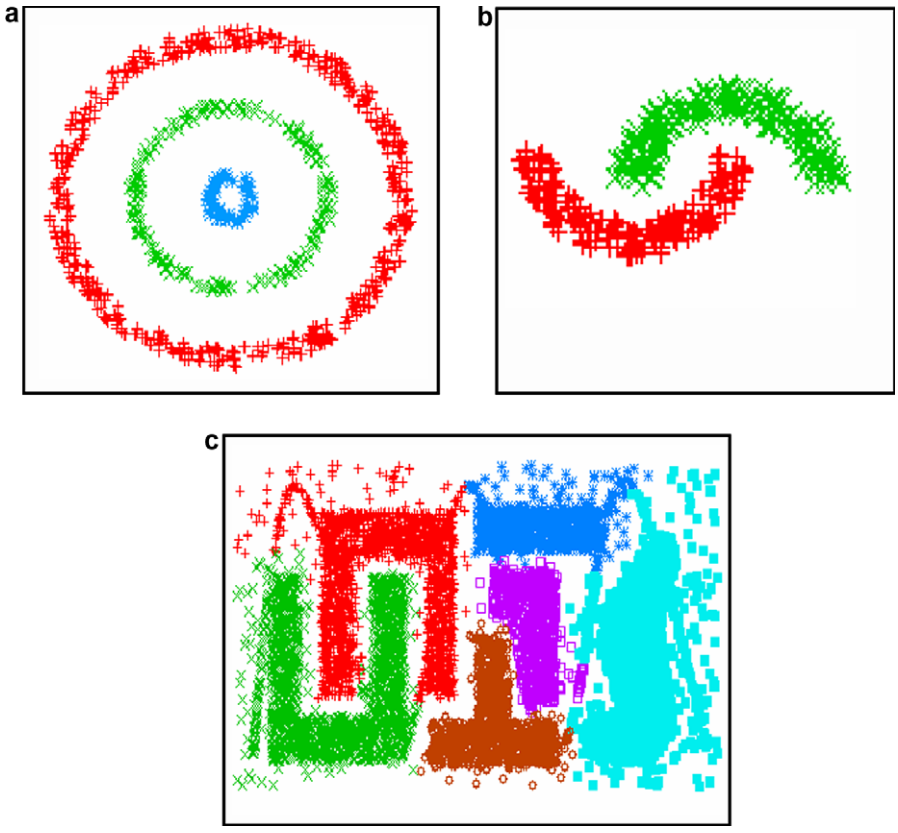

Fig. 2. Different shape, densities and sizes. Left to right: (a) 3 CRings, (b) 2 HRings and (c) DS1 database.

Table 1
Stability index for the synthetic databases with H-Density clustering algorithm.

# Clust	4Gauss	6Gauss	6Gauss-II	3CRings	2HRings	DS1
2	0.00944	0.027000	0.02400	0.0014	**0.00001**	0.217
3	0.01310	0.000080	0.01000	**0.0011**	0.02400	0.013
4	**0.00004**	0.002000	0.00006	0.0547	0.02300	0.023
5	0.01608	0.003000	**0.00003**	0.0356	0.00900	0.023
6	0.01945	**0.000008**		0.0210		**0.002**
7	0.01760			0.0332		0.013
8	0.09400			0.0249		0.042
9						0.015
10						0.015

Table 2
Stability index for the synthetic databases with EM clustering algorithm.

# Clust	4Gauss	6Gauss	6Gauss-II	3CRings	2HRings	DS1
2	0.010000	0.000020	0.00430	0.0026	0.0130	0.03000
3	0.016000	**0.000008**	0.00370	**0.0017**	0.0057	0.00800
4	**0.000001**	0.000020	0.00780	0.0029	**0.0003**	0.00140
5	0.012000	0.002000	**0.00003**	0.0045	0.0027	**0.00001**
6	0.081000	0.000009	0.00029	0.0480	0.0014	0.00180
7	0.100000	0.00500	0.0065	0.0035	0.0049	0.00003
8	0.037000	0.02600	0.01700	0.0060	0.015	0.00004
9	0.043000	0.03400	0.04000	0.0070	0.0047	0.00110
10	0.049000	0.01000	0.01500	0.0040	0.0059	0.00410

estimate of mutual information (Eq. (9)) provides an estimate of the stability index over $N = 20$ realisations.

In the case of H-Density algorithm, since this clustering model depends on the parameter R (radius of the vicinity of each point), on some databases the maximum level of clusters starts from less than 10 clusters, for instance, see Table 1 for the case of 4Gauss database, which starts from the level of 8 clusters.

3.1.1. Synthetic databases

Table 1 shows the result of the cluster stability index proposed for the synthetic databases, using the H-Density clustering algorithm. Bold numbers indicate the optimum value regarding to the proposed stability index. Note how the index shows a minimum value for the right number of clusters (see Figs. 1 and 2). In the case of the 6Gauss-II database, due to the completely overlapped couple of Gaussians, H-Density algorithm cannot distinguish the six Gaussians but the level of five classes is found to be the best one with our strategy.

In Table 2 we show the values of our index over the different synthetic databases using the Gaussian mixture clustering model. As can be seen, the correct number of clusters was obtained on 4Gauss database, and in the case of the 6Gauss-II database, the level of five clusters was detected as the optimum whereas the second lowest stability index corresponds to six clusters. As Gaussian mixture cannot discover the correct clustering in the other databases, the stability index does not show the correct number of clusters. For 3CRings, the minimum value was obtained with three clusters, but EM algorithm is not able to discover the real clustering.

With respect to the other clustering algorithm, K-Means is able to detect the four Gaussians (Fig. 1a), but fails on the other examples, due to the overlapping and the fact that clusters are not modelled as spherical Gaussian distributions, which would be more adequate for a K-Means model (Table 3).

3.1.2. Real databases

The House database (Pascual et al., 2006) represents the chromatic *ab* pixel values of the *House* colour image in the *Lab* space (Fig. 3b). Fig. 3 shows the result of the clustering selected by the

Table 3
Stability index for the synthetic databases with K-Means clustering algorithm.

# Clust	4Gauss	6Gauss	6Gauss-II	3CRings	2HRings	DS1
2	0.0000026	**0.00003**	0.0170	0.00009	**0.00004**	**0.0000005**
3	0.0131724	0.00012	0.0044	0.00017	0.00260	0.0001000
4	**0.0000023**	0.00570	0.0042	**0.00006**	0.00120	0.0015000
5	0.0026520	0.01330	**0.0001**	0.00020	0.0011	0.0000100
6	0.0065380	0.00160	0.0066	0.00090	0.0008	0.0000040
7	0.0032110	0.00100	0.0160	0.00200	0.0036	0.0010000
8	0.0160700	0.01370	0.0045	0.00100	0.0034	0.0010000
9	0.0095200	0.003800	0.0120	0.00400	0.0047	0.0005000
10	0.0152000	0.08600	0.0090	0.00100	0.0041	0.0006000

best stability index (see Table 4) when using the H-Density algorithm. Notice the quality of the clustering selected by looking at the pixel labelling that provides the selected solution by the stability index (Fig. 3c).

The method was also tested on the IRIS database, which has 150 samples corresponding to three types of plants. Two out of these classes are highly overlapped, which presents a high difficulty for most clustering algorithms. The other class is clearly separated from the other two. The H-Density algorithm is able to detect, among the levels of the hierarchy, the level of three classes with a high rate of correct classification (see (Pascual et al., 2006) for details). In this case, the stability index introduced here has been able to detect the right number of classes, even in presence of high degree of overlapping of two of them, and the second minimum value of the variance is achieved at the level of two classes (Table 4). The other clustering algorithms do not provide the right clustering solution, because of the inadequate clustering model.

For House database, the EM and K-Means clustering algorithms were not able to discover the true five classes from the different results and consequently, the index did not detect the correct number of clusters. Regarding IRIS database, EM and K-Means chose the level of two classes as the best because of the high overlapping between two classes.

As a result, we can conclude that when using the appropriate clustering models (in this case, H-Density) we obtain the correct number of clusters with our stability index proposed in this paper.

3.2. Comparison with other cluster stability methods

This section describes an experimental comparison between the proposed cluster validity index and six other cluster validity strategies, two of them also based on stability. The algorithms are compared using all the databases introduced in Sections 3.1.1 and 3.1.2, in order to quantitatively show the differences in the solutions provided by the different algorithms.

3.2.1. Related works

The following two reference algorithms in the literature have been chosen for the comparison because they are based on cluster stability criteria, being two of the most popular methods of stability indices present in the literature. Another reason is because they also use somehow the transfer by prediction between clustering solutions on several realisations of two independent sample sets from the original data set.

The first algorithm (Ben-Hur et al., 2002) introduces the dot product between partitions and defines several similarity measures based on this dot product concept. In addition, they consider a clustering algorithm that controls either directly or indirectly the number of clusters k that the algorithm produces. For each value of k from 2 to k_{max} they select N times two sub-samples of data points with a sampling ratio f (fraction of point sampled), not much smaller than 1 ($f \geqslant 0{,}6$), clustering each subsample and determining the similarities between the respective clustering solutions, using the labels of the samples common to both sub-samples. The distribution of similarities for each k provides a criterion for choosing an optimal partition of the data. Plotting the distribution with respect to the number of clusters reveals a transition between distributions of similarities that are concentrated close to 1; which means that the solution is where both partitions are most similar. This can be quantified by the jump on the area under the distribution function or by the jump on the probability that the similarity between partitions into k clusters is bigger than a certain threshold η.

The second algorithm used in the comparison was introduced by Tibshirani and Walther (2005). In a few words, this algorithm also divides the original data set into two independent subsample sets N times. Each time, one of the subsample set is taken as the training set and the other as the test set. Using a cross validation strategy, both subsample sets are clustered into k clusters (k from 2 to k_{max}), and for each k, it is measured how well the training set cluster centres predict the co-memberships in the test set. Eventually, for each pair of test observations that are assigned to the same test cluster, they determine if they are also assigned to the same cluster based on the training centres, and using the prediction

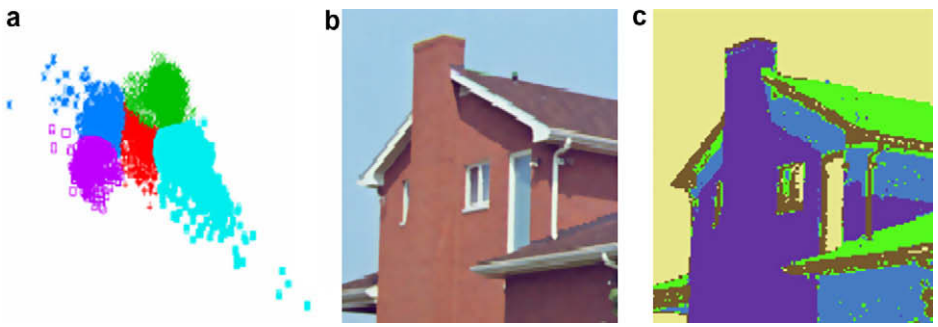

Fig. 3. Left to right: (a) Clusters selected by the stability index when using H-Density algorithm in ab space. (b) "House" image. (c) Pixels labeled by H-Density algorithm result from (a).

Table 4
Stability indices for the House and Iris databases using the H-Density, EM and K-Means clustering algorithms.

# Clusters	H-Density	House			Iris	
		EM	K-Means	H-Density	EM	K-Means
2	0.01800	0.000010	**0.000002**	0.0019	**0.0042**	**0.0016**
3	0.01700	**0.000006**	0.000200	**0.0018**	0.0470	0.0388
4	0.00390	0.000020	0.000020	0.0054	0.0150	0.0513
5	**0.00003**	0.000100	0.000700	0.1918	0.0090	0.0180
6		0.000400	0.023000	0.1827	0.0200	0.0327
7		0.002100	0.014000	0.1599	0.0400	0.0108
8		0.000800	0.007000	0.2385	0.0100	0.0083
9		0.008000	0.021000	0.3062	0.0100	0.0085
		0.015000	0.024000			

strength (ps) defined in that work, they estimate the number of groups by the largest k such that $ps(k) \geqslant \eta$, being η a given threshold.

A third algorithm included in this comparative study is the one proposed by Fischer and Buhman (2003). This chooses the number of clusters that maximizes the difference between the mean cluster assignment probability of the maximum likelihood assignment and the mean cluster assignment probability of random assignments relative to the risk of the random cluster solution. They proposed a fast agglomerative algorithm to minimize the cost function in the path-based clustering, and a resampling scheme like Bootstrap (bagging) to get an empirical probability distribution over the cluster assignments for each object.

Also, we have compared our strategy with several validity indices used for measuring goodness of a clustering results. The Davies Bouldin index (DB) (Davies and Bouldin, 1979) uses a similarity measure of clusters based on the dispersion measure of a cluster and a dissimilarity measure between clusters. The SD validity index (Halkidi et al., 2002) is based on the average scattering of clusters and total separation of clusters; the scattering is calculated by variance of the clusters and variance of the data set in order to measure the compactness of the clusters, while the total separation of clusters is based on the distance between cluster centre points, with the objective of measuring the separation of clusters. The last validity index here considered is S_Dbw proposed by Halkidi et al. (2001), which is based on cluster compactness and separability too, but instead of using the distance between clusters, it employs a criterion of density inter-clusters in order to measure separability.

3.2.2. Experimental setup

All comparative experiments have been carried out over the synthetic (Gaussian and arbitrary shape databases) and real databases described in Sections 3.1 and 3.2, respectively. The clustering algorithms used to test the six validity indices are also the same as the ones used with the proposed stability index, that is, EM, K-Means and H-Density. The proposed stability index will be called hereafter σ-index.

For the Ben-Hur method, a sampling ration $f = 0.8$ was used, over $N = 100$ realisations. As similarity measure between the labelling of two different clustering solution, the correlation or cosine similarity was used, with a threshold of $\eta = 0.9$.

Regarding to Tibshirani and Walther's algorithm, the whole initial database was randomly divided into two partitions, one of them was considered as the training set and the other one as the test set. The algorithm was run $N = 50$ times for each k clusters, with a threshold to detect the jump in the prediction strength of $\eta = 0.8$ (Tibshi index, in Table 5–7).

In the case of FB index (Fischer and Buhmann's algorithm), in order to get an empirical probability distribution over the cluster assignments for each object, we developed the same strategy as the one described in Section 3.1 for the proposed σ-index, and the probability was calculated using Eq. (7).

With respect to the other indices, we got the value of these over the result of each clustering with a number of clusters ranging from 2 to $k_{max} = 10$.

3.2.3. Comparison results

Table 5 shows the number of clusters selected by each index, when using the H-Density algorithm over all the databases considered in this work. For 6Gauss database, Ben-Hur algorithm doubts about choosing either the level of three or six classes because in both cases there is a significant jump on the probabilities. With respect to the 6Gauss-II database, the values of the index also present some doubts about choosing either three or six classes, but now the distribution of values are not so near the ideal value of 1. In the case of DS1 database, the six clusters solution is the one with the highest probability, but in this case the similarity distribution is rather spread, leading to a probability value below 1. Ben-Hur detects the correct number of clusters in the 3CRings, 2HRings, House and 4Gauss databases. Regarding Iris database, it is difficult to determine the optimum level because all index values provided by Ben-Hur algorithm are very similar and close to 1.

Regarding to Tibshi algorithm, the highest value of the prediction strength that are bigger than $\eta = 0.8$ was chosen to determine the number of clusters. When using the H-Density clustering, Tibshi proposes the correct number of clusters for the 4Gauss database, and five clusters for the 6Gauss-II database because H-Density algorithm is not able to separate the two concentric

Table 5
Comparison of the number of clusters for the different strategies over all the databases using H-Density.

Strategy	3CRings	4Gauss	6Gauss	6Gauss-II	DS1	House	Iris	2HRings
σ-Index	3	4	6	5	6	5	3	2
Ben-Hur	3	4	3/6	3/6	–	5	–	2
Tibshi	–	4	3	5	–	–	2	–
FB	2/3	4	3	3	6	5	2	2
DB	10	3	3	4	2	2	2	8
SD	9	3	2	3	2	2	2	5
S_Dbw	–	4	3	5		5	2	8

Table 6
Comparison of the number of clusters results for the different strategies over all the databases using EM.

Strategy	3CRings	4Gauss	6Gauss	6Gauss-II	DS1	House	Iris	2HRings
σ-Index	3	4	3	5	5	3	2	4
Ben-Hur	7	4	7	5	5/6	8	2	4
Tibshi	2	4	2	5	2	2	2	4
FB	7	4	4	5	8	2	2	4
DB	7	3	2	5	2	2	2	8
SD	7	4	5	5	2	3	2	6
S_Dbw	10	4	6	5	10	7	2	9

Table 7
Comparison of the number of clusters results for the different strategies over all the databases using K-Means.

Strategy	3CRings	4Gauss	6Gauss	6Gauss-II	DS1	House	Iris	2HRings
σ-Index	4	4	2	5	2	2	2	2
Ben-Hur	3	4	2	3	3/5	2/4/5	2	2
Tibshi	3	4	2	3	2	2	2	2
FB	7	4	2	3	2	2	2	2
DB	8	4	2	4	2	2	2	6
SD	5	3	2	4	2	5	2	4
S_Dbw	2	5	7	5	10	9	2	10

Gaussians. With respect to the 6Gauss database, it discovered three clusters.

Tibshi method is not able to detect the correct clustering solution, both for the arbitrary shape databases and House database, because all values are below the specified threshold, even when using the most adequate clustering model, that is, H-Density, which provides the right clustering solution.

The drawbacks of this method are even more evident in the case of the 3CRings database. Note that this method is based on a strategy of estimating the transfer by prediction strength using the centroid of each cluster and the stability of the classification depends on the position of these centroids. As the rings are concentric, all centroids collapse in the same point, being unable to undo the ambiguity.

Similarly, in the case of DS1 database, since the shape of the clusters in this database is variable, the classification based on centroids forces to consider the clusters as a compact shape, distorting the real cluster shapes. Therefore, this stability index is affected by this drawback and the method cannot determine the correct number of clusters. The same problem occurs with the other databases used.

The FB index failed over databases strongly overlapped (6Gauss and 6Gauss-II databases) because this method uses probabilities based on path or linkage and random clustering with the objective of normalizing the probabilities. As a result, it is not able to discover the real shape of the clusters. Over 3CRings database, this method got the maximum value of the index both for two and three classes, being not able to detect the level of three classes as the best one. With the other synthetic databases and House database, the index found the correct number of clusters.

In most databases, the other three indices did not discover the correct number of clusters, except S_Dbw that chose the optimum value over House and 4Gauss databases. The main disadvantage of the validity indices SD and S_Dbw is that they select a representative point in each cluster and calculate the distance between clusters using these points and in general, the DB, SD and S_Dbw indices use variance to measure homogeneity within the clusters, thus they are not able to discover the arbitrary shape of the clusters and if the clusters are overlapped, the performance of these indices is not good as can be seen in Table 5. Besides, the S_Dbw index has a drawback in its formulation because the denominator can be zero and even both numerator and denominator, making difficult the selection of the best clustering.

Regarding the Gaussian Mixture (EM) algorithm (see Table 6), all indices detected five classes as the best number of clusters for 6Gauss-II database, whereas in the case of 4Gauss only the DB index did not discover four clusters. Concerning 6Gauss using Ben-Hur index, there are several values close to 1, but using the proposed procedure in (Ben-Hur et al., 2002), it has been selected the solution of seven clusters. The results over the other databases are incorrect because the EM algorithm is not able to discover the real shape of the clusters. The proposed σ-index shows a minimum value for the right number of clusters when using the EM in the case of 4Gauss database and over 6Gauss-II database, while for 6Gauss-II the correct number of clusters corresponds to the second minimum value, as can be seen in Table 2.

Similarly, one can observe in Table 7 that all indices with the K-Means algorithm gave a wrong number of clusters in all databases, except for 4Gauss, due to the characteristics of the K-Means clustering model. In the case of 4Gauss database, only SD and S_Dbw indices did not provide the right number of four classes.

4. Conclusions

The problem of determining the optimal or "natural" number of groups in a database is an important issue for clustering processes, together with the selection of the adequate clustering model for each particular dataset. In this work, a method to select the "natural" number of clusters have been presented, based on cluster stability criterion inspired in an information theoretic approach to assess the variability of clustering solutions due to the different clustering partitions obtained from different data samples of the same problem.

The experiments carried out on several synthetic and real databases, using three different clustering models; show that, when the clustering algorithm is adequate to model the data structure, the stability index proposed can select the right solutions in a wide variety of synthetic and real examples with cluster structures of different shapes, sizes and overlapping degree.

An experimental comparison has been made with other state-of-the-art cluster validity algorithms, most of them based on stability criteria inspired in the idea of transfer by prediction. From the results, we can conclude that the proposed cluster stability index outperforms these methods for most of the databases tested, finding the right number of clusters when using the adequate

clustering model. Moreover, the proposed stability index overcomes some of the drawbacks of those algorithms with respect to assumptions on cluster shapes and separability measures, which produce wrong solutions when the clusters are significantly overlapped. Therefore, the proposed index can be considered as a valid alternative to the existing ones.

As it has also been outlined, another issue beyond the scope of this work is the problem of selecting the adequate clustering model for a given problem, which can affect significantly the stability performance of the clustering algorithm used.

Acknowledgements

This work has been partially supported by Spanish Ministry of Science and Education under Projects ESP2005-00724-C05-05, CSD2007-00018 and PET2005-0643 and Project P1 1B2007-48 by Fundació Caixa-Castelló.

References

Ben-Hur, A., Elisseeff, A., Guyon, I., 2002. A stability based method for discovering structure in clustered data. Pacific Symp. Biocomput., 6–17.

Ben-Hur, A., Guyon, I., 2003. Detecting stable clusters using principal component analysis. In: Brownstein, M., Khodursky, A. (Eds.), Methods in Molecular Biology, Humana Press, pp. 159–182.

Bezdek, J.C., Pal, N.R., 1998. Some new indexes of cluster validity. IEEE Trans. Systems Man Cybernet. – Part B: Cybernetics 28 (3), 301–315.

Bolshakova, N., Azuaje, F., 2003. Cluster validation techniques for genome expression data. Signal Process. 83, 825–833.

Bouguessa, M., Wang, S., Sun, H., 2006. An Objective approach to cluster validation. Pattern Recogn. Lett. 27, 1419–1430.

Cover, T.M., Thomas, J.A., 1991. Elements of Information Theory. Wiley.

Davies, D.L., Bouldin, D.W., 1979. A cluster separation measure. IEEE Trans. Pattern Anal. Machine Intell. 1 (2), 95–104.

Dunn, J.C., 1974. Well separated clusters and optimal fuzzy partitions. J. Cybernetica 4, 95–104.

Ertoz, L., Steinbach, M., Kumar, V., 2003. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In: Proceedings of Third SIAM International Conference on data Mining.

Fischer, B., Buhman, J.M., 2003. Bagging for path-based clustering. IEEE Trans. Pattern Recogn. Anal. Machine Intell. 25 (11), 1411–1415.

Halkidi, M., Vazirgiannis, M., Batistakis, Y., 2000. Quality scheme assessment in the clustering process. In: Proceedings of the Fourth European Conference on Principles of Data Mining and Knowledge Discovery, pp. 265–276.

Halkidi, M., Vazirgiannis, M., 2001. Clustering validity assessment: Finding the optimal partitioning of a data set. In: Proceedings of ICDM 2001, pp. 187–194.

Halkidi, M., Batistakis, Y., Vazirgiannis, M., 2002. Cluster validity methods: Part I. SIGMOD Rec. 31 (2), 40–45.

Halkidi, M., Batistakis, Y., Vazirgiannis, M., 2002. Cluster validity methods: Part II. SIGMOD Rec. 31 (3), 19–27.

Lange, T., Braun, M.L., Buhmann, J.M., 2004. Stability-based validation of clustering solutions. Neural Comput. 16, 1299–1323.

Li, J., 1997. Divergence measures based on Shannon entropy. Information Theory 37 (1), 145–151.

Milligan, G.W., Cooper, M.C., 1985. An examination of procedures for determining the number of clusters in a data set. Psycometrika 50, 159–179.

Mufti, G.B., Bertrand, P., Moubarki, L.E., 2005. Determining the number of groups from measures of cluster validity. In: Proceedings of ASMDA2005, pp. 404–414.

Pal, N.R., Bezdek, J.C., 1995. On cluster validity for the fuzzy c-means model. IEEE Trans. Fuzzy Syst 3 (3), 370–379.

Pascual, D., Pla, F., Sánchez, J.S., 2006. Non parametric local density-based clustering for multimodal overlapping distributions, IDEAL 2006, LNCS, vol. 4224, pp. 671–678.

Sugar, C., 1998. Techniques for clustering and classification with applications to medical problems. Ph.D. Dissertation, Stanford University, Stanford.

Sugar, C., Lenert, L., Olshen, R., 1999. An application of cluster analysis to health services research: Empirically defined health states for depression from the sf-12. Technical Report, Stanford University, Stanford.

Tibshirani, R., Walther, G., 2005. Cluster validation by prediction strength. J. Comput. Graph. Statist. 14, 511–528.

Tibshirani, R., Walther, G., Hastie, T., 2004. Estimating the number of clusters in a data set via the gap statistic. J.R. Statist. Soc. B, Part 2, 411–423.