Some New Indexes of Cluster Validity

James C. Bezdek, *Fellow, IEEE*, and Nikhil R. Pal

***Abstract*—We review two clustering algorithms (hard c-means and single linkage) and three indexes of crisp cluster validity (Hubert's statistics, the Davies–Bouldin index, and Dunn's index). We illustrate two deficiencies of Dunn's index which make it overly sensitive to noisy clusters and propose several generalizations of it that are not as brittle to outliers in the clusters. Our numerical examples show that the standard measure of interset distance (the minimum distance between points in a pair of sets) is the *worst* (least reliable) measure upon which to base cluster validation indexes when the clusters are expected to form volumetric clouds. Experimental results also suggest that intercluster separation plays a more important role in cluster validation than cluster diameter. Our simulations show that while Dunn's original index has operational flaws, the concept it embodies provides a rich paradigm for validation of partitions that have cloud-like clusters. Five of our generalized Dunn's indexes provide the best validation results for the simulations presented.**

***Index Terms*—Cluster validity, Davies–Bouldin index, generalized Dunn's index, hard c-means, modified Hubert statistic, single linkage.**

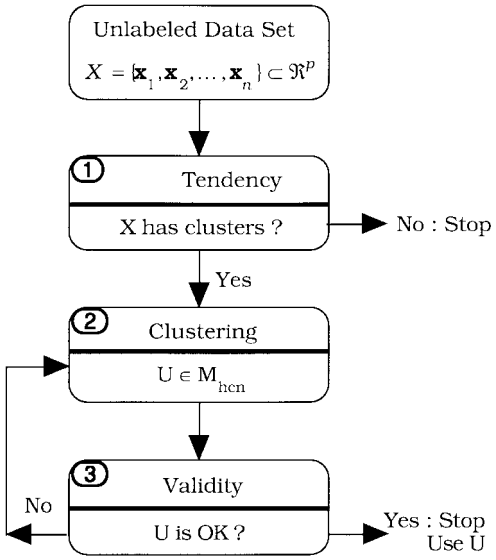


Fig. 1. Exploratory data analysis.

I. Introduction

Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\} \subset \Re^p$ be a set of n feature vectors in p-space. Suppose the vectors in X have crisp (or hard) labels that mark them as representatives of c *nonempty* classes of objects, say $O = \{o_1, o_2, \cdots, o_c\}$. Let $\boldsymbol{e}_i = (0, 0, \cdots, \underbrace{1}_{i}, \cdots, 0)^T$ be the crisp label for class $i, 1 \leq i \leq c$. The n label vectors associated with X can be arrayed as the columns of a $c \times n$ *partition matrix* $U(X) = U = [u_{ik}]$. The value u_{ik} is the membership of $\boldsymbol{x}_k$ in class i. Letting $\boldsymbol{U}_k, 1 \leq k \leq n$, denote the kth column of U, we have $\boldsymbol{U}_k = \boldsymbol{e}_i \Leftrightarrow \boldsymbol{x}_k$ is in class i. We denote the set of all hard c-partitions of X as

$$M_{\text{hcn}} = \left\{ U \in \Re^{cn}\text{: for } k = 1 \text{ to } n, \quad \boldsymbol{U}_k = \boldsymbol{e}_i \exists i; \sum_{k=1}^{n} u_{ik} > 0 \ \forall i \right\}. \tag{1}$$

An equivalent characterization of U in M_{hcn} is in terms of the c subsets that are defined by the rows of U. Specifically, we may write $U \leftrightarrow X = X_1 \cup \cdots X_i \cup \cdots X_c$, where $X_i \cap X_j = \emptyset$ whenever $i \neq j$. When no $U(X)$ is associated with X, the data are *unlabeled*. In this case, there are three questions about X as illustrated in blocks 1, 2, and 3 of Fig. 1.

Q_1 is called *assessment of clustering tendency*. Tendency assessment attempts to determine whether the data have structure in them or not without explicitly looking for clusters in the data. The only crisp partition of X at $c = 1$ is represented uniquely by the 1-partition $\mathbf{1}_n = [\underbrace{1 \quad 1 \quad \cdots 1}_{n \text{ times}}]$, which asserts that all n objects belong to a single cluster. At the other extreme, $c = n$ is represented uniquely by $U = I_n$, the $n \times n$ identity matrix, up to a permutation of columns. In this case, each object is in its own singleton cluster. Choosing $c = 1$ or $c = n$ rejects the hypothesis that X contains clusters. See Jain and Dubes [1] or Everitt [2] for formal and informal treatments of assessment prior to clustering.

Q_2 is called *cluster analysis*. There are many models and algorithms for clustering based on crisp [3], fuzzy [4], probabilistic [5], and possibilistic methods [6]. We use the well known *hard c-means* (HCM) and *single linkage* (SL) models to generate crisp c-partitions of unlabeled data sets in our examples.

Q_3 is called *cluster validity*. Once a c-partition is found, do we believe it? Should we use it? Is there a better one we didn't find? Our paper is about Q_3. We will review three well known validation methods, and then define several

Manuscript received December 11, 1994; revised November 10, 1996 and April 7, 1997. This work was supported by the ONR under Grant N00014-96-1-0642.

J. C. Bezdek is with the Department of Computer Science, University of West Florida, Pensacola, FL 32514 USA (e-mail: jbezdek@argo.cs.uwf.edu).

N. R. Pal is with Machine Intelligence Unit, Indian Statistical Institute, Calcutta 700035, India.

Publisher Item Identifier S 1083-4419(98)02610-7.

generalizations of an index due to Dunn [7]. The main purpose of the paper is to propose generalizations of Dunn's indexes, and show via numerical experiments that they provide a more accurate assessment of partition quality than the original index does.

Clustering algorithms are functions $\boldsymbol{\mathcal{C}}$: $X \mapsto \mathcal{R}_{\boldsymbol{\mathcal{C}}}$, where $\mathcal{R}_{\boldsymbol{\mathcal{C}}}$ is the range of $\boldsymbol{\mathcal{C}}$. When the output of $\boldsymbol{\mathcal{C}}$ is *just* a crisp partition (SL, for example), $\mathcal{R}_{\boldsymbol{\mathcal{C}}} = M_{\text{hcn}}$. Many clustering algorithms produce outputs besides partitions. The most common example is a second set of parameters called *point prototypes* (or cluster centers) $\boldsymbol{V} = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_c\}, \boldsymbol{v}_i \in \Re^p \ \forall i$. For example, HCM is defined jointly in the paired variables $(U, \boldsymbol{V})$, and for these cases, $\mathcal{R}_{\boldsymbol{\mathcal{C}}} = M_{\text{hcn}} \times \Re^{cp}$.

Let $\hat{\boldsymbol{\mathcal{P}}} = \{U_i \in M_{\text{hcn}}\colon\ i \le i \le N\}$ denote N different partitions (with or without extra parameters such as $\boldsymbol{V}$) of a fixed data set X that may arise as a result of: i) clustering X with one algorithm $\boldsymbol{\mathcal{C}}$ at various values of c; ii) clustering X over other algorithmic parameters of $\boldsymbol{\mathcal{C}}$; iii) applying different $\{\boldsymbol{\mathcal{C}}_j\}$ to X, each with various parameters; or iv) all of the above. The general situation can be represented as follows:

$$U = \boldsymbol{\mathcal{C}}_i\{X\colon (\underbrace{p_{i1}}_{c}, p_{i2}, \cdots, p_{ik_i})\}, \quad i = 1, 2, \cdots, M \qquad (2)$$

where $\{p_{ij}\}$ are the k_i parameters of algorithm $\boldsymbol{\mathcal{C}}_i$. For example, the parameter list for HCM is $\{c =$ number of clusters; $T =$ maximum number of iterations; $\varepsilon =$ tolerance for termination; $||*||_A =$ norm for distance calculations; $||*||_{\text{err}} =$ norm for error calculations; $\boldsymbol{V}_0 =$ initial centroids$\}$.The handful of partitions that you can feasibly generate for an unlabeled data set is a function of the M algorithms you choose to use, each of which is itself a function of its k_i parameters.

The only guaranteed common denominator of the algorithms $\{\boldsymbol{\mathcal{C}}_i\}$ is the parameter c, the number of clusters to choose. Moreover, for a fixed X, c *is the most important parameter*, in the sense that other parameters of the algorithm really have what might be called second order effects on U compared to the effect of changing the number of clusters in the data. That is, it is clearly more important to be looking in the right solution space (within c) than it is to be comparing partitions across c because c specifies the number of clusters to look for, while the other parameters control the search for these substructures. Thus, the most effective strategy for clustering is to first decide what seems to be the most reasonable estimate of the correct number of clusters by choosing one C_i, and fixing all of its parameters except c. This results in the problem most often called cluster validity: given

$$\boldsymbol{\mathcal{P}} = \{U_i(c) \in M_{\text{hcn}}\colon\ U_i(c) = \underbrace{\boldsymbol{\mathcal{C}}_i}_{\text{fixed}}(X; (c, \underbrace{p_{i2}, \cdots, p_{ik_i}}_{\text{fixed}})); \quad c = 2, 3, \cdots, c_{\max}\} \qquad (3)$$

find the *best value* for c by examining each $U_i(c)$ in $\boldsymbol{\mathcal{P}}$. There is little guidance in the literature about $c_{\max}$. A rule of thumb that many investigators use is $c_{\max} \le \sqrt{n}$. But in many cases, some auxiliary information may be available for fixing a better estimate of $c_{\max}$. For example, in an image of size $m \times n, c_{\max}$ will be much smaller than $\sqrt{mn}$.

At first glance, it seems like the criterion that defines clusters for any $\boldsymbol{\mathcal{C}}$ should be able to rank the partitions it identifies. However, it is well known that even the global extremum of objective functions such as J_1 for HCM can lead to very unrealistic partitions of X (see [3, p. 220] for an example of this). Moreover, some of the intuitively desirable properties that we want a partition to have may not be captured by a functional that is easily optimized. These are the two most compelling reasons for introducing crisp cluster validity functionals.

Validity functionals, $\mathcal{V}$: $\mathcal{D}_{\mathcal{V}} \mapsto \Re, \mathcal{D}_{\mathcal{V}}$ denoting the domain of $\mathcal{V}$, are used to numerically rank $U_i \in \boldsymbol{\mathcal{P}}$. $\mathcal{D}_{\mathcal{V}}$ is usually (but not necessarily) chosen to match the range of $\boldsymbol{\mathcal{C}}, \mathcal{D}_{\mathcal{V}} = \mathcal{R}_{\boldsymbol{\mathcal{C}}}$. When $\mathcal{D}_{\mathcal{V}} = M_{hcn}$, we call $\mathcal{V}$ a *direct measure* because it assesses properties of crisp (real) clusters or subsets in X; otherwise, it is *indirect*.

There are two ways to view $\boldsymbol{\mathcal{C}}$, and hence, two ways to approach the problem of how to define the best partition of X. First, it is possible to regard $\boldsymbol{\mathcal{C}}$ as a *parametric estimation method*: U and any additional parameters such as $\boldsymbol{V}$ in HCM are being estimated by $\boldsymbol{\mathcal{C}}$ using X. In this case $\mathcal{V}$ is regarded as a measure of goodness of fit of the estimated parameters (to a true but *unknown* set). When $\mathcal{D}_{\mathcal{V}} = M_{\text{hcn}} \times \underbrace{\text{other parameters}}_{\text{e.g., } \boldsymbol{V} \in \Re^{cp}}$, the test $\mathcal{V}$ performs is still direct. The second interpretation of $\boldsymbol{\mathcal{C}}$ is in the sense of exploratory data analysis in unlabeled data. When $\mathcal{V}$ assesses U alone, $\mathcal{V}$ is interpreted as a measure of the quality of U in the sense of partitioning for substructure (exploratory data analysis).

II. The Hard c-Means Clustering Algorithm

We will use HCM to generate partitions of X in M_{hcn}, so we describe the batch *hard c-means* (HCM) model and algorithm. *Batch* hard c (or k) means is the algorithm described in Tou and Gonzalez [8, p. 94], or by Bezdek [4, p. 55]. Confusion sometimes arises both over the use of c instead of k, and because many writers refer to sequential versions of this procedure simply as k-means, dropping the word adaptive or sequential. The HCM *model* is the constrained optimization problem

$$\min_{(U, \boldsymbol{V})} \left\{ J_1(U, \boldsymbol{V}\colon\ X) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik} ||\boldsymbol{x}_k - \boldsymbol{v}_i||_A^2 \right\} \qquad (4)$$

where $U \in M_{\text{hcn}}, \boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_c)$ is a vector of (unknown) cluster centers (weights or prototypes), $\boldsymbol{v}_i \in \Re^p$ for $1 \le i \le c$, and $|| * ||_A$ is any inner product norm ($||\boldsymbol{x}||_A^2 = \boldsymbol{x}^T A \boldsymbol{x}, A$ positive definite). Optimal HCM partitions of X are taken from optimal pairs $(U, \boldsymbol{V})$ that solve (4). Approximate solutions of (4) can be often found by the HCM *algorithm*, which is based on first order necessary conditions for local extrema of J_1.

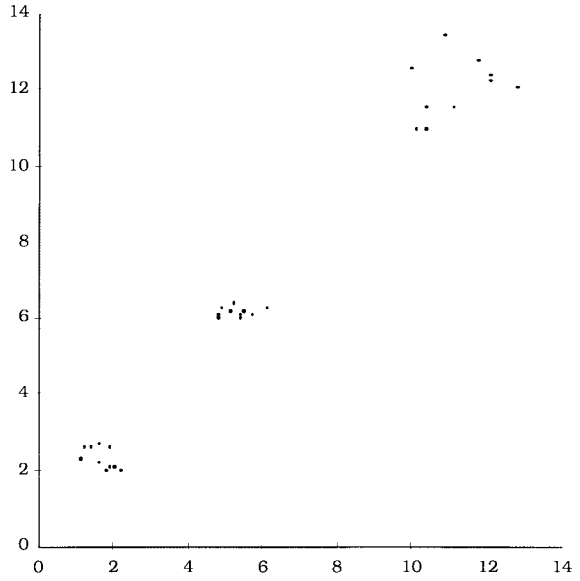


Fig. 2. Data set $\boldsymbol{X}_{30}$.

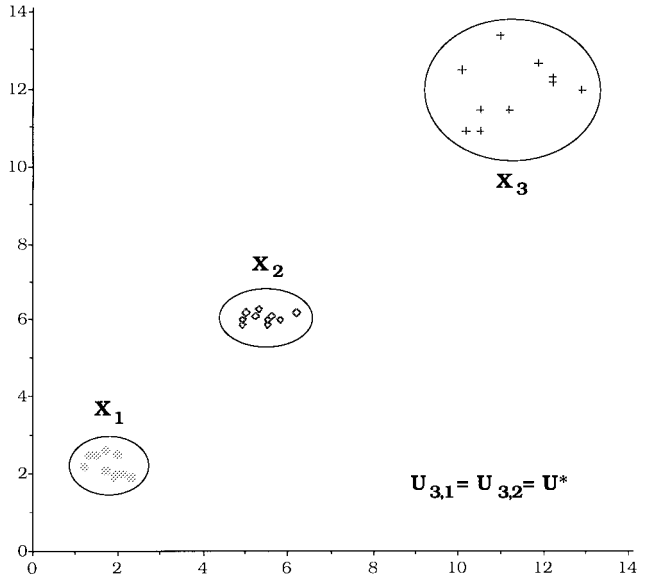


Fig. 3. Terminal HCM clusters in $\boldsymbol{X}_{30}$ for $c = 3$ with two initializations.

Batch Hard c-Means (HCM) Theorem [4]: $(U, \boldsymbol{V}) \in M_{\text{hcn}} \times \Re^{cp}$ *may minimize* J_1 only if

$$u_{ik} = \begin{Bmatrix} 1; & ||\boldsymbol{x}_k - \boldsymbol{v}_i||_A \le ||\boldsymbol{x}_k - \boldsymbol{v}_j||_A, \\ & j = 1 \cdots, c; j \ne i \\ 0; & \text{otherwise} \end{Bmatrix},$$
$$1 \le 1 \le c; 1 \le k \le n; \tag{5a}$$

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{n} (u_{ik}) \boldsymbol{x}_k}{\sum_{k=1}^{n} (u_{ik})} = \frac{\sum_{\boldsymbol{x}_k \in X_i} \boldsymbol{x}_k}{n_i}, \qquad 1 \le i \le c. \tag{5b}$$

Singularities, manifested as ties in (5a), are resolved arbitrarily. Equation (5a) shows that HCM produces crisp partitions of X by assigning all of the membership of each $\boldsymbol{x}_k$ to class i when prototype $\boldsymbol{v}_i$ is nearest to it. The second form for $\boldsymbol{v}_i$ in (5b) emphasizes that it is simply the mean vector of the points currently in crisp cluster i ($n_i = \Sigma_{k=1}^{n} u_{ik}$ is the number of points in the ith row of U—that is, the number of points in the ith cluster X_i in X).

Many validity indexes use the sample means of each subset in crisp partitions of the data, even when the clustering algorithm does not explicitly produce them. For convenience we shall refer to the construction of these vectors from (5b) and any U in M_{hcn} as $\overline{\boldsymbol{V}}(U)$. This notation indicates that the $\boldsymbol{v}_i$'s from (5b) are cluster means, and that they can be computed from (associated with) any U in M_{hcn}, and not just the HCM partition constructed from (5a).

The HCM algorithm is based on iteration through the necessary conditions at (5). This is often called *alternating optimization* (AO) as it simply loops through one cycle of estimates for $\boldsymbol{V}_{t-1} \to U_t \to \boldsymbol{V}_t$ and then checks $||\boldsymbol{V}_t - \boldsymbol{V}_{t-1}||_{\text{err}} \le \varepsilon$. Equivalently, the entire procedure can be shifted one half cycle, so that initialization is done on U_0, and the iterates become $U_{t-1} \to \boldsymbol{V}_t \to U_t$, with the alternate termination criterion $||U_t - U_{t-1}||_{\text{err}} \le \varepsilon$. The literature contains both specifications; the convergence theory is the same in either case. All our computational examples use the protocols shown in Appendix A, and the initial prototypes $\boldsymbol{V}_0$ for each run are c randomly selected distinct data points from X.

Using HCM as just described, we illustrate the need for cluster validation by a simple example. Fig. 2 scatterplots a data set named $\boldsymbol{X}_{30}$ with $n = 30$ points in $\Re^2$.

You may agree that $\boldsymbol{X}_{30}$ has $c = 3$ compact, well-separated clusters of ten points each. We call these three visually attractive clusters X_1, X_2, and X_3 in Fig. 3, where we have marked the points in each cluster with a different symbol and captured them with a crisp boundary. In other words, Fig. 3 corresponds to the (visually) correct crisp labeling of $\boldsymbol{X}_{30}$. The partition of $\boldsymbol{X}_{30}$ corresponding to the labeling in Fig. 3 is

$$U^* = \begin{bmatrix} \underbrace{1 \cdots 1}_{X_1:1-10} & \underbrace{0 \cdots 0}_{X_2:1-20} & \underbrace{0 \cdots 0}_{X_3:21-30} \\ 0 \cdots 0 & 1 \cdots 1 & 0 \cdots 0 \\ 0 \cdots 0 & 0 \cdots 0 & 1 \cdots 1 \end{bmatrix}.$$

We processed $\boldsymbol{X}_{30}$ with HCM six times using two initializations each for $c = 2, 3$, and 4. Fig. 3 shows the terminal clusters of $\boldsymbol{X}_{30}$ obtained by HCM at $c = 3$ from two different initializations. In both cases HCM quickly terminated at the visually correct partition, i.e., $U_{3,1} = U_{3,2} = U^*$ at (5). Here $U_{c,j}$ is a c-partition obtained from initialization j.

Fig. 4 shows terminal clusters obtained by HCM with $c = 2$ and 4 using two different initializations for the prototypes at each of these values. For $c = 2$, X_2 and X_3 merge to form a single cluster in partition $U_{2,1}$. But in partition $U_{2,2}$ X_1 and X_2 are merged instead. For $c = 4$, one initialization of HCM leads to splitting X_3 into two clusters with five points each in partition $U_{4,1}$, while the second initialization leads to splitting X_2 into two five-point clusters in $U_{4,2}$.

Now imagine that, instead of being data in $\Re^2$, unlabeled data set $\boldsymbol{X}_{30}$ is, say, *four-dimensional*. In this case you cannot

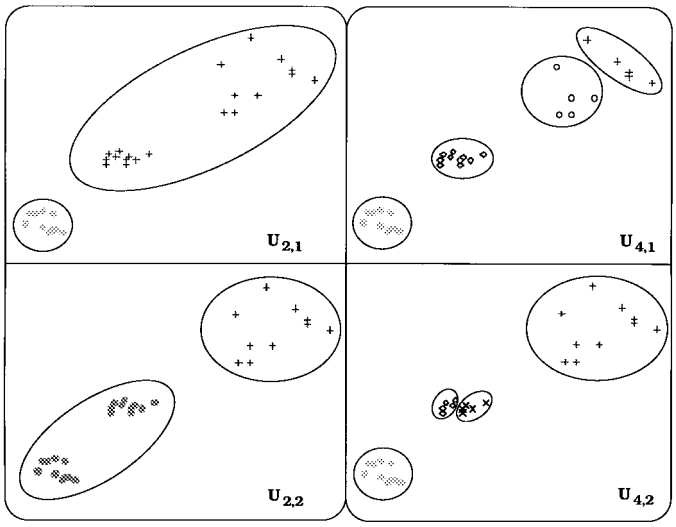


Fig. 4. Terminal HCM clusters in X_{30} with $c = 2, 4$ for two initializations.

examine a scatterplot of the data, so there is no way to know how many clusters to look for, or which points belong to which group. This is true even if the data really have three compact and well-separated clusters. Suppose application of HCM (or any other crisp clustering algorithm) to this hypothetical data set led to five partitions like those in Figs. 3 and 4. How will you choose "the best" one? The wrong choice from among the partitions shown in Figs. 3 and 4 would lead to a very bad interpretation of the data. This is the problem we attack in this paper.

III. The Single Linkage Clustering Algorithm

The second method we use to generate crisp clusters in X is a noniterative method called single linkage [1]. This method is based on a local connectivity criterion, and is usually regarded as a graph-theoretic model, in contrast with objective function models such as HCM at (4). Instead of an object data set X, SL processes sets of (n^2) numerical *relationships*, say $\{r_{jk}\}$, between *pairs* of objects represented by the data. The number r_{jk} represents the extent to which objects j and k are related in the sense of some binary relation ρ. It is convenient to array the relational values as an $(n \times n)$ *relation matrix* $R = [r_{jk}] = [\rho(o_j, o_k)]$. We often call matrix R the relation, even though function ρ is the actual relation. Many functions can convert object data into relational data. For example, every metric (distance measure) d on $\Re^p \times \Re^p$ produces a (dis)-similarity relation matrix $D = [d_{jk}] = [d(\boldsymbol{x}_j, \boldsymbol{x}_k)]$. For dissimilarity relations, low values indicate similar objects, higher values more dissimilar ones.

Single linkage is a special case of the *sequential agglomerative hierarchical nested* (SAHN) model, which is the general name for a family of crisp clustering methods based on the following approach. Our description is limited to the case where similarity is defined by distance. Given $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\} \subset \Re^p$. Choose a (metric) measure of dissimilarity d: $\Re^p \times \Re^p \mapsto \Re^+$ between pairs of points in $\Re^p \times \Re^p$. Each of the object data sets used in our numerical examples was converted to relational data for submission to

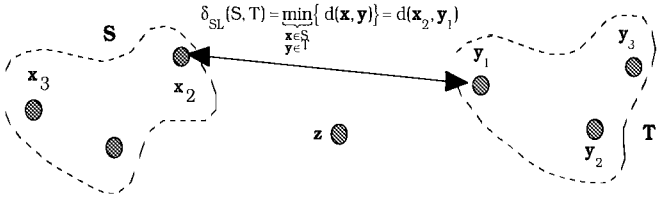


Fig. 5. Intercluster distance for single linkage.

SL by computing d_{jk} as the Euclidean distance between $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$, i.e., $d_{jk} = ||\boldsymbol{x}_j - \boldsymbol{x}_k|| = \sqrt{\boldsymbol{x}_j^T \boldsymbol{x}_k}, 1 \le j, k \le n$. Next, let the power set of $\Re^p$ be denoted by $P(\Re^p)$, and let δ denote any positive semi-definite, symmetric (*set distance*) function on $P(\Re^p) \times P(\Re^p)$. Different linkage models correspond to different choices for δ. For single linkage, this measure of the distance between two subsets S and T of X is the standard distance between a pair of sets, viz., $\delta_{\mathrm{SL}}(S,T) = \underbrace{\min}_{\substack{\boldsymbol{x} \in S \\ \boldsymbol{y} \in T}} \{d(\boldsymbol{x}, \boldsymbol{y})\}$.

Fig. 5 illustrates $\delta_{\mathrm{SL}}(S,T)$ for two sets of three points each. Look at the point $\boldsymbol{z}$ in Fig. 5. If $\boldsymbol{z}$ is included in S or T, $\delta_{\mathrm{SL}}(S,T)$ will be roughly halved. This should convince you that $\delta_{\mathrm{SL}}(S,T)$ is not a reliable measure of the distance between sets when clusters are being sought, because the insertion or deletion of a single point in S or T can radically alter its value. This measure ignores central tendencies in the data, recognizing instead the extreme behavior of bridges (inliers) or outliers. This instability to what may be a very small number of points in the data is one reason that Dunn's index can give misleading validity results.

Now we can describe the SL algorithm. To begin, put $c = n$ so each data point starts out in its own cluster $U_n = I_n$. Compute $D_n = [d_{jk}] = [d(\boldsymbol{x}_j, \boldsymbol{x}_k) = \delta_{\mathrm{SL}}(\boldsymbol{x}_j, \boldsymbol{x}_k)]$, the $n \times n$ symmetric distance matrix for the vectors (which are clusters) in X. In steps beyond this, D_c denotes the $c \times c$ symmetric distance matrix for the clusters in $X, D_c = [\delta_{\mathrm{SL},jk}] = [\delta_{\mathrm{SL}}(X_j, X_k)]$, where X_j and X_k are part (or clusters) of the current c-partition of X. Here are the steps that are repeated to termination at $U_1 = \mathbf{1}_n = X$, i.e., when all points are in one cluster.

1) Search D_c for the nearest pair of clusters in X; find $(s,t) = \underbrace{\operatorname{argmin}}_{j \ne k} \{\delta_{\mathrm{SL}}(X_j, X_k)\}$. Call the distance corresponding to this pair of indexes $\delta_{\min}$.
2) Merge X_s and X_t, labeling the new cluster X_{st}.
3) Update D_c by deleting the rows and columns corresponding to X_s and X_t, and adding a row and column for the distances $\delta_{\mathrm{SL}}(X_{st}, X_q), q \ne st$, between the new cluster X_{st} and the other $(c-2)$ clusters $(X_1 \cup \cdots \cup X_c) - (X_s \cup X_t)$ in X.
4) Repeat steps 1–3 until $c = 1, U_1 = \mathbf{1}_n$, and all n objects belong to the single cluster X.

During this procedure ties are resolved arbitrarily. SL finds *at most* one partition of X at each value of c. U_c and the level of similarity at which mergers occur is recorded at each step. From this information it is customary to construct a visual

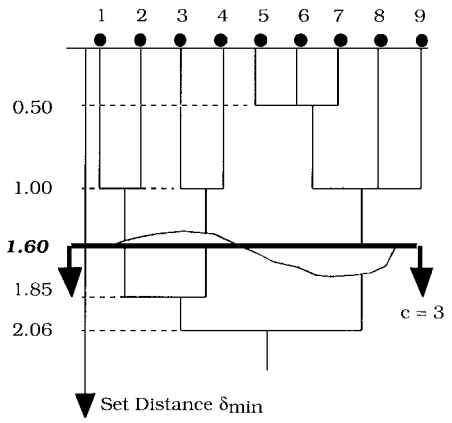


Fig. 6. Single linkage dendogram of clusters in R_9 (and X_9).

display of the results in the form of a dendogram such as the one in Fig. 6, which was made by applying SL to the data set R_9 listed in Appendix B. These relational data are the Euclidean distances between pairs of points in the data set X_9 shown in Appendix B. Fig. 6 exhibits several features of SL clustering. First, the clusters are nested—once merged, points are never split. Second, it is not necessarily the case that a unique partition of the data will be produced at each value of c. In Fig. 6, for example, points 5, 6, and 7 are merged at the same time because their distances are all equal to the minimum ($\delta_{\min} = 0,50$) at this step. Consequently, the first merger apparently reduces c from $c = 9$ to $c = 7$. In the implementation of SL, however, this will happen in two steps at the same merger level, so there will be a partition at $c = 8$, but it is unique only up to the tie-breaking rule used. This is an important point for validity considerations, since the partitions of X at $c = 8$ and $c = 7$ are obviously different, but are equally valid from the point of view of the internal SL criterion.

The cut line shown at $\delta_{\min} = 1.6$ illustrates the general situation at any value of the minimum set distance: $c = 3$ for this value of δ_{SL}. All clusters are merged at 2.06, terminating SL at $U_1 = \mathbf{1}_9$. Since dendograms are useful only for fairly small values of n, we will not show outputs from SL this way in the numerical examples.

Now question Q_3 arises for the clusters associated with Fig. 6: which partition of the nine objects is most valid? The internal method of validity associated with SL is to look for the largest jump $\Delta\delta_{\min}$ in values of $\delta_{\min}$. This is taken as an indicator that the *previous* value of c is most natural, on the presumption that SL works hardest to merge clusters that cause the biggest jump. Note that the biggest jump can be severely influenced by the presence of a few outliers. In Fig. 6, successive jumps are 0.50, 0.50, 0.85, and 0.21. The largest jump, (0.85 from $c = 3$ to $c = 2$) identifies $X = \{1,2\} \cup \{3,4\} \cup \{5,6,7,8,9\}, c = 3$ clusters at $\delta_{\min} = 1.00$, as the most natural ones. Fig. 11 in Appendix B seems to confirm this visually, although a case can be made that $c = 2$ is just as natural. The real point is that, just as in HCM, the criterion that helps find the clusters ($\Delta\delta_{\min}$ here, J_1 for HCM) may or may not also indicate the best ones amongst various candidates generated by the algorithm. This is the reason cluster validity is an important problem.

HCM and SL are known to work best on data structures that have very different properties. HCM with the Euclidean norm performs well when clusters are roughly hyperspherical, well separated, and have nearly equal subsample sizes. SL likes to find well separated stringy clusters such as points along a pair of parallel roads. This behavior is discussed in [3, ch. 6]. We mention this to advertise the fact that our choice of clustering algorithms was quite deliberate. The two algorithms chosen may find very different partitions of the same data at the same value for c. This is good when looking for ways to validate partitions, since useful validity measures should also point to bad partitions when an algorithm finds them.

IV. Three Cluster Validity Methods

How many validation methods for crisp partitions are there? Thirteen years ago Hubert and Arabie began a paper on this topic by saying "We will not try to review this literature comprehensively since that task would require the length of a monograph" [9, p. 193]. Since it is not feasible to attempt a comprehensive comparison of our generalized Dunn's indexes with many others, we have instead chosen three of the better known indexes for this purpose. These three measures have rather different properties and rationales, and should serve as an adequate basis for evaluating our generalizations of Dunn's index.

Modified Hubert's statistic (MH): Hubert's Γ statistic [9] assesses the fit between the data and any crisp structure imposed on it by U in M_{hcn}. Basically then, the rationale underlying this measure is a statistical goodness-of-fit test. Let $P = [p_{ij}]$ be an $n \times n$ proximity matrix; p_{ij} is the observed proximity between objects i and j (for example, $p_{ij} = ||\boldsymbol{x}_i - \boldsymbol{x}_j||$ in any norm). $Q = [q_{ij}]$ is an $n \times n$ matrix defined in terms of any hard c-partition U of X

$$|Q(U)]_{ij} = q_{ij} = \begin{Bmatrix} 0, & u_{ki} = u_{kj} = 1 \ \exists \text{ class } k \\ 1, & \text{otherwise} \end{Bmatrix}. \quad (6)$$

Hubert's Γ statistic is the point serial correlation coefficient between any two matrices. When the two matrices are symmetric, Γ can be written in its raw form as

$$\Gamma(P, Q(U)) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} p_{ij} q_{ij}. \quad (7)$$

In its normalized form, Γ becomes the sample correlation coefficient between the entries of P and Q

$$\hat{\Gamma}(P, Q(U)) = \left\{ \frac{1}{M} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (p_{ij} - \overline{p})(q_{ij} - \overline{q}) / (s_P s_Q) \right. \quad (8)$$

where $M = n(n-1)/2$ is the total number of entries under the double summation,

$$\overline{p} = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} p_{ij}; \quad \overline{q} = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} q_{ij};$$

$$s_P^2 = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} p_{ij}^2 - \overline{p}^2 \quad \text{and}$$

$$s_q^2 = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} q_{ij}^2 - \overline{q}^2.$$

For the normalized index $-1 \le \hat{\Gamma} \le 1$. If P and Q are not symmetric then all summations are extended over all n^2 entries and $M = n^2$. $\hat{\Gamma}$ measures the degree of linear correspondence between the entries of P and Q. A positive value of $\hat{\Gamma}$ close to 1 indicates that P and Q are (more or less) linearly correlated.

For cluster validation we can use Γ *or* $\hat{\Gamma}$ to test whether the association between P and Q is *unusually* large under the *random label hypothesis* (RLH), which is:

RLH: *All permutations of row (and column) labels of Q are equally likely*.

We want to test whether Γ *or* $\hat{\Gamma}$ can be obtained by a chance labeling. Although the value of Γ or $\hat{\Gamma}$ gives some information about the match between P and Q, the distribution of Γ *or* $\hat{\Gamma}$ under the RLH is needed to decide whether $Q(U)$ matches the actual proximity matrix unusually well. The distribution of Γ or $\hat{\Gamma}$ can be found by computing it for all $n!$ permutations and then finding its histogram. But this method is computationally prohibitive. (For example, a data set with ten objects yields 3 628 800 values!)

Other alternatives include approximation of the distribution of Γ or $\hat{\Gamma}$ by Monte Carlo methods, and computation of the mean and standard deviation under the RLH, assuming that the underlying distribution is normal. For the second method, of course, an explicit expression for the moments are required. For these reasons, a more tractable form of Hubert's statistic, called the *modified Hubert's statistic* (MH) is usually used for cluster validation. The modified statistic abandons the goodness of fit strategy, and replaces it with a geometric method that is based on intuitively natural principles.

Let $L(i) = k$ if the ith object is in the kth cluster. Let $||\boldsymbol{v}_i - \boldsymbol{v}_j||_2$ be the Euclidean distance between the cluster centers $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ in $\overline{\boldsymbol{V}}(U)$ computed by (5b) for any U in M_{hcn}. Now define the $n \times n$ matrix $Q(U, \overline{\boldsymbol{V}}(U))$ as

$$Q(U, \overline{\boldsymbol{V}}(U)) = [q_{(U,\overline{\boldsymbol{V}}(U),ij}] = [||\boldsymbol{v}_{L(i)} - \boldsymbol{v}_{L(j)}||_2], \quad i,j = 1,2,\cdots,n. \tag{9}$$

Using (9) instead of (6) in (7) and (8) yields

$$\mathcal{V}_{\text{MH}\Gamma}(U, \overline{\boldsymbol{V}}(U)) = \Gamma(P, Q(U, \overline{\boldsymbol{V}}(U)) \quad \text{[Hubert's modified raw statistic]; and} \tag{10a}$$

$$\mathcal{V}_{\text{MH}\hat{\Gamma}}(U, \overline{\boldsymbol{V}}(U)) = \hat{\Gamma}(P, Q(U, \overline{\boldsymbol{V}}(U)) \quad \text{[Hubert's modified normalized statistic]}. \tag{10b}$$

It is known from computational experience that these indexes tend to increase with an increase of c. They are not defined on $\mathbf{1}_n$ when $c = 1$ and $\mathcal{V}_{\text{MH}\hat{\Gamma}}(I_n) = 1$ for the trivial clustering of X at $c = n$. Because of this, it is not the *value* of $\mathcal{V}_{\text{MH}\Gamma}$ or $\mathcal{V}_{\text{MH}\hat{\Gamma}}$ that is used to choose c; rather it is the *change in the value* as a function of c that is examined. For well separated clusters, a sharp knee (cf., Fig. 10) is expected at the partition $U_i(c)$ which contains the number of clusters that provide the best fit to the data as measured by this statistic. This strategy is like examination of $\Delta\delta_{\min}$ as discussed in Section III in connection with validation of SL partitions.

Davies–Bouldin Index: This index is a function of the ratio of the sum of within-cluster scatter to between-cluster separation [10], and like Hubert's measure, it also uses both the clusters and their sample means $\overline{\boldsymbol{V}}(U)$. Since scatter matrices depend on the geometry of the clusters, this index has both a statistical and geometric rationale. Define the *within ith cluster scatter* and the *between ith and jth cluster distance* as

$$S_{i,q} = \left(\frac{1}{|X_i|}\sum_{\boldsymbol{x}\in X_i} ||\boldsymbol{x} - \boldsymbol{v}_i||_2^q\right)^{1/q} \tag{11}$$

and

$$d_{ij,t} = \left\{\sum_{s=1}^{p} |v_{si} - v_{sj}|^t\right\}^{1/t} = ||\boldsymbol{v}_i - \boldsymbol{v}_j||_t. \tag{12}$$

For a given U in M_{hcn}, $\boldsymbol{v}_i$ is the vector at (5b), $q, t \ge 1$, q is an integer and q, t can be selected independently of each other. $S_{i,q}$ is the qth root of the qth moment of the points in cluster i with respect to their mean, and is a measure of dispersion of the points in cluster i. $S_{i,1}$ is the average Euclidean distance of the vectors in class i to the centroid of class i. $S_{i,2}$ is the square root of the mean square error of the points in the ith cluster with respect to the centroid of the ith class, and so on. $d_{ij,t}$ is the Minkowski distance of order t between the centroids which characterize clusters i and j. Next, define

$$R_{i,qt}(U, \overline{\boldsymbol{V}}(U)) = \max_{j, j\neq i}\left\{\frac{S_{i,q}(U) + S_{j,q}(U)}{d_{ij,t}(U)}\right\}. \tag{13}$$

Now the *Davies–Bouldin* (DB) index can be defined as

$$\mathcal{V}_{\text{DB},qt}(U, \overline{\boldsymbol{V}}(U)) = \frac{1}{c}\sum_{i=1}^{c} R_{i,qt}(U). \tag{14}$$

It is geometrically plausible to seek clusters that have minimum within-cluster scatter [the numerator in (13)] and maximum between-class separation [the denominator in (13)], so the number of clusters c^* that *minimizes* $\mathcal{V}_{\text{DB},qt}$ is taken as the optimal value of c. $\mathcal{V}_{\text{DB},qt}$ is not defined on $\mathbf{1}_n$ when $c = 1$. For well-separated clusters $\mathcal{V}_{\text{DB},qt}$ is expected to decrease monotonically as c increases until the correct number of clusters is achieved (however, $\mathcal{V}_{\text{DB},qt(\text{In})} = 0$). $\mathcal{V}_{\text{DB},qt}$ is easier to use than $\mathcal{V}_{\text{MH}\Gamma}$ or $\mathcal{V}_{\text{MH}\hat{\Gamma}}$ because finding the minimum of $(c_{\max} - 1)$ values is less ambiguous than finding a knee or sharp change in slope in the piece wise linear graph that connects them.

Dunn's Indexes: Dunn's index (DI) is based on geometrical considerations that have the same basic rationale as the DBI in

TABLE I
FOUR CRISP CLUSTER VALIDITY INDEXES

Validity Index	Name	Variables	On { 2, 3, ..., c_{max}}
Modified Hubert Statistic (Raw)	$\nu_{\text{MH}\Gamma}$	(X; U, $\overline{\mathbf{V}}$(U))	Look for sharp knee
Modified Hubert Statistic (Norm.)	$\nu_{\text{MH}\hat{\Gamma}}$	(X; U, $\overline{\mathbf{V}}$(U))	Look for sharp knee
Davies-Bouldin	$\nu_{\text{DB,qt}}$	(X; U, $\overline{\mathbf{V}}$(U))	Minimize
Dunn's CS Index	ν_{D}	(X; U)	Maximize

that they are both designed to identify sets of clusters that are compact and well separated [7]. To understand this index let S and T be non-empty subsets of $\Re^p$, and let d: $\Re^p \times \Re^p \mapsto \Re^+$ be any metric. The standard definitions of the *diameter* Δ of S and the *set distance* δ between S and T are

$$\Delta(s) = \underbrace{\max}_{\boldsymbol{x},\boldsymbol{y} \in S} \{d(\boldsymbol{x},\boldsymbol{y})\} \tag{15}$$

and

$$\delta(S,T) = \underbrace{\min}_{\substack{\boldsymbol{x} \in S \\ \boldsymbol{y} \in T}} \{d(\boldsymbol{x},\boldsymbol{y})\} = \delta_{\text{SL}}(S,T). \tag{16}$$

In (16), we emphasize that the standard distance between S and T is just the distance illustrated in Fig. 5 in connection with our discussion on the SL algorithm. For any partition $U \leftrightarrow X = X_1 \cup \cdots X_i \cup \cdots X_c$, Dunn defined the *separation index* of U as

$$\mathcal{V}_D(U) = \underbrace{\min}_{1 \le i \le c} \left\{ \underbrace{\min}_{\substack{1 \le j \le c \\ j \ne i}} \left\{ \frac{\delta(X_i, X_j)}{\underbrace{\max}_{1 \le k \le c} \{\Delta(X_k)\}} \right\} \right\}. \tag{17}$$

The quantity $\delta(X_i, X_j)$ in the numerator of $\mathcal{V}_D$ is analogous to $d_{ij,t}$ in the denominator of $\mathcal{V}_{\text{DB},qt}$; the former measures the distance between clusters directly on the points in the clusters, whereas the latter uses the distance between their cluster centers in $\overline{\boldsymbol{V}}(U)$ for the same purpose. The use of $\Delta(X_k)$ in the denominator of (17) is analogous to $S_{k,q}$ in the numerator of $\mathcal{V}_{\text{DB},qt}$; both are measures of scatter volume for cluster X_k. Thus, extrema of $\mathcal{V}_D$ and $\mathcal{V}_{\text{DB},qt}$ share roughly the same geometric objective: maximizing intercluster distances whilst minimizing intracluster distances. Since the measures of separation and compactness in (17) occur "upside down" from their appearance in (13), *large* values of $\mathcal{V}_D$ correspond to good clusters. Hence, the number of clusters c^* that *maximizes* $\mathcal{V}_D$ is taken as the optimal value of c. $\mathcal{V}_D$ is not defined on $\mathbf{1}_n$ when $c = 1$ or on I_n when $c = n$.

Dunn called U *compact and separated* (CS) relative to d if and only if the following property is satisfied: for all s, q, and r with $q \ne r$, any pair of points $\boldsymbol{x}, \boldsymbol{y} \in X_s$ are closer together (with respect to d) than any pair $\boldsymbol{u}, \boldsymbol{v}$ with $\boldsymbol{u} \in X_q$ and $\boldsymbol{v} \in X_r$. Dunn proved that X can be clustered into a compact and separated c-partition with respect to d if and only if $\underbrace{\max}_{U \in M_{\text{hcn}}} \{\mathcal{V}_D(U)\} > 1$. Dunn defined a second index of separation for *compact and well separated* (CWS) clusters. He called a partition CWS with respect to d if and only if the following property is satisfied: for all s, q, and r with $q \ne r$, any pair of points $\boldsymbol{x}, \boldsymbol{y}$ with $\boldsymbol{x} \in X_s, \boldsymbol{y} \in \text{conv}(X_s)$ are closer together as measured by d than any pair $\boldsymbol{u}, \boldsymbol{v}$ with $\boldsymbol{u} \in X_q$ and $\boldsymbol{v} \in \text{conv}(X_r)$, where $\text{conv}(S)$ is the *convex hull* of S in $\Re^p$. Dunn's index for CWS clusters is obtained by replacing X_j in (17) with $\text{conv}(X_j)$ as

$$\mathcal{V}_{\hat{D}}(U) = \underbrace{\max}_{1 \le i \le c} \left\{ \underbrace{\min}_{\substack{1 \le j \le c \\ j \ne i}} \left\{ \frac{\delta(X_i, \text{conv}(X_j))}{\underbrace{\max}_{1 \le k \le c} \{\Delta(X_k)\}} \right\} \right\}. \tag{18}$$

Dunn proved that X can be partitioned into CWS clusters relative to d if and only if $\underbrace{\max}_{U \in M_{\text{hcn}}} \{\mathcal{V}_{\hat{D}}(U)\} > 1$. $\mathcal{V}_{\hat{D}}$ sets very attractive geometrical requirements for good CWS clusters. However, estimation of $\text{conv}(X_j)$ for even $p = 2$ is very difficult computationally, so $\mathcal{V}_{\hat{D}}$ finds little use in practice and will not be considered further here. Table I summarizes the indexes discussed in this section.

V. GENERALIZATION OF DUNN'S INDEX

The numerator and denominator of $\mathcal{V}_D$ are both overly sensitive to changes in cluster structure. We have already illustrated the problem for δ_{SL} in Fig. 5: this measure of interset distance can be dramatically altered by the addition or deletion of a single point in either S or T. The denominator suffers from the same problem—for example, adding one point to S can easily scale $\Delta(S)$ by an order of magnitude. Consequently, $\mathcal{V}_D$ can be greatly influenced by a few noisy points (that is, outliers or inliers to the main cluster structure) in X, and is thus very sensitive to what can be a very small minority in the data. However, (17) provides a very general paradigm for defining cluster validity indexes. Appropriate definitions of δ and Δ lead to validity indexes suitable for different types (e.g., clouds or shells) of clusters.

Our objective in formulating generalizations of Dunn's index here is to ameliorate its sensitivity to aberrant data for the case when clusters are expected to form volumetric clouds (as opposed to boundaries, shells or surfaces) in the feature space. There are several principles that can be used as guides. First, *all of the data* should be explicitly involved in the computation of the index. And second, most of the better indexes also use the cluster means $\overline{\boldsymbol{V}}(U)$ in their definition (cf., Table I—only Dunn's index does not). Using $\overline{\boldsymbol{V}}(U)$ implicitly involves all of X, and further insulates indexes from being brittle to a few points in the data.

$\mathcal{V}_D$ can be generalized by using other definitions for the diameter of a set at (15) or the distance between sets at (16). Let $P(\Re^p)$ denote the power set of $\Re^p, \delta_i$ denote any positive semi-definite, symmetric (*set distance*) function on $P(\Re^p) \times P(\Re^p)$, and Δ_j be any positive semi-definite (*diameter*) function on $P(\Re^p)$. The general form of $\mathcal{V}_D$ using δ_i and Δ_j is

$$\mathcal{V}_{GD}(U) \doteq \mathcal{V}_{\delta_i \Delta_j}(U) = \underbrace{\min}_{1 \le s \le c} \left\{ \underbrace{\min}_{\substack{1 \le t \le c \\ t \ne s}} \left\{ \frac{\delta_i(X_s, X_t)}{\underbrace{\max}_{1 \le k \le c} \{\Delta_j(X_k)\}} \right\} \right\}. \quad (19)$$

Let S and T be finite non empty elements of $P(\Re^p)$, and let d: $\Re^p \times \Re^p \mapsto \Re^+$ be any metric. Five set distance functions that can be used in (19) are

$$\delta_1(S,T) = \delta_{\min}(S,T) = \underbrace{\min}_{\substack{\boldsymbol{x} \in S \\ \boldsymbol{y} \in T}} \{d(\boldsymbol{x}, \boldsymbol{y})\} = \delta_{\mathrm{SL}}(S,T) \quad (20)$$

$$\delta_2(S,T) = \delta_{\max}(S,T) = \underbrace{\max}_{\substack{\boldsymbol{x} \in S \\ \boldsymbol{y} \in T}} \{d(\boldsymbol{x}, \boldsymbol{y})\} = \delta_{\mathrm{CL}}(S,T) \quad (21)$$

$$\delta_3(S,T) = \delta_{\mathrm{avg}}(S,T) = \frac{1}{|S||T|} \sum_{\substack{\boldsymbol{x} \in S \\ \boldsymbol{y} \in T}} d(\boldsymbol{x}, \boldsymbol{y}) = \delta_{\mathrm{AL}}(S,T) \quad (22)$$

$$\delta_4(S,T) = d(\boldsymbol{v}_S, \boldsymbol{v}_T)$$

where

$$\boldsymbol{v}_S = \frac{1}{|S|} \sum_{\boldsymbol{x} \in S} \boldsymbol{x}$$

and

$$\boldsymbol{v}_T = \frac{1}{|T|} \sum_{\boldsymbol{y} \in T} \boldsymbol{y} \quad (23)$$

$$\delta_5(S,T) = \frac{1}{|S| + |T|} \left(\sum_{\boldsymbol{x} \in S} d(\boldsymbol{x}, \boldsymbol{v}_T) + \sum_{\boldsymbol{y} \in T} d(\boldsymbol{y}, \boldsymbol{v}_S) \right). \quad (24)$$

In (23) and (24), $\boldsymbol{v}_S$ and $\boldsymbol{v}_T$ are computed with (5b). Function δ_1 at (20) is the same as (16). Functions δ_2 and δ_3 correspond, respectively, to the set distance functions used in the *complete linkage* (CL) and *average linkage* (AL) clustering algorithms [1]. When $\mathcal{V}_{\delta\Delta}$ uses either δ_1 or δ_2 it may be strongly affected by noisy points because neither δ_1 nor δ_2 uses all the points in S and T. Although single and complete linkage share this property, complete linkage is often preferred. Sneath and Sokal [11] assert that complete linkage generally finds tight, hyperspherical, clusters that join others only with difficulty and at relatively low overall similarity values. Jain and Dubes [1] state that complete linkage produces more useful hierarchies in many applications than the single linkage method. These remarks encourage us to speculate that $\mathcal{V}_{\delta_2\Delta}$ will be more useful for validation than $\mathcal{V}_{\delta_1\Delta}$ when clusters form volumetric clouds. Moreover, we expect $\mathcal{V}_{\delta_3\Delta}$, which uses the average of *all* interpoint distances between S and T to be more effective than either $\mathcal{V}_{\delta_2\Delta}$ or $\mathcal{V}_{\delta_1\Delta}$.

δ_4 depends implicitly on every point in S and T through $\boldsymbol{v}_S$ and $\boldsymbol{v}_T$, so the effect of adding or deleting points to or from S or T is ameliorated by averaging. As the number of points in S or T increases, averaging will decrease the sensitivity of δ_4 to a few aberrant data. Moreover, δ_4 has a lower computational overhead than $\delta_1 - \delta_3$. δ_5 is a set distance that combines the averaging concept of δ_3 with the prototype idea of δ_4.

$\delta_1 - \delta_5$ can be used as set distance functions, but none are metrics on $P(X)$. The sixth set distance we propose is the well known Hausdorff metric [12]

$$\delta_6(S,T) = \delta_{\mathrm{Hausdorff}}(S,T) = \max\{\delta(S,T), \delta(T,S)\}$$

where

$$\delta(S,T) = \underbrace{\max}_{\boldsymbol{x} \in S} \{\underbrace{\min}_{\boldsymbol{y} \in T} \{d(\boldsymbol{x}, \boldsymbol{y})\}\}$$
$$\delta(T,S) = \underbrace{\max}_{\boldsymbol{y} \in T} \{\underbrace{\min}_{\boldsymbol{x} \in S} \{d(\boldsymbol{x}, \boldsymbol{y})\}\}. \quad (25)$$

We expect δ_6 to be relatively insensitive to noisy points. It is easy to see that when the same metric d is used in (20), (21), and (25), $\delta_1 \le \delta_6 \le \delta_2$. Notice also that each of these functions can use any metric d, so there are an infinite number of realizations for each one.

$\Delta(S)$ at (15) used by Dunn is the standard diameter of the set S. As previously mentioned, this makes $\Delta(S)$ very sensitive to noisy points. We repeat (15) as (26), now indexed for convenience, and then give two additional definitions for functions related to diameters that are useful in defining measures of cluster validity

$$\Delta_1(S) = \mathrm{diam}(S) = \underbrace{\max}_{\boldsymbol{x}, \boldsymbol{y} \in S} \{d(\boldsymbol{x}, \boldsymbol{y})\} \quad (26)$$

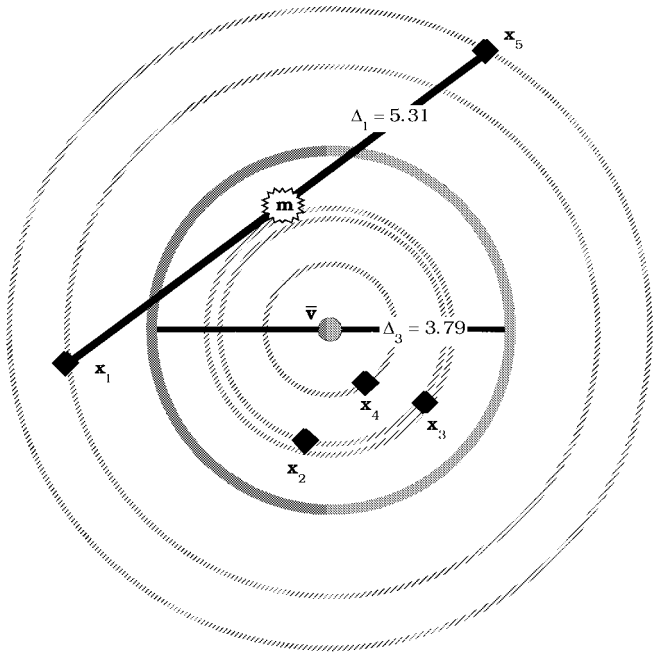
$$\Delta_2(S) = \frac{1}{|S| \cdot (|S| - 1)} \sum_{\substack{\boldsymbol{x}, \boldsymbol{y} \in S \\ \boldsymbol{x} \ne \boldsymbol{y}}} d(\boldsymbol{x}, \boldsymbol{y}) \quad (27)$$

$$\Delta_3(S) = 2 \left(\frac{\sum_{\boldsymbol{x} \in S} d(\boldsymbol{x}, \overline{\boldsymbol{v}})}{|S|} \right)$$

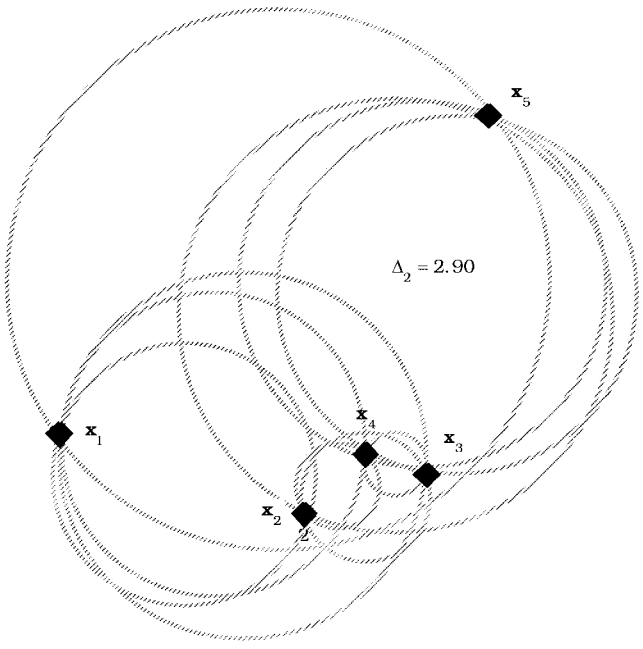
where

$$\overline{\boldsymbol{v}} = \frac{1}{|S|} \sum_{\boldsymbol{x} \in S} \boldsymbol{x}. \quad (28)$$

Fig. 7 depicts the geometric meaning of these three set functions on the set of five data points in $\Re^2$ the coordinates of which are $\boldsymbol{x}_1 = (0,1), \boldsymbol{x}_2 = (2,0), \boldsymbol{x}_3 = (3,0.5), \boldsymbol{x}_4 = (2.5, 0.75)$, and $\boldsymbol{x}_5 = (3.5, 5)$. Distances in this example are Euclidean. Fig. 7(a) shows the distance $\Delta_1 = 5.31$ from $\boldsymbol{x}_1$ to $\boldsymbol{x}_5$. This is traditionally called the diameter of the set of points, but it is not clear what circle it would be the diameter of, for there is no "centering" concept attached to the calculation in (26). A circle of *radius* Δ_1 centered at any point in X will capture all of its points. The circle of diameter Δ_1 centered at



(a)



(b)

Fig. 7. (a) Illustration of the set functions Δ_1 and Δ_3. (b) Illustration of the set function Δ_2.

$\boldsymbol{m}$, the midpoint of the vector connecting the points $\boldsymbol{x}$ and $\boldsymbol{y}$ that solve (26) may not contain any other point in X.

Also shown in Fig. 7(a) is $\Delta_3 = 3.79$, the *average* diameter of the five circles centered at the mean vector $\overline{\boldsymbol{v}} = (2.20, 1.45)$ that pass through these five points with radii as in the numerator of (28). The multiplier of 2 in (28) is used to convert each radius to a diameter. Fig. 7(a) should convince you that $\Delta_1 = \Delta_3$ only if the data are symmetric with respect to the mean vector $\overline{\boldsymbol{v}}$.

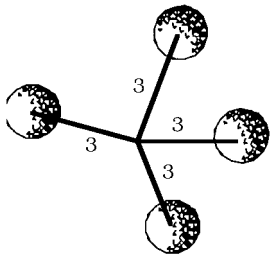


Fig. 8. Schematic illustration of **Normal** 4×4.

Fig. 7(b) shows the ten circles that are associated with the ten distances $\{d(\boldsymbol{x}, \boldsymbol{y})\}$ in (27), which is the average of the ten *diameters* defined by circles centered at the midpoint of the vectors $\{(\boldsymbol{x} - \boldsymbol{y})\}$. Division by 2 to correct for symmetry in $d(\boldsymbol{x}, \boldsymbol{y})$ in (27) is *not done* so that the ten radii computed in the numerator are diameters instead. As with (26), this set function does not have a centering concept, so it is difficult to draw the circle with diameter $\Delta_2 = 2.90$ on Fig. 7(b).

Of the three measures of set size Δ_3 is probably the most reliable for cluster validation because it is the average of the diameters of the smallest hyperspheres (centered at $\overline{\boldsymbol{v}}$) that include the points in the cluster. As seen in Fig. 7(a), the hypersphere of diameter Δ_3 centered at $\overline{\boldsymbol{v}}$ may not contain all points in the cluster. Δ_1 and Δ_2 do not use the cluster centroid $\overline{\boldsymbol{v}}$. Of the two, we expect Δ_1 to work better when $\overline{\boldsymbol{v}}$ is near the middle [$\boldsymbol{m}$ in Fig. 7(a)] of the line joining the two farthest points in the set. In this case the hypersphere with diameter Δ_1 centered at $\overline{\boldsymbol{v}}$ may contain most of the points in the set. However, in the presence of outliers (noisy points) this is not likely to happen and Δ_2 and Δ_3 will be more stable than Δ_1 because averaging has a smoothing effect on both of these measures of dispersion. Our intuition based on the geometry of Fig. 7 is that Δ_3 will provide the best performance, for tight, well formed clouds of points; and for this case, Δ_2 will probably be the least effective measure of set size.

VI. Data Sets and Computational Protocols

Data sets: Six data sets are used in our examples. First we consider $\boldsymbol{X}_{30}$, plotted in Fig. 2. Second, we will use **Iris**, the ubiquitous $n = 150$ points in $\Re^4$ that are divided into $c = 3$ (*physically labeled*) clusters of 50 points each [13]. Our third data set contains $n = 800$ points consisting of 200 points each drawn from the four components of a mixture of $c = 4, p = 4$-variate normal distributions. The population mean vector and covariance matrix for each component of the normal mixture were $\mu_i = 3\boldsymbol{e}_i$ and $\Sigma_i = I_4, i = 1, 2, 3, 4$. We call this data set **Normal** $\boldsymbol{4 \times 4}$. Fig. 8 depicts what **Normal** $\boldsymbol{4 \times 4}$ might look like if it could be seen in three dimensions and if the sampling of each component produced very compact clusters. Because the standard deviation of each population component is 1, we can expect about 68.2% of each 200 samples to be within one unit of their mean.

To study the efficacy of validation with these indexes, we transformed **Normal** $\boldsymbol{4 \times 4}$ three times, creating data sets $\boldsymbol{X}_4, \boldsymbol{X}_5$, and $\boldsymbol{X}_6$ from it by subtracting, respectively, 1, 2, or 2.5 from every value in **Normal** $\boldsymbol{4 \times 4}$. This moves the clusters in **Normal** $\boldsymbol{4 \times 4}$ successively closer together, creating

TABLE II
VALIDITY INDEXES FOR U_{HCM} PARTITIONS OF $\boldsymbol{X}_{30}$

c	2 : $U_{2,2}$	3 : U^*	4 : $U_{4,1}$	5	6	7	8	9	10
$\mathcal{V}_{\mathrm{MH\Gamma}}$	53.77	**63.10**	63.70	63.75	63.76	63.82	63.83	63.84	63.85
$\mathcal{V}_{\mathrm{MH\hat{\Gamma}}}$	0.85	**0.99**	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$\mathcal{V}_{\mathrm{DB},22}$	0.35	**0.18**	0.48	0.63	0.79	0.87	0.82	0.88	0.82
$\mathcal{V}_{11}=\mathcal{V}_{\mathrm{D}}$	0.96	**1.53**	0.52	0.12	0.04	0.04	0.04	0.04	0.04
$\mathcal{V}_{21}$	**2.38**	2.20	1.24	0.49	0.28	0.28	0.28	0.21	0.21
$\mathcal{V}_{31}$	1.72	**1.82**	0.89	0.31	0.17	0.17	0.17	0.16	0.16
$\mathcal{V}_{41}$	1.72	**1.81**	0.79	0.29	0.15	0.15	0.15	0.11	0.11
$\mathcal{V}_{51}$	1.72	**1.82**	0.84	0.30	0.16	0.16	0.16	0.14	0.14
$\mathcal{V}_{61}$	**1.95**	1.92	1.01	0.36	0.22	0.22	0.22	0.21	0.21
$\mathcal{V}_{12}$	2.02	**2.75**	1.14	0.26	0.09	0.09	0.09	0.09	0.09
$\mathcal{V}_{22}$	**4.99**	3.93	2.72	1.09	0.63	0.63	0.63	0.47	0.47
$\mathcal{V}_{32}$	**3.60**	3.26	1.95	0.69	0.38	0.38	0.38	0.34	0.34
$\mathcal{V}_{42}$	**3.59**	3.25	1.74	0.63	0.32	0.32	0.32	0.25	0.25
$\mathcal{V}_{52}$	**3.60**	3.25	1.85	0.66	0.36	0.36	0.36	0.32	0.32
$\mathcal{V}_{62}$	**4.09**	3.44	2.23	0.79	0.48	0.48	0.48	0.47	0.47
$\mathcal{V}_{13}$	1.16	**1.94**	0.95	0.22	0.08	0.08	0.08	0.08	0.08
$\mathcal{V}_{23}$	**2.87**	2.79	2.26	0.90	0.52	0.52	0.52	0.39	0.39
$\mathcal{V}_{33}$	2.07	**2.31**	1.62	0.57	0.32	0.32	0.32	0.28	0.28
$\mathcal{V}_{43}$	2.06	**2.30**	1.45	0.53	0.27	0.27	0.27	0.21	0.21
$\mathcal{V}_{53}$	2.06	**2.30**	1.54	0.55	0.30	0.30	0.30	0.26	0.26
$\mathcal{V}_{63}$	2.35	**2.43**	1.85	0.66	0.40	0.40	0.40	0.39	0.39

more and more overlap as the clusters become less and less well separated. HCM and SL will encounter more and more difficulty in finding good clusters as we move from $\boldsymbol{X}_3$ to $\boldsymbol{X}_6$. This in turn provides a successively more difficult test for the validation indexes.

Computing Protocols: The metric used in all our algorithms wherever a vector distance is called for is Euclidean distance, $d(\boldsymbol{x},\boldsymbol{y}) = ||\boldsymbol{x}-\boldsymbol{y}|| = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^T(\boldsymbol{x}-\boldsymbol{y})}$. Other protocols for FCM are given in Appendix A. Single linkage was executed as specified in Section III. For $\mathcal{V}_{\mathrm{DB},qt}$, we used $q=2$ and $t=2$ in all computations. Each column of Tables II–VII is computed by applying the 21 indexes to the same crisp c-partition of X.

A very important point to remember in the examples that follow is that we show validity function outputs rounded to only two significant digits to keep the tables to a reportable size. Consequently, many comparisons we draw seem to ignore what look like ties. However, *there were NO TIES in the outputs at six significant digits*, so when we report an optimal value for some index, it is always with respect to the original values, not their rounded-off equivalents that appear in the tables below.

VII. NUMERICAL EXAMPLES

Example 7.1A: $\boldsymbol{X}_{30}$ with HCM: Table II shows values of the 21 validity indexes for HCM partitions of $\boldsymbol{X}_{30}$ at each value of c for $c=2$ to $c=10$. The partitions for columns at $c=2,3$, and 4 in Table II are those identified as $U_{2,2}, U_{3,1} = U_{3,2} = U^*$, and $U_{4,1}$ in Figs. 3 and 4.

In Table II and others to follow the highlighted (**bold and shaded**) entries correspond to optimal values of the indexes. The optimal values highlighted in the tables are determined using the *unrounded* values of the indexes, which, to remind you again, did not have ties at six digit accuracy. For example, our roundoff policy renders the values for $\mathcal{V}_{\mathrm{MH\hat{\Gamma}}}$ from $c=4$ to $c=10$ identical in Table II; there were slight differences at six-digit accuracy.

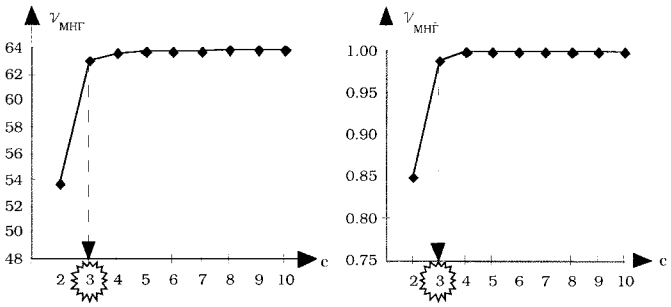


Fig. 9. Plot of $\mathcal{V}_{\mathrm{MH\Gamma}}$ and $\mathcal{V}_{\mathrm{MH\hat{\Gamma}}}$ for U_{HCM} of $\boldsymbol{X}_{30}$.

TABLE III
VALIDITY INDEXES FOR U_{SL} PARTITIONS OF $\boldsymbol{X}_{30}$

c	2 : $U_{2,2}$	3 : U^*	4 : $U_{4,1}$	5	6	7	8	9	10
$\mathcal{V}_{\mathrm{MH\Gamma}}$	53.77	**63.10**	63.70	63.79	63.86	63.88	63.92	63.94	63.94
$\mathcal{V}_{\mathrm{MH\hat{\Gamma}}}$	0.85	**0.99**	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$\mathcal{V}_{\mathrm{DB},22}$	0.35	**0.18**	0.48	0.36	0.31	0.27	0.25	0.21	0.14
$\mathcal{V}_{11}=\mathcal{V}_{\mathrm{D}}$	0.96	**1.53**	0.52	0.69	0.81	0.55	0.52	0.45	0.37
$\mathcal{V}_{21}$	**2.38**	2.20	1.24	1.43	0.95	0.91	0.87	0.50	0.44
$\mathcal{V}_{31}$	1.72	**1.82**	0.89	1.03	0.95	0.68	0.68	0.48	0.41
$\mathcal{V}_{41}$	1.72	**1.81**	0.79	1.03	0.95	0.67	0.67	0.46	0.41
$\mathcal{V}_{51}$	1.72	**1.82**	0.84	1.03	0.95	0.68	0.68	0.47	0.41
$\mathcal{V}_{61}$	**1.95**	1.92	1.01	1.43	0.95	0.91	0.87	0.50	0.44
$\mathcal{V}_{12}$	2.02	**2.75**	1.14	1.12	1.48	1.00	1.20	1.03	0.86
$\mathcal{V}_{22}$	**4.99**	3.93	2.72	2.32	1.75	1.68	2.01	1.15	1.00
$\mathcal{V}_{32}$	**3.60**	3.26	1.95	1.68	1.75	1.24	1.55	1.09	0.93
$\mathcal{V}_{42}$	**3.59**	3.25	1.74	1.67	1.75	1.23	1.54	1.06	0.93
$\mathcal{V}_{52}$	**3.60**	3.25	1.85	1.68	1.75	1.24	1.55	1.08	0.93
$\mathcal{V}_{62}$	**4.09**	3.44	2.23	2.32	1.75	1.68	2.01	1.15	1.00
$\mathcal{V}_{13}$	1.16	**1.94**	0.95	0.93	1.19	0.81	0.86	0.74	0.61
$\mathcal{V}_{23}$	**2.87**	2.79	2.26	1.93	1.41	1.35	1.43	0.82	0.72
$\mathcal{V}_{33}$	2.07	**2.31**	1.62	1.40	1.41	1.00	1.11	0.78	0.66
$\mathcal{V}_{43}$	2.06	**2.30**	1.45	1.40	1.41	0.99	1.10	0.76	0.66
$\mathcal{V}_{53}$	2.06	**2.30**	1.54	1.40	1.41	1.00	1.11	0.77	0.66
$\mathcal{V}_{63}$	2.35	**2.43**	1.85	1.93	1.41	1.35	1.43	0.82	0.72

$\mathcal{V}_{\mathrm{MH\Gamma}}$ and $\mathcal{V}_{\mathrm{MH\hat{\Gamma}}}$ are expected to show sharp knees at $c=3$, and Fig. 9 shows that they both do. The *scale* of a Hubert plot is *very important* in the assessment of Hubert knees. Almost any set of values will have "well defined" knees as long as the vertical scale has sufficient resolution to show it. This visual subjectivity makes Hubert's method somewhat unrepeatable.

$\mathcal{V}_{\mathrm{DB},22}$ clearly indicates $c=3$, having a strong minimum value of 0.18. Of the 18 generalized Dunn's indexes $\{\mathcal{V}_{\delta_i\Delta_j}\}$—each of which is to be maximized—ten indicate the correct value $c=3$ while the remaining eight favor $c=2$ (partition $U_{2,2}$ in Fig. 4). This happens even though the visually correct partition U^* of X_{30} shown in Figs. 2 and 3 is found by HCM at $c=3$. In other words, these indexes fail to solve the problem indicated by Figs. 3 and 4.

TABLE IV
VALIDITY INDEXES FOR U_{HCM} PARTITIONS OF **Iris**

c	2	3	4	5	6	7	8	9	10
$\mathcal{V}_{\mathrm{MH\Gamma}}$	7.22	8.22	**8.50**	8.57	8.60	8.65	8.65	8.66	8.66
$\mathcal{V}_{\mathrm{MH\hat{\Gamma}}}$	0.83	0.92	**0.95**	0.95	0.95	0.96	0.96	0.96	0.96
$\mathcal{V}_{\mathrm{DB},22}$	**0.47**	0.73	0.84	0.99	1.00	0.96	1.09	1.25	1.23
$\mathcal{V}_{11} = \mathcal{V}_{\mathrm{D}}$	0.08	**0.10**	0.08	0.06	0.09	0.10	0.08	0.06	0.06
$\mathcal{V}_{21}$	1.50	**1.81**	1.15	1.07	0.63	0.63	0.59	0.41	0.29
$\mathcal{V}_{31}$	**0.85**	0.73	0.57	0.39	0.33	0.33	0.24	0.21	0.19
$\mathcal{V}_{41}$	**0.83**	0.67	0.50	0.34	0.28	0.28	0.17	0.14	0.13
$\mathcal{V}_{51}$	**0.84**	0.70	0.54	0.36	0.30	0.30	0.21	0.18	0.16
$\mathcal{V}_{61}$	**1.01**	1.00	0.70	0.58	0.37	0.37	0.28	0.24	0.20
$\mathcal{V}_{12}$	**0.26**	0.25	0.20	0.14	0.20	0.23	0.17	0.14	0.14
$\mathcal{V}_{22}$	**5.02**	4.68	2.83	2.45	1.43	1.43	1.36	0.94	0.65
$\mathcal{V}_{32}$	**2.85**	1.89	1.40	0.88	0.75	0.75	0.55	0.47	0.44
$\mathcal{V}_{42}$	**2.78**	1.74	1.22	0.78	0.64	0.64	0.40	0.32	0.30
$\mathcal{V}_{52}$	**2.82**	1.81	1.31	0.83	0.70	0.70	0.48	0.40	0.37
$\mathcal{V}_{62}$	**3.36**	2.59	1.72	1.34	0.84	0.84	0.63	0.54	0.45
$\mathcal{V}_{13}$	**0.18**	0.18	0.14	0.10	0.14	0.16	0.12	0.10	0.10
$\mathcal{V}_{23}$	**3.53**	3.28	2.00	1.74	1.01	1.01	0.96	0.66	0.46
$\mathcal{V}_{33}$	**2.00**	1.32	0.99	0.63	0.53	0.53	0.39	0.34	0.31
$\mathcal{V}_{43}$	**1.96**	1.22	0.87	0.55	0.45	0.45	0.28	0.23	0.21
$\mathcal{V}_{53}$	**1.98**	1.27	0.93	0.59	0.49	0.49	0.34	0.28	0.26
$\mathcal{V}_{63}$	**2.37**	1.81	1.22	0.94	0.59	0.59	0.45	0.38	0.32

Examining the geometry of partition $U_{2,2}$, for example, consider index $\mathcal{V}_{22} = \min\{\delta_2\}/\max\{\Delta_2\}$. For $c = 2$, clusters X_1 and X_2 are merged in $U_{2,2}$ (see Fig. 4). Consequently, the minimum of δ_2 for $c = 2$ (let us call it ${}_{m2}\delta_2$) is much larger than the minimum of δ_2 for $c = 3$, say ${}_{m3}\delta_2$. On the other hand, although the maximum of Δ_2 for $c = 2$ (say ${}_{M2}\Delta_2$) is greater than the maximum of Δ_2 for $c = 3({}_{M3}\Delta_2)$, their numerical values are such that $({}_{m2}\delta_2/{}_{M2}\Delta_2) > ({}_{m3}\delta_2/{}_{M3}\Delta_2)$. Finally, note that the five indexes $\mathcal{V}_{22} - \mathcal{V}_{62}$ have fairly *strong* pointers to the incorrect value for c. This illustrates that indexes can lead to the wrong conclusion even for data sets such as $\boldsymbol{X}_{30}$ that have compact, well-separated clusters, and even when the preferred solution is among the candidates in $\mathcal{P}$.

Example 7.1B: $\boldsymbol{X}_{30}$ with SL: Table III shows values of the 21 validity indexes for SL partitions of $\boldsymbol{X}_{30}$ at each value of c for $c = 2$ to $c = 10$.

Columns in Tables II and III for $c = 2, 3$, and 4 are identical because HCM and SL find the same partitions at these values for c. For $c \geq 5$, the two algorithms find different partitions, so the values of the indexes are in many cases quite different too. Examples 7.1A and 7.1B allow us to conclude that HCM and SL often find the same (correct) partitions of X when its clusters are relatively compact and well-separated. And further, that nearly half of the $\mathcal{V}_{\delta_1\Delta_j}$s fail to indicate this.

Example 7.2A: Iris with HCM: Table IV lists the values of our 21 validity indexes on HCM c-partitions of **Iris**.

Although **Iris** contains observations from three physical classes, classes 2 and 3 are known to overlap in their numeric representations, while the 50 points from class 1 are very well separated from the remaining 100. Geometrically, the primary structure in **Iris** is probably $c = 2$, but the physical labels insist

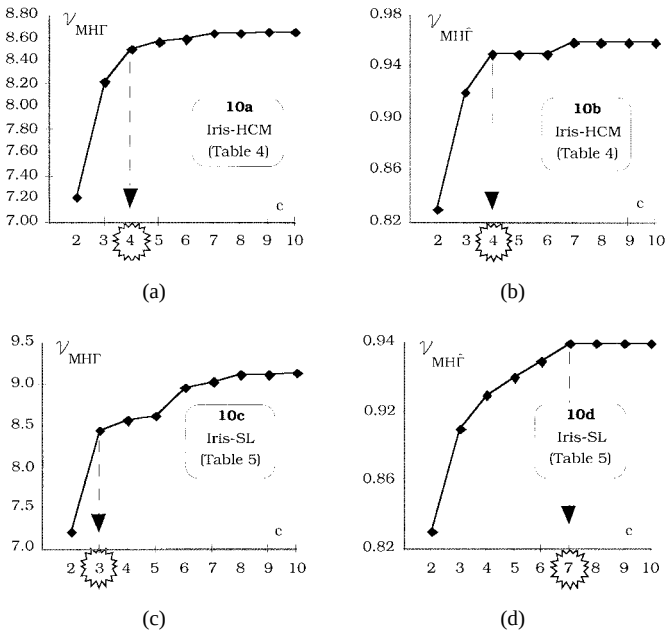


Fig. 10. Plots of $\mathcal{V}_{\mathrm{MH\Gamma}}$ and $\mathcal{V}_{\mathrm{MH\hat{\Gamma}}}$ for U_{HCM} and U_{SL} of **Iris**.

that $c = 3$. Consequently, the best value for c is debatable. Since clusters are defined by mathematical properties within models that depend on data representations that agree with the model, we take $c = 2$ as the correct choice for **Iris**, because what matters to an algorithm is how much cluster information is captured by the numeric representation of objects. Table IV shows that 16 of the 19 algebraic indexes indicate $c = 2$ for HCM partitions of Iris, which in our opinion is the preferred choice.

Both modified Hubert statistics in Table IV have knees at $c = 4$ (see Fig. 10). These are the only indexes that point to $c = 4$ in this experiment. Only two of the 21 indexes prefer $c = 3$, the physically correct number of clusters in **Iris**. And all but two of the 21 indexes identify either $c = 2$ (best choice, and perhaps the geometrically correct answer) or $c = 3$ (next best choice, and the physically correct answer).

Example 7.2B: Iris with SL: Table V lists the values of the 21 validity indexes on SL c-partitions of **Iris**. HCM and SL find slightly different partitions of Iris at every value of c.

$\mathcal{V}_{\mathrm{MH\Gamma}}$ in Table V points to $c = 3$, but its normalized form points to $c = 7$ (see Fig. 10). Of the 19 algebraic indexes, only Dunn's index does not point to $c = 2$ or 3; instead, it also chooses $c = 7$. Values of $\mathcal{V}_{\mathrm{MH\hat{\Gamma}}}$ across c are so close to each other for all c's except $c = 2$ that we might well call this test inconclusive (the same holds for $\mathcal{V}_D$). Of the 21 indexes on the HCM and SL partitions in Tables IV and V, there are 15 votes for $c = 2$ in matched cells. Agreement of many indexes across several rather different clustering models can be taken as a good sign that the structure of the data is being clustered and assessed correctly.

Fig. 10 shows the Hubert plots for the HCM and SL partitions of Iris. This figure illustrates the difficulty in choosing the distinguished value of c by observing a "sharp knee." View 10a is a pretty smooth curve, but does seem to have a knee

TABLE V
VALIDITY INDEXES FOR U_{SL} PARTITIONS OF **Iris**

c	2	3	4	5	6	7	8	9	10
$\mathcal{V}_{MH\Gamma}$	7.27	**8.44**	8.57	8.63	8.97	9.04	9.13	9.12	9.15
$\mathcal{V}_{MH\hat{\Gamma}}$	0.84	0.89	0.91	0.92	0.93	**0.94**	0.94	0.94	0.94
$\mathcal{V}_{DB,22}$	**0.46**	0.67	0.97	0.95	1.30	1.28	1.29	1.38	1.32
$\mathcal{V}_{11} = \mathcal{V}_D$	0.03	0.06	0.06	0.06	0.06	**0.07**	0.07	0.07	0.07
$\mathcal{V}_{21}$	**1.49**	1.36	1.31	0.65	0.68	0.79	0.82	0.72	0.67
$\mathcal{V}_{31}$	**0.85**	0.64	0.62	0.34	0.36	0.42	0.43	0.41	0.38
$\mathcal{V}_{41}$	**0.84**	0.72	0.52	0.56	0.36	0.42	0.43	0.37	0.37
$\mathcal{V}_{51}$	**0.85**	0.76	0.59	0.49	0.52	0.60	0.56	0.51	0.46
$\mathcal{V}_{61}$	**1.01**	1.00	0.81	0.43	0.45	0.53	0.55	0.43	0.49
$\mathcal{V}_{12}$	0.10	**0.18**	0.16	0.14	0.15	0.15	0.16	0.16	0.16
$\mathcal{V}_{22}$	**4.95**	3.91	3.41	1.63	1.77	1.77	1.89	1.66	1.54
$\mathcal{V}_{32}$	**2.82**	1.83	1.61	0.85	0.93	0.93	0.99	0.94	0.86
$\mathcal{V}_{42}$	**2.78**	2.07	1.36	1.39	0.93	0.93	0.99	0.86	0.86
$\mathcal{V}_{52}$	**2.81**	2.18	1.53	1.24	1.34	1.34	1.29	1.17	1.05
$\mathcal{V}_{62}$	**3.34**	2.87	2.10	1.08	1.18	1.18	1.25	0.98	1.14
$\mathcal{V}_{13}$	0.07	**0.12**	0.08	0.08	0.09	0.09	0.09	0.09	0.09
$\mathcal{V}_{23}$	**3.49**	2.46	1.72	0.92	1.02	1.02	1.07	0.93	0.87
$\mathcal{V}_{33}$	**1.99**	1.15	0.81	0.48	0.53	0.53	0.56	0.53	0.49
$\mathcal{V}_{43}$	**1.96**	1.30	0.68	0.78	0.53	0.53	0.56	0.48	0.48
$\mathcal{V}_{53}$	**1.98**	1.37	0.77	0.70	0.77	0.77	0.73	0.66	0.59
$\mathcal{V}_{63}$	**2.36**	1.80	1.06	0.61	0.67	0.67	0.71	0.55	0.64

TABLE VI
VALIDITY INDEXES FOR U_{HCM} PARTITIONS OF **Normal** 4×4

c	2	3	4	5	6	7	8	9	10
$\mathcal{V}_{MH\Gamma}$	7.27	11.91	**15.82**	16.24	16.61	16.86	16.93	17.41	17.49
$\mathcal{V}_{MH\hat{\Gamma}}$	0.34	0.49	**0.63**	0.67	0.67	0.68	0.69	0.71	0.71
$\mathcal{V}_{DB,22}$	1.81	1.32	**0.94**	1.36	1.60	1.66	1.45	1.68	1.53
$\mathcal{V}_{11} = \mathcal{V}_D$	0.03	0.03	0.06	**0.06**	0.03	0.04	0.04	0.02	0.03
$\mathcal{V}_{21}$	1.04	1.04	**1.12**	0.85	0.85	0.76	0.78	0.75	0.75
$\mathcal{V}_{31}$	0.46	0.49	**0.56**	0.39	0.35	0.35	0.31	0.34	0.32
$\mathcal{V}_{41}$	0.30	0.36	**0.48**	0.26	0.23	0.23	0.21	0.22	0.22
$\mathcal{V}_{51}$	0.39	0.43	**0.52**	0.33	0.30	0.30	0.27	0.29	0.27
$\mathcal{V}_{61}$	0.48	0.50	**0.60**	0.45	0.37	0.39	0.41	0.36	0.34
$\mathcal{V}_{12}$	0.08	0.09	0.18	**0.18**	0.10	0.13	0.13	0.08	0.11
$\mathcal{V}_{22}$	2.82	2.81	**3.60**	2.66	2.70	2.52	2.49	2.53	2.56
$\mathcal{V}_{32}$	1.26	1.31	**1.80**	1.20	1.12	1.16	1.00	1.16	1.09
$\mathcal{V}_{42}$	0.80	0.98	**1.55**	0.80	0.72	0.75	0.67	0.73	0.77
$\mathcal{V}_{52}$	1.06	1.18	**1.68**	1.02	0.94	0.98	0.86	0.98	0.94
$\mathcal{V}_{62}$	1.31	1.34	**1.93**	1.40	1.18	1.30	1.31	1.22	1.15
$\mathcal{V}_{13}$	0.06	0.06	0.13	**0.13**	0.07	0.09	0.10	0.06	0.08
$\mathcal{V}_{23}$	1.95	1.92	**2.56**	1.89	1.92	1.78	1.76	1.81	1.83
$\mathcal{V}_{33}$	0.87	0.90	**1.28**	0.85	0.80	0.82	0.71	0.83	0.77
$\mathcal{V}_{43}$	0.55	0.67	**1.10**	0.57	0.52	0.53	0.47	0.52	0.55
$\mathcal{V}_{53}$	0.73	0.80	**1.19**	0.73	0.67	0.69	0.60	0.70	0.67
$\mathcal{V}_{63}$	0.90	0.92	**1.37**	0.99	0.84	0.92	0.92	0.87	0.82

at $c = 4$. View 10b has a "strong knee" at $c = 4$ and a weaker knee at $c = 7$; panel 10c shows the same type of graph, pointing first to $c = 3$, and then to $c = 6$. Finally, view 10d seems to have the most pronounced knee at $c = 7$. Our analysis of this figure, however, is obviously subjective. If the *largest change in values* is used (like the internal SL criterion) instead, all four of these graphs would point to either $c = 2$ or $c = 3$ depending on your interpretation of the meaning underlying this strategy. Another point worth noting is that $\mathcal{V}_{MH\Gamma}$ and $\mathcal{V}_{MH\hat{\Gamma}}$ do not always lead to the same value for c. This is seen in Fig. 10(c) and 10(d), and we also observed this in other tests as well.

Example 7.3A: Normal 4 × 4 with HCM: Table VI lists the validity indexes on HCM c-partitions of **Normal** 4×4. All indexes except $\mathcal{V}_{11}, \mathcal{V}_{12}$, and $\mathcal{V}_{13}$ indicate $c = 4$, the preferred choice for this data set. The three failures use $\delta_1 = \delta_{SL}$ (20) as the interset distance. However, the optimal values for these three indexes, shown at $c = 5$ (they look identical to $c = 4$, but remember our roundoff policy), are very close to their values for the correct c. $\mathcal{V}_{MH\Gamma}$ and $\mathcal{V}_{MH\hat{\Gamma}}$ both point nicely to $c = 4$. The structure of **Normal** 4×4 should be fairly well defined since ∼95% of the 200 samples in each cluster in it are captured in spheres of radius 2 about their centers, which are three units from the origin of $\Re^4$.

Example 7.3B: Normal 4 × 4 with SL: Table VII lists the validity indexes on SL c-partitions of **Normal** 4×4. The values of the indexes in Table VII are vastly different from those in Table VI, implying that SL has found very different partitions of this data set than HCM. Moreover, of the 21 indexes, *only* $\mathcal{V}_{21}$ points to $c = 4$! The failures represented by Table VII can be understood by remembering the type of data structures that HCM and SL prefer. **Normal** 4×4 has essentially compact cluster *cores*, but the sampling process undoubtedly produces a few bridge points between clusters. This enables SL to (mistakenly) leap across the neck between the Gaussian clusters, and partitions produced by it are predictably bad. We can't see this data set, so this speculation is based on our knowledge of the clustering algorithm. The worrisome thing here is, of course, that if you didn't know the right answer, Table VII would lead you to strongly consider $c = 2$ as the best choice for this data set. And this would be a misleading inference about the (unknown) structure possessed by the data.

The failure of all but one index to pick the right c is not due to the incapability of the indexes. Rather, we attribute this to the bad partitions generated by the SL clustering algorithm. The most compelling evidence for rejecting the suggestion presented by Table VII is to put Tables VI and VII side by side. In this case, one set of values points largely to $c = 4$, while the other set points to $c = 2$. Here we know why, but if X were unlabeled, what you should conclude from a comparison of the two tables is that (unlike Tables IV and V, which showed good consistency for **Iris**) something is badly amiss in algorithmic interpretations of this data. Neither conclusion ($c = 2$ or $c = 4$) should be given a lot of weight without further study.

Example 7.4: Table VIII summarizes the values of c^* suggested by the 21 indexes on HCM partitions of each of six data sets. In the table, "inc." for $\mathcal{V}_{MH\Gamma}$ and $\mathcal{V}_{MH\hat{\Gamma}}$ means *inconclusive*. The last column of Table VIII shows the number of times in six tries that c agrees with the preferred value of c, called $\hat{c}$ in the table. Table VIII indicates that the three indexes $\mathcal{V}_{\delta_1\Delta_j}$ that use δ_1 as the interset distance fail to point to good partitions, irrespective of the diameter (Δ_i) definition used.

TABLE VII
VALIDITY INDEXES FOR U_{SL} PARTITIONS OF **Normal** 4×4

c	2	3	4	5	6	7	8	9	10
$\mathcal{V}_{MH\Gamma}$	0.97	4.93	8.13	**11.87**	13.10	13.89	15.05	15.42	15.92
$\mathcal{V}_{MH\hat{\Gamma}}$	0.14	0.23	0.29	**0.37**	0.39	0.41	0.43	0.44	0.46
$\mathcal{V}_{DB,22}$	**1.07**	1.28	1.29	1.30	1.23	1.51	1.57	1.85	1.83
$\mathcal{V}_{11} = \mathcal{V}_D$	**0.07**	0.04	0.04	0.04	0.04	0.03	0.03	0.03	0.03
$\mathcal{V}_{21}$	1.05	1.02	**1.08**	0.87	0.80	0.80	0.78	0.78	0.63
$\mathcal{V}_{31}$	**0.55**	0.48	0.51	0.49	0.40	0.40	0.38	0.38	0.30
$\mathcal{V}_{41}$	**0.64**	0.44	0.47	0.47	0.47	0.31	0.33	0.24	0.24
$\mathcal{V}_{51}$	**0.70**	0.52	0.56	0.54	0.54	0.42	0.43	0.41	0.39
$\mathcal{V}_{61}$	**0.77**	0.68	0.69	0.69	0.48	0.48	0.44	0.44	0.35
$\mathcal{V}_{12}$	**0.18**	0.10	0.09	0.09	0.09	0.08	0.08	0.08	0.08
$\mathcal{V}_{22}$	**2.57**	2.52	2.54	2.13	1.97	2.01	1.88	1.90	1.55
$\mathcal{V}_{32}$	**1.34**	1.18	1.19	1.19	0.99	1.00	0.91	0.92	0.74
$\mathcal{V}_{42}$	**1.57**	1.08	1.11	1.15	1.15	0.79	0.80	0.59	0.60
$\mathcal{V}_{52}$	**1.71**	1.29	1.31	1.31	1.33	1.04	1.04	1.00	0.97
$\mathcal{V}_{62}$	**1.87**	1.68	1.62	1.68	1.19	1.20	1.07	1.08	0.87
$\mathcal{V}_{13}$	**0.11**	0.06	0.05	0.05	0.06	0.05	0.05	0.05	0.05
$\mathcal{V}_{23}$	**1.48**	1.45	1.42	1.32	1.25	1.27	1.16	1.16	0.94
$\mathcal{V}_{33}$	**0.77**	0.68	0.67	0.73	0.62	0.63	0.56	0.56	0.45
$\mathcal{V}_{43}$	**0.90**	0.62	0.62	0.71	0.73	0.50	0.50	0.36	0.36
$\mathcal{V}_{53}$	**0.99**	0.74	0.73	0.81	0.84	0.66	0.64	0.61	0.59
$\mathcal{V}_{63}$	**1.08**	0.97	0.91	1.04	0.75	0.76	0.66	0.66	0.53

TABLE VIII
VALUES OF c SUGGESTED BY THE 21 INDEXES ON HCM PARTITIONS OF SIX DATA SETS

$\hat{c}$	3	2	4	4	4	4	
Data	$\mathbf{X}_{30}$	**Iris**	$\mathbf{X}_3$ **Normal 4x4**	$\mathbf{X}_4$	$\mathbf{X}_5$	$\mathbf{X}_6$	# Correct in 6 tries
$\mathcal{V}_{MH\Gamma}$	3	3	4	4	inc.	inc.	**3**
$\mathcal{V}_{MH\hat{\Gamma}}$	3	3	4	4	inc.	inc.	**3**
$\mathcal{V}_{DB,22}$	3	2	4	4	9	7	**4**
$\mathcal{V}_{11} = \mathcal{V}_D$	3	3	5	8	10	7	**1**
$\mathcal{V}_{21}$	2	3	4	3	4	5	**2**
$\mathcal{V}_{31}$	3	2	4	4	4	5	**5**
$\mathcal{V}_{41}$	3	2	4	4	4	5	**5**
$\mathcal{V}_{51}$	3	2	4	4	4	5	**5**
$\mathcal{V}_{61}$	2	2	4	4	4	5	**4**
$\mathcal{V}_{12}$	3	3	5	8	10	10	**1**
$\mathcal{V}_{22}$	2	2	4	4	2	2	**3**
$\mathcal{V}_{32}$	2	2	4	4	4	7	**4**
$\mathcal{V}_{42}$	2	2	4	4	4	7	**4**
$\mathcal{V}_{52}$	2	2	4	4	4	7	**4**
$\mathcal{V}_{62}$	2	2	4	4	2	5	**3**
$\mathcal{V}_{13}$	3	2	5	8	10	10	**2**
$\mathcal{V}_{23}$	2	2	4	4	2	2	**3**
$\mathcal{V}_{33}$	3	2	4	4	4	7	**5**
$\mathcal{V}_{43}$	3	2	4	4	9	7	**4**
$\mathcal{V}_{53}$	3	2	4	4	4	7	**5**
$\mathcal{V}_{63}$	3	2	4	4	2	7	**4**

Not surprisingly, the correct number of clusters ($\hat{c} = 4$) is *never* indicated for $\boldsymbol{X}_6$ ($\mathcal{V}_{MH\hat{\Gamma}}$ had a very weak Hubert knee here, which we decided to label as inconclusive). Recall that $\boldsymbol{X}_6$ is derived from $\boldsymbol{X}_3 = \mathbf{Normal\ 4 \times 4}$ by subtracting 2.5 from every coordinate of $\boldsymbol{X}_3$. Since the means of $\boldsymbol{X}_3$ were three units from the origin of $\Re^4$, it is very likely that cluster structure has more or less disappeared under this transformation.

VIII. DISCUSSION AND CONCLUSIONS

We have reviewed three crisp indexes of cluster validity: the modified Hubert Statistic, Davies–Bouldin, and Dunn's index. We then proposed five new set distance and two new set diameter functions, and used them to define a family of 18 cluster validity indexes that generalize Dunn's index. Computational examples on six data sets were used to compare the 21 indexes described in this paper. Here are our conclusions, which we emphasize again, are specialized to the case where clusters are expected to form volumetric clouds as follows.

1) The modified Hubert Statistics $\mathcal{V}_{MH\Gamma}$ and $\mathcal{V}_{MH\hat{\Gamma}}$ and the Davies–Bouldin index $\mathcal{V}_{DB,22}$ produced either three or four successes in six tries (cf., Table VIII). We conclude that they are more or less equally effective. Visual identification of Hubert knees is very subjective and scale dependent. We think the DB index is preferable to $\mathcal{V}_{MH\Gamma}$ and $\mathcal{V}_{MH\hat{\Gamma}}$ because of this.
2) The three generalized Dunn's indexes using δ_1 have the worst records in these trials. Table VIII supports the assertion that the standard measure of interset distance, $\delta_1(S,T) = \delta_{\min}(S,T) = \underbrace{\min}_{\substack{\boldsymbol{x} \in S \\ \boldsymbol{y} \in T}} \{d(\boldsymbol{x},\boldsymbol{y})\}$, is the *worst* among the family $\{\delta_i\}$. Dunn's index, which also uses Δ_1, is the least successful index among the 21 indexes tested. The performance of the three GDI's using δ_2 also is pretty bad. This is because δ_2, like δ_1, depends only on the distance between a pair of points. We conclude that intraset distances should use all the data points. Δ_1, although sensitive to noisy points, produces good performance when used with δ_3, δ_4, and δ_5. This is because the data used in our study possess relatively compact clusters the means $(\overline{\boldsymbol{v}})$ of which are close to the vectors $(\boldsymbol{m})$ as shown in Fig. 7(a). Moreover, we believe that interclass separation plays a more significant role in cluster validation than within cluster dispersion (size or diameter of the cluster).
3) Five of the 18 GDI's produced five successes in six tries on HCM partitions: $\mathcal{V}_{31}, \mathcal{V}_{41}, \mathcal{V}_{51}, \mathcal{V}_{33}$, and $\mathcal{V}_{53}$. From this study we conclude that δ_3 and δ_5 provide the most reliable measures of intercluster distance. In other studies, δ_6 has also been very effective [14]. Not surprisingly, Δ_2 gives the least reliable measure of set diameter [cf., Fig. 7(b)]. We think our simulations show that some of the GDI's are better than any of the crisp validity functions to which they were compared in this study.
4) The indexes discussed here may not be good for chain or shell type of clusters because the definitions (particularly the set diameters discussed in Section V) implicitly characterize cloud type clusters. However, Dunn's index can be suitably generalized in a slightly different way than the method presented here so that it is applicable to chain or shell type clusters. For example Pal and Biswas [15] used graph theoretic concepts (minimal spanning

trees, relative neighborhood graphs and Gabriel graphs) to define the diameter of clusters.

In summary, Dunn's index in its original form is not very suitable for cluster validation because of its sensitivity to noisy points. But it provides a rich and very general structure for defining cluster validity indexes for different types of clusters. With suitable interset distances and set diameters generalizations of Dunn's index can be used to validate hyperspherical/cloud and shell type clusters.

Finally, we add some comments for practitioners. Clustering is a very useful tool that has many well documented and important applications: to name a few, data mining, image segmentation and extraction of rules for fuzzy controllers. The problem of validation for truly unlabeled data is an important consideration in all of these applications, each of which has developed its own set of partially successful validation schemes. Our experience is that no one index is likely to provide consistent results across different clustering algorithms and data structures. One popular approach to overcoming this dilemma is to use many validation indexes, and conduct some sort of vote among them about the best value for c. Many votes for the same value tend to increase your confidence, but even this does not prevent mistakes (cf., Tables VI and VII). We feel that the best strategy is to use several very different clustering models (such as HCM and SL), vary the parameters of each, and collect many votes from various indexes. If the results across various trials are consistent (as in Tables IV and V), the user may assume that meaningful structure in the data is being found. But if the results are inconsistent (Tables VI and VII), more simulations are needed before much confidence in algorithmically suggested substructure is warranted.

APPENDIX A
THE BATCH HARD c-MEANS (HCM) ALGORITHM [4]

Store	Unlabeled Object Data $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}\ c\ \Re^p$
Pick	• $1 < c < n$ • $T =$ iteration limit $= 100$ • $\varepsilon = 0.000\,01$ • Euclidean norm for clustering criterion J_1: $\|\|\boldsymbol{x}\|\|_2 = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$ • Norm for termination error $E_t = \|\|\boldsymbol{V}_t - \boldsymbol{V}_{t-1}\|\|_2$ $= \sqrt{(\boldsymbol{V}_t - \boldsymbol{V}_{t-1})^T(\boldsymbol{V}_t - \boldsymbol{V}_{t-1})}$
Guess	$\boldsymbol{V}_0 = (\boldsymbol{v}_{1,0}, \boldsymbol{v}_{2,0}, \cdots, \boldsymbol{v}_{c,0}) \in \Re^{cp}$
Iterate	For $t = 1$ to T: Calculate U_t with $\boldsymbol{V}_{t-1}$ and (5a) Calculate $\boldsymbol{V}_t$ with U_t and (5b) If $E_t \le \varepsilon$, stop; Else Next t $(U, V) \leftarrow (U_t, \boldsymbol{V}_t)$
Use	Prototypes $\boldsymbol{V}$ and/or Labels U

APPENDIX B
RELATIONAL DATA FOR THE SL EXAMPLE

(See Tables IX and X and Fig. 11.)

TABLE IX
COORDINATES OF $\boldsymbol{X}_9$

	Cluster X_1				Cluster X_2				
Data Vector	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$
Feature 1	1	2	2	1	4	5	4.5	5	4
Feature 2	1	1	3	3	1.5	1.5	1.5	2.5	2.5

TABLE X
RELATIONAL DATA $\boldsymbol{R}_9$ CREATED FROM $\boldsymbol{X}_9$: $r_{jk} = \|\|\boldsymbol{x}_j - \boldsymbol{x}_k\|\|_2$

	1	2	3	4	5	6	7	8	9
1	0								
2	1.00	0							
3	2.24	2.00	0						
4	2.00	2.24	1.00	0					
5	3.04	2.06	2.50	3.35	0				
6	4.03	3.04	3.35	4.27	1.00	0			
7	3.54	2.25	2.92	3.81	0.50	0.50	0		
8	4.27	3.35	3.04	4.03	1.41	1.00	1.12	0	
9	3.35	2.50	2.06	3.04	1.00	1.41	1.12	1.00	0

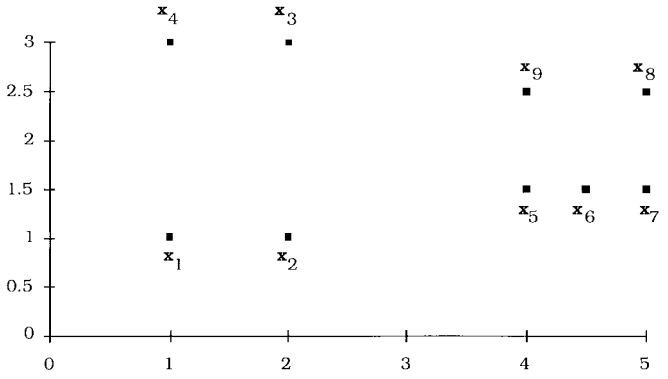


Fig. 11 Data set X_9.

REFERENCES

[1] A. Jain and R. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice Hall, 1988.
[2] B. S. Everitt, *Graphical Techniques for Multivariate Data*. New York: North Holland, 1978.
[3] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[4] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
[5] D. Titterington, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions*. New York: Wiley, 1985.
[6] R. Krishnapuram and J. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 4, pp. 98–110, 1993.

[7] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1973.
[8] J. Tou and R. Gonzalez, *Pattern Recognition Principles*. Reading, MA: Addison-Wesley, 1974.
[9] L. J. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, pp. 193–218, 1985.
[10] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 1, no. 4, pp. 224–227, 1979.
[11] P. Sneath and R. Sokal, *Numerical Taxonomy*. San Francisco, CA: Freeman, 1973.
[12] F. Preparata and M. Shamos, *Computational Geometry: An Introduction*. New York: Springer-Verlag, 1987.
[13] E. Anderson, "The Irises of the Gaspe peninsula," *Bull. Amer. Iris Soc.*, vol. 59, pp. 2–5, 1935.
[14] J. C. Bezdek, W. Q. Li, Y. Attikiouzel, and M. Windham, "A geometric approach to cluster validity for normal mixtures," *Soft Comput.*, vol. 1, pp. 166–179, 1997.
[15] N. R. Pal and J. Biswas, "Cluster validation using graph theoretic concepts," *Pattern Recognit.*, vol. 30, no. 6, pp. 847–857.

James C. Bezdek (M'80–SM'90–F'92) received the Ph.D. degree from Cornell University, Ithaca, NY, in 1973.

His interests include pattern recognition, fishing, computational neural networks, skiing, image processing, blues music, medical computing, and motorcycles.

Dr. Bezdek is the founding Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS.

Nikhil R. Pal obtained the B.Sc. (Hons.) degree in physics and the M.B.M. degree in operations research in 1979 and 1982, respectively, from the University of Calcutta, Calcutta, India. He received the M.Tech. and Ph.D. degrees in computer science from the Indian Statistical Institute, Calcutta, in 1984 and 1991, respectively.

He is an Associate Professor in the Machine Intelligence Unit, Indian Statistical Institute. He was with Hindusthan Motors Ltd., W. B., from 1984 to 1985 and Dunlop India Ltd., W. B., from 1985 to 1987. In 1987, he joined the Computer Science Unit, Indian Statistical Institute. From August 1991 to February 1993, July 1994 to December 1994, and October 1996 to December 1996, he visited the University of West Florida, Pensacola. He was a guest lecturer at the University of Calcutta. His research interests include image processing, pattern recognition, fuzzy sets and systems, uncertainty measures, genetic algorithms, fuzzy control, and neural networks. He is an associate editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and the *International Journal of Approximate Reasoning*.