2nd Hellenic Conf. on AI, SETN-2002, 11-12 April 2002, Thessaloniki, Greece,
Proceedings, Companion Volume, pp. 237-248

Clustering validity assessment using multi representatives

Maria Halkidi, Michalis Vazirgiannis
Dept of Informatics, Athens University of Economics & Business
76 Patission Str, 10434
Email:{mhalk, mvazirg}@aueb.gr

Abstract. Clustering is a mostly unsupervised procedure and the majority of the clustering algorithms depend on certain assumptions in order to define the subgroups present in a data set. Moreover, they may behave in a different way depending on the features of the data set and their input parameters' values. Therefore, in most applications the resulting clustering scheme requires some sort of evaluation as regards its validity.
In this paper a new validity index, *CDbw*, is defined based on well-known clustering criteria enabling: i. the finding of the optimal input parameters' values for a clustering algorithm that results in the optimal partitioning of a data set, ii. the selection of the algorithm that provides the optimal partitioning of a data set. *CDbw* puts emphasis on the geometric features of clusters, handling efficiently arbitrary shaped clusters. It achieves this by representing each cluster by a certain fixed number of clusters rather than a single center point. Our experimental results confirm the reliability of our index showing that it performs favorably in all cases selecting independently of clustering algorithm the scheme that best fits the data under consideration.

1. Introduction

In the literature a wide variety of algorithms have been proposed for different applications and types of data sets [14]. The application of an algorithm to a data set aims at, assuming that the data set offers a clustering tendency, discovering its real partitions. This implies that i. all the points that naturally belong to the same cluster will eventually be attached to it by the algorithm, ii. no additional data set points (i.e., outliers or points of another cluster) will be attached to the cluster.

In most algorithms' experimental evaluations [1, 6, 10, 11, 12, 17] 2D-data sets are used in order the reader is able to visually verify the validity of the results (i.e., how well the clustering algorithm discovered the clusters of the data set). It is clear that the visualization of the data set is a crucial verification of the clustering results. In the case of large multidimensional data sets (e.g. more than three dimensions) effective visualization of the data set can be difficult. Moreover the perception of clusters using available visualization tools is a difficult task for the humans that are not accustomed to higher dimensional spaces.

It is obvious then that a major problem in clustering is to decide the optimal number of clusters that fits a data set. The various clustering algorithms behave in a different way depending on: i) the features of the data set (geometry and density distribution of clusters), ii) the input parameters values

Assuming that the data set includes distinct partitions (i.e., inherently supports clustering tendency), the second issue becomes very important. In the following we show that different input parameters values of clustering algorithms may result in good or bad results in partitioning the data set.

For instance in Figure 1 we can see the way a specific algorithm (e.g., K-means[2]) partitions a data set having different input parameter values. It is clear that the data set is falsely partitioned in most of the cases. Only some specific values for the algorithms' input parameters lead to optimal partitioning of the data set. Moreover, in Figure 2 we can see the way different algorithms (DBSCAN [6], K-Means [2]) partition a data set. It is clear from Figure 2a and Figure 2b that K-means may partition the data set into the correct number of clusters (i.e., three clusters) but in a wrong way. On the other hand, DBSCAN (see Figure2c) is more efficient since it partitioned the data set in the inherent

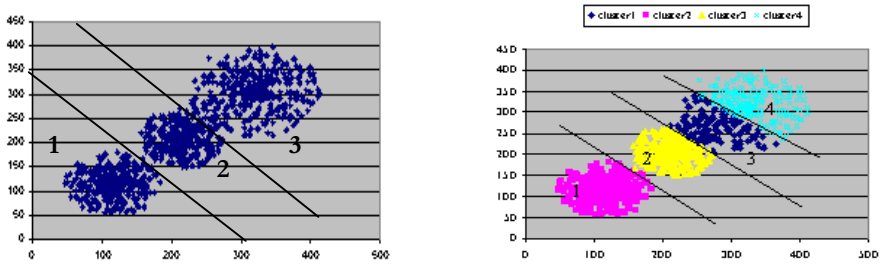


Figure 1. The different partitions resulting from running K-Means with different input parameter values.

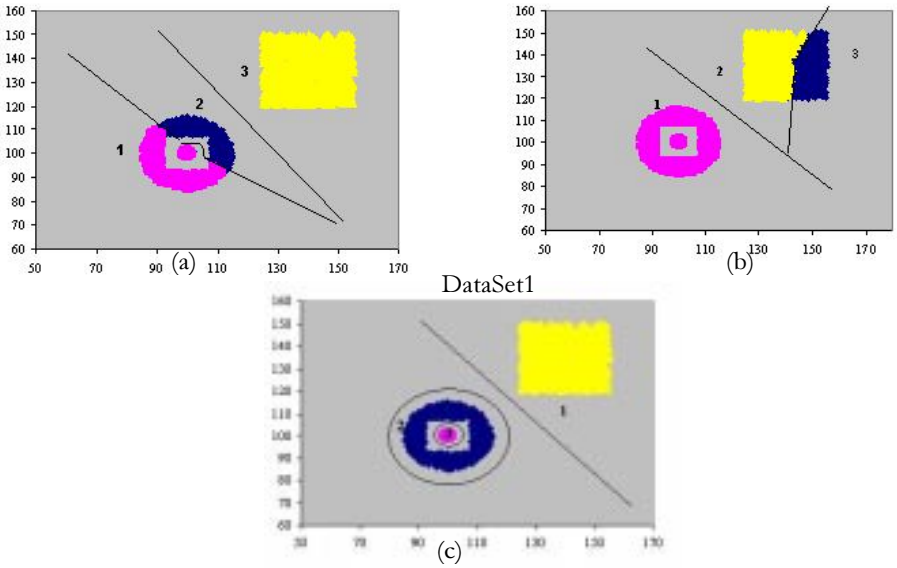


Figure 2: DataSet1 - Partitioning of DataSet4 into three clusters as defined by (a) Kmeans, (b) CURE and (c) DBSCAN

three clusters under the consideration of the suitable input parameters' values. As it is evident, if there is no visual perception of the clusters it is impossible to assess the validity of the partitioning. It is important then to be able to choose the optimal partitioning of a data set as a result of applying different algorithms with different input parameter values.

What is then needed is a visual-aids-free assessment of some objective criterion, indicating the validity of the results of a clustering algorithm application on a potentially high dimensional data set. In this paper we define and evaluate a cluster validity index, *CDbw* (Compose Density between and within clusters). Assuming a data set S, the index enables the selection of optimal input parameter values for a clustering algorithm that best partition S. Moreover, *CDbw* adjusts well to non-spherical clusters contrary to the validity indices proposed in the literature. It achieves this by considering multi-representative points per cluster.

The rest of the paper is organized as follows. Section 2 surveys the related work. We motivate and define the validity index in Section 3. Furthermore, in Section 4 we present an experimental study of our approach using synthetic dataset while we compare our approach to other validity indices. In Section 5 we conclude by briefly presenting our contributions and indicate directions for further research.

2. Related Work

The fundamental clustering problem is to partition a given data set into groups (clusters), such that the data points in a cluster are more similar to each other than points in different clusters [10]. In the clustering process, there are no predefined classes and no examples that would show what kind of desirable relations should be valid among the data [2]. This is what distinguishes clustering from classification [7, 8].

There is a multitude of clustering methods available in the literature, which can be broadly classified into the following types [11, 14]: i) *Partitional clustering ii)Hierarchical clustering, iii) Density-based clustering,* iv) *Grid-based clustering*.

For each of these types there exists a wealth of subtypes and different algorithms [1, 11, 12, 13, 14, 17, 19, 22, 26] for finding the clusters. In general terms, the clustering algorithms are based on a criterion for judging the validity of a given partitioning. Moreover, they define a partitioning of a data set based on certain assumptions and *not* the optimal one that fits the data set.

Since clustering algorithms discover clusters, which are not known a priori, the final partition of a data set requires some sort of evaluation in most applications [18]. A particularly difficult problem, which is often ignored in clustering algorithms is "how many clusters are there in the data set?".

Previously described requirements for the evaluation of clustering results is well known in the research community and a number of efforts have been made especially in the area of pattern recognition [22]. However, the issue of cluster validity is rather under-addressed in the area of databases and data mining applications, even though recognized as important. In general terms, there are three approaches to investigate cluster validity [22]. The first is based on *external criteria*. This implies that we evaluate the results of a clustering algorithm based on a pre-specified structure, which is imposed on a data set and reflects our intuition about the clustering structure of the data set. The second approach is based on *internal criteria*. We may evaluate the results of a clustering algorithm in terms of quantities that involve the vectors of the data set themselves (e.g., proximity matrix). The third approach of clustering validity is based on *relative criteria*. Here the basic idea is the evaluation of a clustering structure by comparing it with other clustering schemes, resulting by the same algorithm but with different parameter values. A number of validity indices have been defined and proposed in the literature for each of above approaches [22]. A cluster validity index for crisp clustering proposed in [4], attempts to identify "compact and well-separated clusters". Other validity indices for crisp clustering have been proposed in [3] and [16]. The implementation of most of these indices is very computationally expensive, especially when the number of clusters and number of objects in the data set grows very large [26]. In [15], an evaluation study of thirty validity indices proposed in the literature is presented. The results of this study place Caliski and Harabasz(1974), Je(2)/Je(1) (1984), C-index (1976), Gamma and Beale among the six best indices. However, it is noted that although the results concerning these methods are encouraging they are likely to be data dependent. For fuzzy clustering [22], Bezdek proposed the partition coefficient (1974) and the classification entropy (1984). The limitations of these indices are [3]: i) their monotonous dependency on the number of clusters, and ii) the lack of direct connection to the geometry of the data. Other fuzzy validity indices are proposed in [9, 26, 18]. We should mention that the evaluation of proposed indices and the analysis of their reliability are limited.

Another approach for finding the best number of cluster of a data set proposed in [21]. It introduces a practical clustering algorithm based on Monte Carlo cross-validation. This approach differs significantly from the one we propose. While we evaluate clustering schemes based on widely recognized validity criteria of clustering, the evaluation approach proposed in [21] is based on density functions considered for the data set. Thus, it uses concepts related to probabilistic models in order to estimate

the number of clusters, better fitting a data set, while we use concepts directly related to the data.

3. Introducing a new validity index

According to the literature [22] the clustering validity criteria are classified into: i. internal, ii. external, and iii. relative. Criteria of the categories i. and ii. are quite complex due to usage of Monte Carlo simulation which scans the data sets multiple times resulting in exponential complexities. In this research effort we focused on *relative* criteria where the algorithm is running repetitively using different input values and the resulting clusters are compared as for their validity.

The criteria widely accepted for partitioning a data set into a number of clusters are: i. the *separation of the clusters*, and ii. *their compactness*. Thus these criteria are obviously good candidates for checking the validity of clustering results.

Input parameter values. The examples discussed in Section 1 (Figure 1 and Figure 2) illustrate that the clustering algorithm's input parameter values are crucial for discovering the optimal partitioning of a data set during the clustering process. The data set is falsely partitioned in most of the cases (K-means, DBSCAN), whereas only specific values for the algorithms' input parameters lead to optimal partitioning of the data set. Here the term "optimal" implies parameters that lead to partitions that are as close as possible (in terms of similarity) to the real partitions of the data set.

Therefore our o*bjective* is the definition of a relative [22] algorithm-independent validity index, for assessing the quality of partitioning for each set of the input values. Such a validity index should be able to select for each algorithm under consideration the optimal set of input parameters with regard to a specific data set.

The criteria (i.e., compactness and separation) on which the proposed index is partially based are the fundamental criteria of clustering. However, the algorithms aim at satisfying these criteria based on initial assumptions (e.g. initial locations of the cluster centers) or input parameter values (e.g. the number of clusters, minimum diameter or number of points in a cluster). For instance the algorithm DBSCAN[6] defines clusters based on density variations, considering values for the cardinality and radius of an object's neighborhood. It finds the best partitions for the given input values but we don't know if the resulting partitions are the optimal or even the ones presented in the underlying data set.

The above motivated us to take into account density variations among clusters. We define our validity index, *CDbw*, combining both clustering criteria (compactness and separation) in terms of inter- and intra- cluster density. Moreover, the cluster indices proposed in the literature cannot handle properly arbitrary shaped clusters. In this work, we consider multi-representative points to represent the clusters defined by an algorithm. The result is a better description of the clusters' structure than this achieved by others approaches, which consider a single center point. Having more than one representative points per cluster allows CDbw to adjust well to the geometry of non-spherical shapes.

3.1 Index definition

In the sequel, we formalize our clustering validity index based on:

i. *clusters' compactness* (in terms of intra-cluster density), and
ii. *clusters' separation* (combining the distance between clusters and the inter-cluster density).

Let $D=\{V_1,\ldots, V_c\}$ a partitioning of a data set S into *c* convex clusters where V_i is the set of representative points of cluster i, that is, $V_i= \{v_{i1},\ldots, v_{ir} \mid r=\text{number of representatives per cluster}\}$ where v_{ij} is the jth representative of cluster i as it results from applying a clustering algorithm to S. Each cluster is represented by a certain fixed

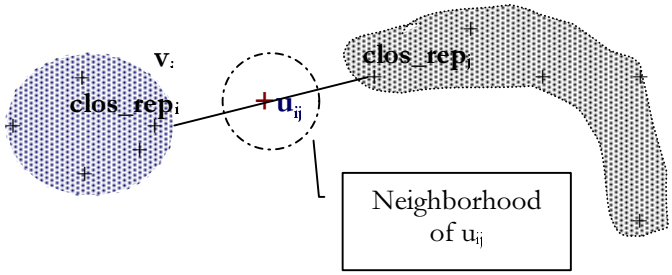


Figure 3. Inter-cluster density definition

number of points that are generated by selecting well-scattered points of the cluster. The procedure for defining the representatives of a cluster c_j is an iterative procedure. In the first iteration, the point farthest from the mean of the cluster under consideration is chosen as the first scattered point. In each subsequent iteration, a point form the cluster is chosen that is farthest from the previously chosen scattered points.

Let *stdev* be the average standard deviation of clusters defined as: $stdev = \frac{1}{c}\sqrt{\sum_{i=1}^{c}\|\sigma(v_i)\|}$

Further the term ||x|| is defined as : $\|x\| = (x^T x)^{1/2}$, where x is a vector. Then the overall inter-cluster density is defined as:

Definition 1. *Inter-cluster Density* (ID) - It evaluates the average density in the region among clusters. The goal is the density in the area among clusters to be significant low. Then, considering a partitioning of the data set into more than two clusters (i.e., c>1) the inter-cluster density is defined as follows:

$$Inter\_dens(c) = \sum_{i=1}^{c}\sum_{\substack{j=1 \\ i\neq j}}^{c}\left(\frac{d(clos\_rep_i, clos\_rep_j)}{stdev_i + stdev_j}\cdot density(u_{ij})\right), \quad c>1, c\neq n \qquad (1)$$

where $clos\_rep_i$, $clos\_rep_j$ are the closest representative points between clusters i and j and n the number of points in a data set.

Also, u_{ij} is the middle point of the line segment defined by the closest clusters' representatives $clos\_rep_i$, $clos\_rep_j$ (see Figure 3). The term density(u_{ij}) is defined in equation(2):

$$density(u_{ij}) = \frac{\sum_{l=1}^{n_i+n_j} f(x_l, u_{ij})}{n_i + n_j}, \qquad (2)$$

where $clos\_rep_i$, $clos\_rep_j$ are the closest representative points between cluster c_i and c_j and n the number of points in a data set. It represents the percentage of points in the cluster i and the cluster j that belong to the neighborhood of u_{ij}. The neighborhood of a data point, u_{ij}, is defined to be a hyper-sphere with center u_{ij} and radius the average standard deviation of the clusters between which we estimate the density. Also, the function $f(x, u_{ij})$ is defined as:

$$f(x, u_{ij}) = \begin{cases} 0, & \text{if } d(x, u_{ij}) > (stdev_i + stdev_j)/2 \\ 1, & \text{otherwise} \end{cases} \qquad (3)$$

It is obvious that a point belongs in the neighborhood of u_{ij} if its distance from u_{ij} is smaller than the average standard deviation of clusters.

However, the actual area between clusters, whose density we are interested to estimate, is defined to be the area between the closest representative points. Thus, we consider as

density between clusters (i.e., *Itrer-cluster density*) the $\frac{d(clos\_rep_i, clos\_rep_j)}{stedev_i + stdev_j}$ percentage of points that belong to the neighborhood of u_{ij}.
In case that each point is a separate cluster, i.e., c=n the $stedv_i = 0$, $\forall$ i=1,…,c and thus *density(u_{ij})=0*. Then, we consider that the inter cluster density is equal to 0. That is

$$Inter\_dens(n)=0$$

Definition 2. *Clusters' separation (Sep).* It evaluates the separation of clusters taking into account both the distances between the closest clusters and the Inter-cluster density. The goal is the distances among clusters to be high while the density in the area among them to be low. Then, the clusters' separation is given by the equation (4):

$$Sep(c) = \frac{\sum_{i=1}^{c}\sum_{\substack{j=1 \\ i\neq j}}^{c} \min\{d(clos\_rep_i, clos\_rep_j)\}}{1 + Inter\_dens(c)}, c > 1 \quad (4)$$

where $clos\_rep_i$, $clos\_rep_j$ are the closest representative points between clusters c_i and c_j.

Definition 3. *Intra-cluster density*. The average density within clusters is defined as the percentage of points that belong to the neighborhood of representative points of the considered clusters. The goal is the density within clusters to be significant high. It is given by the following equation:

$$Intra\_dens(c) = \frac{1}{c}\sum_{i=1}^{c}\frac{1}{r}\sum_{j=1}^{r}\frac{density(\underline{v}_{ij})}{stdev}, \quad c > 1,\ c \neq 0 \quad (5)$$

where $\underline{v}_{ij}$ corresponds to the j representative point of the cluster i shrinked toward the center of the cluster, v_{ij}, by a specified fraction.
The term density ($\underline{v}_{ij}$) is defined in equation (6):

$$density(\underline{v}_{ij}) = \sum_{l=1}^{n_i} f(x_l, \underline{v}_{ij}), \quad (6)$$

where n_i is the number of tuples that belong to the cluster c_i, i.e., $x_l \in c_i \subseteq S$. It represents the number of points in the neighborhood of the $\underline{v}_{ij}$ representative of the cluster i. In our work, the neighborhood of a data point, $\underline{v}_{ij}$, is defined to be a hyper-sphere with center $\underline{v}_{ij}$ and radius the average standard deviation of the clusters, *stdev*. The function *f(x,* $\underline{v}_{ij}$*)* is defined as,

$$f(x, \underline{v}_{ij}) = \begin{cases} 0, & \text{if } d(x, \underline{v}_{ij}) > stdev \\ 1, & \text{otherwise} \end{cases} \quad (7)$$

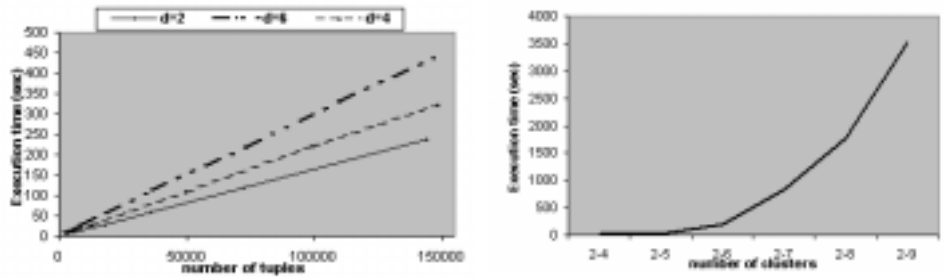
In case that each point is a separate cluster, i.e., c=n the $stedv_i = 0$, $\forall$ i=1,…,c and thus *density($\underline{v}_{ij}$)=1*. Then, we consider that the intra-cluster density is equal to 1. That is,

$$Itra\_dens(n) = 1$$

Then the validity index *CDbw* is defined as:

$$CDbw(c) = Intra\_dens(c) \cdot Sep(c), \ c > 1 \quad (8)$$

The above definitions refer to the case that a cluster presents clustering tendency, i.e., it can be partitioned into at least two clusters. The index is not defined for c=1.
In a good clustering scheme both terms of *CDbw* present high values which converges to a maximum for the optimal partitioning. Also, *CDbw* exhibits no trends with regards to the number of clusters and thus in the plot of *CDbw* versus the number of clusters we



(a) (b)

Figure 4: Execution time in seconds as function of (a) the number of points (b) the number of clusters

seek the maximum value of *CDbw*. The absence of a local maximum in the plot is an indication that the data set under consideration possesses no clustering structure. Also, *CDbw* exhibits no trends with regard to the number of clusters and thus in the plot of *CDbw* versus the number of clusters we seek the maximum value of *CDbw*. The absence of a local maximum in the plot is an indication that the data set under consideration possesses no clustering structure.

3.2 Time Complexity

The complexity of the validity index *CDbw*, is based on the complexity of its two terms as defined in equations (1) and (5). Assuming d the number of attributes (data set dimension); c is the number of clusters; n is the number of database tuples; r the number of a cluster' s representatives. Then the complexity of selecting the closest representative points of c clusters is $O(dc^2r^2)$. The intra-cluster density complexity is $O(ncrd)$ while the complexity of inter-cluster density is $O(ndc^2)$. Then *CDbw* complexity is $O(ndr^2c^2)$. Usually, c, d, $r << n$, therefore the complexity of our index for a specific clustering scheme is $O(n)$. The graphs in Figure 4 show the results of an experimental study referring to the execution time of our approach. The considered data sets for these experiments are synthetically generated according to the normal distribution. Figure 4a demonstrates that the execution time is almost linear to the number of points as expected from the preceding complexity study. Furthermore, we measured the execution time for data sets of higher dimensionality (two, four and six dimensions). Figure 4b shows the execution time as function of the number of clusters. The execution time, as expected, is nearly quadratic with respect to the number of clusters but as c is usually a small integer, it creates no problem.

4. Experimental evaluation

In this section *CDbw* is experimentally tested using representative clustering algorithms of different categories, partitional, hierarchical and density-based.

We experiment with synthetic multidimensional data sets containing different number of clusters. In all cases our approach performs favorably selecting the best partitioning among these proposed by an algorithm. Additionally we compare *CDbw* to other validity indices found in the literature. In the sequel, due to lack of space, we present only some representative examples of our experimental study.

4.1 Selection of the optimal partitioning defined by a clustering algorithm

The goal of this experiment is to evaluate our index with regards to the selection of the optimal clustering scheme by a specific clustering algorithm. More specifically, we consider a 2-dimensional data set consisting of four clusters (see Figure 5a). We define a number of different clustering schemes of our data set using the K-Means algorithm,

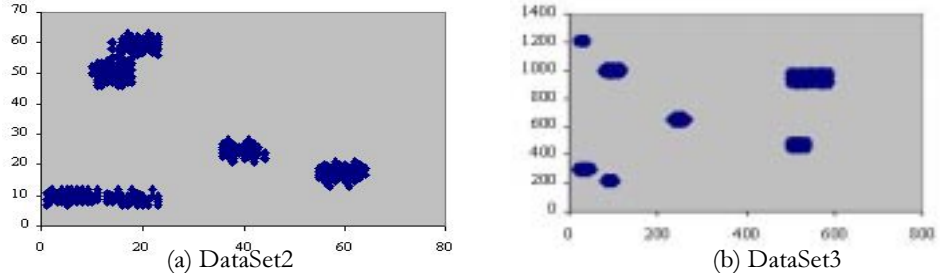


Figure 5: Sample Synthetic Data Sets for testing clustering schemes defined by a specific clustering algorithm.

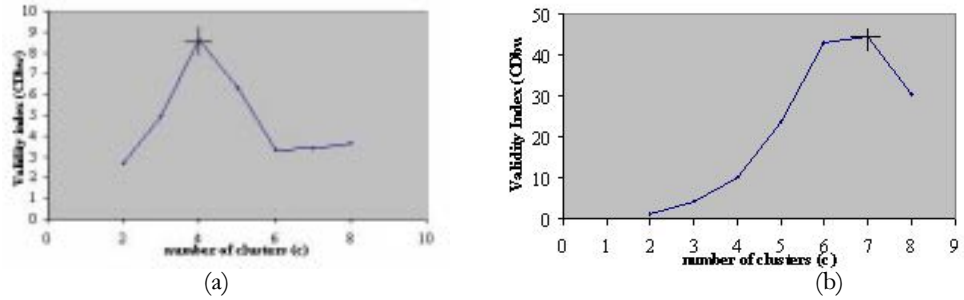


Figure 6. CDbw as a function of number of clusters for (a) DataSet2 and (b) DataSet3.

with its input parameters (number of clusters) ranging between 2 and 8. The behavior of *CDbw* is depicted in Figure 6a. It is clear that the correct number of clusters is proposed (i.e., four), as at this value the index reaches its maximum.

Similarly, we assume the clustering schemes of DataSet3 (see Figure 5b) as defined by CURE when the number of clusters ranges between 2 and 8. Then, we evaluated the defined clustering schemes based on the *CDbw* index so as to find which of them best fits the underlying data. As Figure 6b shows the clustering scheme of seven clusters is proposed as the best partitioning of DataSet3

A multidimensional data set. In the sequel, we demonstrate that our index works properly in multidimensional data sets. The validity of clustering results (i.e., that the set has been optimally partitioned) can be visually verified only in 2D or 3D cases. In higher dimensions it is difficult to verify the resulting clusters. The proposed index, *CDbw*, offers a solution to this problem giving an indication of the optimal clustering scheme without visualization of the data set. We consider a synthetic six-dimensional

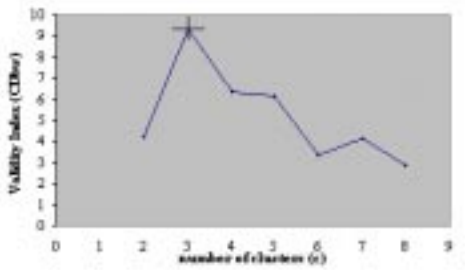


Figure 7: CDbw as a function of the number of clusters for a six-dimensional data set consisting of three clusters.

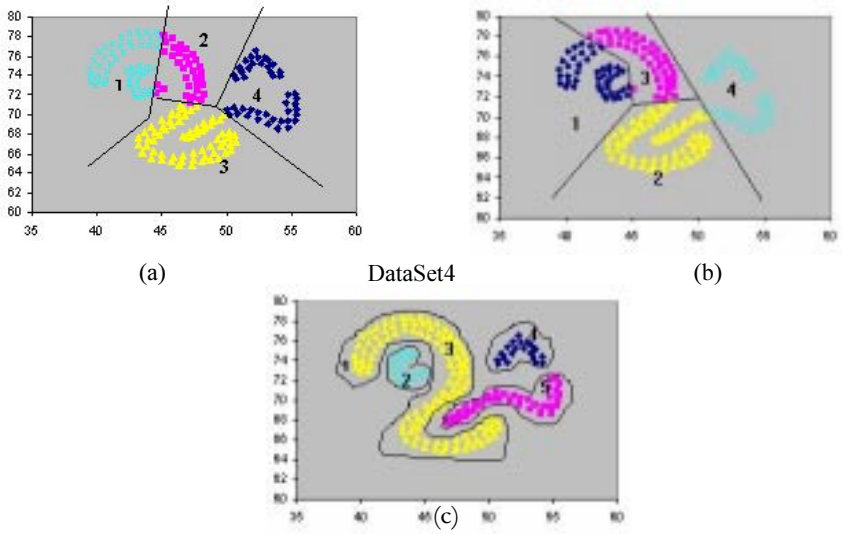


Figure 8. Partitioning of DataSet4 into four clusters as defined by (a) Kmeans, (b) CURE and (c)DBSCAN

data set, further referred as MD_Set, containing three distinct clusters. This is also verified by *CDbw*. As Figure 7 depicts, *CDbw* finds the correct number of clusters as it takes its maximum value when c=3.

4.2 The index is independent of clustering algorithm

As mentioned in previous sections, different input values for clustering algorithms applied to a data set result in different partitioning schemes. In the following we show that *CDbw* selects the optimal partitioning among those defined by a clustering algorithm independently of the algorithm used. Also, the clustering algorithm that defines the partitioning best fitting the data can be selected. We use three well-known algorithms, one from each of the popular algorithm categories: K-Means (partitional), DBSCAN (density based) and CURE (hierarchical).

Table 1a presents the *CDbw* values for the clustering schemes of the synthetic data set DataSet2 (see Figure 5a) as defined by K-Means, DBSCAN and CURE respectively. More specifically, we consider the clustering schemes revealed by the algorithms mentioned above while their input parameters values are depicted in Table 1. In the case of DataSet2, all three algorithms propose four clusters as the optimal clustering schemes (see Table 1a).

In some cases, however, an algorithm may partition a data set into the correct number of clusters but in a wrong way. *CDbw* can be considered to evaluate the results of different clustering algorithms and select the optimal partitioning among those proposed, i.e., to select the optimal algorithm for our data set. According to Table 1b, in case of DataSet4, *CDbw* takes its maximum value for the partitioning of four clusters defined by DBSCAN. This is also the number of actual clusters in the data set. Figure 8c presents the partitioning of Dataset4 into four clusters as defined by DBSCAN while the clustering result of K-Means and CURE into four clusters is presented in Figure 8a and Figure 8b respectively. It is obvious that K-Means and CURE fails to partition DataSet4 properly even in case that the correct number of clusters (i.e., c=4) is considered.

Similarly, we considered another data set, Dataset1, containing clusters with strange geometries. As Figure 2 depicts, the actual clusters in DataSet1 are three. However, the majority of clustering algorithms fail to partition it in a right way. Figure 2(a), Figure

	K-means		DBSCAN		CURE r =10, a=0.3	
No clusters	**Input**	**CDbw Value**	**Input**	**CDbw Value**	**Input**	**CDbw Value**
6	C=6	3.353	Eps=2, MinC =8	3.777	C=6	3.323
5	C=5	6.268	Eps=2, MinC =4	6.678	C=5	6.126
4	**C=4**	**8.163**	**Eps=10, MinC=15**	**8.163**	**C=4**	**8.163**
3	C=3	4.549	Eps=15, MinC=15	4.549	C=3	4.549
2	C=2	2.575	Eps=20,MinC=15	2.575	C=2	2.575

(a) DataSet2

	K-Means		DBSCAN		CURE r =10, a=0.3	
No clusters	**Input**	**CDbw Value**	**Input**	**CDbw Value**	**Input**	**CDbw Value**
6	C=6	0.0457	-	-	C=6	0.1304
5	C=5	0.046	_	_	**C=5**	**0.5656**
4	C=4	0.0293	**Eps=1,MinPts=4**	**1.0758**	C=4	0.317
3	C=3	0.0246	Eps=2 ,inPts=15	0.0053	C=3	0.2489
2	**C=2**	**0.0597**	Eps=2 ,inPts=10	0.789	C=2	0.4857

(bDataSet1

	K-Means		DBSCAN		CURE r =10, a=0.3	
No clusters	**Input**	**CDbw Value**	**Input**	**CDbw Value**	**Input**	**CDbw Value**
6	C=6	0.316	-	-	C=6	0.255
5	C=5	0.9805	_	_	**C=5**	**1.228**
4	C=4	1.006	_	_	C=4	1.118
3	C=3	0.8457	**Eps=2 , MinPts=4**	**1.4335**	C=3	1.077
2	C=2	1.368	Eps=10 , MinPts=4	1.3687	C=2	1.368

(c)DataSet4

Table 1: Optimal partitioning found by CDbw for different clustering algorithms

2(b) and Figure 2 (c) present the proposed partitioning of DataSet1 into three clusters as defined by K-Means, CURE and DBSCAN respectively. It is obvious that DBSCAN is the only algorithm that achieves to identify the actual clusters (i.e., the clusters that fits DataSet1). This is also verified by Table1c, which presents the values of *CDbw* for the clustering schemes defined by the considered clustering algorithms. *CDbw* takes its maximum value for the clustering scheme of three clusters as defined by DBSCAN.

Based on the above experimental study we may conclude that *CDbw* does not only select the optimal partitioning among the results of a specific clustering algorithm but can also assist to find the partitioning that best fits the considered data among the results of different algorithms. Thus, it selects the algorithm and its parameters values for which the optimal partitioning of a data set is defined. Moreover, *CDbw* handles efficiently arbitrary shaped clusters since its definition is based on multi-representative points describing the structure of clusters.

4.3 Comparison to other validity indices

We consider the known validity indices proposed in the literature, such as *RS-RMSSTD*[20], *DB*[22], *SD*[23] and the most recent one *S_Dbw* [24]. *RMSSTD* and *RS* have to be taken into account simultaneously in order to find the correct number of clusters. The optimal values of the number of clusters are those for which a significant local change in values of *RS* and *RMSSTD* occurs. As regards *DB, SD* and *S_Dbw* an indication of the optimal clustering scheme is the point at which it takes its minimum

	DataSet2	DataSet4	MD_Set
Optimal number of clusters	**4**	**4**	**3**
RS, RMSSTD	3	3	3
DB	6	3	3
SD	4	2	3
S_Dbw	4	3	3
CDbw	4	4	3

Table 2: Optimal number of clusters proposed by validity indices compared with *CDbw*

value. We carried an evaluation study comparing *CDbw* to the indices mentioned above. We used the 2-dimensional data sets DataSet2 (see Figure 5a) and DataSet4 (see Figure 8). Also we consider the six-dimensional data set, MD_Set, described in Section 4.1.

Table 2 summarizes the results of the validity indices (*RS, RMSSDT, DB, SD* and *S_Dbw*), for different clustering schemes of the above-mentioned data sets as resulting from a clustering algorithm (K-Means, CURE or DBSCAN). In case of DataSet2 and Nd_Set we use the results of the algorithm K-Means and CURE. Indices *RS, RMSSTD* propose the partitioning of DataSet2 into three clusters while DB selects six clusters as the best partitioning. On the other hand, *SD* and *S_Dbw, CDbw* select four clusters as the optimal partitioning for DataSet2, which is also the correct number of clusters fitting the underlying data. As regards MD_Set all indices propose three clusters as its best partitioning, which is also the actual clusters in the data set. In the case of DataSet4, we consider the results of DBSCAN since it handles efficiently arbitrary shaped clusters. Thus *CDbw* finds the correct number of clusters (four) for DataSet4, on the contrary to *RS – RMSSTD, S_Dbw* and *DB* indices, which propose three clusters as the best partitioning and *SD* that proposes the partitioning of two clusters.
In all cases *CDbw* finds the optimal number of clusters fitting a data set, while other validity indices fail in some cases.

5. Conclusions and Further Work

In this paper we addressed the important issue of *assessing the validity of clustering algorithms' results*, i.e., how close are the results to the real partitions of the data set (assuming that the data set presents clustering tendency). In most of the cases the users visually verify the clustering results. However, in the case of voluminous and/or multidimensional data sets where efficient visualization is difficult or even impossible, it becomes tedious to know if the results of clustering are valid or not.We have defined a new validity index, *CDbw*, for assessing the results of clustering algorithms. The index is optimized for data sets that include compact and well-separated clusters. The compactness of the data set is measured by the intra-cluster density whereas the separation by the density between clusters. We have proved *CDbw* reliability and value using various data sets of non-standard geometries covering also the multidimensional case. The index results, as indicated by experiments, are not dependent on the clustering algorithm used, and always indicate the optimal input parameters for the algorithm used in each case.

As further work, we plan an extension of this effort to be directed towards an integrated algorithm for cluster discovery putting emphasis on the geometric features of clusters, using sets of representative points, or even multidimensional curves rather than a single center point.

Acknowledgements

We are grateful to C. Rodopoulos and C. Amanatidis for the implementation of CURE algorithm as well as to Drs Joerg Sander and Eui-Hong (Sam) Han for providing information and the source code for DBSCAN and CURE algorithms respectively.

References

[1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, Prabhakar Raghavan, “Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications”, *Proceedings of SIGMOD*, 1998.

[2] Michael J. A. Berry, Gordon Linoff. Data Mining Techniques For marketing, Sales and Customer Support. John Willey & Sons, Inc, 1996.

[3] Rajesh N. Dave. "Validating fuzzy partitions obtained through c-shells clustering", *Pattern Recognition Letters*, Vol .17, pp613-623, 1996

[4] J. C. Dunn. "Well separated clusters and optimal fuzzy partitions", *J. Cybern.* Vol.4, pp. 95-104, 1974

[5] Martin Ester, Hans-Peter Kriegel, Jorg Sander, Michael Wimmer, Xiaowei Xu. "Incremental Clustering for Mining in a Data Warehousing Environment", *Proceedings of 24th VLDB Conference,* New York, USA, 1998.

[6] Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", *Proceedings of 2nd Int. Conf. On Knowledge Discovery and Data Mining,* Portland, OR, pp. 226-231, 1996.

[7] Usama M. Fayyad, Gregory Piatesky-Shapiro, Padhraic Smuth and Ramasamy Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI Press 1996

[8] Usama Fayyad, Ramasamy Uthurusamy. "Data Mining and Knowledge Discovery in Databases", *Communications of the ACM.* Vol.39, No11, November 1996.

[9] I. Gath, B. Geva. "Unsupervised Optimal Fuzzy Clustering". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 11, No7, July 1989.

[10] Sudipto Guha, Rajeev Rastogi, Kyueseok Shim. "CURE: An Efficient Clustering Algorithm for Large Databases", *Published in the Proceedings of the ACM SIGMOD Conference*, 1998.

[11] Sudipto Guha, Rajeev Rastogi, Kyueseok Shim. "ROCK: A Robust Clustering Algorithm for Categorical Attributes", *Published in the Proceedings of the IEEE Conference on Data Engineering*, 1999.

[12] Alexander Hinneburg, Daniel Keim. "An Efficient Approach to Clustering in Large Multimedia Databases with Noise". *Proceeding of KDD '98,* 1998.

[13] Zhexue Huang. "A Fast Clustering Algorithm to Cluster very Large Categorical Data Sets in Data Mining", *DMKD,* 1997

[14] A.K Jain, M.N. Murty, P.J. Flyn. “Data Clustering: A Review”, *ACM Computing Surveys*, Vol. 31, No3, September 1999.

[15] Milligan, G.W. and Cooper, M.C. (1985), "An Examination of Procedures for Determining the Number of Clusters in a Data Set", *Psychometrika*, 50, 159-179.

[16] Milligan G. W., Soon S.C., Sokol L. M. “The effect of cluster size, dimensionality and the number of clusters on recovery of true cluster structure”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5, pp. 40-47, 1983

[17] Raymond Ng, Jiawei Han. "Efficient and Effective Clustering Methods for Spatial Data Mining". *Proceeding of the 20th VLDB Conference,* Santiago, Chile, 1994.

[18] Ramze Rezaee, B.P.F. Lelieveldt, J.H.C Reiber. "A new cluster validity index for the fuzzy c-mean", *Pattern Recognition Letters*, 19, pp237-246, 1998.

[19] C. Sheikholeslami, S. Chatterjee, A. Zhang. "WaveCluster: A-MultiResolution Clustering Approach for Very Large Spatial Database". *Proceedings of 24th VLDB Conference, New York*, USA, 1998.

[20] Sharma S.C. *Applied Multivariate Techniques.* John Willwy & Sons, 1996.

[21] Padhraic Smyth. "Clustering using Monte Carlo Cross-Validation". *KDD 1996*, 126-133.

[22] S. Theodoridis, K. Koutroubas. *Pattern recognition,* Academic Press, 1999

[23] M. Halkidi, M. Vazirgiannis, Y. Batistakis. "Quality scheme assessment in the clustering process", *In Proceedings of PKDD*, Lyon, France, 2000.

[24] M. Halkidi, M. Vazirgiannis, “Clustering Validity Assessment: Finding the optimal partitioning of a data set”, In the Proceedings of ICDM Conference,California, USA, November 2001.

[25] Tian Zhang, Raghu Ramakrishnman, Miron Linvy. "BIRCH: An Efficient Method for Very Large Databases", *ACM SIGMOD' 96,* Montreal, Canada, 1996.

[26] Xunali Lisa Xie, Genardo Beni. "A Validity measure for Fuzzy Clustering", *IEEE Transactions on Pattern Analysis and machine Intelligence,* Vol13, No4, August 1991.