A Bounded Index for Cluster Validity

Sandro Saitta, Benny Raphael, and Ian F.C. Smith

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Station 18, 1015 Lausanne, Switzerland
sandro.saitta@epfl.ch,bdgbr@nus.edu.sg,ian.smith@epfl.ch

Abstract. Clustering is one of the most well known types of unsupervised learning. Evaluating the quality of results and determining the number of clusters in data is an important issue. Most current validity indices only cover a subset of important aspects of clusters. Moreover, these indices are relevant only for data sets containing at least two clusters. In this paper, a new bounded index for cluster validity, called the score function (SF), is introduced. The score function is based on standard cluster properties. Several artificial and real-life data sets are used to evaluate the performance of the score function. The score function is tested against four existing validity indices. The index proposed in this paper is found to be always as good or better than these indices in the case of hyperspheroidal clusters. It is shown to work well on multidimensional data sets and is able to accommodate unique and sub-cluster cases.

Key words: clustering, cluster validity, validity index, k-means

1 Introduction

The goal of clustering [1, 2] is to group data points that are similar according to a chosen similarity metric (Euclidean distance is commonly used). Clustering techniques have been applied in domains such as text mining [3], intrusion detection [4] and object recognition [5]. In these fields, as in many others, the number of clusters is usually not known in advance.

Several clustering techniques can be found in the literature. They usually belong to one of the following categories [6]: partitional clustering, hierarchical clustering, density-based clustering and grid-based clustering. An additional category is the mixture of Gaussian approach. Since its computational complexity is high, it is not likely to be used in practice. All these categories have drawbacks. For example, hierarchical clustering has a higher complexity. Density-based clustering algorithms often require tuning non-intuitive parameters. Finally, density-based clustering algorithms do not always reveal clusters of good quality. The K-means [1] algorithm, part of the partitional clustering, is the most widely used. Advantages of K-means include computational efficiency, fast implementation and easy mathematical background. However, K-means also has limitations. They include a random choice of centroid locations at the beginning of the procedure, treatment of categorical variables and an unknown number of clusters

k. Concerning the first limitation, multiple runs may be a solution. The paper by [7] contains a possible solution to the second limitation through the use of a matching dissimilarity measure to handle categorical parameters. Finally, the third issue is related to the number of clusters and therefore cluster validity.

Clustering is by definition a subjective task and this is what makes it difficult [8]. Examples of challenges in clustering include i) the number of clusters present in the data and ii) the quality of clustering [9]. Elements of answers to these two issues can be found in the field of cluster validation. Other challenges such as initial conditions and high dimensional data sets are of importance in clustering. The aim of cluster validation techniques is to evaluate clustering results [6, 8, 10]. This evaluation can be used to determine the number of clusters within a data set. Current literature contains several examples of validity indices [9, 11–13]. Recent work has also been done on evaluating them [14].

The Dunn index [11] combines dissimilarity between clusters and their diameters to estimate the most reliable number of clusters. As stated in [6], the Dunn index is computationally expensive and sensitive to noisy data. The concepts of dispersion of a cluster and dissimilarity between clusters are used to compute the Davies-Bouldin index [12]. The Davies-Bouldin index has been found to be among the best indices [14]. The Silhouette index [13] uses average dissimilarity between points to identify the structure of the data and highlights possible clusters. The Silhouette index is only suitable for estimating the first choice or the best partition [15]. Finally, the Maulik-Bandyopadhyay index [9] is related to the Dunn index and involves tuning of a parameter.

All of these indices require the specification of at least two clusters. As noted in [16], the one cluster case is important and is likely to happen in practice. As a prerequisite to the identification of a single cluster, a definition of what is a cluster is important. Among those that exist in the literature, a possible definition is given in [17]. Briefly, it states that a cluster is considered to be "real" if it is significantly compact and isolated. Concepts of compactness and isolation are based on two parameters that define internal properties of a cluster. While this definition is precise, it is often too restrictive since few data sets satisfy such criteria. More details of single cluster tests can be found in [16]. Other validity indices exist in the literature. Some are computationally expensive [6] while others are unable to discover the real number of clusters in all data sets [14]. This paper proposes a new validity index that helps overcome such limitations.

This article is organized as follows. Section 2 describes existing validity indices from the literature. Section 3 proposes a new validity index, named the score function. Performance of the score function is described in Section 4. The last Section provides conclusions and directions for future work.

2 Existing Indices

In this Section, four validity indices suitable for hard partitional clustering are described. These indices serve as a basis for evaluating results from the score function on benchmark data sets. Notation for these indices have been adapted to provide a coherent basis. The metric used on the normalized data is the standard Euclidean distance defined as $||x-y|| = \sqrt{\sum_{i=1}^{d}(x_i-y_i)^2}$ where x and y are data points and d is the number of dimensions.

Dunn index: One of the most cited indices is proposed by [11]. The Dunn index (DU) identifies clusters which are well separated and compact. The goal is therefore to maximize the inter-cluster distance while minimizing the intra-cluster distance. The Dunn index for k clusters is defined by Equation 1:

$$DU_k = \min_{i=1,\ldots,k}\left\{\min_{j=1+1,\ldots,k}\left(\frac{diss(c_i,c_j)}{\max_{m=1,\ldots,k} diam(c_m)}\right)\right\} \tag{1}$$

where $diss(c_i,c_j) = \min_{x\in c_i, y\in c_j} ||x-y||$ is the dissimilarity between clusters c_i and c_j and $diam(C) = \max_{x,y\in C} ||x-y||$ is the intra-cluster function (or diameter) of the cluster. If Dunn index is large, it means that compact and well separated clusters exist. Therefore, the maximum is observed for k equal to the most probable number of clusters in the data set.

Davies-Bouldin index: Similar to the Dunn index, Davies-Bouldin index [12] identifies clusters which are far from each other and compact. Davies-Bouldin index (DB) is defined according to Equation 2:

$$DB_k = \frac{1}{k}\sum_{i=1}^{k}\max_{j=1,\ldots,k,i\neq j}\left\{\frac{diam(c_i)+diam(c_j)}{||c_i-c_j||}\right\} \tag{2}$$

where, in this case, the diameter of a cluster is defined as:

$$diam(c_i) = \left(\frac{1}{n_i}\sum_{x\in c_i}||x-z_i||^2\right)^{1/2} \tag{3}$$

with n_i the number of points and z_i the centroid of cluster c_i. Since the objective is to obtain clusters with minimum intra-cluster distances, small values for DB are interesting. Therefore, this index is minimized when looking for the best number of clusters.

Silhouette index: The silhouette statistic [13] is another well known way of estimating the number of groups in a data set. The Silhouette index (SI) computes for each point a width depending on its membership in any cluster. This silhouette width is then an average over all observations. This leads to Equation 4:

$$SI_k = \frac{1}{n}\sum_{i=1}^{n}\frac{(b_i-a_i)}{\max(a_i,b_i)} \tag{4}$$

where n is the total number of points, a_i is the average distance between point i and all other points in its own cluster and b_i is the minimum of the average dissimilarities between i and points in other clusters. Finally, the partition with the highest SI is taken to be optimal.

Maulik-Bandyopadhyay index: A more recently developed index is named the I index [9]. For consistence with other indices it is renamed MB. This index, which is a combination of three terms, is given through Equation 5:

$$MB_k = \left(\frac{1}{k} \cdot \frac{E_1}{E_k} \cdot D_k\right)^p \tag{5}$$

where the intra-cluster distance is defined by $E_k = \sum_{i=1}^{k} \sum_{x \in c_i} ||x - z_i||$ and the inter-cluster distance by $D_k = \max_{i,j=1}^{k} ||z_i - z_j||$. As previously, z_i is the center of cluster c_i. The correct number of clusters is estimated by maximizing Equation 5. In this work, p is chosen to be two.

Discussion: Although all these indices are useful in certain situations, they are not of general-purpose. For example, Dunn index is computationally heavy and has difficulty to deal with noisy data. It is useful for identifying clean clusters in data sets containing no more than hundreds of points. Davies-Bouldin index gives good results for distinct groups. However, it is not designed to accommodate overlapping clusters. The Silhouette index is only able to identify the first choice and therefore should not be applied to data sets with sub-clusters. The Maulik-Bandyopadhyay index has the particularity of being dependent on a user specified parameter.

3 Score Function

In this paper, we propose a function to estimate the number of clusters in a data set. The proposed index, namely the score function (SF), is based on inter-cluster and intra-cluster distances. The score function is used for two purposes: i) to estimate the number of clusters and ii) to evaluate the quality of the clustering results. The score function is a function combining two terms: the distance between clusters and the distance inside a cluster. The first notion is defined as the "between class distance" (*bcd*) whereas the second is the "within class distance" (*wcd*) .

Three common approaches exist to measure the distance between two clusters: single linkage, complete linkage and comparison of centroids. This proposal is based on the third concept since the first two have computational costs that are too high [6]. In this work, the *bcd* is defined by Equation 6:

$$bcd = \frac{\sum_{i=1}^{k} ||z_i - z_{tot}|| \cdot n_i}{n \cdot k} \tag{6}$$

where k is the number of clusters, z_i the centroid of the current cluster and z_{tot} the centroid of all the clusters. The size of a cluster, n_i is given by the number of points inside it. The most important quantity in the *bcd* is the distance between z_i and z_{tot}. To limit the influence of outliers, each distance is weighted by the cluster size. This has the effect to reduce the sensitivity to noise. Through n, the *bcd* sensitivity to the total number of points is avoided. Finally, values for k are used to penalize the addition of a new cluster. This way, the limit of one point per cluster is avoided. The *wcd* is given in Equation 7:

$$wcd = \sum_{i=1}^{k} \left(\frac{1}{n_i} \sum_{x \in c_i} ||x - z_i|| \right) \tag{7}$$

Computing values for *wcd* involves determining the distance between each point to the centroid of its cluster. This is summed over the k clusters. Note that $||z_i - x||$ already takes into account the size of the corresponding cluster. As in *bcd* (Equation 6), the cluster size in the denominator avoids the sensibility to the total number of points. With Equations 6 and 7, *bcd* and *wcd* are independent of the number of data points.

For the score function to be effective, it should i) maximize the *bcd*, ii) minimize the *wcd* and iii) be bounded. Maximizing Equation 8 satisfies the above conditions:

$$SF = 1 - \frac{1}{e^{e^{bcd-wcd}}} \tag{8}$$

The higher the value of the SF, the more suitable the number of clusters. Therefore, with the proposed SF, it is now possible to estimate the number of clusters for a given set of models. Difficulties such as perfect clusters ($wcd = 0$) and unique cluster ($bcd = 0$) are overcome. Moreover, the proposed score function is bounded by $]0, 1[$. The upper bound allows the examination of how close the current data set is to the perfect cluster case. Thus we seek to maximize Equation 8 to obtain the most reliable number of clusters. As can be seen through Equations 6 and 7, computational complexity is linear. If n is the number of data points, then the proposed score function has a complexity of O(n). In the next Section, the score function is tested on several benchmark problems and compared with existing indices.

4 Results

In this Section, the performance of validity indices are compared. For this purpose, the standard K-means algorithm is used. K-means is a procedure that iterates over k clusters in order to minimize their intra-cluster distances. The K-means procedure is as follows. First, k centroids are chosen randomly over all the points. The data set is then partitioned according to the minimum squared distance. New centroid positions are calculated according to the points inside clusters. The process of partitioning and updating is repeated until a stopping criterion is reached. This happens when either the cluster centers or the intra-cluster distances do not significantly change over two consecutive iterations.

To control the randomness of K-means, it is launched 10 times from k_{min} to k_{max} clusters. The optimum - minimum or maximum, depending on the index - is chosen as the most suitable number of clusters. The indices for comparison have been chosen according to their performance and usage reported in the literature (see Section 1). Selected indices are Dunn (DU), Davies-Bouldin (DB), Silhouette (SI) and Maulik-Bandyopadhyay (MB). These are compared with the

Score Function (SF). Results according to the number of clusters identified for the proposed benchmarks are shown next. Particularities of the score function such as perfect and unique clusters as well as hierarchy of clusters are then tested. Finally, examples of limitations concerning the score function are given.

4.1 Number of clusters

The score function has been tested on benchmark data sets and results are compared with other indices. k_{min} and k_{max} are taken to be respectively 2 and 10. Artificial data sets used in this Section are composed of 1000 points in two dimensions.

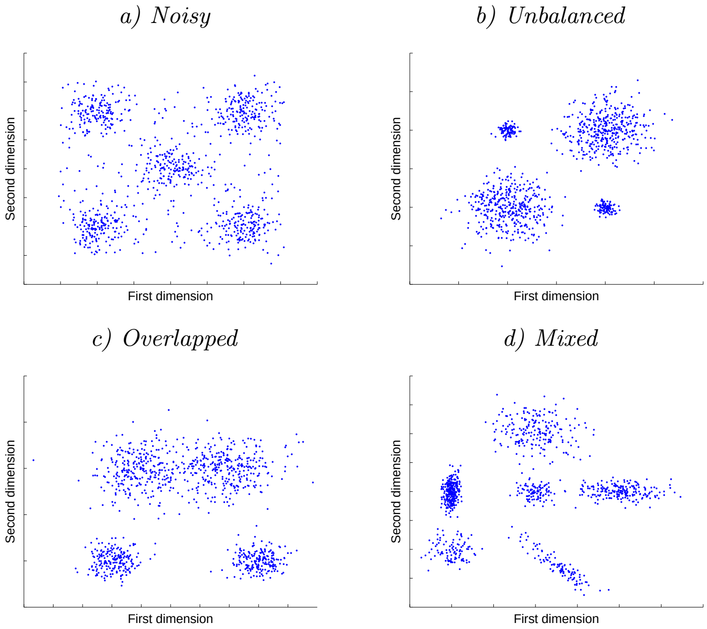


Fig. 1. Four artificial data sets, namely *Noisy*, *Unbalanced*, *Overlapped* and *Mixed*. All of these data sets contains 1000 points in 2D space.

Example 1: In the first data set, *Noisy*, five clusters in a noisy environment are present (see Figure 1a). It is improbable that a data set contains no noise. Therefore, clusters are frequently surrounded by noise. Table 1 shows that, unlike other indices, the Dunn index is not able to estimate correctly the number of clusters (five). This result confirms the idea that the Dunn index is sensitive to noise [6].

Example 2: The second data set, *Unbalanced*, consists of four clusters (see Figure 1b). These clusters are of different sizes and densities. According to [18],

k	2	3	4	5	6	7	8	9	10
DU	0.018	0.016	0.019	0.019	0.032	**0.035**	0.027	0.028	0.023
DB	1.060	0.636	0.532	**0.440**	0.564	0.645	0.665	0.713	0.729
SI	0.534	0.573	0.719	**0.821**	0.785	0.768	0.733	0.706	0.669
MB	1.314	2.509	3.353	**5.037**	4.167	3.323	2.898	2.515	2.261
SF	0.424	0.489	0.553	**0.592**	0.584	0.578	0.575	0.573	0.572

Table 1. Results of the five validity indices on the *Noisy* data set (example 1). The data set is shown in Figure 1a. Bold numbers show maximum values for all indices except DB, where minimum values is desired. This indication is used for Tables 1-6. The correct number of clusters is five.

clusters of varying densities are of importance. Table 2 shows the results for this data set. Whereas DU underestimates the number of clusters, MB overestimates it. This is not the case for DB, SI and SF which correctly identify four clusters.

k	2	3	4	5	6	7	8	9	10
DU	**0.154**	0.066	0.025	0.024	0.016	0.018	0.014	0.012	0.016
DB	0.739	0.522	**0.347**	0.552	0.633	0.712	0.713	0.722	0.733
SI	0.709	0.688	**0.803**	0.689	0.704	0.701	0.679	0.683	0.590
MB	3.900	3.686	4.795	4.751	**4.941**	4.844	4.540	3.575	3.794
SF	0.549	0.563	**0.601**	0.593	0.591	0.589	0.589	0.588	0.589

Table 2. Results of the five validity indices on the *Unbalanced* data set (example 2). The data set is shown in Figure 1b. The correct number of clusters is four.

Example 3: This data set, named *Overlapped*, contains four clusters, two of them overlap. It can be seen in Figure 1c. Two clusters are likely to overlap in real-life data sets. Therefore, the ability to deal with overlapping cluster is one of the best ways to compare indices [19]. Table 3 contains the results for this data set. It can be seen that DU and DB underestimate the correct number of clusters. Only SI, MB and SF are able to identify four clusters.

k	2	3	4	5	6	7	8	9	10
DU	**0.030**	0.025	0.013	0.013	0.012	0.019	0.021	0.012	0.012
DB	0.925	**0.451**	0.482	0.556	0.701	0.753	0.743	0.774	0.761
SI	0.635	0.740	**0.818**	0.728	0.713	0.669	0.683	0.669	0.656
MB	1.909	3.322	**5.755**	5.068	4.217	3.730	3.527	3.150	3.009
SF	0.452	0.555	**0.610**	0.601	0.593	0.589	0.588	0.585	0.584

Table 3. Results of the five validity indices on the *Overlapped* data set (example 3). The data set is shown in Figure 1c. The correct number of clusters is four.

Example 4: The following data set, named *Mixed*, contains six clusters. They have different size, compactness and shape. The data set is shown in Figure 1d. Table 4 presents results. First, it can be seen that DU is maximum for two consecutive values (although not the correct ones). MB is the only index to overestimate the correct number of clusters. Finally, only DB, SI and SF are able to identify correctly six clusters.

k	2	3	4	5	6	7	8	9	10
DU	0.015	**0.041**	**0.041**	0.027	0.018	0.020	0.014	0.018	0.017
DB	1.110	0.751	0.630	0.575	**0.504**	0.554	0.596	0.641	0.662
SI	0.578	0.616	0.696	0.705	**0.766**	0.744	0.758	0.730	0.687
MB	1.523	1.574	2.379	2.813	3.389	3.661	**3.857**	3.490	3.236
SF	0.442	0.492	0.540	0.559	**0.583**	0.579	0.577	0.576	0.579

Table 4. Results of the five validity indices on the *Mixed* data set (example 4). The data set is shown in Figure 1d. The correct number of clusters is six.

Example 5: The data set used in this example, *Iris* is one of the most used real-life data sets in the machine learning and data mining communities [20]. It is composed of 150 points in four dimensions. *Iris* contains three clusters (two of them are not linearly separable). It is a good example of a case where the dimension is more than two and clusters overlap. Table 5 shows the index values for this data set. In this case, only SF is able to correctly identify the three clusters. The overlap is too strong for other tested indices to enumerate the clusters.

k	2	3	4	5	6	7	8	9	10
DU	**0.267**	0.053	0.070	0.087	0.095	0.090	0.111	0.091	0.119
DB	**0.687**	0.716	0.739	0.744	0.772	0.791	0.833	0.752	0.778
SI	**0.771**	0.673	0.597	0.588	0.569	0.561	0.570	0.535	0.580
MB	**8.605**	8.038	6.473	6.696	5.815	5.453	4.489	4.011	4.068
SF	0.517	**0.521**	0.506	0.507	0.503	0.503	0.497	0.510	0.513

Table 5. Results of the five validity indices on the *Iris* data set (example 5). The data set is made by 150 points in a 4D space. The correct number of clusters is three (two of them overlap).

Example 6: The next data set, named *Wine*, is also a real-life data set [20]. It contains 178 points in 13 dimensions. *Wine* data set contains three clusters. Results of the five indices are given in Table 6. Whereas DU overestimates the correct number of clusters, MB underestimates it. DB, SI and SF are able to discover the three clusters.

k	2	3	4	5	6	7	8	9	10
DU	0.160	0.232	0.210	0.201	0.202	0.208	**0.235**	0.206	0.214
DB	1.505	**1.257**	1.499	1.491	1.315	1.545	1.498	1.490	1.403
SI	0.426	**0.451**	0.416	0.394	0.387	0.347	0.324	0.340	0.288
MB	**5.689**	5.391	3.548	2.612	2.302	2.124	1.729	1.563	1.387
SF	0.131	**0.161**	0.151	0.146	0.143	0.145	0.147	0.149	0.150

Table 6. Results of the five validity indices on the *Wine* data set (example 6). The data set is made of 178 points in a 13 dimension space. The correct number of clusters is three.

Table 7 summarizes the results of the application of the five indices to four artificial and two real-life data sets. Among the five indices tested, SF has the best performance. SF correctly identified the number of clusters in all six data sets. The SF successfully processes the standard case with clusters and noise (*Noisy*), clusters of different size and compactness (*Unbalanced*), overlapped clusters (*Overlapped*), multiple kind of clusters (*Mixed*) and multidimensional data (*Iris* and *Wine*).

Data Sets	**DU**	**DB**	**SI**	**MB**	**SF**
Noisy	7(X)	5(O)	5(O)	5(O)	5(O)
Unbalanced	2(X)	4(O)	4(O)	6(X)	4(O)
Overlapped	2(X)	3(X)	4(O)	4(O)	4(O)
Mixed	3/4(X)	6(O)	6(O)	8(X)	6(O)
Iris	2(X)	2(X)	2(X)	2(X)	3(O)
Wine	8(X)	3(O)	3(O)	2(X)	3(O)

Table 7. Estimated number of clusters for six data sets and five cluster validity indices. Notation indicates when the correct number of clusters has been found (O) or not (X).

4.2 Perfect Clusters

Since the score function is bounded, its upper limit (1.0) can be used to estimate the closeness of data sets to perfect clusters. The next two data sets are used to test how the SF deals with perfect clusters. The data sets *Perfect3* and *Perfect5* are made of 1000 points in 2D and contain three and five clusters respectively which are near to perfect (i.e. with a very high compactness). Although the number of clusters is correctly identified, it is interesting to note that the maximum value for the SF is different in both cases. In the three cluster case, the maximum (0.795) is higher than in the second one (0.722). This is due to the dependence of the SF on the number of clusters k. This can be seen in the denominator of Equation 6. Nevertheless, the SF gives an idea of how good clusters are through the proximity of the value of the index to its upper bound of unity.

4.3 Unique Cluster

An objective of the SF is to accommodate the unique cluster case. This case is not usually treated by others. In this subsection, k_{min} and k_{max} are taken to be respectively, 1 and 8. When the SF is plotted against the number of clusters, two situations may occur. Either the number of clusters is clearly located with a local maximum (Figure 2, left) or the SF grows monotonically between k_{min} and k_{max} (Figure 2, right).

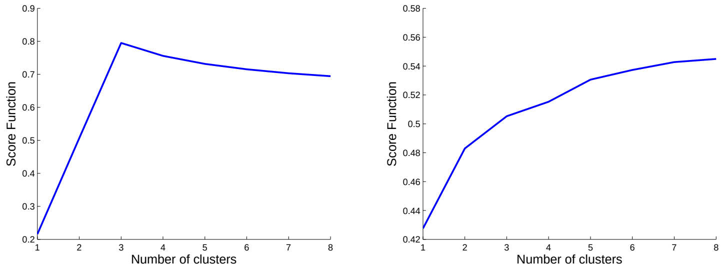


Fig. 2. Difference of the SF trend with a data set containing three clusters (left) and one cluster (right).

Since in the first situation, the number of clusters is identifiable, the challenge lies in the second situation. There are three possible cases. They are: i) no structure in the data, ii) data that forms one cluster and iii) the correct number of clusters is higher than k_{max}. The first situation is out of the scope of this article. More details of whether the data is structured or not, known as cluster tendency, can be found in [1]. In the last two situations, the SF grows monotonically with the number of clusters.

Two observations have been noticed. First, in the unique cluster cases, the value of the SF when $k = 2$, denoted as SF_2 is closer to the value for $k = 1$ (SF_1) than in other data sets. Second, the SF is dependent upon the dimensionality of the data set. Therefore, the slope between SF_2 and SF_1 weighted by the dimensionality of the data set is used as an indicator. To test the unique cluster case, two new data sets are introduced: *UniqueN* is a unique cluster with an added noise and *Unique30* is a unique cluster in a 30 dimensional space. Results of this indicator on all data sets are given in Table 8.

According to Table 8, it is empirically stated that the data set is likely to contain more than one cluster if Equation 9 is satisfied.

$$(SF_2 - SF_1) \cdot d > 0.2 \qquad (9)$$

where d is the dimensionality of the data, SF_2 and SF_1 are respectively the value for SF when $k = 2$ and $k = 1$. Only two data sets containing unique

Data sets	**Indicator**	**Data sets**	**Indicator**
Noisy	0.37	*UniqueN*	**0.11**
Unbalanced	0.65	*Unique30*	**0.10**
Overlapped	0.45	*Iris*	1.49
Mixed	0.41	*Wine*	1.31

Table 8. Results of the indicator $(SF_2 - SF_1) \cdot d$ for eight benchmark data sets.

clusters do not satisfy the condition in Equation 9. Therefore, the index SF is the only one, among all tested indices, that is able to identify a unique cluster situation.

4.4 Sub-clusters

Another interesting study concerns the sub-cluster case. This situation occurs when existing clusters can be seen as a cluster hierarchy. If this hierarchy can be captured by the validity index, more information about the structure of the data can be given to the user. Data set *Sub-cluster* in Figure 3 is an example of this situation. The index SF is compared with the previously mentioned indices on this topic. Figure 3 shows the evolution of each validity index with respect to the number of clusters.

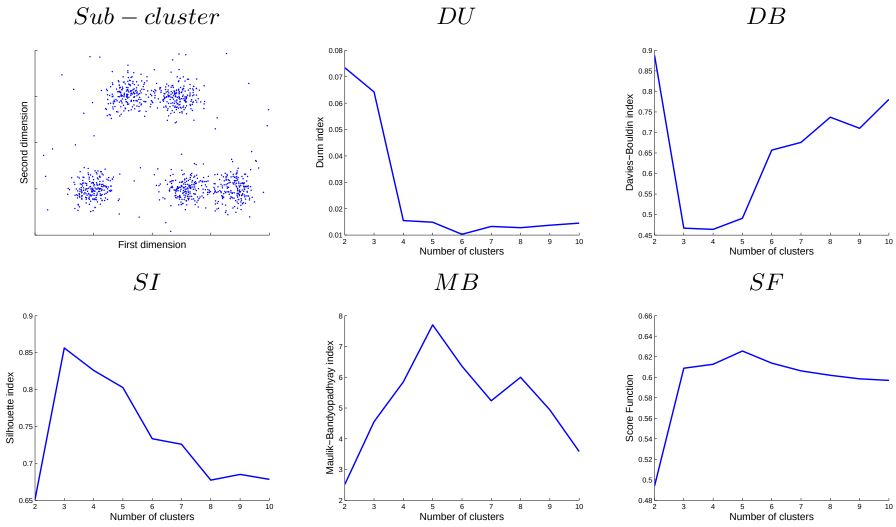
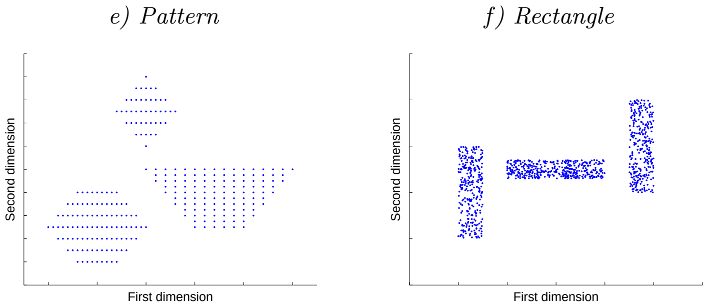


Fig. 3. Comparison of DU, DB, SI, MB and SF for the sub-cluster case. DB must be minimized.

DU is not able to find the correct number of clusters (neither the sub-clusters, nor the overall clusters). Although MB finds the sub-clusters, no information about the hierarchy is visible. In the case of DB, even if it is not able to find the five clusters (it finds four), the sub-cluster hierarchy is visible because the value of the index drops rapidly at three clusters. The SI index is not able to recover the correct number of clusters (i.e. the sub-clusters) although it can find the three overall clusters. Finally, the only index which is capable of giving the correct five clusters as well as an indication for the three overall clusters is SF.

4.5 Limitations

In the above subsections, data sets used to test the different indices contain hyperspheroidal clusters. In this subsection, arbitrarily-shaped clusters are briefly studied using two new data sets. *Pattern* is a data set containing 258 points in 2D. It contains three clusters with a specific pattern and different shapes. *Rectangle* is made of 1000 points in 2D that represent three rectangular clusters. These data sets are shown in Figure 4.


e) Pattern

f) Rectangle

Fig. 4. Two new artificial data sets. *Pattern* and *Rectangle* contain respectively 258 and 1000 points in 2D.

Regarding the *Pattern* data set, all indices are able to find the correct number of clusters (3). The proposed shapes and the pattern do not reveal weaknesses in any index. Concerning the *Rectlangle* data set, results are different. The proposed score function is not able to discover the three clusters. All other tested indices fail as well. All indices overestimates the correct number of clusters: DU (9), DB (7), SI (8), MB (8) and SF (10). A likely explanation is that clusters are far from hyperspheroidal shaped. Therefore, a limitation of the score function, as well as other tested indices, is their restriction to data sets containing hyperspheroidal clusters.

5 Conclusions

Although there are several proposals for validity indices in the literature, most of them succeed only in certain situations. A new index for hard clustering - the score function (SF) - is presented and studied in this paper. The proposed index is based on a combination of the within and between class distances. It can accommodate special cases such as the unique cluster and perfect cluster cases. The SF is able to estimate correctly the number of clusters in several artificial and real-life data sets. The SF has successfully estimated the number of clusters in data sets containing unbalanced, overlapped and noisy clusters. In addition, the SF has been tested successfully on multidimensional real-life data sets. No other index performed as well on all data sets. Finally, in the case of sub-cluster hierarchies, only the SF was able to estimate five clusters and overall, three groups. Therefore, the index SF outperforms four other validity indices (Dunn, Davies-Bouldin, Silhouette and Maulik-Bandyopadhyay) for the k-means algorithm on hyperspheroidal clusters. The proposed index can also accommodate perfect and unique cluster cases. In order to identify the one cluster case, an empirical condition has been formulated. Finally, determining values for the index is computationally efficient.

Several extensions to the present work are in progress. For example, a theoretical justification for the unique cluster condition (Equation 9) is under study. More extensive testing on arbitrarily shaped clusters is necessary. Finally, studies of other clustering algoritms are also under way.

Acknowledgments

This research is funded by the Swiss National Science Foundation (grant no 200020-109257). The authors recognize Dr. Fleuret for fruitful discussions and the two anonymous reviewers for their helpful comments.

References

1. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice Hall (1988)
2. Webb, A.: Statistical Pattern Recognition. Wiley (2002)
3. SanJuan, E., Ibekwe-SanJuan, F.: Text mining without document context. Inf. Process. Manage. **42**(6) (2006) 1532–1552
4. Perdisci, R., Giacinto, G., Roli, F.: Alarm clustering for intrusion detection systems in computer networks. Engineering Applications of Artificial Intelligence **19**(4) (2006) 429–438
5. Jaenichen, S., Perner, P.: Acquisition of concept descriptions by conceptual clustering. In Perner, P., Amiya, A., eds.: MLDM 2005. LNAI 3587, Springer-Verlag Berlin Heidelberg (2005) 153–162
6. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Journal of Intelligent Information Systems **17**(2-3) (2001) 107–145
7. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery **2**(3) (1998) 283–304

8. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys **31**(3) (1999) 264–323
9. Maulik, U., Bandyopadhyay, S.: Performance evaluation of some clustering algorithms and validity indices. IEEE Transactions Pattern Analysis Machine Intelligence **24**(12) (2002) 1650–1654
10. Bezdek, J., Pal, N.: Some new indexes of cluster validity. IEEE Transactions on Systems, Man and Cybernetics **28**(3) (1998) 301–315
11. Dunn, J.: Well separated clusters and optimal fuzzy partitions. Journal of Cybernetics **4** (1974) 95–104
12. Davies, D., Bouldin, W.: A cluster separation measure. IEEE PAMI **1** (1979) 224–227
13. Kaufman, L., Rousseeuw, P.: Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons (1990)
14. Kim, M., Ramakrishna, R.: New indices for cluster validity assessment. Pattern Recognition Letters **26**(15) (2005) 2353–2363
15. Bolshakova, N., Azuaje, F.: Cluster validation techniques for genome expression data. Signal Processing **83**(4) (2003) 825–833
16. Gordon, A.: Cluster Validation. In: Data science, classification and related methods (eds. Hayashi, C. and Yajima, K. and Bock H.H. and Ohsumi, N. and Tanaka, Y. and Baba, Y.). Springer (1996) 22–39
17. Ling, R.: On the theory and construction of k-clusters. Computer Journal **15** (1972) 326–332
18. Chou, C., Su, M., Lai, E.: A new cluster validity measure and its application to image compression. Pattern Analysis Applications **7**(2) (2004) 205–220
19. Bouguessa, M., Wang, S., Sun, H.: An objective approach to cluster validation. Pattern Recognition Letters **27**(13) (2006) 1419–1430
20. Merz, C., Murphy, P.: *UCI* machine learning repository (1996) http://www.ics.uci.edu/∼mlearn/MLSummary.html.