A Validity Index for Prototype-Based Clustering of Data Sets With Complex Cluster Structures

Kadim Taşdemir, *Member, IEEE*, and Erzsébet Merényi, *Senior Member, IEEE*

***Abstract*—Evaluation of how well the extracted clusters fit the true partitions of a data set is one of the fundamental challenges in unsupervised clustering because the data structure and the number of clusters are unknown *a priori*. Cluster validity indices are commonly used to select the best partitioning from different clustering results; however, they are often inadequate unless clusters are well separated or have parametrical shapes. Prototype-based clustering (finding of clusters by grouping the prototypes obtained by vector quantization of the data), which is becoming increasingly important for its effectiveness in the analysis of large high-dimensional data sets, adds another dimension to this challenge. For validity assessment of prototype-based clusterings, previously proposed indexes—mostly devised for the evaluation of point-based clusterings—usually perform poorly. The poor performance is made worse when the validity indexes are applied to large data sets with complicated cluster structure. In this paper, we propose a new index, $Conn\_Index$, which can be applied to data sets with a wide variety of clusters of different shapes, sizes, densities, or overlaps. We construct $Conn\_Index$ based on inter- and intra-cluster connectivities of prototypes. Connectivities are defined through a "connectivity matrix", which is a weighted Delaunay graph where the weights indicate the local data distribution. Experiments on synthetic and real data indicate that $Conn\_Index$ outperforms existing validity indices, used in this paper, for the evaluation of prototype-based clustering results.**

***Index Terms*—Cluster validity index, complex data structure, connectivity, Conn_Index, prototype-based clustering.**

I. Introduction

UNSUPERVISED clustering aims to extract the natural partitions in a data set without *a priori* class information. It groups the data samples into subsets so that samples within a subset are more similar to each other than to samples in other subsets. Any given clustering method can produce a different partitioning depending on its parameters and criteria. This leads to one of the main challenges in clustering—to determine, without auxiliary information, how well the obtained clusters fit the natural partitions of the data set. The common approach for this evaluation is to use validity indices. A meaningful validity index is of great importance; however, an index that accurately evaluates clusterings of complicated data sets (data sets with many clusters of varying statistics) has not been developed yet. The objective of this paper is to propose such an index for prototype-based clustering of large data sets.

Existing cluster validity indices, discussed in Section II, work well for data with simple structures or for scenarios where the user is seeking well-behaved superclusters that can be readily derived from a simple and scalable algorithm, such as k-means, instead of extracting detailed structure of complex clusters. Two reasons for seeking satisfactory performance on this level are difficulty to search for more complex structures due to many attributes and noise and the difficulty to interpret those complex structures even if they are extracted. However, many real-world applications are increasingly dependent on finding complex structures even if interpretation may be, at least initially, challenging. Prototype-based clusterings, among them self-organizing maps (SOM) in particular, are successful for finding detailed structure, and are gaining importance for large data sets that are collected to characterize many real-world problems and to enable the discovery of new knowledge. Currently, evaluation of complex clusterings can be done only through expert knowledge and ground truth. This necessitates sophisticated indexes for validity assessment of complex cluster structures, and motivates the exploitation of specific aspects of prototype-based clustering.

We introduce a validity index $Conn\_Index$ that can evaluate prototype-based clusterings of data sets with a wide variety of cluster types. $Conn\_Index$ takes advantage of the knowledge encapsulated in the prototypes of a quantized data set and uses new measures for separation between clusters and scatter within clusters based on data topology on the prototype level. The data topology is represented by the "connectivity matrix" $CONN$ introduced in [1] as a weighted version of the Delaunay graph of the prototypes. The weights (the elements of $CONN$) express the data density local to the prototypes. This will be further explained in Section III.

To evaluate the effectiveness of $Conn\_Index$, we use two synthetic data sets with clusters of different shapes, sizes, dimensionalities, and densities. We also use four real data sets, the Breast Cancer Wisconsin (9-D), Iris (4-D), Wine (13-D) data from the UCI repository [2], and Ocean City, a remote sensing spectral image. We obtain prototypes with SOMs and cluster these prototypes with various methods—k-means and two interactive clusterings. We compare the performance of $Conn\_Index$ to the performances of commonly used indices by evaluation of which clustering results are favored as the best by each of the indices used in this paper. The outline of the paper is as follows: Section II gives a background information on cluster validity indices and common approaches for index

Manuscript received February 8, 2010; revised June 17, 2010 and September 29, 2010; accepted December 5, 2010. This work was partially supported by Grant NNG05GA94G from the Applied Information Research Program of NASA, Science Mission Directorate. This paper was recommended by Associate Editor J. Basak.

K. Taşdemir was with Rice University, Houston, TX 77005 USA. He is currently with the European Commission Joint Research Centre, Institute for Environment and Sustainability, 21027 Ispra, Italy (e-mail: kadim.tasdemir@jrc.ec.europa.eu).

E. Merényi is with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005 USA (e-mail: erzsebet@rice.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2010.2104319

construction, Section III briefly reviews the prototype-based clustering, describes the "connectivity matrix", and introduces $Conn\_Index$. Sections IV and V give examples for the performance of $Conn\_Index$ on synthetic data sets and on the real data sets, respectively. In addition, Section V shows that $Conn\_Index$ can also provide a meaningful measure when different prototypes may be left unclustered in different clusterings. Section VI concludes the paper. An Appendix provides estimates on computational complexities of the indexes compared.

II. Background on Cluster Validity Indices

A cluster validity index can be constructed by using one of the following three criteria: 1) external crtieria; internal criteria; and 3) relative criteria [3]. External criteria are used to compare clustering results to a pre-specified structure. Internal criteria are for comparison to a proximity matrix of the data samples. The common approach is to use relative criteria, which is to compare the validity of several clustering results based on a combined measure of between-cluster separation and within-cluster scatter. There are many different methods to determine the validity of crisp clustering (where each data sample belongs to only one cluster) [4]–[11] or that of fuzzy clustering (where each data sample has a degree of membership in several clusters) [12]–[16]. Some validity indices are specific to the clustering method. For example, the indices in [17], [18] are proposed for support vector clustering whereas the indices proposed in [16] are for generalized fuzzy c-means clustering. In this paper, we focus on crisp clustering algorithms and we refer to Kim *et al.*[14] for a detailed analysis of the cluster validity indices for fuzzy clustering, where an index (based on the data distribution at overlapping regions) is also proposed.

For crisp clustering, the Davies–Bouldin index (DBI) [4] and the generalized Dunn Index (GDI) [5] are two commonly used indices. Two other indices are the Silhouette width criterion [19] (selected best in a recent study [20]), and the Calinski–Harabasz variance ratio criterion (CH-VRC) [21] (selected best among 30 indices in [9]). A recent index shown to be useful is PBM [10]. All these indices provide meaningful measures for well-separated or parametrical clusters but they may fail for complicated data structures with clusters of different shapes or sizes or with overlaps. This is because available distance measures for separation between clusters and scatter within clusters may be ineffective for complicated data sets due to the fact that the cluster boundaries are usually defined not only by the distances between the data samples but also by how the samples are distributed within the clusters. Several indices proposed in recent years integrate the data distribution and the distance metrics [6], [14], [22]. One of these, CDbw (composite density between and within clusters) [6] is promising for clusters of different shapes and with homogeneous density distribution. Brief explanations of these indices are given below along with the discussion on their constructions.

A. Construction of Cluster Validity Indices

The separation and scatter measures, used in the index construction, are often computed from various distances, some of which are illustrated in Fig. 1. A general approach is to use centroid-based distance metrics (d_{b_cent} and d_{w_cent}) for separation and scatter [4], [9], [10], [12], [13], [15], which favor (hyper)spherical or (hyper)ellipsoidal clusters. The most reliable results for validity indices are obtained when all data samples in the clusters are considered in the computation of the distances for index construction [5]. In the following, N will denote the number of data vectors in a data set, K will denote the number of clusters in the clustering, and, where applicable, P will denote the number of prototypes that result from a vector quantization (SOM or other) of a data set.

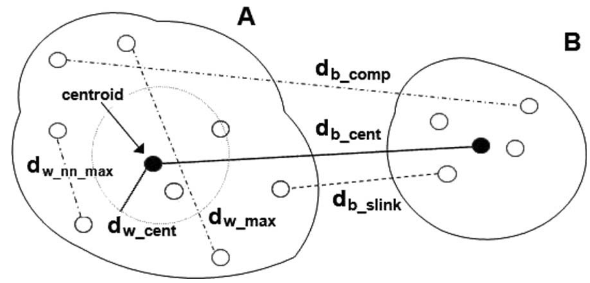


Fig. 1. Several metrics for within-cluster ($d_{w\_cent}, d_{w\_max}, d_{w\_nn\_max}$) and between-cluster ($d_{b\_cent}, d_{b\_comp}, d_{b\_slink}$) distances. d_{w_cent} is the average distance to the cluster centroid, d_{w_max} is the maximum distance between the points within the cluster, $d_{w\_nn\_max}$ is the maximum of the nearest neighbor distances. d_{b_cent} is the distance between the cluster centroids, $d_{b\_comp}(d_{b\_slink})$ is the maximum (minimum) distance between the points across the clusters. Among them, d_{b_cent} and d_{w_cent} are the commonly used metrics.

In addition to the choice of distance metrics for separation and scatter measures, how the index is constructed from these measures is also important. One way to construct the index is to calculate the ratio between the total or maximum within-cluster scatter and minimum separation between clusters such as in the Dunn index [7], or in the GDI [5]. For example, the GDI is calculated as follows:

$$GDI = \min_m \left\{ \min_n \left\{ \frac{d_{b\_i}(C_m, C_n)}{\max_k \{d_{w\_j}(C_k)\}} \right\} \right\} \tag{1}$$

where C_m, C_n, and C_k are clusters; d_{b_i} is a between-cluster separation measure and d_{w_j} is a within-cluster scatter measure with i, j indicating choices of distances. The choices for d_{b_i} and d_{w_j} can be metrics from Fig. 1 or any other that the user selects. The index constructed this way heavily depends on the cluster with the maximum scatter and on the pair of clusters with the minimum separation. If there is a large cluster or there are two small clusters which are very close to each other, the index will be dominated by their scatter or separation and will be insensitive to the separation or scatter of other clusters, thus producing an incorrect measure.

Another way to construct the index is to consider the scatter and separation measures of all clusters. A good example is the DBI, which is computed by averaging the ratio of the within-cluster scatter to the between-cluster separation over all clusters. This type of construction is useful when the separation and the scatter measures together provide a meaningful geometric interpretation of the cluster structure. The DBI is calculated with the distances between cluster centroids (d_{b_cent}) and average distances of data samples to their cluster centroid (d_{w_cent})

(from Fig. 1) as follows:

$$DBI = \frac{1}{K}\sum_{k=1}^{K}\max_{m}\left(\frac{d_{w\_cent}(C_k)+d_{w\_cent}(C_m)}{d_{b\_cent}(C_k,C_m)}\right). \quad (2)$$

With this construction, the DBI provides correct interpretation for data sets with hyperspherical clusters or with hyperellipsoidal clusters if Mahalanobis distance is chosen instead of Euclidean. A similar approach has been used in the Silhouette width criterion [19] where the average distance of a data sample i to the samples within its own cluster (d_{avg_i}) is considered along with the minimum distance of i to samples in other clusters (d_{b_i}). The criterion is obtained by averaging over all N samples as follows:

$$Silhouette = \frac{1}{N}\sum_{i=1}^{N}\frac{d_{b\_i}-d_{avg\_i}}{\max(d_{b\_i},d_{avg\_i})}. \quad (3)$$

Another example for this type of index construction is the variance ratio criterion of Calinski and Harabasz [21] (CH-VRC). This criterion is constructed as

$$CHVRC = \frac{BGSS/(K-1)}{WGSS/(N-K)} \quad (4)$$

where $BGSS$ is between-group sum of squares [sum of squared distances of cluster centroids to the geometric center (or centroid) of all data samples], $WGSS$ is within-group sum of squares (sum of squared distances between each data sample and its respective cluster centroid). A recent index PBM [10] also uses a similar approach and is constructed by using three components

$$PBM = \left(\frac{1}{K}\frac{E_1}{E_K}D_K\right)^2. \quad (5)$$

E_1 is the average distance to the geometric center of all samples; E_K is the sum of within-cluster distances (distances of data samples to their respective cluster centroid); and D_K is the maximum distance between the centers of the K clusters.

Instead of using cluster centroids, the CDbw index [6] defines the separation and the scatter based on distances between multiple cluster prototypes and data distribution around them, as follows:

$$CDbw = Intra\_dens \times Sep \quad (6)$$

where $Intra\_dens$, the scatter, is the density within one standard deviation around the prototypes, averaged over all clusters; and Sep, the separation, is the sum of the distances (d_{b_slink}) between all pairs of clusters divided by the sum of densities at the cluster boundaries (number of data samples around the midpoints of the prototypes that form single linkage between clusters). CDbw correctly evaluates clusterings where clusters have homogeneous distribution. However, CDbw fails to represent true inter- and intra-cluster densities when the clusters have inhomogeneous density distribution which is often the case for real data.

Considering the scatter and the separation of all samples or clusters (as in the case of Silhouette, CH-VRC, DBI and CDbw) can provide more reliable results than using the scatter and the separation of selected clusters, because the delineation of cluster boundaries is more dependent on the relationship between neighbor clusters than on the relationship between, for example, the closest pair of clusters. Therefore, the index we propose below utilizes the scatter and separation of all clusters, with new definitions of the scatter and separation based on the local data distribution.

III. $Conn\_Index$: A Validity Index Based on Prototype Level Data Topology

The proposed $Conn\_Index$ is tailored to exploit the information produced by prototype-based clustering methods, which makes $Conn\_Index$ suitable only for those methods. Therefore, we first explain prototype-based clustering, discuss how the data topology on the prototype level can help validity assessment, and then define the new index.

A. *Prototype-Based Clustering for Large Data Sets*

Prototype-based clustering aims to find a number of representative data vectors or prototypes in the data space which faithfully represent the large number of data samples. This is usually done through an iterative minimization of a cost function based on the deviation of the data samples from their closest prototypes, i.e., their best matching units (BMUs). For clustering of large data sets with complex cluster structures, prototype-based clustering is often preferred. Compared to clustering data samples, prototype-based clustering has the advantage that it is easier to deal with a smaller number of prototypes than with a large number of data samples (for reasons of lower computational complexity and less memory demand), and it is robust to noise and outliers. The use of single prototypes to represent a cluster, such as in k-means and fuzzy c-means, is often inadequate to describe complex cluster structures with arbitrary shapes and sizes. Therefore, multiple prototypes per cluster are employed in recent studies based on SOMs [23], [24], neural gas [25], and CURE [26]. In these methods, the number of prototypes is often much larger than the number of expected clusters, yet still much smaller than the number of the data samples. After obtaining the prototypes, they are grouped according to their similarities and data clusters are extracted by assigning each data point to the cluster of its prototype. In particular, SOMs have been successful for extraction of detailed structure [1], [27] because SOMs distribute prototypes in the data space through a topology-preserving mapping in an iterative learning process, which results in as faithful representation of the data distribution as possible with the given number of prototypes. The SOM neural units are, at the same time, indexed in a (usually 2-D) rigid lattice according to their similarity relations; therefore, similar prototypes map close to one another in the lattice and vice versa, and prototypes (weight vectors) of neural units that are neighbors in the SOM lattice represent similar data vectors. Therefore, the visualization and examination of the prototype relationships in the SOM lattice facilitates the extraction of clusters.

We briefly summarize here the SOM learning rule for completeness, details can be found in many text books. Let M be a data set, and S be the fixed SOM lattice with P neural units.

For a given data sample $v \in M$, the BMU w_i is found by

$$\|v - w_i\| \leq \|v - w_j\| \qquad \forall j \in \mathcal{S} \tag{7}$$

and then the BMU w_i and its lattice neighbors (determined by a (often Gaussian) neighborhood function $h_{i,j}(t)$, centered around the BMU w_i) are updated according to

$$w_j(t+1) = w_j(t) + \alpha(t)h_{i,j}(t)\left(v - w_j(t)\right) \tag{8}$$

where $\alpha(t)$ is a learning parameter. Both $\alpha(t)$ and $h_{i,j}(t)$ should decrease with time t. The weight vectors of the neural units become the vector quantization prototypes of the data set, ordered on a rigid lattice.

The data space can be partitioned with respect to the prototypes (obtained by any vector quantization method, SOM included), resulting in a Voronoi tessellation where each prototype is the geometric center or centroid of its Voronoi polyhedron. The Voronoi polyhedron contains the data samples which are closest to its centroid, thus it corresponds to the receptive field (RF) of the respective prototype. A Voronoi polyhedron containing no data samples indicates a discontinuity in the data space (possible separation between clusters).

B. Topology Representation of Quantized Data by Connectivity Matrix ($CONN$)

Each quantization prototype is the BMU for the samples in its receptive field (RF, Voronoi polyhedron). In general, topology can be expressed by the Delaunay graph (the dual of the Voronoi tessellation) which is obtained by connecting the centers of the neighboring Voronoi polyhedra (polyhedra that share an edge). In order to better characterize and summarize the data topology on the prototype level, we introduced the cumulative adjacency matrix, $CADJ$, and the connectivity matrix, $CONN$, in [1]. $CADJ$ and $CONN$ describe, as we formally explain below, the topology of the quantization prototypes but not only their adjacency relations but also their "attractions" to one another, as defined by the local densities of the manifold. They are obtained by assigning weights to edges of the induced Delaunay graph (which is the intersection of the Delaunay graph with the data manifold) that provides the binary adjacency relations of the prototypes. As proposed by Martinetz and Schulten [25], when prototypes are dense enough in the data set, the induced Delaunay graph can be produced by connecting two prototypes p_i and p_j if at least one data sample selects them as a BMU and second BMU pair, i.e., if they are the two closest prototypes to a data sample. (When a data sample is equidistant from multiple prototypes, which is a very rare case, it is assigned to the one with the lowest index i among them.) Analogously, a weighted induced Delaunay graph can be produced by assigning the number of data samples for which p_i and p_j are the BMU and the second BMU pair, as the weight to the edge in the Delaunay graph that connects p_i and p_j. These weights are the elements of the $CONN$ matrix. The weight of the edge between p_i and p_j is $CONN(i,j)$. Obviously, $CONN$ is a symmetric matrix. The cumulative adjacency $CADJ$ is nonsymmetric. $CADJ(i,j)$ is the number of data samples for which p_i is the BMU and p_j is the second BMU. $CADJ(i,j)$ therefore describes the density distribution within the receptive field RF_i of p_i with respect to its neighbors indexed by j. $CONN(i,j)$, which is the sum of $CADJ(i,j)$ and $CADJ(j,i)$, is a similarity measure for prototypes based on local densities. Both CADJ and CONN are $P \times P$ matrices indicating similarities between P prototypes.

Fig. 2 shows a visualized example of the $CONN$ matrix for a 2-D data set called "Clown", created by Vesanto and Alhoniemi [28] by using different parametric models for each cluster and adding noise. This data set has clusters of various shapes and sizes: spherical (right eye), elliptical (nose), U-shaped (mouth), three subclusters in the left eye, a sparse body, and outliers. The prototypes were obtained by a 19 × 17 SOM, also by [28]. $CONN$ makes high-density regions and no-data regions (disconnected parts of the data set) visible. As explained in Fig. 2(b), when $CONN$ is visualized by indicating the connection weights with proportional line width for edges in the Delaunay graph, separations between clusters may become apparent. This outlines the boundaries of some clusters even though the distances between the prototypes at the cluster boundaries may be smaller than the distances between the prototypes within clusters. The illustration in Fig. 2(b) further suggests that $CONN$ can help determine the validity of clustering for prototype based clustering algorithms. We show this in the next sections.

C. Definition of $Conn\_Index$

We define $Conn\_Index$ with the help of two quantities: the intra-cluster connectivity ($Intra\_Conn$) as the within-cluster scatter and the complement of the inter-cluster connectivity ($1 - Inter\_Conn$) as the between-cluster separation measure. First, we introduce these quantities and then we define our new index. Assume K clusters and P prototypes p_i ($i = 1, 2, \ldots, P$) in a data set ($N > P > K$), and let C_k and C_l refer to two different clusters ($1 \leq k, l \leq K$).

Definition 1: The intra-cluster connectivity $Intra\_Conn$ is the average of intra-cluster connectivities $Intra\_Conn(C_k)$ over all clusters

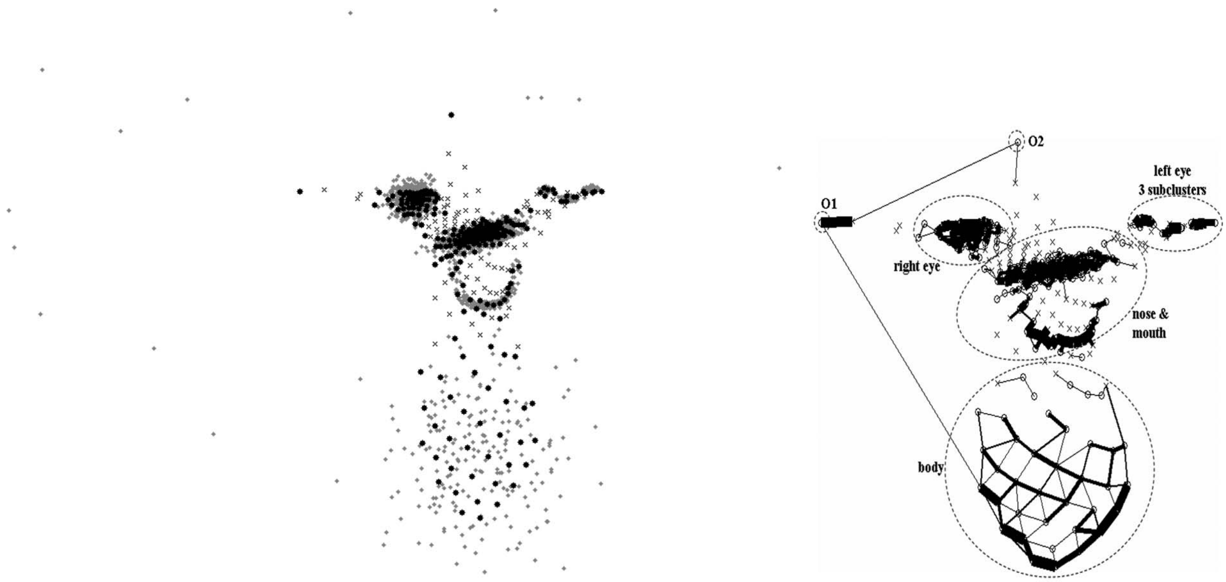
$$Intra\_Conn = \sum_{k}^{K} Intra\_Conn(C_k)/K \tag{9}$$

where $Intra\_Conn(C_k)$ is the ratio of the number of those data samples in C_k which have both their BMU and second BMU in C_k to the total number of data samples in C_k

$$Intra\_Conn(C_k) = \frac{\sum_{i,j}^{P}\left\{CADJ(i,j) : p_i, p_j \in C_k\right\}}{\sum_{i,j}^{P}\left\{CADJ(i,j) : p_i \in C_k\right\}}. \tag{10}$$

The denominator of (10) can be replaced by the sum of receptive field sizes of prototypes $p_i \in C_k$ because, obviously, the receptive field size of p_i is $RF_i = \sum_j^P\{CADJ(i,j)\}$. $Intra\_Conn$ is computed from all data samples in C_k. By definition, $Intra\_Conn(C_k) \in [0,1]$ where a greater value means more connectivity within the cluster, i.e., C_k is more self-contained. If the second BMUs of all data samples in C_k are also in C_k (there is no connection to any other cluster) $Intra\_Conn(C_k) = 1$.

To define the inter-cluster connectivity $Inter\_Conn(C_k, C_l)$ between clusters C_k and C_l, we



(a) (b)

Fig. 2. (a) 2-D data set "Clown" (a mixture of several parametrical distributions) and the SOM prototypes created by [28]. Small gray diamonds indicate data samples. Notice that there are several outliers at the far upper left which are somewhat hard to see. The black dots are prototypes with non-empty receptive fields, while $\times$ are prototypes with empty receptive fields. The data set has different types of clusters, such as spherical (right eye), elliptical (nose), U-shaped (mouth), sparse (body), three small elliptical subclusters (left eye). Variances within clusters and inter-cluster distances are different but the clusters are well separated except for the mouth and nose. (b) Topology representation by connectivity matrix $CONN$. An edge between two prototypes indicates adjacency of their Voronoi cells. The width of a line is proportional to the number of data samples for which the prototypes connected by this line are a BMU and the second BMU pair. The separations between clusters are indicated by unconnected prototypes.

consider the prototypes at the cluster boundaries since those prototypes are the ones which often facilitate the separation between clusters. A prototype at a cluster boundary is the one which may have connections to clusters other than its own.

Definition 2: The inter-cluster connectivity of clusters C_k and C_l $Inter\_Conn(C_k, C_l)$ is the ratio of the sum of the connectivity strengths between C_k and $C_l Conn(C_k, C_l)$ to the sum of the connectivity strengths of those prototypes in C_k which have at least one connection to a prototype in C_l

$$
\begin{aligned}
&Inter\_Conn(C_k, C_l)\\
&\quad = \begin{cases} 0, & \text{if } P_{k,l} = \emptyset \\ \frac{Conn(C_k,C_l)}{\sum_{i,j}^{P}\{CONN(i,j):p_i \in P_{k,l}\}}, & \text{if } P_{k,l} \neq \emptyset \end{cases}\\
&with\; Conn(C_k, C_l)\\
&\quad = \sum_{i,j}^{P} \{CONN(i,j) : p_i \in C_k, p_j \in C_l\}\\
&and\; P_{k,l}\\
&\quad = \{p_i : p_i \in C_k, \exists\, p_j \in C_l : CADJ(i,j) > 0\}\,. \qquad (11)
\end{aligned}
$$

$Inter\_Conn(C_k, C_l)$ shows how similar the prototypes at the boundary of C_k are to the ones at the boundary of C_l in comparison to the similarity of the prototypes within C_k. If C_k and C_l are completely separated in the sense that there are no cross-connections $Inter\_Conn(C_k, C_l) = 0$. A greater $Inter\_Conn(C_k, C_l)$ is an indication of a greater degree of similarity between C_k and C_l. $Inter\_Conn(C_k, C_l) > 0.5$ indicates that those prototypes in C_k which have connections to C_l are more similar to the prototypes in C_l than to the prototypes in C_k. This means they should either be in C_l or C_k and C_l should be combined. The cluster most similar to C_k is the one for which $Inter\_Conn(C_k, C_l)$ is maximum $(l \neq k, 1 \leq l \leq K)$.

Definition 3: The inter-cluster connectivity (average similarity) $Inter\_Conn$ is the average of the inter-cluster connectivities of all clusters $Inter\_Conn(C_k)$

$$Inter\_Conn = \sum_{k}^{K} Inter\_Conn(C_k)/K \qquad (12)$$

where

$$Inter\_Conn(C_k) = \max_{l,l \leq K} Inter\_Conn(C_k, C_l). \qquad (13)$$

Similarly to $Intra\_Conn$, $Inter\_Conn \in [0,1]$ by definition. Since $Inter\_Conn$ is average similarity, $1 - Inter\_Conn$ becomes a dissimilarity (separation) measure. We define our new validity index, the $Conn\_Index$, as

$$Conn\_Index = Intra\_Conn \times (1 - Inter\_Conn). \qquad (14)$$

$Conn\_Index \in [0,1]$ increases with better clustering and has a maximum of one when the clusters are separated. Details of the calculation of $Conn\_Index$ and its components $Intra\_Conn$ and $Inter\_Conn$ were shown through an example in [29].

$Intra\_Conn$ heavily depends on the sizes of the clusters. When clusters have many data samples, the total strength of within-cluster connections will be relatively strong compared to the total strength of between-cluster connections, resulting in a high $Intra\_Conn$ value. As a result, $Intra\_Conn$ will decrease with increasing number of clusters unless the clusters are split along natural cluster boundaries. Contrarily, $Inter\_Conn$

depends only on the connections of prototypes at the cluster boundaries, hence it is independent of the sizes of clusters.

IV. Performance of $Conn\_Index$ on Synthetic Data

When comparing indices, we want to see whether they favor the true clusters as the best partitioning. True (or natural) clusters are those which satisfy the criterion "points in a cluster are closer to a point in the same cluster than to any point in other clusters". Accordingly, "true labels" describe known true clusters in this discussion. We compare the indices computed for the clusterings obtained by different clustering methods to the indices computed for the known true labeling (true clusters). Since different indices have different ranges, some are bounded, some are not, and their nonlinearities are also different, it is not quite straightforward to compare their performance. For example, a better cluster quality is indicated by a smaller DBI while it is indicated by a greater value for other indices in this study. Theoretically, DBI, GDI, CH-VRC, PBM, and CDbw may have values in $[0, \infty)$ while Silhouette is in $[-1, 1]$ and $Conn\_Index \in [0, 1]$. However, DBI and GDI usually have a small range of values (in our experience with different data sets and different distance metrics, their maximum value did not exceed 10), whereas PBM and CDbw span a much larger range of values depending on the number of data samples and their distribution within clusters (for example, CDbw can be more than 100). Therefore, one meaningful approach is to compare the values of the same index obtained for different partitionings of the same data and determine the validity rank of clusterings according to this index and then to compare the validity ranks across different indices.

For performance evaluation, we compare $Conn\_Index$ to the indices mentioned above. We use GDI with centroid linkage (d_{b_cent} in Fig. 1) and average distance of points to cluster centroids (d_{w_cent}) as the inter- and intra-cluster distance metrics, respectively. We also considered other distance metrics (shown in Fig. 1) for GDI but did not include here due to the fact that the GDI with those metrics either performed the same or poorer than the GDI with d_{w_cent} and d_{b_cent} for the data sets in this paper. We also computed the non-prototype-based indices (DBI, GDI, CH-VRC, PBM, and Silhoutte) based on individual data points as well as based on prototypes, in order to observe whether they provide different rankings of clusterings. Due to the fact that the ranking by the various indices came out often the same by both ways of computing the indices, we provide the index values based on prototypes in this paper.

Some specific index values convey important properties. For example, $Conn\_Index = 1$ means that the clusters are completely separated whereas any other $Conn\_Index$ value indicates an overlapping case. As $Conn\_Index$ goes to zero, the degree of overlap increases. For DBI, an index value greater than one means either there are overlapping clusters or the natural partitions are not hyperspherical. However, if DBI is less than one, it does not necessarily indicate well-separated clusters. A positive value (close to one) for Silhoutte width criterion may indicate non-overlapping clusters whereas a negative value surely indicates overlapping clusters. Due to the fact that GDI considers the maximum scatter and minimum separation but not the relative dissimilarity for each cluster, a well-separated case can be represented by any GDI value.

We analyze the performance of Conn_Index on the clusterings of two synthetic data sets: the 2-D Clown data [28] with nine clusters of varying statistics, and a 6-D data set with 11 known classes [30]. These data sets—although far from the complexities real data can produce—represent some of the characteristics that make data complicated. We also show the performance of Conn_Index for real data sets: three simple data sets (Breast cancer Wisconsin, Iris, Wine) from the UCI machine learning repository [2], and an 8-D remote sensing spectral image [30]. In addition, we compare Conn_Index to DBI, GDI, CDbw, silhouette, CH-VRC, and PBM indices. Since Conn_Index does not depend on the dimensionality of the data sets, we do not include data sets with hundreds of features. In our experiments, we select the number of prototypes (P) to be larger than the number of expected clusters (K) in the data sets but much smaller than the large number of data samples (N).

A. 2-D Clown Data

The Clown data set, shown in Fig. 2 and described in Section III-B, has 2220 data samples in nine clusters which are presented in Fig. 3(a). These nine clusters can be naturally grouped into two superclusters: the face and the body.

For performance comparison of the indices, we show a hierarchical clustering produced by [28] in Fig. 3(b). This clustering extracts eight clusters with a few incorrectly labeled prototypes as shown. In Fig. 3(c), we combined two subclusters ($\triangleright$ and $\times$) in the left eye in Fig. 3(b) to measure the effect of small changes in the clustering on the validity indices. Fig. 3(d)–(f) provide the results of the k-means clustering for $\mathrm{k} = 2, 4, 5$. The k-means clustering is only successful for $\mathrm{k} = 2$ where the two clusters are the face and body which have nearly spherical structures. As k becomes larger, the partitioning is less similar to the natural partitions [Fig. 3(e)–(f)].

Table I and Fig. 4 give the indices for the different partitionings of the Clown data in Fig. 3. When we compare the indices for the clusterings in Fig. 3(b) and (c), there is a large increase in GDI in favor of the clustering in Fig. 3(c) over the true labels. This is because GDI depends on the minimum separation (which has increased by merging the two subclusters) rather than on the relative comparison of separations as in DBI, CDbw, and $Conn\_Index$. As we stated in Section II, other indices in Table I are less sensitive to this change because of their averaging property.

$Conn\_Index$ values are similar for k-means clustering with k2 and to those for the true labels. It slightly favors k-means clustering with k2 due to the supercluster structure (face and body) in the data set. This is because face and body are two large clusters connected with a thin connection, whereas known clusters (nose and mouth) are more strongly connected [Fig. 2(b)]. The index value drops slowly up to k4 and significantly for larger k due to more incorrectly labeled prototypes. GDI, Silhouette, and CH-VRC also favor k-means clustering with k2 while DBI and PBM choose k-means clustering with k4 where there are four superclusters with several incorrectly labeled prototypes. Surprisingly, CDbw favors k-means clustering with k5 where the partitioning is quite different from the true labels. One reason can be the incorrect density estimation due to varying statistics of clusters. In summary, as shown in

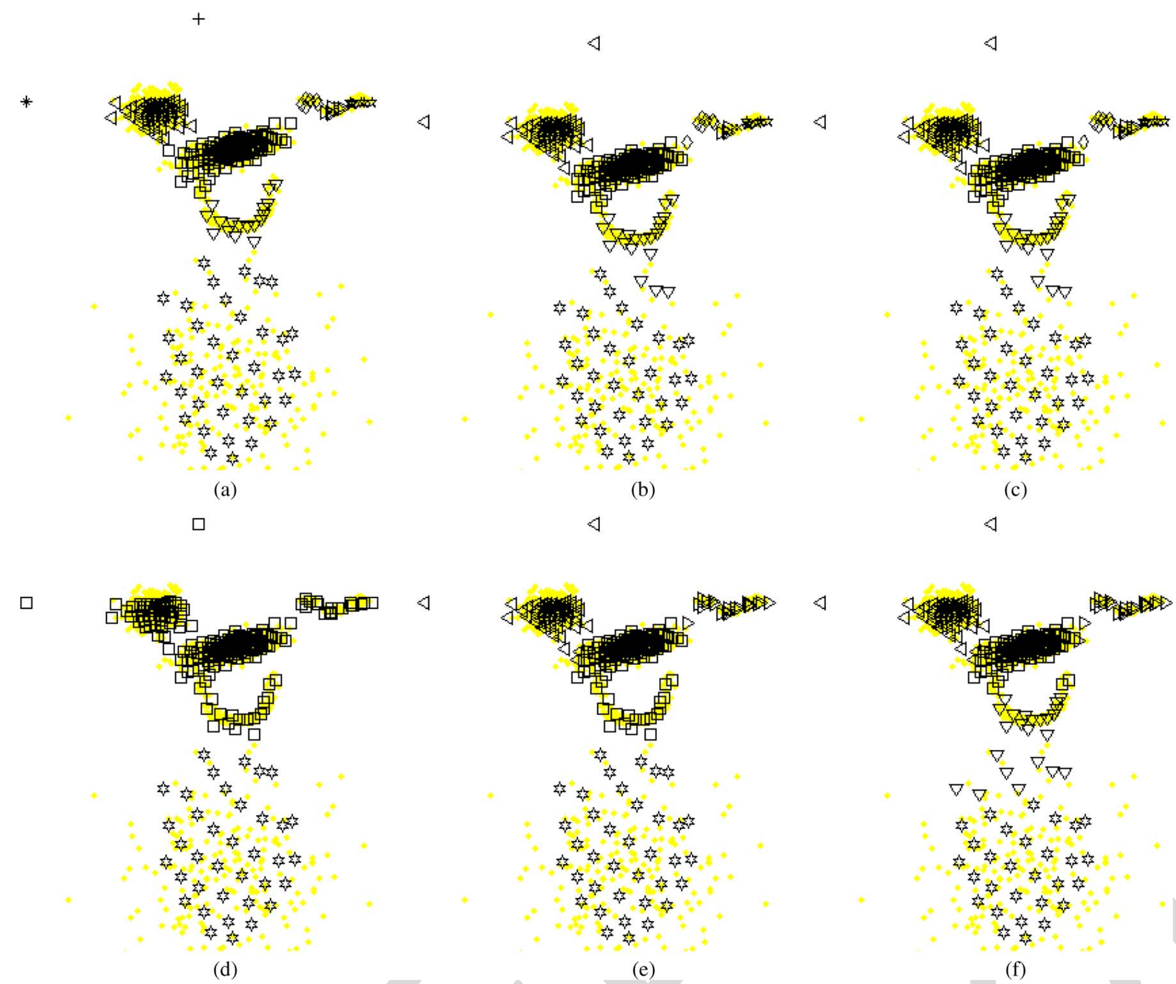


Fig. 3. Clusterings of the Clown data set by clustering of SOM prototypes. The data points are shown with dots and the prototypes are labeled by symbols. Top: (a) Known labels. Seven clusters constitute the Clown: one cluster for the body (David stars), and six clusters for the face: nose (□), mouth (▽), right eye (◁), and three clusters in the left eye (◇, ▷, open star); the remaining two, $+$ and $*$, are singletons, outliers due to noise. (b) Clustering by a hierarchical algorithm by Vesanto and Alhoniemi [28]. The two singletons are merged to the closest cluster. The true cluster in the middle of the left eye is extracted as two subclusters ▷ and $\times$. There are eight clusters with a few incorrect labels. (c) A clustering similar to (b) except the two subclusters ▷ and $\times$ in the middle of left eye are merged and labeled as ▷, in order to analyze how the indices respond to this change. Bottom: k-means clustering with (d) k2, (e) k4, (f) k5. The index values of these clusterings are shown in Table 1.

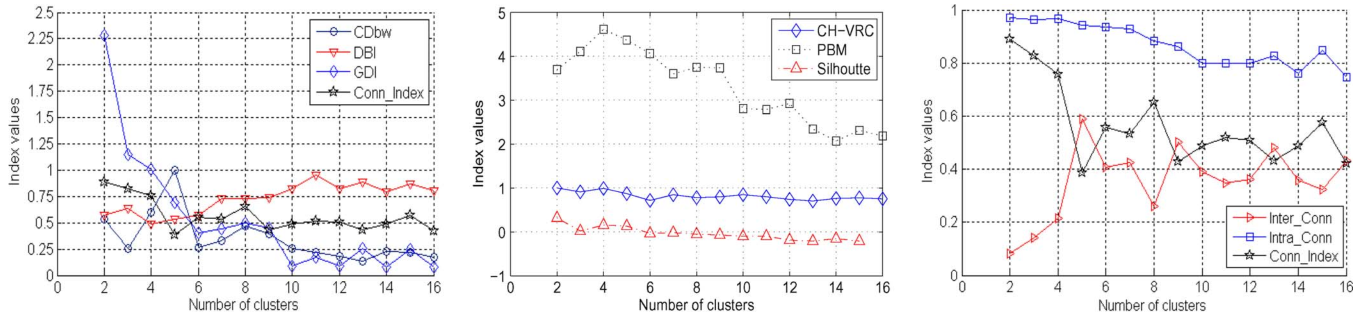
TABLE I
VALIDITY INDICES FOR THE CLUSTERINGS OF THE CLOWN DATA. INDICES FOR THE FAVORED PARTITIONINGS ARE IN BOLD FACE

Cluster validity Index	Clustering method						
	Fig. 3.a	Fig. 3.b	Fig. 3.c	k-means clustering			
	k=9	k=8	k=7	k=2	k=3	k=4	k=5
DBI	0.58	0.61	0.58	0.58	0.64	**0.49**	0.54
GDI	0.15	0.07	0.31	**2.29**	1.15	1.01	0.69
CDbw	0.39	0.49	0.56	4.92	2.32	5.48	**9.18**
Conn_Index	**0.88**	0.74	0.83	**0.89**	0.83	0.76	0.39
Silhoutte	0.22	0.19	0.18	**0.32**	0.02	0.15	0.13
CH-VRC	174	153	184	**236**	215	234	206
PBM	1.34	1.95	2.52	3.70	4.11	**4.62**	4.37

Table I, DBI, GDI, CH-VRC, PBM, and CDbw favor incorrect partitionings of k-means [for example k5, in Fig. 3(f)] over the true labels due to inaccurate density estimation of CDbw and the centroid-based approach of the rest, while Silhouette and $Conn\_Index$ favor the true labels and the supercluster structure determined by the face and the body. We point out, however, that the relative difference of $Conn\_Index$ values for the true labels (0.89) and for the superclusters (0.88) are much closer than the respective Silhoutte index values, i.e., that Silhoutte ranks the true labels lower (on its scale) than $Conn\_Index$.

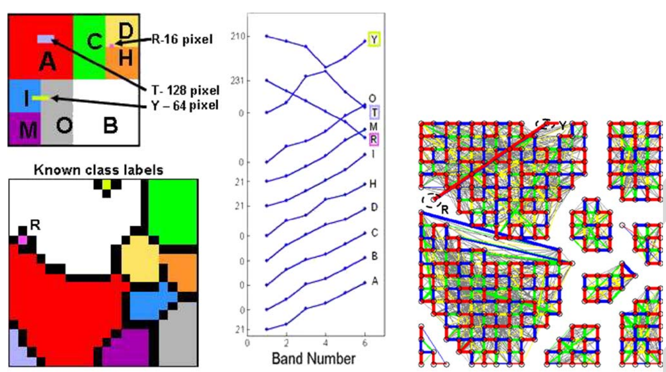
B. 11-Class Data Set

This data set is from a family of 6-D synthetic data cubes used in [30] and described in detail at http://terra.ece.rice.edu. It has 128 $\times$ 128 6-D data samples in a square "image" grouped into 11 classes, three of which are relatively small. Each data sample is a 6-D feature vector (signature) specifying its characteristics. The mean signatures of eight classes are quite similar to each other and the small classes have different signatures (Fig. 5). Because the dimensionality of this data



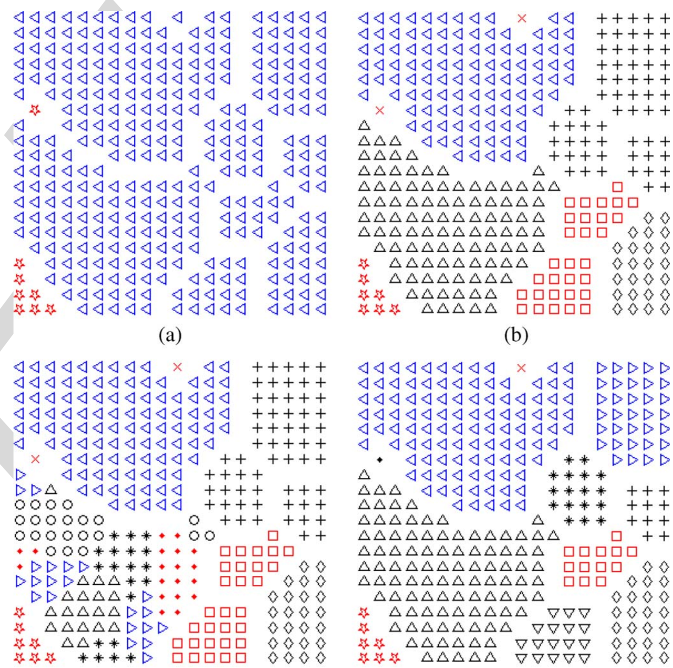
(a) (b) (c)

Fig. 4. Validity indices for k-means clusterings of the Clown data. (a) Comparison of DBI, GDI, CDbw, and $Conn\_Index$. CDbw is normalized by its maximum value 9.18. (b) Comparison with CH-VRC, Silhoutte, and PBM (CH-VRC is normalized to one by its maximum value, 236). (c) $Conn\_Index$ and its subcomponents, $Intra\_Conn$ and $Inter\_Conn$. $Intra\_Conn$ monotonically decreases with increasing k (except for k = 13,15) since greater k does not produce a better partitioning but reduces the size of the extracted clusters. $Inter\_Conn$ is maximum for k = 5 where some strongly connected prototypes are incorrectly labeled [Fig. 3(f)].



(a) (b)

Fig. 5. (a) 6-D synthetic data set with 11 classes, three of which are relatively small. The top left image shows the spatial distribution of the data classes in the 128 × 128 pixel image. The signatures of the 11 classes are shown on the right, offset for clarity. The signatures of the small classes are very different from the rest. The bottom left image represents the known labels of the SOM prototypes. (b) The $CONN$ visualization on the SOM. The classes are well separated except for two small ones, Y and R, each of which are represented by one prototype.

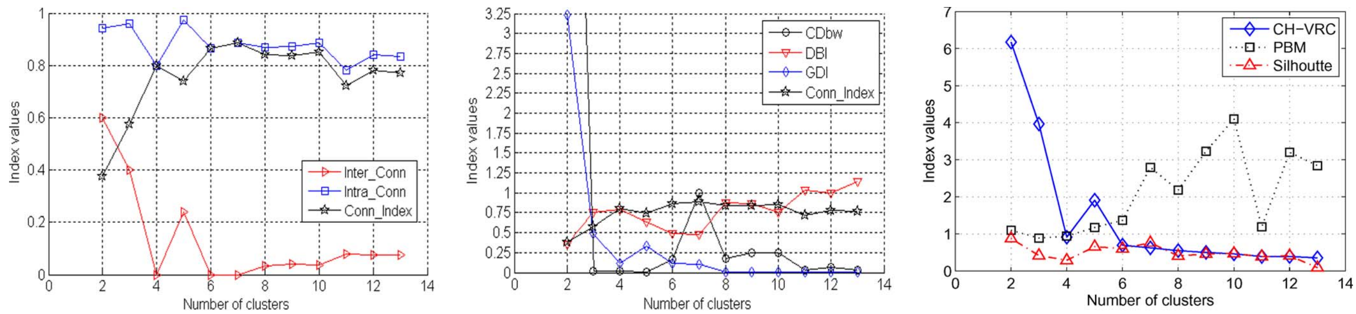


(a) (b) (c) (d)

Fig. 6. k-means clustering of the (20 × 20) SOM prototypes of the 11-class data set and the true labels. (a) k2 (favored by DBI, GDI, and CDbw) (b) k7 (for which the $Conn\_Index$ is maximum). (c) k11 (true number of clusters) (d) true labels of the 11 classes.

set is greater than three, we cannot visualize it in the data space. Therefore, we show the classes (Fig. 5) through $CONN$ visualization (CONNvis) of the prototypes on the SOM lattice. CONNvis is a recent SOM visualization scheme that represents data topology [1] and has the advantage of visualizing higher dimensional data spaces on the SOM lattice regardless of the data dimensionality. CONNvis is obtained by connecting prototypes p_i, p_j whose Voronoi cells are adjacent, with lines of various widths and colors. The width of the connection is proportional to $CONN(i, j)$ whereas the color indicates the ranking of the connections to i.

Fig. 5 shows that the classes are well separated (no connections between the classes) except for two small ones, R and Y. We cluster the 20 × 20 SOM prototypes with k-means. The cluster labels for $k2, 7, 11$ and the true labels are given in Fig. 6. All k values up to seven produce superclusters of the existing 11 classes. Fig. 7 shows the index values for these k-means clusterings with different k values. All indices except $Conn\_Index$ and PBM favor k2 [Fig. 6(a)] as the best k-means partitioning even though the two connected small classes R and Y are grouped into different superclusters. This is because, owing to their small sizes, clusters R and Y have very little effect on those indices. In contrast, $Conn\_Index$ indicates the similarity at the cluster boundaries of these two extracted clusters in Fig. 6(a) by producing a large $Inter\_Conn$ value since the prototype representing cluster R is more similar to the prototype of Y than to any other prototype within its own group [open stars in Fig. 6(a)]. The best k-means clustering according to $Conn\_Index$ is the one with k7 [Fig. 6(b)] which is the second best according to DBI and CDbw. For k7, the two small classes R and Y are grouped into one cluster [× in Fig. 6(b)] and disconnected from the other six clusters. $Inter\_Conn$, shown in Fig. 7(a), indicates that for k4, k6 and k7, there are no cross-connections between the extracted clusters (the clusters are well separated superclusters of the 11 true classes). However, since in those cases, nonspherical clusters are likely formed, other indices may not indicate the clear



(a) (b) (c)

Fig. 7. Validity indices for k-means clustering of the 11-class data set. (a) $Conn\_Index$ and its subcomponents, $Intra\_Conn$ and $Inter\_Conn$. $Inter\_Conn = 0$ at k$4, 6, 6$ indicates that the extracted clusters are well-separated. (b) Comparison with DBI, GDI, CDbw, and $Conn\_Index$ for k-means clusterings. (c) Comparison with Silhoutte, CH-VRC, and PBM indices. For this data set, the indices for true labels are $Conn\_Index = 1.0$, DBI $= 0.16$, GDI $= 8.5$, CDbw $= 4000$, Silhouette $= 0.89$, CH-VRC $= 0.83$, and PBM $= 3.58$.

separation of these superclusters. In comparison, as long as the clusters are separated, it will be reflected by $Conn\_Index$ even if the clusters have different shapes or sizes or uneven data distribution.

When the index values for the true labels are compared to the indices of k-means clusterings in Fig. 7, indices except CH-VRC and PBM strongly favor the true labels over any k-means clustering due to the fact that these 11 clusters are spherical and well-separated. Surprisingly, PBM favors an incorrect partitioning of k-means with ten clusters while CH-VRC favors k-means with k2 or k3 (super clusters) over the 11 known well-separated clusters.

V. Performance of $Conn\_Index$ on Real Data

A. $Conn\_Index$ for Data Sets With Small Number of Data Samples and Few Clusters

We use three of the benchmark data sets in the UCI Machine Learning Repository [2]: Breast Cancer Wisconsin, Iris, and Wine. These have small numbers of data samples and at most three classes. The analyses of the index performance on these data sets provide a necessary step before moving on to complicated data because if the index does not perform well on these data, it may not perform well on more complicated ones. We obtain the quantization prototypes of the data sets with a SOM and cluster the (4 × 4) SOM prototypes by k-means clustering. The validity indices values are listed in Table II.

1) Breast Cancer Wisconsin: This data set consists of 699 samples with ten features grouped into two linearly inseparable classes (benign and malignant). $Conn\_Index$ and Silhouette (Table II) favor the true labels as the best partitioning of the data set and k-means clustering with k2 as the second best. Contrarily, DBI, GDI, and CH-VRC indicate k-means clustering with k2 as the best and the true labels as the second best. This is mainly because the true clusters are nonspherical and these three indices are dependent on centroid distances. Surprisingly, CDbw favors any k-means clustering over the true labels. One reason for this can be the highly connected nature of the SOM where prototypes may exist close to the boundaries of the clusters, which in turn results in incorrect estimation of intra-cluster density by CDbw.

2) Iris: The Iris data set has 150 samples across three species, Setosa, Versicolor, and Virginica. (50 samples per species) The input features are sepal length, sepal width, petal length, and petal width. All indices, listed in Table II, except CH-VRC and PBM, select k-means clustering with k2 as the best fit. This is expected in this case [5] due to the inseparability of Versicolor and Virginica and their clean separation from Setosa. PBM is the only index that (slightly) favors the true clusters. The runner-up is the true partitioning according to GDI, CDbw, and $Conn\_Index$. CH-VRC provides different

TABLE II
Validity Indices for k-Means Clustering of Three Real Data Sets: Breast Cancer Wisconsin, Iris and Wine. Indices for the Favored Partitionings Are in Bold Face

Data Sets	Validity index	Value for true clusters	Indices for k-means k = # of clusters			
			k=2	k=3	k=4	k=5
Breast Cancer Wisconsin (k=2)	DBI	0.69	**0.67**	0.93	0.97	1.00
	GDI	1.43	**1.56**	1.11	0.80	0.40
	CDbw	6.03	**43.7**	20.6	19.3	8.98
	Silhouette	**0.29**	0.25	0.22	0.22	-0.05
	CH-VRC	12.3	**14.3**	13.6	11.7	14.1
	PBM	89	94	**100**	76	71
	Conn_Index	**0.79**	0.78	0.64	0.39	0.30
Iris (k=3)	DBI	0.60	**0.40**	0.60	0.70	0.65
	GDI	2.75	**3.61**	2.62	1.69	1.38
	CDbw	1.06	**4.77**	0.68	0.41	0.30
	Silhouette	0.17	**0.54**	0.22	0.16	0.24
	CH-VRC	33.7	15.4	24.5	**34.3**	23.7
	PBM	**0.56**	0.35	0.54	0.53	0.45
	Conn_Index	0.67	**1.0**	0.62	0.54	0.53
Wine (k=3)	DBI	1.09	**0.85**	0.86	0.88	1.06
	GDI	0.94	**1.47**	1.40	1.16	0.62
	CDbw	0.24	**0.67**	0.51	0.45	0.25
	Silhouette	-0.19	0.06	**0.07**	0.07	-0.09
	CH-VRC	5.1	9.6	10.5	**11.0**	10.4
	PBM	0.08	0.12	**0.14**	0.13	0.14
	Conn_Index	**0.63**	0.45	0.55	0.36	0.23

rankings for Iris data depending on whether it is calculated based on data points or based on prototypes. It strongly favors k-means clustering with k2 over any other ones including the true labels for the former, whereas it strongly favors k-means clustering with k4 (CH-VRC $= 34.3$) and (true labels, CH-VRC $= 33.7$) over any other partitioning for the latter. $Conn\_Index$ is as far from selecting the true clusters as any of the other indices due to the well-known separated cluster from two other overlapping clusters.

$Conn\_Index = 1$ for k-means with k2 reflects the clean separation of the two extracted clusters. The $Conn\_Index$ value of less than 1.0 for the true labels (0.67) and for the k-means with k3 (0.62) indicate overlap among the clusters. The same information can be learned, to some extent, from the GDI and DBI values, which strongly favor k-means clustering with k2 and have a similar percentage change (about 40%) in the index value in response to increasing k to 3. For example, the GDI value is 3.61 for k-means with k2 whereas it is 2.62 for k-means with k3 and 2.75 for true labels. However, we cannot directly learn from the GDI and DBI values whether the extracted clusters are clearly separated. This is because the GDI is not necessarily constructed from the separation and the scatter of the same cluster (numerator and denominator in (1) may be from different clusters), and the DBI and Silhouette consider the average distance to cluster centroid but not the maximum distance to cluster centroid [(2)].

3) Wine: This data set has 178 13-D samples with three classes. The groups are nonspherical but separable. $Conn\_Index$ is the only index which selects the known labels as the best partitioning. It also produces values less than 0.5 for k-means clusterings with $k2, 4, 5$as an indication of poor partitioning. The other indices choose k-means with different k values while the number of clusters in the Wine data set is 3.

B. $Conn\_Index$ Performance for a Real Remote Sensing Image: Ocean City

For performance evaluation of $Conn\_Index$ on complicated data, we use a remote sensing spectral image of Ocean City, Maryland, comprising 512 $\times$ 512 pixels. Each pixel has an 8-D feature vector called spectrum, associated with it. 28 meaningful physical clusters have been identified in this scene and verified by a domain expert, with field observations and with aerial photographs [24], [30]. Fig. 8(a) shows the spatial layout of different surface cover types in this image through an earlier cluster map [1] which indicates the spectrally different materials by different colors. Some clusters are ocean (blue, I), small bays (medium blue, J), water canals (turquoise, R), lawn, trees and bushes (green, L; and split-pea green, O), dry grass (orange, N), marshlands (brown, P; and ocher, Q), soil (gray, S), road (magenta, G) with a reflective paint (E). The small rows of rectangles are houses with different types of roof materials (A, B, C, D, V, a, c). A detailed discussion on these 28 clusters is given in [1], [24]. Here, we point out that these 28 clusters have widely varying statistical properties and they exhibit a large range of sizes, shapes, and densities [27].

We use the 1600 SOM prototypes created for this data set in [30] and compare clusterings of these prototypes obtained by k-means and by two interactive clusterings produced in earlier works from different SOM visualizations: modified U-matrix (mU-matrix) [30] and $CONN$ visualization (CONNvis) [1]. The mU-matrix is a SOM visualization that shows Euclidean distances between prototypes neighboring in the SOM lattice as well as the number of data samples in their receptive fields, as explained in Fig. 9. CONNvis is the visualization of $CONN$ graph on the SOM lattice. The first interactive clustering [Fig. 9(a)] was obtained from mU-matrix [30]; the second one, shown in Fig. 9(b), was obtained from CONNvis [1]. The clustered image, obtained through CONNvis, is shown in Fig. 8(a). The clustered image produced from the mU-matrix can be seen in [1]. In both cases, the extracted clusters look very similar except the clustering from mU-matrix leaves more prototypes unclustered as seen in Fig. 9(a). Table III gives the index values for the interactive clusterings and for k-means with selected k values whereas Fig. 10 shows the index values for k-means with k values up to 40. For k-means, k4 is favored as the best partitioning by $Conn\_Index$, PBM, and CDbw. These four clusters, shown in Fig. 8(b), appear to be superclusters of the known 28 ones. One supercluster (dark green) comprises the known vegetation classes (lawn, trees, bushes, etc.), one (blue) includes the water classes (ocean, canals, pool, etc.), one (brown) represents soil (marshlands, bare soil, etc.) and one (purple) comprises roads, concrete, and different roof materials. The partitioning of k-means clustering with k2 which is favored by DBI, GDI, and Silhouette combines vegetation and soil into one group and everything else into another group. For larger k values, k-means produces smaller spherical clusters which do not correspond to the true partitioning. This is indicated by increasing DBI and decreasing GDI values as k increases. CDbw and $Conn\_Index$ do not have monotonic relation with increasing k, and they favor the cases where the clusters are relatively more self-contained (a larger number of connected pairs of prototypes reside within clusters). Contrarily, CH-VRC produces greater index values for greater k values (from $k = 10$ to $k = 30$) since BGSS increases and WGSS decreases due to smaller clusters for large k and this cannot be balanced by the $K - 1$ factor in the index formula given in (4) (Fig. 11).

When the indices of k-means clusterings are compared to the indices of the interactive clusterings, we expect them to favor the latter ones because we know from expert evaluation that those correspond better to the true material groups. Another reason for this expectation is that the separation between clusters is increased by the omission of prototypes at the boundaries [black cells in Fig. 9(a) and (b)]. $Conn\_Index$ favors the interactive clusterings over k-means clustering for $k > 4$ since the resulting partitions obtained by k-means with $k > 4$ do not fit the natural ones. For k-means clustering with $k = 2$ or $k = 4$, the clusters become large and they correspond to the superclusters we described above [the $k = 4$ case is shown in Fig. 9(c)]. In these cases, $Intra\_Conn$ is high (0.98 as shown in Table IV) since most of the connected prototypes remain within these large clusters. The high $Intra\_Conn$ value produces a large $Conn\_Index$ [(14)]. Therefore, $Conn\_Index$ favors $k = 2$ or $k = 4$ over the interactive clusterings. DBI, CDbw, Silhouette, and PBM favor any of the k-means clusterings over the interactive ones in spite that k-means clustering for $k > 4$ are not superclusters anymore (do not fit true partitions). GDI, however, indicates the interactive partitioning as better than k-means for $k > 10$ due to the fact that all clusters become smaller in k-means clustering with increasing k. The smaller clusters have

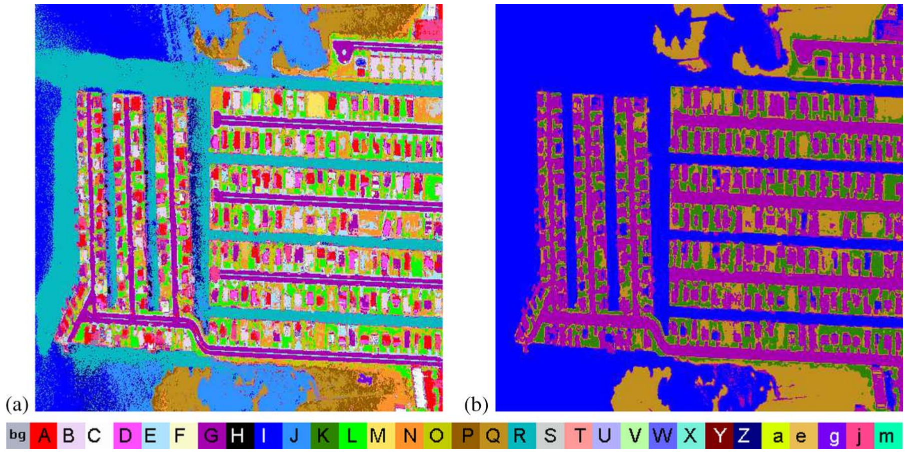


Fig. 8. Cluster map of Ocean City, an 8-band 512 × 512 pixel remote sensing image. 28 clusters were identified, and color coded according to the color wedge (not all colors were used from the color wedge). (a) Cluster map obtained by interactive clustering based on $CONN$ visualization [1]. The cluster labels of the SOM prototypes are shown in Fig. 9(b). (b) Cluster map by k-means clustering, k4.

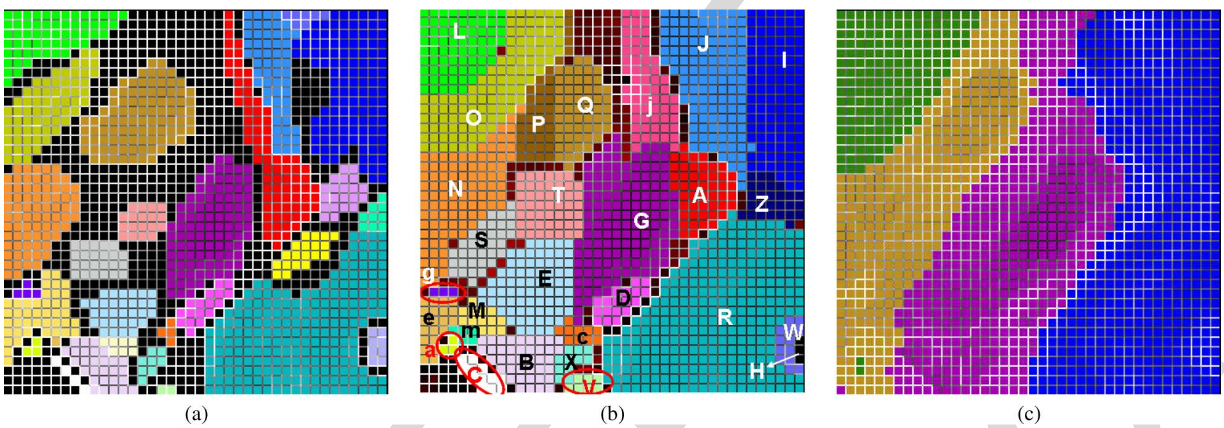


Fig. 9. Clusterings of the 40 × 40 SOM prototypes of Ocean City data. Each cell is a prototype, color coded with a cluster label consistent with Fig. 8. The intensities of the white fences around the cells are proportional to the distances between neighbor prototypes (mU-matrix). Black cells are unclustered prototypes. (a) Clustering obtained from a modified U-matrix visualization [30], (b) Clustering from $CONN$ visualization [1] (c) k-means clustering, k4 (k2 produces two clusters where one is the union of the purple and blue clusters and the other is the union of the brown and green clusters).

TABLE III
VALIDITY INDICES FOR THE CLUSTERINGS OF OCEAN CITY. INDICES FOR THE FAVORED PARTITIONINGS ARE IN BOLD FACE

Type of Clustering	# of clusters (k)	Cluster validity indices						
		DBI	GDI	CDbw	Silhouette	CH-VRC	PBM	Conn_Index
CONNvis [1]	28	1.30	0.55	0.21	-0.47	877	0.03	0.66
mU-mat [30]	28	1.17	0.41	0.18	-0.60	813	0.04	0.63
k-means	2	**0.63**	**2.75**	0.38	**0.07**	405	0.13	0.70
	4	0.65	2.25	**2.33**	-0.11	290	**0.25**	**0.72**
	10	0.86	0.62	1.47	-0.38	422	0.12	0.61
	20	1.14	0.24	0.89	-0.35	652	0.06	0.49
	28	1.18	0.23	0.74	-0.38	776	0.05	0.56
	30	1.22	0.23	0.62	-0.38	**906**	0.04	0.55

relatively smaller within-cluster distances which reduces GDI. Similarly to $Conn\_Index$, GDI favors k-means clusterings with k2 and k4 over the interactive ones, but the GDI values for these k-means clusterings are at least four times higher than the index values for the interactive ones (2.75 and 2.25 versus 0.55 and 0.41 in Table III), whereas the $Conn\_Index$ values are

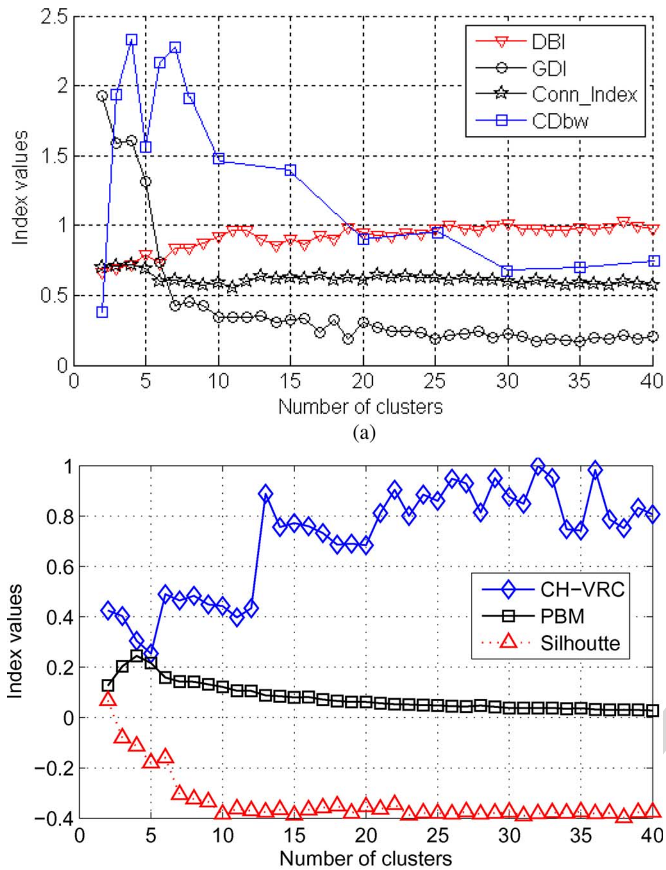


Fig. 10. Validity indices for k-means clustering of the Ocean City data set. (a) Comparison with DBI, GDI, CDbw, and $Conn\_Index$ for k-means clusterings. (c) Comparison with Silhoutte, CH-VRC, and PBM indices. CH-VRC is normalized to 1 by its maximum value 906 (k-means with $k = 30$, Table 3).

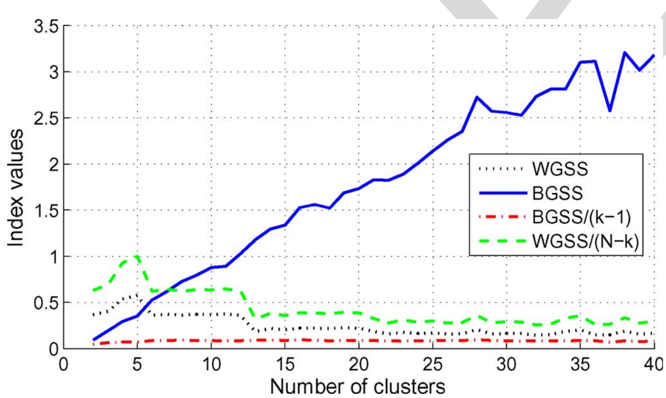


Fig. 11. Analysis of CH-VRC for k-means clustering with different k values up to 40. $WGSS/(N-k)$ in (4) is normalized to one for comparison since N is large. For $k > 10$, it can be seen that average between-cluster distance $(BGSS/(k-1))$ is almost constant whereas within-cluster distances $WGSS/(N-k)$ decreases due to smaller cluster size by increasing k values. This provides large CH-VRC values even if the partitioning is bad.

much similar (0.70 and 0.72 versus 0.66 and 0.63 in Table IV). CH-VRC strongly favors k-means clustering with $k = 30$ as the best even though that is a bad partitioning of the data set. CH-VRC also strongly favors the interactive clusterings [Fig. 9(a) and (b)] as second and third; however, this is mainly due to the large number of clusters which results in decreasing within-cluster distances while keeping the average between-cluster

TABLE IV
$Conn\_Index$ AND ITS COMPONENTS $Intra\_Conn$ AND $Inter\_Conn$ FOR THE CLUSTERINGS OF OCEAN CITY. INDICES FOR THE FAVORED PARTITIONINGS ARE IN BOLD FACE

Type of Clustering	# of clusters (k)	Conn_Index and its components		
		Conn_Index	Intra_Conn	Inter_Conn
CONNvis [1]	28	0.66	0.83	0.21
mU-mat [30]	28	0.63	0.74	**0.17**
k-means	2	0.70	**0.98**	0.26
	4	**0.72**	**0.98**	0.23
	10	0.61	0.92	0.34
	20	0.49	0.81	0.39
	30	0.55	0.79	0.31

distance constant with increasing number of clusters (Fig. 11). To further support this claim, we refer to Table I which shows that for a smaller number of clusters in the Clown data, CH-VRC ranks the true partitioning very low.

To summarize, for the relatively large number of clusters with different shapes and sizes in this data set, DBI, GDI, CDbw, Silhouette, CH-VRC, and PBM may not be helpful in evaluation of cluster validity. $Conn\_Index$ appears to provide more faithful evaluation for this case.

C. Evaluation of Partial Clusterings

SOM visualizations provide tools to extract cluster boundaries and find the cluster structure. However, due to different visualization schemes, knowledge representations, or processing by different users, different prototypes may be left unclustered in various clusterings of the same SOM. Yet, comparison of the quality of such different clusterings can be of great importance. We can argue that for these situations, $Conn\_Index$ and its components provide useful measures.

$Conn\_Index$, $Intra\_Conn$, and $Inter\_Conn$ express the relation of the unclustered prototypes to the clustered ones. Since $Intra\_Conn$ measures how self-contained the clusters are based on the connections among prototypes, it reflects how important the prototypes are for the clusters. For example, assume that p_m is a prototype in cluster C_k, and a and b are the numerator and the denominator of $Intra\_Conn(C_k)$ [(10)], respectively. Let us remove p_m from C_k and recalculate the intra-connectivity of C_k after this removal, denoted by $Intra\_Conn(C_k)^-$

$$Intra\_Conn(C_k)^- = \frac{a - \sum_j^P \{CADJ(m,j) : p_j \in C_k\}}{b - \sum_j^P CADJ(m,j)}. \tag{15}$$

Since $a \leq b$, $Intra\_Conn(C_k)^-$ will be smaller than a/b, i.e., $Intra\_Conn(C_k)$, if

$$\sum_j^P \{CONN(m,j) : p_j \in C_k\} > \frac{a}{b} \sum_j^P CADJ(m,j). \tag{16}$$

If p_m has all its connections to prototypes within its own cluster C_k, then $Intra\_Conn(C_k)^-$ becomes smaller than $Intra\_Conn(C_k)$ since $\sum_j^P\{CADJ(m,j) : p_j \in C_k\} = \sum_j^P CADJ(m,j) = RF_m$. In this case, the decrease in $Intra\_Conn(C_k)$ depends on the RF_m and on the size of C_k. The $Inter\_Conn(C_k)$ remains unchanged after this removal since p_m is not at the cluster boundary [hence not used in either the numerator or the denominator of (13)]. If p_m has connections to the prototypes in C_k and also to prototypes in another cluster, then p_m is at a cluster boundary. If within-cluster connections of p_m and its connections to other clusters have similar strengths, then p_m is in an overlapping region of the clusters. For this case, removal of p_m may not reduce $Intra\_Conn$ because $\sum_j^P\{CADJ(m,j) : p_j \in C_k\}$ is about half of the $\sum_j^P CADJ(m,j)$. Contrarily, this removal decreases $Inter\_Conn(C_k)$ [(13)] since the connections across clusters are reduced, which in turn increases $Conn\_Index$ (a better clustering). If within-cluster connections of p_m are much stronger than its connections to other clusters, removal of p_m reduces both $Intra\_Conn(C_k)$ and $Inter\_Conn(C_k)$. However, since in this case, $C_k - \{p_m\}$ becomes less self-contained due to strong connections with p_m (now outside of C_k), the decrease in $Intra\_Conn$ value will be more significant than in the previous case of overlapping clusters. At the same time, the separation $(1 - Inter\_Conn)$ only slightly increases because the connections of p_m to other clusters are much weaker than its within-cluster connections. This produces a lower $Conn\_Index$ value, indicating decreased clustering quality due to the removal of p_m.

Based on the above discussion, if prototypes at the overlapping regions are left unclustered, $Conn\_Index$ is expected to be higher than in the case they are assigned to a cluster. However, if prototypes are left unclustered at the true boundaries of a cluster, the remaining prototypes in that cluster will have strong connections to these unclustered ones near the edges of the "trimmed" cluster. Hence, in this case, the $Intra\_Conn$ value will be smaller than when the prototypes are included in the right cluster, indicating that the omitted prototypes should be assigned to the respective cluster. $Intra\_Conn$ can also be small for random partitioning. Fortunately, in such cases a high $Inter\_Conn$ value will indicate the incorrect grouping.

The interactive clusterings of the 40 × 40 SOM for Ocean City are shown in Fig. 9. The first one [Fig. 9(a)], obtained from a modified U-matrix [30], has many unclustered prototypes (black cells) due to the user's conservative judgment given the uncertainty about the boundaries in the SOM visualization. The second one [Fig. 9(b)], obtained from $CONN$ visualization [1], has very few omitted prototypes. Table IV shows the $Conn\_Index$ and its components for these cluster maps. Omitting a large number of prototypes in Fig. 9(a) produces smaller $Intra\_Conn$ and $Inter\_Conn$. This is to say, the clusters are more separated in this case but many unclustered prototypes are strongly connected to some clusters, which makes those clusters less self-contained. Table IV shows that the difference between the $Intra\_Conn$ values of the clusterings from the $CONN$ visualization and from the mU-matrix is 0.09 whereas the difference of their $Inter\_Conn$ values is 0.04. In this case, the decrease in $Intra\_Conn$ is more significant than the decrease in $Inter\_Conn$, which results in a decreased $Conn\_Index$ value according to (14). Therefore, $Conn\_Index$ favors the more complete clustering based on $CONN$ visualization over the clustering based on the modified U-matrix.

VI. Summary, Discussion, and Conclusion

$Conn\_Index$ is a new validity index for prototype-based clustering algorithms. Prototype-based clustering is increasingly important in the light of the data volume explosion we experience in real applications and because of the need for extraction of complex structure from data. $Conn\_Index$ utilizes the data topology on the prototype level as its scatter and separation measures. Its within-cluster scatter measure, the intra-cluster connectivity $(Intra\_Conn)$, and between-cluster separation measure, the complement of the inter-cluster connectivity $(1 - Inter\_Conn)$, are obtained from the "connectivity matrix" (a weighted Delaunay triangulation) defined in [1], thus $Conn\_Index$ reflects the cluster validity according to the adjacencies of the prototypes, and to local data distribution within their receptive fields. This makes $Conn\_Index$ applicable for validity evaluation of clustering results for data sets with clusters of different shapes, sizes or densities, or with overlapping clusters. The scope of this index is restricted to prototype-based clusterings due to its construction, and it is not applicable for data mining scenarios where data samples are clustered directly.

$Conn\_Index$ and its components are bounded (all are in [0, 1]). The maximum $Conn\_Index$ value indicates that clusters are well-separated whereas any index value less than 1 shows clusters are overlapping. Due to the constructions of $Intra\_Conn$ (which uses all connections of each cluster) and $Inter\_Conn$ (which uses the connections of the prototypes at the cluster boundaries only), $Conn\_Index$ can also help evaluation of partial clusterings, where different prototypes are left unclustered in different clusterings.

One thing to notice about the $Intra\_Conn$ component of $Conn\_Index$ is its dependence on the size of clusters. We can illuminate this as follows: Assume the body of the Clown in Fig. 2 has more data samples (hence more prototypes) at the bottom of the body, and we are calculating the index for true labels. The sum of the receptive fields $\sum RF_j$ of the body increases with these additional samples but the number of the prototypes that have their second BMU in other clusters [one in the body, the prototype connected to O1 in Fig. 2(b)] remains the same. This produces an equal amount of increase (number of additional samples) in the numerator and the denominator of $Intra\_Conn(body)$ [(10)], resulting in a higher $Intra\_Conn(body)$, hence a higher $Intra\_Conn$ value than the actual $Intra\_Conn$ of the original true labels (0.97, Table I). The body becomes more self-contained than before. However, such addition of data samples does not affect the separation of the body from others because the separation measure $[1 - Inter\_Conn$, (13)] depends only on the prototypes at the cluster boundaries. Yet, $Conn\_Index$ becomes slightly larger which indicates a better clustering because of a slightly more self-contained cluster. The averaging of $Intra\_Conn(C_k)$ values [(9)] will diminish the effect of few large clusters in case of many existing clusters. However, partitioning large data sets into a few clusters will produce a high $Intra\_Conn$ value since

$Intra\_Conn(C_k)$ [(10)] tends to one as the size of cluster C_k increases, even if those clusters do not correspond to the true partitions. For such cases, the quality of extracted clusters is determined by the $Inter\_Conn$ value which is independent of the size of the clusters but dependent on the similarities at the cluster boundaries.

The computational complexity of $Conn\_Index$ is of $O(P^2)$ and only dependent on the number of prototypes P. It is similar to or less complex than the computational complexities of other indices in this paper. We refer to the Appendix for a detailed complexity analysis.

One important aspect of the application of $Conn\_Index$ is that the number of prototypes should be significantly lower than the number of data samples and much greater than the number of clusters. If the number of prototypes (with nonempty receptive fields) is very close to the number of data samples, the index becomes meaningless due to the fact that the matrices $CADJ$ and $CONN$, from which the index is constructed, represent the topology of prototypes with the local data distribution. If the number of prototypes is very close to the number of clusters, then many prototypes will be singleton clusters, which in turn produces invalid $Inter\_Conn$ measures. However, both of these cases are in contradiction to the idea of prototype-based clustering and should not arise in connection with the use of $Conn\_Index$. Apart from the above extremes, $Conn\_Index$ should provide a significant tool for measuring the quality of prototype-based clustering of complex data sets, specifically when the number of prototypes P is much less than the number of data samples N, (P is of $O(\sqrt{N})$, but much larger than the number of clusters K (P is of $O(K^2)$), as it is the case for the data sets in this paper.

Finally, we want to emphasize that while we present this paper in the context of SOM prototypes and k-means clustering of these prototypes, the construction of $Conn\_Index$ is not specific to SOM prototypes or to the clustering algorithm. The construction of the $Conn\_Index$ is based on the Voronoi tessellation of the data space with respect to a given set of prototypes (obtained with any clustering algorithm, or in any other manner). Therefore, $Conn\_Index$ is applicable to the evaluation of any prototype-based clustering where prototypes are produced by a vector quantization algorithm.

APPENDIX
COMPLEXITY OF $Conn\_Index$

In this section, we discuss the computational complexity of the proposed Conn_Index and compare it to the computational complexities of various indices used in this paper. Due to the fact that this paper is focused on the evaluation of the quality of clustering, the computational cost of prototype-based clustering algorithm, which is the same for any index used for the evaluation of cluster validity, is ignored.

The complexity of $Conn\_Index$ is computed from the complexity of the two subcomponents $Inter\_Conn$ and $Intra\_Conn$. Let N, P, and K be the number of data points, the number of prototypes, and the number of clusters, respectively, and let P_k and N_k be the number of prototypes and data points in cluster C_k, respectively. D will denote the dimensionality (number of features) of the data points. For P_k prototypes in cluster C_k, finding $Intra\_Conn$ will need $\sum_k P_k * (P_k - 1)/2 (< P^2)$ operations. To find $Inter\_Conn$, we need to find, for each pair of clusters, $Inter\_Conn(k,l)$, the connectivities across cluster boundaries (this costs, for each pair of clusters C_k and C_l, at most $P_k * P_m$ operations) and we need the within-cluster connectivities of the prototypes at the boundaries (at most $\sum_k P_k * (P_k - 1)/2$ operations, assuming each prototype has connections to prototypes in another cluster). Calculation of $Inter\_Conn$ from $Inter\_Conn(k,l)$ requires $O(K^2) \ll O(P^2)$ operations. Thus, $Conn\_Index$ has a complexity of at most $O(P^2)$. (Note that the calculation of matrices $CADJ$ and $CONN$ do not carry any additional computational cost since they are formed during assignment of data samples to the prototypes, which is a mandatory step in prototype-based clustering.) The complexity depends only on the number of prototypes and does not depend on the number of data samples or on the dimensionality of the data points, which makes $Conn\_Index$ easily applicable for large and high-dimensional data sets.

The complexity of GDI [5] [(1)] based on average distance to cluster centroid as within-cluster distance requires $\sum_k P_k * (P_k - 1)/2$ operations to find cluster centroids and $\sum_k P_k = P$ operations to find the within-cluster distances if it is calculated based on the prototypes (at most of $O(DP^2)$), and $\sum_k N_k * (N_k - 1)/2$ operations (of $O(DN^2)$) if it is calculated based on the data samples. The calculation of average linkage requires $K * (K-1)/2$ operations after finding centroids, whereas the calculation of single linkage requires $\sum_k \sum_m P_k * P_m (< P^2)$ operations. Thus GDI has a computational complexity of $O(DP^2)$ when calculated from prototypes and $O(DN^2)$ when based on data samples. The computational complexity of the DBI which uses average distance to cluster centroid and average linkage [(1)]; of the Silhouette width criterion that uses average distance between samples in the cluster and single linkage [(3)]; and of CH-VRC that uses average distance to cluster centroid and average linkage [(4)] is similar to the complexity of GDI. While the complexity of $Conn\_Index$, $O(P^2)$, is comparable to $O(DP^2)$, it is much less than $O(DN^2)$ since for the data sets used in this paper, P is typically in the order of a few times the square root of the number of data samples $(\sqrt{N})$, that is $O(DN^2) \approx O(DP^4)$. (For example, the Clown data set has 2220 data samples, 254 prototypes with nonempty receptive fields, and 9 clusters; the Iris data set has 150 samples, 16 prototypes, and 3 clusters; Ocean City has 262 144 [512 × 512] samples, 1600 prototypes and about 30 clusters.) Assuming an equal number of prototypes per cluster, $P_k = P/K$, the complexity of $CDbw$[6] is $O(NDP_k^2K^2) = O(NDP^2) \approx O(DP^4)$, obviously higher than the complexity of $Conn\_Index$, and the gap widens for large values of N and D.

ACKNOWLEDGMENT

This paper is partly based on the first author's Ph.D. thesis at Rice University and completed at the European Commission Joint Research Centre. The authors would like to thank Prof. B. Csathó, from the Department of Geology, University of Buffalo, for the Ocean City image and ground truth, and Dr. Alhoniemi, from the Department of Information Technology, Turku University, for the Clown data and the 19 × 17 SOM prototypes.

REFERENCES

[1] K. Tademir and E. Merényi, "Exploiting data topology in visualization and clustering of self-organizing maps," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 549–562, Apr. 2009.

[2] A. Asuncion and D. Newman, UCI machine learning repository, 2007. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[3] S. Theodoridis and K. Koutroubas, *Pattern Recognition*. New York: Academic, 1999.

[4] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[5] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 3, pp. 301–315, Jun. 1998.

[6] M. Halkidi and M. Vazirgiannis, "A density-based cluster validity approach using multi-representatives," *Pattern Recognit. Lett.*, vol. 29, no. 6, pp. 773–786, Apr. 2008.

[7] J. C. Dunn, "Well separated clusters and optimal fuzzy partitions," *J. Cybern.*, vol. 4, no. 1, pp. 95–104, 1974.

[8] M. Kim and R. S. Ramakrishna, "New indices for cluster validity assessment," *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2353–2363, Nov. 2005.

[9] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, Jun. 1985.

[10] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, "Validity index for crisp and fuzzy clusters," *Pattern Recognit.*, vol. 37, no. 3, pp. 487–501, Mar. 2004.

[11] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1650–1654, Dec. 2002.

[12] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 3, pp. 841–847, Aug. 1991.

[13] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, Aug. 1995.

[14] D. Kim, K. H. Lee, and D. Lee, "On cluster validity index for estimation of the optimal number of fuzzy clusters," *Pattern Recognit.*, vol. 37, no. 10, pp. 2009–2025, Oct. 2004.

[15] M. Bouguessa, S. Wang, and H. Sun, "An objective approach to cluster validation," *Pattern Recognit. Lett.*, vol. 27, no. 13, pp. 1419–1430, Oct. 2006.

[16] P. Maji and S. K. Pal, "Rough set based generalized fuzzy C -means algorithm and quantitative indices," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 6, pp. 1529–1540, Dec. 2007.

[17] J. Lee and D. Lee, "An improved cluster labeling method for support vector clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 461–464, Mar. 2005.

[18] J.-S. Wang and J.-C. Chiang, "A cluster validity measure with outlier detection for support vector clustering," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 1, pp. 78–89, Feb. 2008.

[19] L. Kaufman and P. Rousseauw, *Finding Groups in Data*. Hoboken, NJ: Wiley, 1990.

[20] L. Vendramin, R. J. G. B. Campello, and E. R. Hruschka, "A robust methodology for comparing performances of clustering validity criteria," in *Proc. SBIA*, vol. 5249, *LNAI*, 2008, pp. 237–247.

[21] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Stat.*, vol. 3, no. 1, pp. 1–27, 1974.

[22] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intell. Inf. Syst.*, vol. 17, no. 2/3, pp. 107–145, Dec. 2001.

[23] T. Villmann, E. Merényi, and B. Hammer, "Neural maps in remote sensing image analysis," *Neural Netw., Special Issue Self-Organizing Maps for Anal. Complex Sci. Data*, vol. 16, no. 3/4, pp. 389–403, Apr. 2003.

[24] E. Merényi, B. Csathó, and K. Tademir, "Knowledge discovery in urban environments from fused multi-dimensional imagery," in *Proc. 4th IEEE GRSS/ISPRS Joint Workshop Remote Sens. Data Fusion Over Urban Areas (URBAN)*, P. Gamba and M. Crawford, Ed.s, Paris, France, Apr. 11–13, 2007, pp. 1–13.

[25] T. Martinetz and K. Schulten, "Topology representing networks," *Neural Netw.*, vol. 7, no. 3, pp. 507–522, 1994.

[26] S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases," in *Proc. ACM SIGMOD Int. Conf. Manag. Data (SIGMOD)*, 1996, pp. 73–84.

[27] E. Merényi, K. Tademir, and L. Zhang, "Learning highly structured manifolds: harnessing the power of soms," in *Similarity Based Clustering*, M. Biehl, B. Hammer, M. Verleysen, and T. Villmann, Eds. New York: Springer-Verlag, 2009, pp. 138–168.

[28] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, May 2000.

[29] K. Tademir and E. Merényi, "A new cluster validity index for prototype based clustering algorithms based on inter- and intra-cluster density," in *Proc. IJCNN*, Orlando, FL, Aug. 12–17, 2007, pp. 2205–2211.

[30] E. Merényi, A. Jain, and T. Villmann, "Explicit magnification control of self-organizing maps for 'forbidden' data," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 786–797, May 2007.

Kadim Taşdemir (M'XX) received the B.S. degree in electrical and electronics engineering from Bogazici University, Istanbul, Turkey, in 2001, the M.S. degree in computer science from Istanbul Technical University, Istanbul, Turkey, in 2003, and the Ph.D. degree in electrical and computer engineering from Rice University, Houston, TX, in 2008.

After receipt of the Ph.D. degree, he was an Assistant Professor in the Department of Computer Engineering, Yaşar University, Izmir, Turkey.

Currently, he is a Researcher at the European Commission Joint Research Centre, Institute for Environment and Sustainability, Ispra, Italy. His research interests include detailed knowledge discovery from high-dimensional and large data sets, particularly multi- and hyperspectral imagery, artificial neural networks, self-organized learning, manifold learning, data mining, and pattern recognition. He is currently working on developing advanced control methods for monitoring agricultural resources using remote sensing imagery.

Erzsébet Merényi (SM'XX) received the M.Sc. degree in mathematics and the Ph.D. degree in computational science from Szeged (Attila Jozsef) University, Szeged, Hungary, in 1975 and 1980, respectively.

She is a Research Professor in the Departments of Statistics, and Electrical and Computer Engineering, Rice University, Houston, TX. Previously, she worked at the Central Research Institute for Physics of the Hungarian Academy of Science, and at the Lunar and Planetary Laboratory of the University of Arizona, Tucson. Her current work focuses on self-organized machine learning, artificial neural networks, manifold learning, clustering and classification of high-dimensional, complex patterns, data fusion, variable selection, data mining, and knowledge discovery. Application areas include identification of surface materials from remote sensing hyperspectral imagery on earth and other planets, medical diagnostics from microscopic hyperspectral imagery, and lately compiler optimization.

Dr. Merényi's work has been funded by NASA's Applied Information Systems Research, Mars Data Analysis, and Solid Earth and Natural Hazards Programs, the Baylor College of Medicine, DARPA, and various collaborations. She is a member of the IEEE Computational Intelligence Society, the International Neural Network Society, and the Division of Planetary Science of the American Geophysical Union.

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

Note that your paper will incur overlength page charges of $175 per page. The page limit for regular papers is 12 pages, and the page limit for correspondence papers is 6 pages.

AQ1 = Please provide IEEE membership updates.
AQ2 = Please provide IEEE membership updates.

END OF ALL QUERIES

A Validity Index for Prototype-Based Clustering of Data Sets With Complex Cluster Structures

Kadim Taşdemir, *Member, IEEE*, and Erzsébet Merényi, *Senior Member, IEEE*

***Abstract*—Evaluation of how well the extracted clusters fit the true partitions of a data set is one of the fundamental challenges in unsupervised clustering because the data structure and the number of clusters are unknown *a priori*. Cluster validity indices are commonly used to select the best partitioning from different clustering results; however, they are often inadequate unless clusters are well separated or have parametrical shapes. Prototype-based clustering (finding of clusters by grouping the prototypes obtained by vector quantization of the data), which is becoming increasingly important for its effectiveness in the analysis of large high-dimensional data sets, adds another dimension to this challenge. For validity assessment of prototype-based clusterings, previously proposed indexes—mostly devised for the evaluation of point-based clusterings—usually perform poorly. The poor performance is made worse when the validity indexes are applied to large data sets with complicated cluster structure. In this paper, we propose a new index, $Conn\_Index$, which can be applied to data sets with a wide variety of clusters of different shapes, sizes, densities, or overlaps. We construct $Conn\_Index$ based on inter- and intra-cluster connectivities of prototypes. Connectivities are defined through a "connectivity matrix", which is a weighted Delaunay graph where the weights indicate the local data distribution. Experiments on synthetic and real data indicate that $Conn\_Index$ outperforms existing validity indices, used in this paper, for the evaluation of prototype-based clustering results.**

***Index Terms*—Cluster validity index, complex data structure, connectivity, Conn_Index, prototype-based clustering.**

I. Introduction

UNSUPERVISED clustering aims to extract the natural partitions in a data set without *a priori* class information. It groups the data samples into subsets so that samples within a subset are more similar to each other than to samples in other subsets. Any given clustering method can produce a different partitioning depending on its parameters and criteria. This leads to one of the main challenges in clustering—to determine, without auxiliary information, how well the obtained clusters fit the natural partitions of the data set. The common approach for this evaluation is to use validity indices. A meaningful validity index is of great importance; however, an index that accurately evaluates clusterings of complicated data sets (data sets with many clusters of varying statistics) has not been developed yet. The objective of this paper is to propose such an index for prototype-based clustering of large data sets.

Existing cluster validity indices, discussed in Section II, work well for data with simple structures or for scenarios where the user is seeking well-behaved superclusters that can be readily derived from a simple and scalable algorithm, such as k-means, instead of extracting detailed structure of complex clusters. Two reasons for seeking satisfactory performance on this level are difficulty to search for more complex structures due to many attributes and noise and the difficulty to interpret those complex structures even if they are extracted. However, many real-world applications are increasingly dependent on finding complex structures even if interpretation may be, at least initially, challenging. Prototype-based clusterings, among them self-organizing maps (SOM) in particular, are successful for finding detailed structure, and are gaining importance for large data sets that are collected to characterize many real-world problems and to enable the discovery of new knowledge. Currently, evaluation of complex clusterings can be done only through expert knowledge and ground truth. This necessitates sophisticated indexes for validity assessment of complex cluster structures, and motivates the exploitation of specific aspects of prototype-based clustering.

We introduce a validity index $Conn\_Index$ that can evaluate prototype-based clusterings of data sets with a wide variety of cluster types. $Conn\_Index$ takes advantage of the knowledge encapsulated in the prototypes of a quantized data set and uses new measures for separation between clusters and scatter within clusters based on data topology on the prototype level. The data topology is represented by the "connectivity matrix" $CONN$ introduced in [1] as a weighted version of the Delaunay graph of the prototypes. The weights (the elements of $CONN$) express the data density local to the prototypes. This will be further explained in Section III.

To evaluate the effectiveness of $Conn\_Index$, we use two synthetic data sets with clusters of different shapes, sizes, dimensionalities, and densities. We also use four real data sets, the Breast Cancer Wisconsin (9-D), Iris (4-D), Wine (13-D) data from the UCI repository [2], and Ocean City, a remote sensing spectral image. We obtain prototypes with SOMs and cluster these prototypes with various methods—k-means and two interactive clusterings. We compare the performance of $Conn\_Index$ to the performances of commonly used indices by evaluation of which clustering results are favored as the best by each of the indices used in this paper. The outline of the paper is as follows: Section II gives a background information on cluster validity indices and common approaches for index

Manuscript received February 8, 2010; revised June 17, 2010 and September 29, 2010; accepted December 5, 2010. This work was partially supported by Grant NNG05GA94G from the Applied Information Research Program of NASA, Science Mission Directorate. This paper was recommended by Associate Editor J. Basak.

K. Taşdemir was with Rice University, Houston, TX 77005 USA. He is currently with the European Commission Joint Research Centre, Institute for Environment and Sustainability, 21027 Ispra, Italy (e-mail: kadim.tasdemir@jrc.ec.europa.eu).

E. Merényi is with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005 USA (e-mail: erzsebet@rice.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2010.2104319

construction, Section III briefly reviews the prototype-based clustering, describes the "connectivity matrix", and introduces $Conn\_Index$. Sections IV and V give examples for the performance of $Conn\_Index$ on synthetic data sets and on the real data sets, respectively. In addition, Section V shows that $Conn\_Index$ can also provide a meaningful measure when different prototypes may be left unclustered in different clusterings. Section VI concludes the paper. An Appendix provides estimates on computational complexities of the indexes compared.

II. Background on Cluster Validity Indices

A cluster validity index can be constructed by using one of the following three criteria: 1) external crtieria; internal criteria; and 3) relative criteria [3]. External criteria are used to compare clustering results to a pre-specified structure. Internal criteria are for comparison to a proximity matrix of the data samples. The common approach is to use relative criteria, which is to compare the validity of several clustering results based on a combined measure of between-cluster separation and within-cluster scatter. There are many different methods to determine the validity of crisp clustering (where each data sample belongs to only one cluster) [4]–[11] or that of fuzzy clustering (where each data sample has a degree of membership in several clusters) [12]–[16]. Some validity indices are specific to the clustering method. For example, the indices in [17], [18] are proposed for support vector clustering whereas the indices proposed in [16] are for generalized fuzzy c-means clustering. In this paper, we focus on crisp clustering algorithms and we refer to Kim *et al.*[14] for a detailed analysis of the cluster validity indices for fuzzy clustering, where an index (based on the data distribution at overlapping regions) is also proposed.

For crisp clustering, the Davies–Bouldin index (DBI) [4] and the generalized Dunn Index (GDI) [5] are two commonly used indices. Two other indices are the Silhouette width criterion [19] (selected best in a recent study [20]), and the Calinski–Harabasz variance ratio criterion (CH-VRC) [21] (selected best among 30 indices in [9]). A recent index shown to be useful is PBM [10]. All these indices provide meaningful measures for well-separated or parametrical clusters but they may fail for complicated data structures with clusters of different shapes or sizes or with overlaps. This is because available distance measures for separation between clusters and scatter within clusters may be ineffective for complicated data sets due to the fact that the cluster boundaries are usually defined not only by the distances between the data samples but also by how the samples are distributed within the clusters. Several indices proposed in recent years integrate the data distribution and the distance metrics [6], [14], [22]. One of these, CDbw (composite density between and within clusters) [6] is promising for clusters of different shapes and with homogeneous density distribution. Brief explanations of these indices are given below along with the discussion on their constructions.

A. Construction of Cluster Validity Indices

The separation and scatter measures, used in the index construction, are often computed from various distances, some of which are illustrated in Fig. 1. A general approach is to use centroid-based distance metrics (d_{b_cent} and d_{w_cent}) for separation and scatter [4], [9], [10], [12], [13], [15], which favor (hyper)spherical or (hyper)ellipsoidal clusters. The most reliable results for validity indices are obtained when all data samples in the clusters are considered in the computation of the distances for index construction [5]. In the following, N will denote the number of data vectors in a data set, K will denote the number of clusters in the clustering, and, where applicable, P will denote the number of prototypes that result from a vector quantization (SOM or other) of a data set.

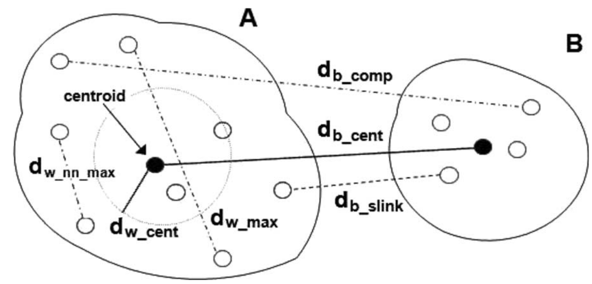


Fig. 1. Several metrics for within-cluster ($d_{w\_cent}, d_{w\_max}, d_{w\_nn\_max}$) and between-cluster ($d_{b\_cent}, d_{b\_comp}, d_{b\_slink}$) distances. d_{w_cent} is the average distance to the cluster centroid, d_{w_max} is the maximum distance between the points within the cluster, $d_{w\_nn\_max}$ is the maximum of the nearest neighbor distances. d_{b_cent} is the distance between the cluster centroids, $d_{b\_comp}(d_{b\_slink})$ is the maximum (minimum) distance between the points across the clusters. Among them, d_{b_cent} and d_{w_cent} are the commonly used metrics.

In addition to the choice of distance metrics for separation and scatter measures, how the index is constructed from these measures is also important. One way to construct the index is to calculate the ratio between the total or maximum within-cluster scatter and minimum separation between clusters such as in the Dunn index [7], or in the GDI [5]. For example, the GDI is calculated as follows:

$$GDI = \min_m \left\{ \min_n \left\{ \frac{d_{b\_i}(C_m, C_n)}{\max_k \{d_{w\_j}(C_k)\}} \right\} \right\} \tag{1}$$

where C_m, C_n, and C_k are clusters; d_{b_i} is a between-cluster separation measure and d_{w_j} is a within-cluster scatter measure with i, j indicating choices of distances. The choices for d_{b_i} and d_{w_j} can be metrics from Fig. 1 or any other that the user selects. The index constructed this way heavily depends on the cluster with the maximum scatter and on the pair of clusters with the minimum separation. If there is a large cluster or there are two small clusters which are very close to each other, the index will be dominated by their scatter or separation and will be insensitive to the separation or scatter of other clusters, thus producing an incorrect measure.

Another way to construct the index is to consider the scatter and separation measures of all clusters. A good example is the DBI, which is computed by averaging the ratio of the within-cluster scatter to the between-cluster separation over all clusters. This type of construction is useful when the separation and the scatter measures together provide a meaningful geometric interpretation of the cluster structure. The DBI is calculated with the distances between cluster centroids (d_{b_cent}) and average distances of data samples to their cluster centroid (d_{w_cent})

(from Fig. 1) as follows:

$$DBI = \frac{1}{K}\sum_{k=1}^{K}\max_{m}\left(\frac{d_{w\_cent}(C_k) + d_{w\_cent}(C_m)}{d_{b\_cent}(C_k, C_m)}\right). \quad (2)$$

With this construction, the DBI provides correct interpretation for data sets with hyperspherical clusters or with hyperellipsoidal clusters if Mahalanobis distance is chosen instead of Euclidean. A similar approach has been used in the Silhouette width criterion [19] where the average distance of a data sample i to the samples within its own cluster (d_{avg_i}) is considered along with the minimum distance of i to samples in other clusters (d_{b_i}). The criterion is obtained by averaging over all N samples as follows:

$$Silhouette = \frac{1}{N}\sum_{i=1}^{N}\frac{d_{b\_i} - d_{avg\_i}}{\max(d_{b\_i}, d_{avg\_i})}. \quad (3)$$

Another example for this type of index construction is the variance ratio criterion of Calinski and Harabasz [21] (CH-VRC). This criterion is constructed as

$$CHVRC = \frac{BGSS/(K-1)}{WGSS/(N-K)} \quad (4)$$

where $BGSS$ is between-group sum of squares [sum of squared distances of cluster centroids to the geometric center (or centroid) of all data samples], $WGSS$ is within-group sum of squares (sum of squared distances between each data sample and its respective cluster centroid). A recent index PBM [10] also uses a similar approach and is constructed by using three components

$$PBM = \left(\frac{1}{K}\frac{E_1}{E_K}D_K\right)^2. \quad (5)$$

E_1 is the average distance to the geometric center of all samples; E_K is the sum of within-cluster distances (distances of data samples to their respective cluster centroid); and D_K is the maximum distance between the centers of the K clusters.

Instead of using cluster centroids, the CDbw index [6] defines the separation and the scatter based on distances between multiple cluster prototypes and data distribution around them, as follows:

$$CDbw = Intra\_dens \times Sep \quad (6)$$

where $Intra\_dens$, the scatter, is the density within one standard deviation around the prototypes, averaged over all clusters; and Sep, the separation, is the sum of the distances (d_{b_slink}) between all pairs of clusters divided by the sum of densities at the cluster boundaries (number of data samples around the midpoints of the prototypes that form single linkage between clusters). CDbw correctly evaluates clusterings where clusters have homogeneous distribution. However, CDbw fails to represent true inter- and intra-cluster densities when the clusters have inhomogeneous density distribution which is often the case for real data.

Considering the scatter and the separation of all samples or clusters (as in the case of Silhouette, CH-VRC, DBI and CDbw) can provide more reliable results than using the scatter and the separation of selected clusters, because the delineation of cluster boundaries is more dependent on the relationship between neighbor clusters than on the relationship between, for example, the closest pair of clusters. Therefore, the index we propose below utilizes the scatter and separation of all clusters, with new definitions of the scatter and separation based on the local data distribution.

III. $Conn\_Index$: A VALIDITY INDEX BASED ON PROTOTYPE LEVEL DATA TOPOLOGY

The proposed $Conn\_Index$ is tailored to exploit the information produced by prototype-based clustering methods, which makes $Conn\_Index$ suitable only for those methods. Therefore, we first explain prototype-based clustering, discuss how the data topology on the prototype level can help validity assessment, and then define the new index.

A. *Prototype-Based Clustering for Large Data Sets*

Prototype-based clustering aims to find a number of representative data vectors or prototypes in the data space which faithfully represent the large number of data samples. This is usually done through an iterative minimization of a cost function based on the deviation of the data samples from their closest prototypes, i.e., their best matching units (BMUs). For clustering of large data sets with complex cluster structures, prototype-based clustering is often preferred. Compared to clustering data samples, prototype-based clustering has the advantage that it is easier to deal with a smaller number of prototypes than with a large number of data samples (for reasons of lower computational complexity and less memory demand), and it is robust to noise and outliers. The use of single prototypes to represent a cluster, such as in k-means and fuzzy c-means, is often inadequate to describe complex cluster structures with arbitrary shapes and sizes. Therefore, multiple prototypes per cluster are employed in recent studies based on SOMs [23], [24], neural gas [25], and CURE [26]. In these methods, the number of prototypes is often much larger than the number of expected clusters, yet still much smaller than the number of the data samples. After obtaining the prototypes, they are grouped according to their similarities and data clusters are extracted by assigning each data point to the cluster of its prototype. In particular, SOMs have been successful for extraction of detailed structure [1], [27] because SOMs distribute prototypes in the data space through a topology-preserving mapping in an iterative learning process, which results in as faithful representation of the data distribution as possible with the given number of prototypes. The SOM neural units are, at the same time, indexed in a (usually 2-D) rigid lattice according to their similarity relations; therefore, similar prototypes map close to one another in the lattice and vice versa, and prototypes (weight vectors) of neural units that are neighbors in the SOM lattice represent similar data vectors. Therefore, the visualization and examination of the prototype relationships in the SOM lattice facilitates the extraction of clusters.

We briefly summarize here the SOM learning rule for completeness, details can be found in many text books. Let M be a data set, and S be the fixed SOM lattice with P neural units.

For a given data sample $v \in M$, the BMU w_i is found by

$$\|v - w_i\| \leq \|v - w_j\| \qquad \forall j \in \mathcal{S} \tag{7}$$

and then the BMU w_i and its lattice neighbors (determined by a (often Gaussian) neighborhood function $h_{i,j}(t)$, centered around the BMU w_i) are updated according to

$$w_j(t+1) = w_j(t) + \alpha(t)h_{i,j}(t)\left(v - w_j(t)\right) \tag{8}$$

where $\alpha(t)$ is a learning parameter. Both $\alpha(t)$ and $h_{i,j}(t)$ should decrease with time t. The weight vectors of the neural units become the vector quantization prototypes of the data set, ordered on a rigid lattice.

The data space can be partitioned with respect to the prototypes (obtained by any vector quantization method, SOM included), resulting in a Voronoi tessellation where each prototype is the geometric center or centroid of its Voronoi polyhedron. The Voronoi polyhedron contains the data samples which are closest to its centroid, thus it corresponds to the receptive field (RF) of the respective prototype. A Voronoi polyhedron containing no data samples indicates a discontinuity in the data space (possible separation between clusters).

B. *Topology Representation of Quantized Data by Connectivity Matrix* ($CONN$)

Each quantization prototype is the BMU for the samples in its receptive field (RF, Voronoi polyhedron). In general, topology can be expressed by the Delaunay graph (the dual of the Voronoi tessellation) which is obtained by connecting the centers of the neighboring Voronoi polyhedra (polyhedra that share an edge). In order to better characterize and summarize the data topology on the prototype level, we introduced the cumulative adjacency matrix, $CADJ$, and the connectivity matrix, $CONN$, in [1]. $CADJ$ and $CONN$ describe, as we formally explain below, the topology of the quantization prototypes but not only their adjacency relations but also their "attractions" to one another, as defined by the local densities of the manifold. They are obtained by assigning weights to edges of the induced Delaunay graph (which is the intersection of the Delaunay graph with the data manifold) that provides the binary adjacency relations of the prototypes. As proposed by Martinetz and Schulten [25], when prototypes are dense enough in the data set, the induced Delaunay graph can be produced by connecting two prototypes p_i and p_j if at least one data sample selects them as a BMU and second BMU pair, i.e., if they are the two closest prototypes to a data sample. (When a data sample is equidistant from multiple prototypes, which is a very rare case, it is assigned to the one with the lowest index i among them.) Analogously, a weighted induced Delaunay graph can be produced by assigning the number of data samples for which p_i and p_j are the BMU and the second BMU pair, as the weight to the edge in the Delaunay graph that connects p_i and p_j. These weights are the elements of the $CONN$ matrix. The weight of the edge between p_i and p_j is $CONN(i,j)$. Obviously, $CONN$ is a symmetric matrix. The cumulative adjacency $CADJ$ is nonsymmetric. $CADJ(i,j)$ is the number of data samples for which p_i is the BMU and p_j is the second BMU. $CADJ(i,j)$ therefore describes the density distribution within the receptive field RF_i of p_i with respect to its neighbors indexed by j. $CONN(i,j)$, which is the sum of $CADJ(i,j)$ and $CADJ(j,i)$, is a similarity measure for prototypes based on local densities. Both CADJ and CONN are $P \times P$ matrices indicating similarities between P prototypes.

Fig. 2 shows a visualized example of the $CONN$ matrix for a 2-D data set called "Clown", created by Vesanto and Alhoniemi [28] by using different parametric models for each cluster and adding noise. This data set has clusters of various shapes and sizes: spherical (right eye), elliptical (nose), U-shaped (mouth), three subclusters in the left eye, a sparse body, and outliers. The prototypes were obtained by a 19×17 SOM, also by [28]. $CONN$ makes high-density regions and no-data regions (disconnected parts of the data set) visible. As explained in Fig. 2(b), when $CONN$ is visualized by indicating the connection weights with proportional line width for edges in the Delaunay graph, separations between clusters may become apparent. This outlines the boundaries of some clusters even though the distances between the prototypes at the cluster boundaries may be smaller than the distances between the prototypes within clusters. The illustration in Fig. 2(b) further suggests that $CONN$ can help determine the validity of clustering for prototype based clustering algorithms. We show this in the next sections.

C. *Definition of* $Conn\_Index$

We define $Conn\_Index$ with the help of two quantities: the intra-cluster connectivity ($Intra\_Conn$) as the within-cluster scatter and the complement of the inter-cluster connectivity ($1 - Inter\_Conn$) as the between-cluster separation measure. First, we introduce these quantities and then we define our new index. Assume K clusters and P prototypes p_i ($i = 1, 2, \ldots, P$) in a data set ($N > P > K$), and let C_k and C_l refer to two different clusters ($1 \leq k, l \leq K$).

Definition 1: The intra-cluster connectivity $Intra\_Conn$ is the average of intra-cluster connectivities $Intra\_Conn(C_k)$ over all clusters

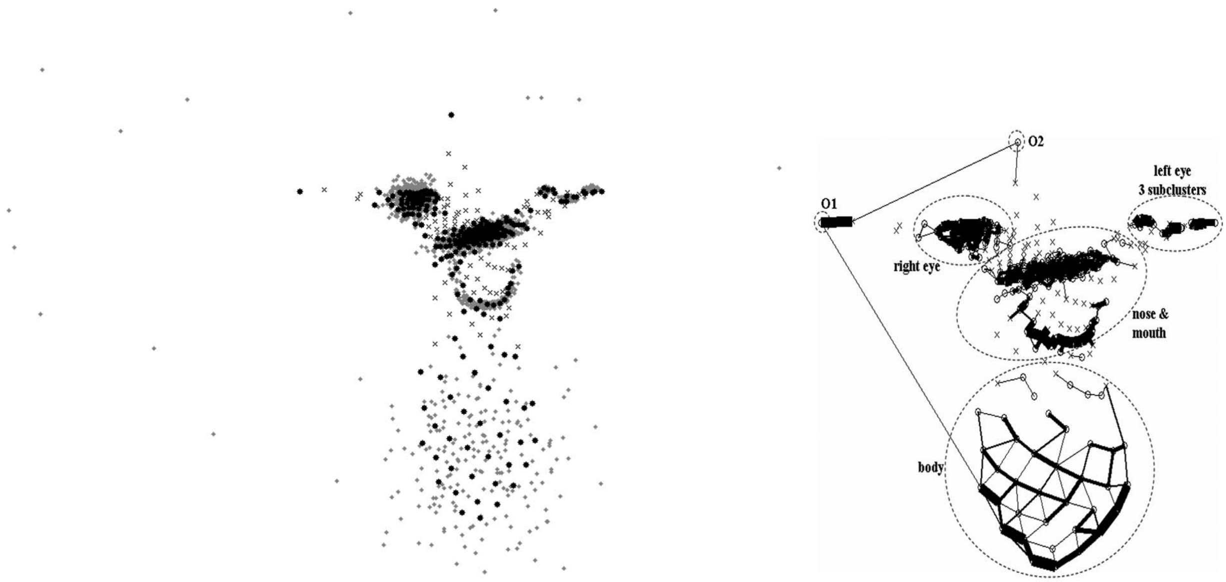
$$Intra\_Conn = \sum_{k}^{K} Intra\_Conn(C_k)/K \tag{9}$$

where $Intra\_Conn(C_k)$ is the ratio of the number of those data samples in C_k which have both their BMU and second BMU in C_k to the total number of data samples in C_k

$$Intra\_Conn(C_k) = \frac{\sum_{i,j}^{P}\left\{CADJ(i,j) : p_i, p_j \in C_k\right\}}{\sum_{i,j}^{P}\left\{CADJ(i,j) : p_i \in C_k\right\}}. \tag{10}$$

The denominator of (10) can be replaced by the sum of receptive field sizes of prototypes $p_i \in C_k$ because, obviously, the receptive field size of p_i is $RF_i = \sum_{j}^{P}\{CADJ(i,j)\}$. $Intra\_Conn$ is computed from all data samples in C_k. By definition, $Intra\_Conn(C_k) \in [0,1]$ where a greater value means more connectivity within the cluster, i.e., C_k is more self-contained. If the second BMUs of all data samples in C_k are also in C_k (there is no connection to any other cluster) $Intra\_Conn(C_k) = 1$.

To define the inter-cluster connectivity $Inter\_Conn(C_k, C_l)$ between clusters C_k and C_l, we



(a) (b)

Fig. 2. (a) 2-D data set "Clown" (a mixture of several parametrical distributions) and the SOM prototypes created by [28]. Small gray diamonds indicate data samples. Notice that there are several outliers at the far upper left which are somewhat hard to see. The black dots are prototypes with non-empty receptive fields, while $\times$ are prototypes with empty receptive fields. The data set has different types of clusters, such as spherical (right eye), elliptical (nose), U-shaped (mouth), sparse (body), three small elliptical subclusters (left eye). Variances within clusters and inter-cluster distances are different but the clusters are well separated except for the mouth and nose. (b) Topology representation by connectivity matrix $CONN$. An edge between two prototypes indicates adjacency of their Voronoi cells. The width of a line is proportional to the number of data samples for which the prototypes connected by this line are a BMU and the second BMU pair. The separations between clusters are indicated by unconnected prototypes.

consider the prototypes at the cluster boundaries since those prototypes are the ones which often facilitate the separation between clusters. A prototype at a cluster boundary is the one which may have connections to clusters other than its own.

Definition 2: The inter-cluster connectivity of clusters C_k and C_l $Inter\_Conn(C_k, C_l)$ is the ratio of the sum of the connectivity strengths between C_k and $C_l Conn(C_k, C_l)$ to the sum of the connectivity strengths of those prototypes in C_k which have at least one connection to a prototype in C_l

$$\begin{aligned} &Inter\_Conn(C_k, C_l) \\ &\quad = \begin{cases} 0, & \text{if } P_{k,l} = \emptyset \\ \frac{Conn(C_k,C_l)}{\sum_{i,j}^{P}\{CONN(i,j): p_i \in P_{k,l}\}}, & \text{if } P_{k,l} \neq \emptyset \end{cases} \\ &with\ Conn(C_k, C_l) \\ &\quad = \sum_{i,j}^{P} \{CONN(i,j) : p_i \in C_k, p_j \in C_l\} \\ &and\ P_{k,l} \\ &\quad = \{p_i : p_i \in C_k, \exists p_j \in C_l : CADJ(i,j) > 0\}. \qquad (11) \end{aligned}$$

$Inter\_Conn(C_k, C_l)$ shows how similar the prototypes at the boundary of C_k are to the ones at the boundary of C_l in comparison to the similarity of the prototypes within C_k. If C_k and C_l are completely separated in the sense that there are no cross-connections $Inter\_Conn(C_k, C_l) = 0$. A greater $Inter\_Conn(C_k, C_l)$ is an indication of a greater degree of similarity between C_k and C_l. $Inter\_Conn(C_k, C_l) > 0.5$ indicates that those prototypes in C_k which have connections to C_l are more similar to the prototypes in C_l than to the prototypes in C_k. This means they should either be in C_l or C_k and C_l should be combined. The cluster most similar to C_k is the one for which $Inter\_Conn(C_k, C_l)$ is maximum $(l \neq k, 1 \leq l \leq K)$.

Definition 3: The inter-cluster connectivity (average similarity) $Inter\_Conn$ is the average of the inter-cluster connectivities of all clusters $Inter\_Conn(C_k)$

$$Inter\_Conn = \sum_{k}^{K} Inter\_Conn(C_k)/K \qquad (12)$$

where

$$Inter\_Conn(C_k) = \max_{l, l \leq K} Inter\_Conn(C_k, C_l). \qquad (13)$$

Similarly to $Intra\_Conn$, $Inter\_Conn \in [0, 1]$ by definition. Since $Inter\_Conn$ is average similarity, $1 - Inter\_Conn$ becomes a dissimilarity (separation) measure. We define our new validity index, the $Conn\_Index$, as

$$Conn\_Index = Intra\_Conn \times (1 - Inter\_Conn). \quad (14)$$

$Conn\_Index \in [0, 1]$ increases with better clustering and has a maximum of one when the clusters are separated. Details of the calculation of $Conn\_Index$ and its components $Intra\_Conn$ and $Inter\_Conn$ were shown through an example in [29].

$Intra\_Conn$ heavily depends on the sizes of the clusters. When clusters have many data samples, the total strength of within-cluster connections will be relatively strong compared to the total strength of between-cluster connections, resulting in a high $Intra\_Conn$ value. As a result, $Intra\_Conn$ will decrease with increasing number of clusters unless the clusters are split along natural cluster boundaries. Contrarily, $Inter\_Conn$

depends only on the connections of prototypes at the cluster boundaries, hence it is independent of the sizes of clusters.

IV. Performance of $Conn\_Index$ on Synthetic Data

When comparing indices, we want to see whether they favor the true clusters as the best partitioning. True (or natural) clusters are those which satisfy the criterion “points in a cluster are closer to a point in the same cluster than to any point in other clusters”. Accordingly, “true labels” describe known true clusters in this discussion. We compare the indices computed for the clusterings obtained by different clustering methods to the indices computed for the known true labeling (true clusters). Since different indices have different ranges, some are bounded, some are not, and their nonlinearities are also different, it is not quite straightforward to compare their performance. For example, a better cluster quality is indicated by a smaller DBI while it is indicated by a greater value for other indices in this study. Theoretically, DBI, GDI, CH-VRC, PBM, and CDbw may have values in $[0, \infty)$ while Silhouette is in $[-1, 1]$ and $Conn\_Index \in [0, 1]$. However, DBI and GDI usually have a small range of values (in our experience with different data sets and different distance metrics, their maximum value did not exceed 10), whereas PBM and CDbw span a much larger range of values depending on the number of data samples and their distribution within clusters (for example, CDbw can be more than 100). Therefore, one meaningful approach is to compare the values of the same index obtained for different partitionings of the same data and determine the validity rank of clusterings according to this index and then to compare the validity ranks across different indices.

For performance evaluation, we compare $Conn\_Index$ to the indices mentioned above. We use GDI with centroid linkage (d_{b_cent} in Fig. 1) and average distance of points to cluster centroids (d_{w_cent}) as the inter- and intra-cluster distance metrics, respectively. We also considered other distance metrics (shown in Fig. 1) for GDI but did not include here due to the fact that the GDI with those metrics either performed the same or poorer than the GDI with d_{w_cent} and d_{b_cent} for the data sets in this paper. We also computed the non-prototype-based indices (DBI, GDI, CH-VRC, PBM, and Silhoutte) based on individual data points as well as based on prototypes, in order to observe whether they provide different rankings of clusterings. Due to the fact that the ranking by the various indices came out often the same by both ways of computing the indices, we provide the index values based on prototypes in this paper.

Some specific index values convey important properties. For example, $Conn\_Index = 1$ means that the clusters are completely separated whereas any other $Conn\_Index$ value indicates an overlapping case. As $Conn\_Index$ goes to zero, the degree of overlap increases. For DBI, an index value greater than one means either there are overlapping clusters or the natural partitions are not hyperspherical. However, if DBI is less than one, it does not necessarily indicate well-separated clusters. A positive value (close to one) for Silhoutte width criterion may indicate non-overlapping clusters whereas a negative value surely indicates overlapping clusters. Due to the fact that GDI considers the maximum scatter and minimum separation but not the relative dissimilarity for each cluster, a well-separated case can be represented by any GDI value.

We analyze the performance of Conn_Index on the clusterings of two synthetic data sets: the 2-D Clown data [28] with nine clusters of varying statistics, and a 6-D data set with 11 known classes [30]. These data sets—although far from the complexities real data can produce—represent some of the characteristics that make data complicated. We also show the performance of Conn_Index for real data sets: three simple data sets (Breast cancer Wisconsin, Iris, Wine) from the UCI machine learning repository [2], and an 8-D remote sensing spectral image [30]. In addition, we compare Conn_Index to DBI, GDI, CDbw, silhouette, CH-VRC, and PBM indices. Since Conn_Index does not depend on the dimensionality of the data sets, we do not include data sets with hundreds of features. In our experiments, we select the number of prototypes (P) to be larger than the number of expected clusters (K) in the data sets but much smaller than the large number of data samples (N).

A. 2-D Clown Data

The Clown data set, shown in Fig. 2 and described in Section III-B, has 2220 data samples in nine clusters which are presented in Fig. 3(a). These nine clusters can be naturally grouped into two superclusters: the face and the body.

For performance comparison of the indices, we show a hierarchical clustering produced by [28] in Fig. 3(b). This clustering extracts eight clusters with a few incorrectly labeled prototypes as shown. In Fig. 3(c), we combined two subclusters ($\triangleright$ and $\times$) in the left eye in Fig. 3(b) to measure the effect of small changes in the clustering on the validity indices. Fig. 3(d)–(f) provide the results of the k-means clustering for $\mathrm{k} = 2, 4, 5$. The k-means clustering is only successful for $\mathrm{k} = 2$ where the two clusters are the face and body which have nearly spherical structures. As k becomes larger, the partitioning is less similar to the natural partitions [Fig. 3(e)–(f)].

Table I and Fig. 4 give the indices for the different partitionings of the Clown data in Fig. 3. When we compare the indices for the clusterings in Fig. 3(b) and (c), there is a large increase in GDI in favor of the clustering in Fig. 3(c) over the true labels. This is because GDI depends on the minimum separation (which has increased by merging the two subclusters) rather than on the relative comparison of separations as in DBI, CDbw, and $Conn\_Index$. As we stated in Section II, other indices in Table I are less sensitive to this change because of their averaging property.

$Conn\_Index$ values are similar for k-means clustering with k2 and to those for the true labels. It slightly favors k-means clustering with k2 due to the supercluster structure (face and body) in the data set. This is because face and body are two large clusters connected with a thin connection, whereas known clusters (nose and mouth) are more strongly connected [Fig. 2(b)]. The index value drops slowly up to k4 and significantly for larger k due to more incorrectly labeled prototypes. GDI, Silhouette, and CH-VRC also favor k-means clustering with k2 while DBI and PBM choose k-means clustering with k4 where there are four superclusters with several incorrectly labeled prototypes. Surprisingly, CDbw favors k-means clustering with k5 where the partitioning is quite different from the true labels. One reason can be the incorrect density estimation due to varying statistics of clusters. In summary, as shown in

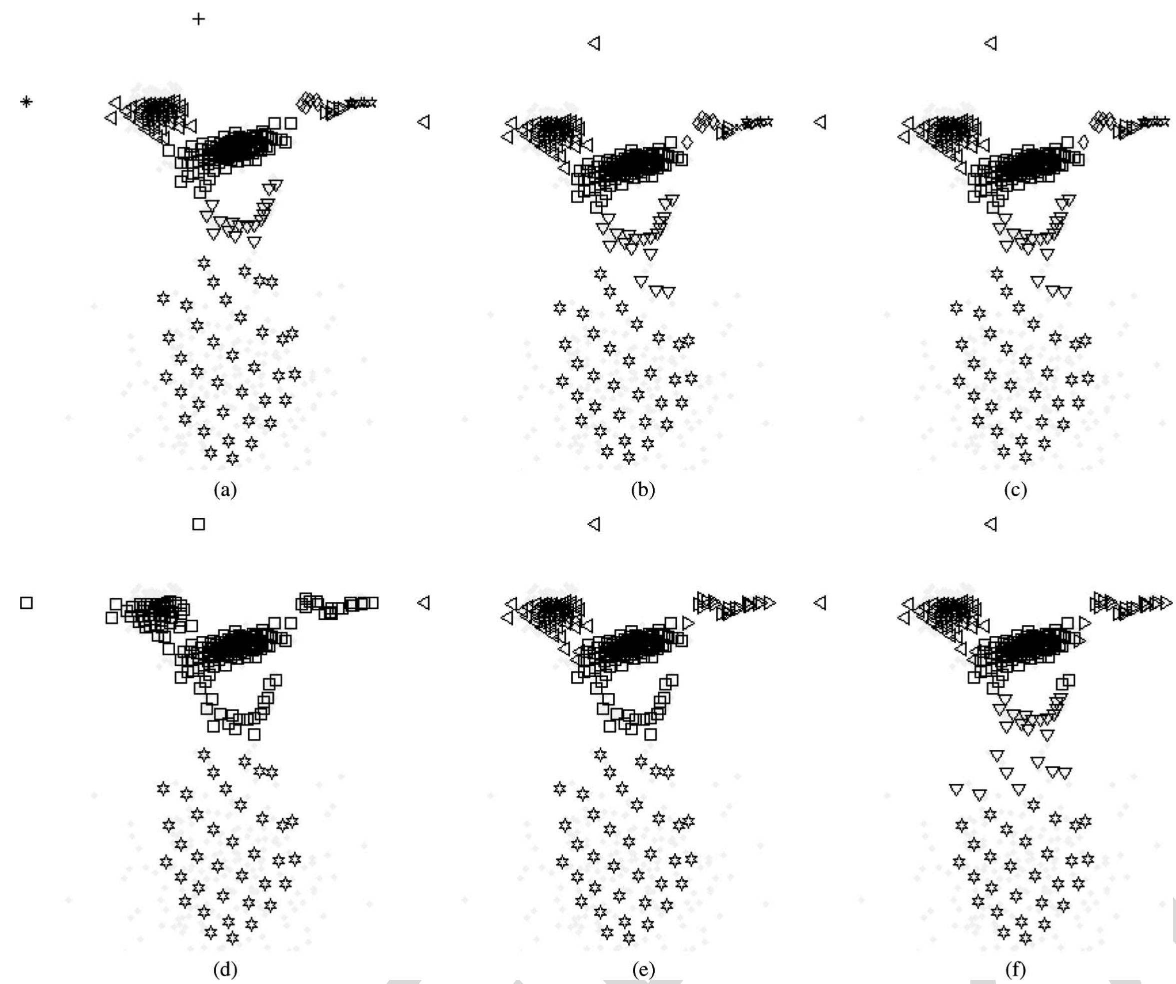


Fig. 3. Clusterings of the Clown data set by clustering of SOM prototypes. The data points are shown with dots and the prototypes are labeled by symbols. Top: (a) Known labels. Seven clusters constitute the Clown: one cluster for the body (David stars), and six clusters for the face: nose (□), mouth (▽), right eye (◁), and three clusters in the left eye (◇, ▷, open star); the remaining two, $+$ and $*$, are singletons, outliers due to noise. (b) Clustering by a hierarchical algorithm by Vesanto and Alhoniemi [28]. The two singletons are merged to the closest cluster. The true cluster in the middle of the left eye is extracted as two subclusters ▷ and $\times$. There are eight clusters with a few incorrect labels. (c) A clustering similar to (b) except the two subclusters ▷ and $\times$ in the middle of left eye are merged and labeled as ▷, in order to analyze how the indices respond to this change. Bottom: k-means clustering with (d) k2, (e) k4, (f) k5. The index values of these clusterings are shown in Table 1.

TABLE I
VALIDITY INDICES FOR THE CLUSTERINGS OF THE CLOWN DATA. INDICES FOR THE FAVORED PARTITIONINGS ARE IN BOLD FACE

Cluster validity Index	Clustering method						
	Fig. 3.a	Fig. 3.b	Fig. 3.c	k-means clustering			
	k=9	k=8	k=7	k=2	k=3	k=4	k=5
DBI	0.58	0.61	0.58	0.58	0.64	**0.49**	0.54
GDI	0.15	0.07	0.31	**2.29**	1.15	1.01	0.69
CDbw	0.39	0.49	0.56	4.92	2.32	5.48	**9.18**
Conn_Index	**0.88**	0.74	0.83	**0.89**	0.83	0.76	0.39
Silhoutte	0.22	0.19	0.18	**0.32**	0.02	0.15	0.13
CH-VRC	174	153	184	**236**	215	234	206
PBM	1.34	1.95	2.52	3.70	4.11	**4.62**	4.37

Table I, DBI, GDI, CH-VRC, PBM, and CDbw favor incorrect partitionings of k-means [for example k5, in Fig. 3(f)] over the true labels due to inaccurate density estimation of CDbw and the centroid-based approach of the rest, while Silhouette and $Conn\_Index$ favor the true labels and the supercluster structure determined by the face and the body. We point out, however, that the relative difference of $Conn\_Index$ values for the true labels (0.89) and for the superclusters (0.88) are much closer than the respective Silhoutte index values, i.e., that Silhoutte ranks the true labels lower (on its scale) than $Conn\_Index$.

B. 11-Class Data Set

This data set is from a family of 6-D synthetic data cubes used in [30] and described in detail at http://terra.ece.rice.edu. It has 128 $\times$ 128 6-D data samples in a square "image" grouped into 11 classes, three of which are relatively small. Each data sample is a 6-D feature vector (signature) specifying its characteristics. The mean signatures of eight classes are quite similar to each other and the small classes have different signatures (Fig. 5). Because the dimensionality of this data

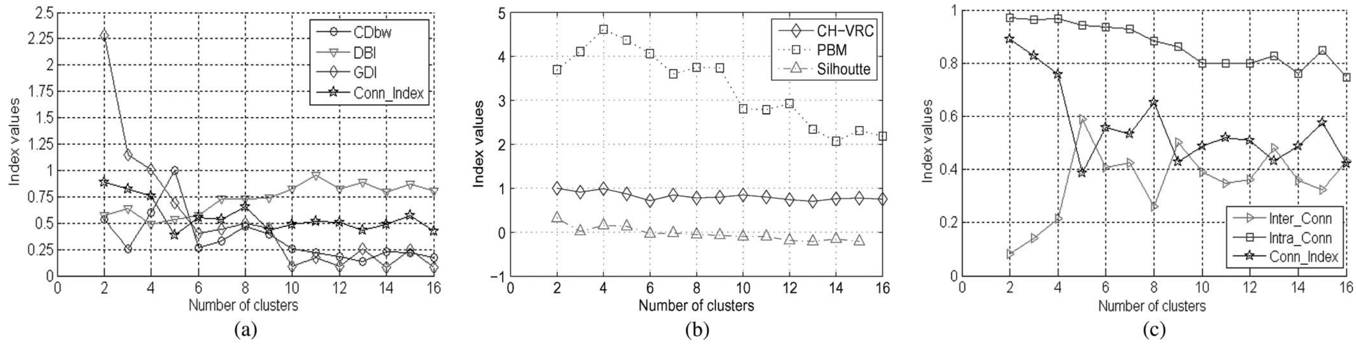


Fig. 4. Validity indices for k-means clusterings of the Clown data. (a) Comparison of DBI, GDI, CDbw, and $Conn\_Index$. CDbw is normalized by its maximum value 9.18. (b) Comparison with CH-VRC, Silhoutte, and PBM (CH-VRC is normalized to one by its maximum value, 236). (c) $Conn\_Index$ and its subcomponents, $Intra\_Conn$ and $Inter\_Conn$. $Intra\_Conn$ monotonically decreases with increasing k (except for k = 13,15) since greater k does not produce a better partitioning but reduces the size of the extracted clusters. $Inter\_Conn$ is maximum for k = 5 where some strongly connected prototypes are incorrectly labeled [Fig. 3(f)].

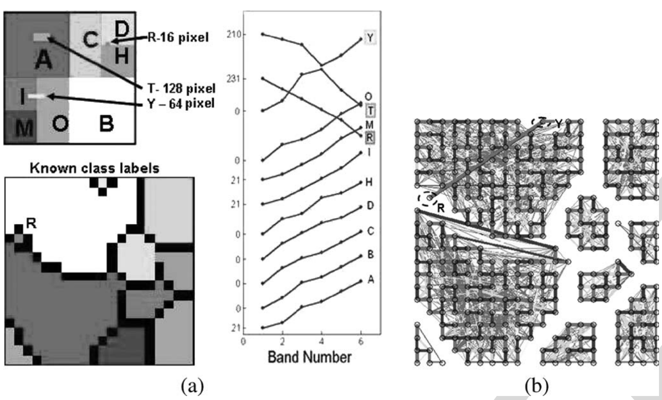


Fig. 5. (a) 6-D synthetic data set with 11 classes, three of which are relatively small. The top left image shows the spatial distribution of the data classes in the 128 × 128 pixel image. The signatures of the 11 classes are shown on the right, offset for clarity. The signatures of the small classes are very different from the rest. The bottom left image represents the known labels of the SOM prototypes. (b) The $CONN$ visualization on the SOM. The classes are well separated except for two small ones, Y and R, each of which are represented by one prototype.

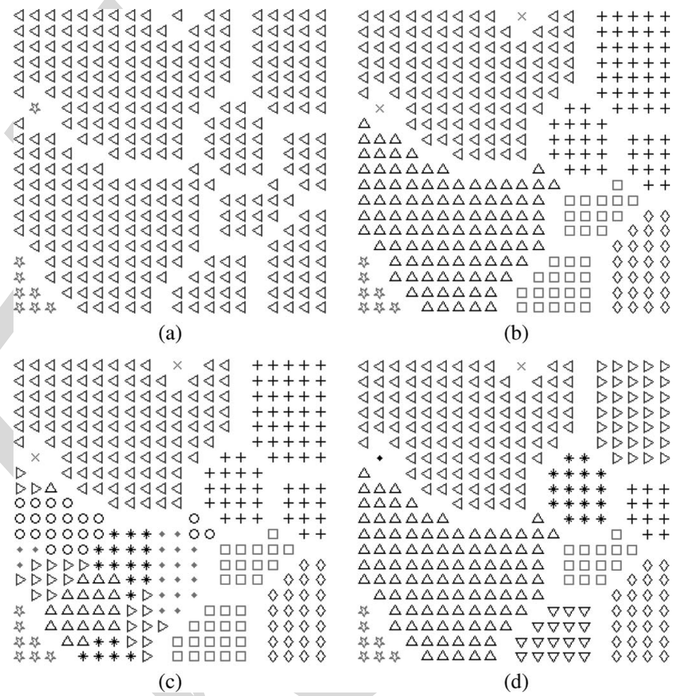


Fig. 6. k-means clustering of the (20 × 20) SOM prototypes of the 11-class data set and the true labels. (a) k2 (favored by DBI, GDI, and CDbw) (b) k7 (for which the $Conn\_Index$ is maximum). (c) k11 (true number of clusters) (d) true labels of the 11 classes.

set is greater than three, we cannot visualize it in the data space. Therefore, we show the classes (Fig. 5) through $CONN$ visualization (CONNvis) of the prototypes on the SOM lattice. CONNvis is a recent SOM visualization scheme that represents data topology [1] and has the advantage of visualizing higher dimensional data spaces on the SOM lattice regardless of the data dimensionality. CONNvis is obtained by connecting prototypes p_i, p_j whose Voronoi cells are adjacent, with lines of various widths and colors. The width of the connection is proportional to $CONN(i, j)$ whereas the color indicates the ranking of the connections to i.

Fig. 5 shows that the classes are well separated (no connections between the classes) except for two small ones, R and Y. We cluster the 20 × 20 SOM prototypes with k-means. The cluster labels for k2, 7, 11 and the true labels are given in Fig. 6. All k values up to seven produce superclusters of the existing 11 classes. Fig. 7 shows the index values for these k-means clusterings with different k values. All indices except $Conn\_Index$ and PBM favor k2 [Fig. 6(a)] as the best k-means partitioning even though the two connected small classes R and Y are grouped into different superclusters. This is because, owing to their small sizes, clusters R and Y have very little effect on those indices. In contrast, $Conn\_Index$ indicates the similarity at the cluster boundaries of these two extracted clusters in Fig. 6(a) by producing a large $Inter\_Conn$ value since the prototype representing cluster R is more similar to the prototype of Y than to any other prototype within its own group [open stars in Fig. 6(a)]. The best k-means clustering according to $Conn\_Index$ is the one with k7 [Fig. 6(b)] which is the second best according to DBI and CDbw. For k7, the two small classes R and Y are grouped into one cluster [× in Fig. 6(b)] and disconnected from the other six clusters. $Inter\_Conn$, shown in Fig. 7(a), indicates that for k4, k6 and k7, there are no cross-connections between the extracted clusters (the clusters are well separated superclusters of the 11 true classes). However, since in those cases, nonspherical clusters are likely formed, other indices may not indicate the clear

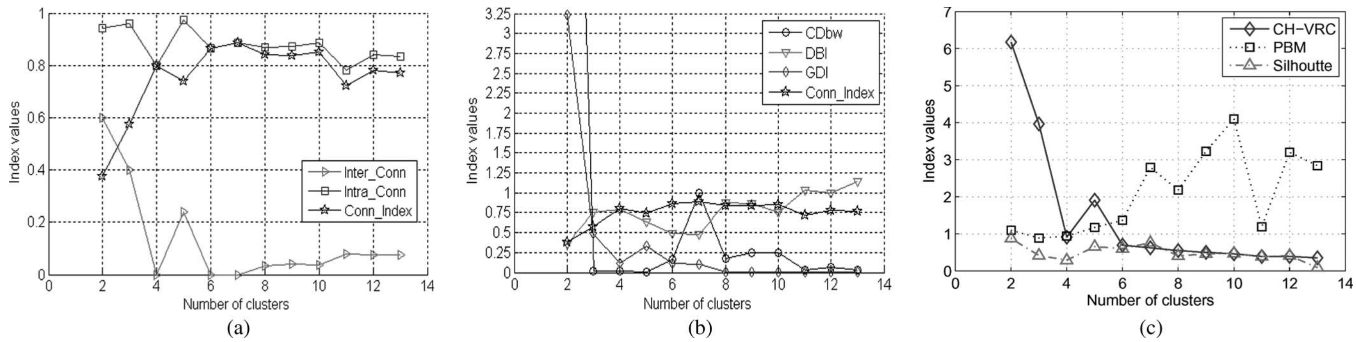


Fig. 7. Validity indices for k-means clustering of the 11-class data set. (a) $Conn\_Index$ and its subcomponents, $Intra\_Conn$ and $Inter\_Conn$. $Inter\_Conn = 0$ at k4, 6, 6 indicates that the extracted clusters are well-separated. (b) Comparison with DBI, GDI, CDbw, and $Conn\_Index$ for k-means clusterings. (c) Comparison with Silhoutte, CH-VRC, and PBM indices. For this data set, the indices for true labels are $Conn\_Index = 1.0$, DBI $= 0.16$, GDI $= 8.5$, CDbw $= 4000$, Silhouette $= 0.89$, CH-VRC $= 0.83$, and PBM $= 3.58$.

separation of these superclusters. In comparison, as long as the clusters are separated, it will be reflected by $Conn\_Index$ even if the clusters have different shapes or sizes or uneven data distribution.

When the index values for the true labels are compared to the indices of k-means clusterings in Fig. 7, indices except CH-VRC and PBM strongly favor the true labels over any k-means clustering due to the fact that these 11 clusters are spherical and well-separated. Surprisingly, PBM favors an incorrect partitioning of k-means with ten clusters while CH-VRC favors k-means with k2 or k3 (super clusters) over the 11 known well-separated clusters.

V. Performance of $Conn\_Index$ on Real Data

A. $Conn\_Index$ for Data Sets With Small Number of Data Samples and Few Clusters

We use three of the benchmark data sets in the UCI Machine Learning Repository [2]: Breast Cancer Wisconsin, Iris, and Wine. These have small numbers of data samples and at most three classes. The analyses of the index performance on these data sets provide a necessary step before moving on to complicated data because if the index does not perform well on these data, it may not perform well on more complicated ones. We obtain the quantization prototypes of the data sets with a SOM and cluster the (4×4) SOM prototypes by k-means clustering. The validity indices values are listed in Table II.

1) Breast Cancer Wisconsin: This data set consists of 699 samples with ten features grouped into two linearly inseparable classes (benign and malignant). $Conn\_Index$ and Silhouette (Table II) favor the true labels as the best partitioning of the data set and k-means clustering with k2 as the second best. Contrarily, DBI, GDI, and CH-VRC indicate k-means clustering with k2 as the best and the true labels as the second best. This is mainly because the true clusters are nonspherical and these three indices are dependent on centroid distances. Surprisingly, CDbw favors any k-means clustering over the true labels. One reason for this can be the highly connected nature of the SOM where prototypes may exist close to the boundaries of the clusters, which in turn results in incorrect estimation of intra-cluster density by CDbw.

2) Iris: The Iris data set has 150 samples across three species, Setosa, Versicolor, and Virginica. (50 samples per species) The input features are sepal length, sepal width, petal length, and petal width. All indices, listed in Table II, except CH-VRC and PBM, select k-means clustering with k2 as the best fit. This is expected in this case [5] due to the inseparability of Versicolor and Virginica and their clean separation from Setosa. PBM is the only index that (slightly) favors the true clusters. The runner-up is the true partitioning according to GDI, CDbw, and $Conn\_Index$. CH-VRC provides different

TABLE II
VALIDITY INDICES FOR k-MEANS CLUSTERING OF THREE REAL DATA SETS: BREAST CANCER WISCONSIN, IRIS AND WINE. INDICES FOR THE FAVORED PARTITIONINGS ARE IN BOLD FACE

Data Sets	Validity index	Value for true clusters	Indices for k-means k = # of clusters			
			k=2	k=3	k=4	k=5
Breast Cancer Wisconsin (k=2)	DBI	0.69	**0.67**	0.93	0.97	1.00
	GDI	1.43	**1.56**	1.11	0.80	0.40
	CDbw	6.03	**43.7**	20.6	19.3	8.98
	Silhouette	**0.29**	0.25	0.22	0.22	-0.05
	CH-VRC	12.3	**14.3**	13.6	11.7	14.1
	PBM	89	94	**100**	76	71
	Conn_Index	**0.79**	0.78	0.64	0.39	0.30
Iris (k=3)	DBI	0.60	**0.40**	0.60	0.70	0.65
	GDI	2.75	**3.61**	2.62	1.69	1.38
	CDbw	1.06	**4.77**	0.68	0.41	0.30
	Silhouette	0.17	**0.54**	0.22	0.16	0.24
	CH-VRC	33.7	15.4	24.5	**34.3**	23.7
	PBM	**0.56**	0.35	0.54	0.53	0.45
	Conn_Index	0.67	**1.0**	0.62	0.54	0.53
Wine (k=3)	DBI	1.09	**0.85**	0.86	0.88	1.06
	GDI	0.94	**1.47**	1.40	1.16	0.62
	CDbw	0.24	**0.67**	0.51	0.45	0.25
	Silhouette	-0.19	0.06	**0.07**	0.07	-0.09
	CH-VRC	5.1	9.6	10.5	**11.0**	10.4
	PBM	0.08	0.12	**0.14**	0.13	0.14
	Conn_Index	**0.63**	0.45	0.55	0.36	0.23

rankings for Iris data depending on whether it is calculated based on data points or based on prototypes. It strongly favors k-means clustering with k2 over any other ones including the true labels for the former, whereas it strongly favors k-means clustering with k4 (CH-VRC $= 34.3$) and (true labels, CH-VRC $= 33.7$) over any other partitioning for the latter. $Conn\_Index$ is as far from selecting the true clusters as any of the other indices due to the well-known separated cluster from two other overlapping clusters.

$Conn\_Index = 1$ for k-means with k2 reflects the clean separation of the two extracted clusters. The $Conn\_Index$ value of less than 1.0 for the true labels (0.67) and for the k-means with k3 (0.62) indicate overlap among the clusters. The same information can be learned, to some extent, from the GDI and DBI values, which strongly favor k-means clustering with k2 and have a similar percentage change (about 40%) in the index value in response to increasing k to 3. For example, the GDI value is 3.61 for k-means with k2 whereas it is 2.62 for k-means with k3 and 2.75 for true labels. However, we cannot directly learn from the GDI and DBI values whether the extracted clusters are clearly separated. This is because the GDI is not necessarily constructed from the separation and the scatter of the same cluster (numerator and denominator in (1) may be from different clusters), and the DBI and Silhouette consider the average distance to cluster centroid but not the maximum distance to cluster centroid [(2)].

3) Wine: This data set has 178 13-D samples with three classes. The groups are nonspherical but separable. $Conn\_Index$ is the only index which selects the known labels as the best partitioning. It also produces values less than 0.5 for k-means clusterings with $\mathrm{k}2, 4, 5$as an indication of poor partitioning. The other indices choose k-means with different k values while the number of clusters in the Wine data set is 3.

B. $Conn\_Index$ Performance for a Real Remote Sensing Image: Ocean City

For performance evaluation of $Conn\_Index$ on complicated data, we use a remote sensing spectral image of Ocean City, Maryland, comprising 512 $\times$ 512 pixels. Each pixel has an 8-D feature vector called spectrum, associated with it. 28 meaningful physical clusters have been identified in this scene and verified by a domain expert, with field observations and with aerial photographs [24], [30]. Fig. 8(a) shows the spatial layout of different surface cover types in this image through an earlier cluster map [1] which indicates the spectrally different materials by different colors. Some clusters are ocean (blue, I), small bays (medium blue, J), water canals (turquoise, R), lawn, trees and bushes (green, L; and split-pea green, O), dry grass (orange, N), marshlands (brown, P; and ocher, Q), soil (gray, S), road (magenta, G) with a reflective paint (E). The small rows of rectangles are houses with different types of roof materials (A, B, C, D, V, a, c). A detailed discussion on these 28 clusters is given in [1], [24]. Here, we point out that these 28 clusters have widely varying statistical properties and they exhibit a large range of sizes, shapes, and densities [27].

We use the 1600 SOM prototypes created for this data set in [30] and compare clusterings of these prototypes obtained by k-means and by two interactive clusterings produced in earlier works from different SOM visualizations: modified U-matrix (mU-matrix) [30] and $CONN$ visualization (CONNvis) [1]. The mU-matrix is a SOM visualization that shows Euclidean distances between prototypes neighboring in the SOM lattice as well as the number of data samples in their receptive fields, as explained in Fig. 9. CONNvis is the visualization of $CONN$ graph on the SOM lattice. The first interactive clustering [Fig. 9(a)] was obtained from mU-matrix [30]; the second one, shown in Fig. 9(b), was obtained from CONNvis [1]. The clustered image, obtained through CONNvis, is shown in Fig. 8(a). The clustered image produced from the mU-matrix can be seen in [1]. In both cases, the extracted clusters look very similar except the clustering from mU-matrix leaves more prototypes unclustered as seen in Fig. 9(a). Table III gives the index values for the interactive clusterings and for k-means with selected k values whereas Fig. 10 shows the index values for k-means with k values up to 40. For k-means, k4 is favored as the best partitioning by $Conn\_Index$, PBM, and CDbw. These four clusters, shown in Fig. 8(b), appear to be superclusters of the known 28 ones. One supercluster (dark green) comprises the known vegetation classes (lawn, trees, bushes, etc.), one (blue) includes the water classes (ocean, canals, pool, etc.), one (brown) represents soil (marshlands, bare soil, etc.) and one (purple) comprises roads, concrete, and different roof materials. The partitioning of k-means clustering with k2 which is favored by DBI, GDI, and Silhouette combines vegetation and soil into one group and everything else into another group. For larger k values, k-means produces smaller spherical clusters which do not correspond to the true partitioning. This is indicated by increasing DBI and decreasing GDI values as k increases. CDbw and $Conn\_Index$ do not have monotonic relation with increasing k, and they favor the cases where the clusters are relatively more self-contained (a larger number of connected pairs of prototypes reside within clusters). Contrarily, CH-VRC produces greater index values for greater k values (from $k = 10$ to $k = 30$) since BGSS increases and WGSS decreases due to smaller clusters for large k and this cannot be balanced by the $K - 1$ factor in the index formula given in (4) (Fig. 11).

When the indices of k-means clusterings are compared to the indices of the interactive clusterings, we expect them to favor the latter ones because we know from expert evaluation that those correspond better to the true material groups. Another reason for this expectation is that the separation between clusters is increased by the omission of prototypes at the boundaries [black cells in Fig. 9(a) and (b)]. $Conn\_Index$ favors the interactive clusterings over k-means clustering for $k > 4$ since the resulting partitions obtained by k-means with $k > 4$ do not fit the natural ones. For k-means clustering with $k = 2$ or $k = 4$, the clusters become large and they correspond to the superclusters we described above [the $k = 4$ case is shown in Fig. 9(c)]. In these cases, $Intra\_Conn$ is high (0.98 as shown in Table IV) since most of the connected prototypes remain within these large clusters. The high $Intra\_Conn$ value produces a large $Conn\_Index$ [(14)]. Therefore, $Conn\_Index$ favors $k = 2$ or $k = 4$ over the interactive clusterings. DBI, CDbw, Silhouette, and PBM favor any of the k-means clusterings over the interactive ones in spite that k-means clustering for $k > 4$ are not superclusters anymore (do not fit true partitions). GDI, however, indicates the interactive partitioning as better than k-means for $k > 10$ due to the fact that all clusters become smaller in k-means clustering with increasing k. The smaller clusters have

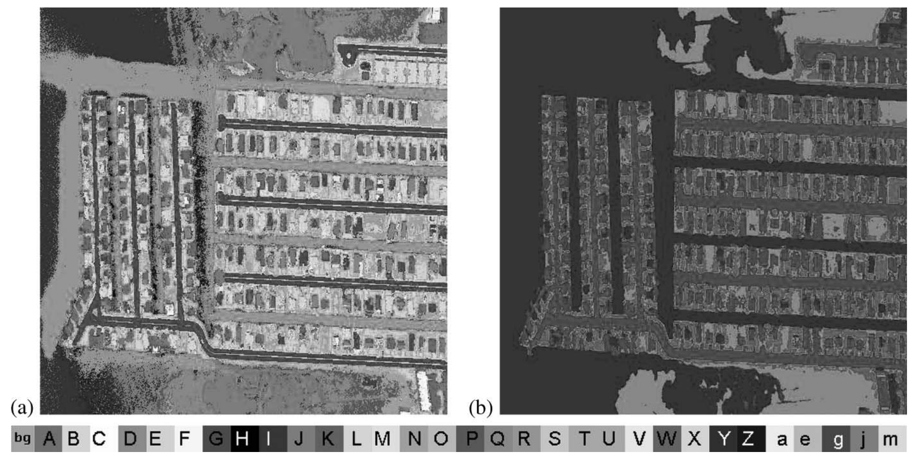


Fig. 8. Cluster map of Ocean City, an 8-band 512 × 512 pixel remote sensing image. 28 clusters were identified, and color coded according to the color wedge (not all colors were used from the color wedge). (a) Cluster map obtained by interactive clustering based on $CONN$ visualization [1]. The cluster labels of the SOM prototypes are shown in Fig. 9(b). (b) Cluster map by k-means clustering, k4.

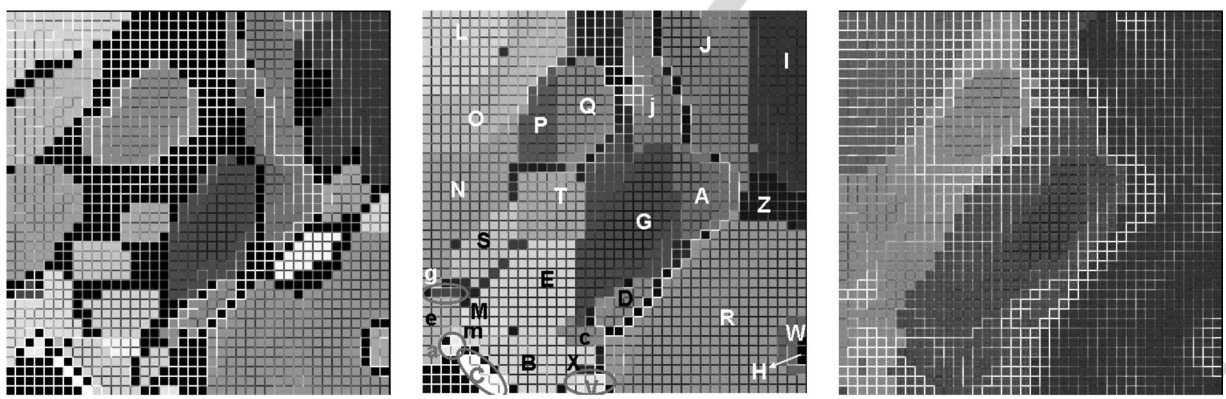


Fig. 9. Clusterings of the 40 × 40 SOM prototypes of Ocean City data. Each cell is a prototype, color coded with a cluster label consistent with Fig. 8. The intensities of the white fences around the cells are proportional to the distances between neighbor prototypes (mU-matrix). Black cells are unclustered prototypes. (a) Clustering obtained from a modified U-matrix visualization [30], (b) Clustering from $CONN$ visualization [1] (c) k-means clustering, k4 (k2 produces two clusters where one is the union of the purple and blue clusters and the other is the union of the brown and green clusters).

TABLE III
VALIDITY INDICES FOR THE CLUSTERINGS OF OCEAN CITY. INDICES FOR THE FAVORED PARTITIONINGS ARE IN BOLD FACE

Type of Clustering	# of clusters (k)	Cluster validity indices						
		DBI	GDI	CDbw	Silhouette	CH-VRC	PBM	Conn_Index
CONNvis [1]	28	1.30	0.55	0.21	-0.47	877	0.03	0.66
mU-mat [30]	28	1.17	0.41	0.18	-0.60	813	0.04	0.63
k-means	2	**0.63**	**2.75**	0.38	**0.07**	405	0.13	0.70
	4	0.65	2.25	**2.33**	-0.11	290	**0.25**	**0.72**
	10	0.86	0.62	1.47	-0.38	422	0.12	0.61
	20	1.14	0.24	0.89	-0.35	652	0.06	0.49
	28	1.18	0.23	0.74	-0.38	776	0.05	0.56
	30	1.22	0.23	0.62	-0.38	**906**	0.04	0.55

relatively smaller within-cluster distances which reduces GDI. Similarly to $Conn\_Index$, GDI favors k-means clusterings with k2 and k4 over the interactive ones, but the GDI values for these k-means clusterings are at least four times higher than the index values for the interactive ones (2.75 and 2.25 versus 0.55 and 0.41 in Table III), whereas the $Conn\_Index$ values are

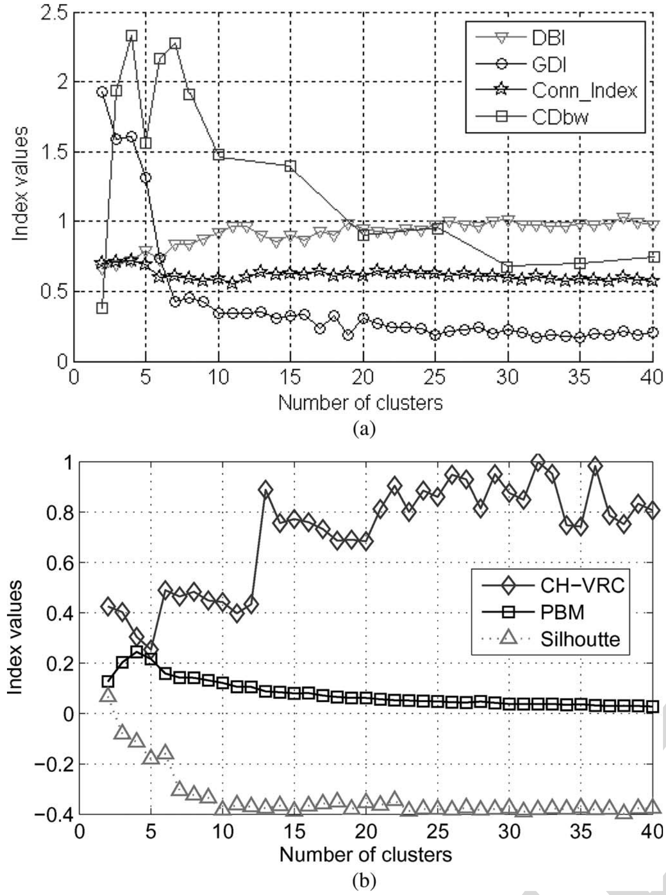


Fig. 10. Validity indices for k-means clustering of the Ocean City data set. (a) Comparison with DBI, GDI, CDbw, and $Conn\_Index$ for k-means clusterings. (c) Comparison with Silhoutte, CH-VRC, and PBM indices. CH-VRC is normalized to 1 by its maximum value 906 (k-means with $k = 30$, Table 3).

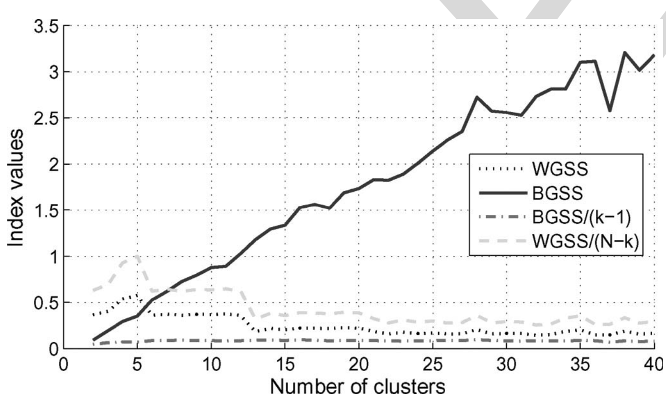


Fig. 11. Analysis of CH-VRC for k-means clustering with different k values up to 40. $WGSS/(N-k)$ in (4) is normalized to one for comparison since N is large. For $k > 10$, it can be seen that average between-cluster distance ($BGSS/(k-1)$) is almost constant whereas within-cluster distances $WGSS/(N-k)$ decreases due to smaller cluster size by increasing k values. This provides large CH-VRC values even if the partitioning is bad.

much similar (0.70 and 0.72 versus 0.66 and 0.63 in Table IV). CH-VRC strongly favors k-means clustering with $k = 30$ as the best even though that is a bad partitioning of the data set. CH-VRC also strongly favors the interactive clusterings [Fig. 9(a) and (b)] as second and third; however, this is mainly due to the large number of clusters which results in decreasing within-cluster distances while keeping the average between-cluster

TABLE IV
$Conn_Index$AND ITS COMPONENTS $Intra_Conn$AND $Inter_Conn$FOR THE CLUSTERINGS OF OCEAN CITY. INDICES FOR THE FAVORED PARTITIONINGS ARE IN BOLD FACE

Type of Clustering	# of clusters (k)	Conn_Index and its components		
		Conn_Index	Intra_Conn	Inter_Conn
CONNvis [1]	28	0.66	0.83	0.21
mU-mat [30]	28	0.63	0.74	**0.17**
k-means	2	0.70	**0.98**	0.26
	4	**0.72**	**0.98**	0.23
	10	0.61	0.92	0.34
	20	0.49	0.81	0.39
	30	0.55	0.79	0.31

distance constant with increasing number of clusters (Fig. 11). To further support this claim, we refer to Table I which shows that for a smaller number of clusters in the Clown data, CH-VRC ranks the true partitioning very low.

To summarize, for the relatively large number of clusters with different shapes and sizes in this data set, DBI, GDI, CDbw, Silhouette, CH-VRC, and PBM may not be helpful in evaluation of cluster validity. $Conn\_Index$ appears to provide more faithful evaluation for this case.

C. Evaluation of Partial Clusterings

SOM visualizations provide tools to extract cluster boundaries and find the cluster structure. However, due to different visualization schemes, knowledge representations, or processing by different users, different prototypes may be left unclustered in various clusterings of the same SOM. Yet, comparison of the quality of such different clusterings can be of great importance. We can argue that for these situations, $Conn\_Index$ and its components provide useful measures.

$Conn\_Index$, $Intra\_Conn$, and $Inter\_Conn$ express the relation of the unclustered prototypes to the clustered ones. Since $Intra\_Conn$ measures how self-contained the clusters are based on the connections among prototypes, it reflects how important the prototypes are for the clusters. For example, assume that p_m is a prototype in cluster C_k, and a and b are the numerator and the denominator of $Intra\_Conn(C_k)$ [(10)], respectively. Let us remove p_m from C_k and recalculate the intra-connectivity of C_k after this removal, denoted by $Intra\_Conn(C_k)^-$

$$Intra\_Conn(C_k)^- = \frac{a - \sum_j^P \{CADJ(m,j) : p_j \in C_k\}}{b - \sum_j^P CADJ(m,j)}. \tag{15}$$

Since $a \leq b$, $Intra\_Conn(C_k)^-$ will be smaller than a/b, i.e., $Intra\_Conn(C_k)$, if

$$\sum_j^P \{CONN(m,j) : p_j \in C_k\} > \frac{a}{b} \sum_j^P CADJ(m,j). \tag{16}$$

If p_m has all its connections to prototypes within its own cluster C_k, then $Intra\_Conn(C_k)^-$ becomes smaller than $Intra\_Conn(C_k)$ since $\sum_j^P \{CADJ(m,j) : p_j \in C_k\} = \sum_j^P CADJ(m,j) = RF_m$. In this case, the decrease in $Intra\_Conn(C_k)$ depends on the RF_m and on the size of C_k. The $Inter\_Conn(C_k)$ remains unchanged after this removal since p_m is not at the cluster boundary [hence not used in either the numerator or the denominator of (13)]. If p_m has connections to the prototypes in C_k and also to prototypes in another cluster, then p_m is at a cluster boundary. If within-cluster connections of p_m and its connections to other clusters have similar strengths, then p_m is in an overlapping region of the clusters. For this case, removal of p_m may not reduce $Intra\_Conn$ because $\sum_j^P \{CADJ(m,j) : p_j \in C_k\}$ is about half of the $\sum_j^P CADJ(m,j)$. Contrarily, this removal decreases $Inter\_Conn(C_k)$ [(13)] since the connections across clusters are reduced, which in turn increases $Conn\_Index$ (a better clustering). If within-cluster connections of p_m are much stronger than its connections to other clusters, removal of p_m reduces both $Intra\_Conn(C_k)$ and $Inter\_Conn(C_k)$. However, since in this case, $C_k - \{p_m\}$ becomes less self-contained due to strong connections with p_m (now outside of C_k), the decrease in $Intra\_Conn$ value will be more significant than in the previous case of overlapping clusters. At the same time, the separation $(1 - Inter\_Conn)$ only slightly increases because the connections of p_m to other clusters are much weaker than its within-cluster connections. This produces a lower $Conn\_Index$ value, indicating decreased clustering quality due to the removal of p_m.

Based on the above discussion, if prototypes at the overlapping regions are left unclustered, $Conn\_Index$ is expected to be higher than in the case they are assigned to a cluster. However, if prototypes are left unclustered at the true boundaries of a cluster, the remaining prototypes in that cluster will have strong connections to these unclustered ones near the edges of the "trimmed" cluster. Hence, in this case, the $Intra\_Conn$ value will be smaller than when the prototypes are included in the right cluster, indicating that the omitted prototypes should be assigned to the respective cluster. $Intra\_Conn$ can also be small for random partitioning. Fortunately, in such cases a high $Inter\_Conn$ value will indicate the incorrect grouping.

The interactive clusterings of the 40 × 40 SOM for Ocean City are shown in Fig. 9. The first one [Fig. 9(a)], obtained from a modified U-matrix [30], has many unclustered prototypes (black cells) due to the user's conservative judgment given the uncertainty about the boundaries in the SOM visualization. The second one [Fig. 9(b)], obtained from $CONN$ visualization [1], has very few omitted prototypes. Table IV shows the $Conn\_Index$ and its components for these cluster maps. Omitting a large number of prototypes in Fig. 9(a) produces smaller $Intra\_Conn$ and $Inter\_Conn$. This is to say, the clusters are more separated in this case but many unclustered prototypes are strongly connected to some clusters, which makes those clusters less self-contained. Table IV shows that the difference between the $Intra\_Conn$ values of the clusterings from the $CONN$ visualization and from the mU-matrix is 0.09 whereas the difference of their $Inter\_Conn$ values is 0.04. In this case, the decrease in $Intra\_Conn$ is more significant than the decrease in $Inter\_Conn$, which results in a decreased $Conn\_Index$ value according to (14). Therefore, $Conn\_Index$ favors the more complete clustering based on $CONN$ visualization over the clustering based on the modified U-matrix.

VI. Summary, Discussion, and Conclusion

$Conn\_Index$ is a new validity index for prototype-based clustering algorithms. Prototype-based clustering is increasingly important in the light of the data volume explosion we experience in real applications and because of the need for extraction of complex structure from data. $Conn\_Index$ utilizes the data topology on the prototype level as its scatter and separation measures. Its within-cluster scatter measure, the intra-cluster connectivity $(Intra\_Conn)$, and between-cluster separation measure, the complement of the inter-cluster connectivity $(1 - Inter\_Conn)$, are obtained from the "connectivity matrix" (a weighted Delaunay triangulation) defined in [1], thus $Conn\_Index$ reflects the cluster validity according to the adjacencies of the prototypes, and to local data distribution within their receptive fields. This makes $Conn\_Index$ applicable for validity evaluation of clustering results for data sets with clusters of different shapes, sizes or densities, or with overlapping clusters. The scope of this index is restricted to prototype-based clusterings due to its construction, and it is not applicable for data mining scenarios where data samples are clustered directly.

$Conn\_Index$ and its components are bounded (all are in [0, 1]). The maximum $Conn\_Index$ value indicates that clusters are well-separated whereas any index value less than 1 shows clusters are overlapping. Due to the constructions of $Intra\_Conn$ (which uses all connections of each cluster) and $Inter\_Conn$ (which uses the connections of the prototypes at the cluster boundaries only), $Conn\_Index$ can also help evaluation of partial clusterings, where different prototypes are left unclustered in different clusterings.

One thing to notice about the $Intra\_Conn$ component of $Conn\_Index$ is its dependence on the size of clusters. We can illuminate this as follows: Assume the body of the Clown in Fig. 2 has more data samples (hence more prototypes) at the bottom of the body, and we are calculating the index for true labels. The sum of the receptive fields $\sum RF_j$ of the body increases with these additional samples but the number of the prototypes that have their second BMU in other clusters [one in the body, the prototype connected to O1 in Fig. 2(b)] remains the same. This produces an equal amount of increase (number of additional samples) in the numerator and the denominator of $Intra\_Conn(body)$ [(10)], resulting in a higher $Intra\_Conn(body)$, hence a higher $Intra\_Conn$ value than the actual $Intra\_Conn$ of the original true labels (0.97, Table I). The body becomes more self-contained than before. However, such addition of data samples does not affect the separation of the body from others because the separation measure $[1 - Inter\_Conn$, (13)] depends only on the prototypes at the cluster boundaries. Yet, $Conn\_Index$ becomes slightly larger which indicates a better clustering because of a slightly more self-contained cluster. The averaging of $Intra\_Conn(C_k)$ values [(9)] will diminish the effect of few large clusters in case of many existing clusters. However, partitioning large data sets into a few clusters will produce a high $Intra\_Conn$ value since

$Intra\_Conn(C_k)$ [(10)] tends to one as the size of cluster C_k increases, even if those clusters do not correspond to the true partitions. For such cases, the quality of extracted clusters is determined by the $Inter\_Conn$ value which is independent of the size of the clusters but dependent on the similarities at the cluster boundaries.

The computational complexity of $Conn\_Index$ is of $O(P^2)$ and only dependent on the number of prototypes P. It is similar to or less complex than the computational complexities of other indices in this paper. We refer to the Appendix for a detailed complexity analysis.

One important aspect of the application of $Conn\_Index$ is that the number of prototypes should be significantly lower than the number of data samples and much greater than the number of clusters. If the number of prototypes (with nonempty receptive fields) is very close to the number of data samples, the index becomes meaningless due to the fact that the matrices $CADJ$ and $CONN$, from which the index is constructed, represent the topology of prototypes with the local data distribution. If the number of prototypes is very close to the number of clusters, then many prototypes will be singleton clusters, which in turn produces invalid $Inter\_Conn$ measures. However, both of these cases are in contradiction to the idea of prototype-based clustering and should not arise in connection with the use of $Conn\_Index$. Apart from the above extremes, $Conn\_Index$ should provide a significant tool for measuring the quality of prototype-based clustering of complex data sets, specifically when the number of prototypes P is much less than the number of data samples N, (P is of $O(\sqrt{N})$, but much larger than the number of clusters K (P is of $O(K^2)$), as it is the case for the data sets in this paper.

Finally, we want to emphasize that while we present this paper in the context of SOM prototypes and k-means clustering of these prototypes, the construction of $Conn\_Index$ is not specific to SOM prototypes or to the clustering algorithm. The construction of the $Conn\_Index$ is based on the Voronoi tessellation of the data space with respect to a given set of prototypes (obtained with any clustering algorithm, or in any other manner). Therefore, $Conn\_Index$ is applicable to the evaluation of any prototype-based clustering where prototypes are produced by a vector quantization algorithm.

APPENDIX
COMPLEXITY OF $Conn\_Index$

In this section, we discuss the computational complexity of the proposed Conn_Index and compare it to the computational complexities of various indices used in this paper. Due to the fact that this paper is focused on the evaluation of the quality of clustering, the computational cost of prototype-based clustering algorithm, which is the same for any index used for the evaluation of cluster validity, is ignored.

The complexity of $Conn\_Index$ is computed from the complexity of the two subcomponents $Inter\_Conn$ and $Intra\_Conn$. Let N, P, and K be the number of data points, the number of prototypes, and the number of clusters, respectively, and let P_k and N_k be the number of prototypes and data points in cluster C_k, respectively. D will denote the dimensionality (number of features) of the data points. For P_k prototypes in cluster C_k, finding $Intra\_Conn$ will need $\sum_k P_k * (P_k - 1)/2 (< P^2)$ operations. To find $Inter\_Conn$, we need to find, for each pair of clusters, $Inter\_Conn(k,l)$, the connectivities across cluster boundaries (this costs, for each pair of clusters C_k and C_l, at most $P_k * P_m$ operations) and we need the within-cluster connectivities of the prototypes at the boundaries (at most $\sum_k P_k * (P_k - 1)/2$ operations, assuming each prototype has connections to prototypes in another cluster). Calculation of $Inter\_Conn$ from $Inter\_Conn(k,l)$ requires $O(K^2) \ll O(P^2)$ operations. Thus, $Conn\_Index$ has a complexity of at most $O(P^2)$. (Note that the calculation of matrices $CADJ$ and $CONN$ do not carry any additional computational cost since they are formed during assignment of data samples to the prototypes, which is a mandatory step in prototype-based clustering.) The complexity depends only on the number of prototypes and does not depend on the number of data samples or on the dimensionality of the data points, which makes $Conn\_Index$ easily applicable for large and high-dimensional data sets.

The complexity of GDI [5] [(1)] based on average distance to cluster centroid as within-cluster distance requires $\sum_k P_k * (P_k - 1)/2$ operations to find cluster centroids and $\sum_k P_k = P$ operations to find the within-cluster distances if it is calculated based on the prototypes (at most of $O(DP^2)$), and $\sum_k N_k * (N_k - 1)/2$ operations (of $O(DN^2)$) if it is calculated based on the data samples. The calculation of average linkage requires $K * (K-1)/2$ operations after finding centroids, whereas the calculation of single linkage requires $\sum_k \sum_m P_k * P_m (< P^2)$ operations. Thus GDI has a computational complexity of $O(DP^2)$ when calculated from prototypes and $O(DN^2)$ when based on data samples. The computational complexity of the DBI which uses average distance to cluster centroid and average linkage [(1)]; of the Silhouette width criterion that uses average distance between samples in the cluster and single linkage [(3)]; and of CH-VRC that uses average distance to cluster centroid and average linkage [(4)] is similar to the complexity of GDI. While the complexity of $Conn\_Index$, $O(P^2)$, is comparable to $O(DP^2)$, it is much less than $O(DN^2)$ since for the data sets used in this paper, P is typically in the order of a few times the square root of the number of data samples ($\sqrt{N}$), that is $O(DN^2) \approx O(DP^4)$. (For example, the Clown data set has 2220 data samples, 254 prototypes with nonempty receptive fields, and 9 clusters; the Iris data set has 150 samples, 16 prototypes, and 3 clusters; Ocean City has 262 144 [512 × 512] samples, 1600 prototypes and about 30 clusters.) Assuming an equal number of prototypes per cluster, $P_k = P/K$, the complexity of $CDbw$[6] is $O(NDP_k^2K^2) = O(NDP^2) \approx O(DP^4)$, obviously higher than the complexity of $Conn\_Index$, and the gap widens for large values of N and D.

ACKNOWLEDGMENT

This paper is partly based on the first author's Ph.D. thesis at Rice University and completed at the European Commission Joint Research Centre. The authors would like to thank Prof. B. Csathó, from the Department of Geology, University of Buffalo, for the Ocean City image and ground truth, and Dr. Alhoniemi, from the Department of Information Technology, Turku University, for the Clown data and the 19 × 17 SOM prototypes.

REFERENCES

[1] K. Tademir and E. Merényi, "Exploiting data topology in visualization and clustering of self-organizing maps," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 549–562, Apr. 2009.

[2] A. Asuncion and D. Newman, UCI machine learning repository, 2007. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[3] S. Theodoridis and K. Koutroubas, *Pattern Recognition*. New York: Academic, 1999.

[4] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[5] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 3, pp. 301–315, Jun. 1998.

[6] M. Halkidi and M. Vazirgiannis, "A density-based cluster validity approach using multi-representatives," *Pattern Recognit. Lett.*, vol. 29, no. 6, pp. 773–786, Apr. 2008.

[7] J. C. Dunn, "Well separated clusters and optimal fuzzy partitions," *J. Cybern.*, vol. 4, no. 1, pp. 95–104, 1974.

[8] M. Kim and R. S. Ramakrishna, "New indices for cluster validity assessment," *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2353–2363, Nov. 2005.

[9] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, Jun. 1985.

[10] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, "Validity index for crisp and fuzzy clusters," *Pattern Recognit.*, vol. 37, no. 3, pp. 487–501, Mar. 2004.

[11] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1650–1654, Dec. 2002.

[12] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 3, pp. 841–847, Aug. 1991.

[13] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, Aug. 1995.

[14] D. Kim, K. H. Lee, and D. Lee, "On cluster validity index for estimation of the optimal number of fuzzy clusters," *Pattern Recognit.*, vol. 37, no. 10, pp. 2009–2025, Oct. 2004.

[15] M. Bouguessa, S. Wang, and H. Sun, "An objective approach to cluster validation," *Pattern Recognit. Lett.*, vol. 27, no. 13, pp. 1419–1430, Oct. 2006.

[16] P. Maji and S. K. Pal, "Rough set based generalized fuzzy C -means algorithm and quantitative indices," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 6, pp. 1529–1540, Dec. 2007.

[17] J. Lee and D. Lee, "An improved cluster labeling method for support vector clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 461–464, Mar. 2005.

[18] J.-S. Wang and J.-C. Chiang, "A cluster validity measure with outlier detection for support vector clustering," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 1, pp. 78–89, Feb. 2008.

[19] L. Kaufman and P. Rousseauw, *Finding Groups in Data*. Hoboken, NJ: Wiley, 1990.

[20] L. Vendramin, R. J. G. B. Campello, and E. R. Hruschka, "A robust methodology for comparing performances of clustering validity criteria," in *Proc. SBIA*, vol. 5249, *LNAI*, 2008, pp. 237–247.

[21] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Stat.*, vol. 3, no. 1, pp. 1–27, 1974.

[22] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intell. Inf. Syst.*, vol. 17, no. 2/3, pp. 107–145, Dec. 2001.

[23] T. Villmann, E. Merényi, and B. Hammer, "Neural maps in remote sensing image analysis," *Neural Netw., Special Issue Self-Organizing Maps for Anal. Complex Sci. Data*, vol. 16, no. 3/4, pp. 389–403, Apr. 2003.

[24] E. Merényi, B. Csathó, and K. Tademir, "Knowledge discovery in urban environments from fused multi-dimensional imagery," in *Proc. 4th IEEE GRSS/ISPRS Joint Workshop Remote Sens. Data Fusion Over Urban Areas (URBAN)*, P. Gamba and M. Crawford, Ed.s, Paris, France, Apr. 11–13, 2007, pp. 1–13.

[25] T. Martinetz and K. Schulten, "Topology representing networks," *Neural Netw.*, vol. 7, no. 3, pp. 507–522, 1994.

[26] S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases," in *Proc. ACM SIGMOD Int. Conf. Manag. Data (SIGMOD)*, 1996, pp. 73–84.

[27] E. Merényi, K. Tademir, and L. Zhang, "Learning highly structured manifolds: harnessing the power of soms," in *Similarity Based Clustering*, M. Biehl, B. Hammer, M. Verleysen, and T. Villmann, Eds. New York: Springer-Verlag, 2009, pp. 138–168.

[28] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, May 2000.

[29] K. Tademir and E. Merényi, "A new cluster validity index for prototype based clustering algorithms based on inter- and intra-cluster density," in *Proc. IJCNN*, Orlando, FL, Aug. 12–17, 2007, pp. 2205–2211.

[30] E. Merényi, A. Jain, and T. Villmann, "Explicit magnification control of self-organizing maps for 'forbidden' data," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 786–797, May 2007.

Kadim Taşdemir (M'XX) received the B.S. degree in electrical and electronics engineering from Bogazici University, Istanbul, Turkey, in 2001, the M.S. degree in computer science from Istanbul Technical University, Istanbul, Turkey, in 2003, and the Ph.D. degree in electrical and computer engineering from Rice University, Houston, TX, in 2008. AQ1

After receipt of the Ph.D. degree, he was an Assistant Professor in the Department of Computer Engineering, Yaşar University, Izmir, Turkey.

Currently, he is a Researcher at the European Commission Joint Research Centre, Institute for Environment and Sustainability, Ispra, Italy. His research interests include detailed knowledge discovery from high-dimensional and large data sets, particularly multi- and hyperspectral imagery, artificial neural networks, self-organized learning, manifold learning, data mining, and pattern recognition. He is currently working on developing advanced control methods for monitoring agricultural resources using remote sensing imagery.

Erzsébet Merényi (SM'XX) received the M.Sc. degree in mathematics and the Ph.D. degree in computational science from Szeged (Attila Jozsef) University, Szeged, Hungary, in 1975 and 1980, respectively. AQ2

She is a Research Professor in the Departments of Statistics, and Electrical and Computer Engineering, Rice University, Houston, TX. Previously, she worked at the Central Research Institute for Physics of the Hungarian Academy of Science, and at the Lunar and Planetary Laboratory of the University of Arizona, Tucson. Her current work focuses on self-organized machine learning, artificial neural networks, manifold learning, clustering and classification of high-dimensional, complex patterns, data fusion, variable selection, data mining, and knowledge discovery. Application areas include identification of surface materials from remote sensing hyperspectral imagery on earth and other planets, medical diagnostics from microscopic hyperspectral imagery, and lately compiler optimization.

Dr. Merényi's work has been funded by NASA's Applied Information Systems Research, Mars Data Analysis, and Solid Earth and Natural Hazards Programs, the Baylor College of Medicine, DARPA, and various collaborations. She is a member of the IEEE Computational Intelligence Society, the International Neural Network Society, and the Division of Planetary Science of the American Geophysical Union.

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

Note that your paper will incur overlength page charges of $175 per page. The page limit for regular papers is 12 pages, and the page limit for correspondence papers is 6 pages.

AQ1 = Please provide IEEE membership updates.
AQ2 = Please provide IEEE membership updates.

END OF ALL QUERIES