

Comparing Soft Clusters and Partitions

Derek T. Anderson, James M. Keller, Ozy

Sjahputera and James C. Bezdek

Electrical and Computer Engineering

University of Missouri

Columbia, MO, USA

andersondt@missouri.edu, kellerj@missouri.edu,
sjahputerao@missouri.edu and jcbzdek@gmail.com

Mihail Popescu

Health Management and Informatics

University of Missouri

Columbia, MO, USA

popescum@missouri.edu

Abstract—Previously, we presented a method for comparing soft partitions (i.e. crisp, probabilistic, fuzzy and possibilistic) to a known crisp reference partition. Many of the classical indices that have been used with outputs of crisp clustering algorithms were generalized so that they are applicable for candidate partitions of any type. In particular, focus was placed on generalizations of the Rand index. In this article, we extend our prior work by (1) investigating the behavior of the soft Rand for comparing non-crisp, specifically possibilistic, partitions and (2) we demonstrate how the possibilistic Rand and visual assessment of (cluster) tendency (VAT) algorithm can be used to discover the number of actual clusters and coincident clusters for outputs from the possibilistic c-means (PCM) algorithm.

Keywords: cluster validity; possibilistic Rand index; comparing soft partitions; cluster correspondence.

I. INTRODUCTION

Let $O = \{o_1, \dots, o_n\}$ denote n objects. When each object in O is represented by a (column) vector $\mathbf{x}$, the set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^p$ is an *object data representation* of O . When each object in $o_i \in O$ has a *physical label*, O is a set of *labeled data*; otherwise, O is unlabeled. Let integer c denote the number of classes, $1 < c < n$. Clustering in unlabeled data is the assignment of one of four types of labels to each object in O . The label vectors of the objects are the columns of c -partitions of O , which are sets of (cn) values $\{u_{ik}\}$ that can be conveniently arrayed as $(c \times n)$ matrices, say $U = [u_{ik}]$. The three sets are:

$$M_{pcn} = \left\{ U \in \mathbb{R}^{cn} : u_{ik} \in [0,1] \forall i,k; 0 < \sum_{k=1}^n u_{ik} \forall i \right\}, \quad (1a)$$

$$M_{fcn} = \left\{ U \in M_{pcn} : \sum_{i=1}^c u_{ik} = 1 \forall k \right\}, \quad (1b)$$

$$M_{hcn} = \{U \in M_{fcn} : u_{ik} \in \{0,1\} \forall i,k\}, \quad (1c)$$

where M_{pcn} (1a) are possibilistic c -partitions, M_{fcn} (1b) are fuzzy or probabilistic c -partitions, and M_{hcn} (1c) are crisp or hard c -partitions. It is convenient to have a single name for the set M_{pcn} - M_{hcn} , which contains the fuzzy, probabilistic, and possibilistic c -partitions of O . We call M_{pcn} - M_{hcn} the *soft c -partitions* of O .

Clustering algorithms map $X \subset \mathbb{R}^p \mapsto M_{pcn}$. Let $CP = \{U_i; 1 \leq i \leq N\}$ denote N different *candidate partitions* of a fixed object set O that may arise as a result of clustering X with one algorithm at various values of its parameters; or more generally, with different algorithms, each with its own parameters. Which $U \in CP$ best explains and represents the (unknown) structure in O ? In [1,2], we introduced one way to answer this question through the use of a *comparison index*, $s(U,V)$. Applications of our prior approach include:

- A1. $s(U,V)$: V ($r \times n$) is a *reference partition* that purports to represent the "true cluster structure" in O . $s(U,V)$ measures the extent to which U 's in CP recover or retrieve the true clusters in O , and hence, the sizes of U and V are equal. Retrieval rates are called *resubstitution error rates* when U and V are crisp, $r = c$, and s is a measure that counts matches between the columns in U and V .
- A2. $s(U,U')$: U is a partition of O found with clustering algorithm A . Perturb each $\mathbf{x}_k$ or r_{ij} with zero mean, unit variance noise, and obtain U' , a clustering of the perturbed data. $s(U, U')$ measures the sensitivity of A to perturbations of the data.
- A3. $s(U,U')$: U, U' are partitions of O found with clustering algorithms A and A' . $s(U, U')$ measures the agreement between the two algorithms about clusters in O .
- A4. $s(U,U')$: U, U' are partitions of O found by successive iterations of clustering algorithms. $s(U, U')$ measures the change in similarity due to the next iteration, and hence, can be used as a stopping criterion for iterative methods.
- A5. $s(U,V)$: Prediction of V given U , or U given V (regression).

In [1,2], our experiments focused on A1. We never have an external reference partition in a real clustering situation which, by definition, involves unlabeled data. So, why do this at all? Well, the only way you can evaluate *any* clustering algorithm before using it in a real situation is to see how well it recovers "true but unknown" reference partitions. If nothing else, good recovery rates on data with "known" cluster structure at least provide some psychological reassurance that the clustering algorithm *can* sometimes recover "good clusters".

In this article, we extend our prior work by exploring the behavior of the soft Rand for comparing non-crisp, specifically possibilistic, partitions. We also demonstrate how the possibilistic Rand index and visual assessment of (cluster) tendency (VAT) algorithm can be coupled in order to estimate the number of clusters and identify coincident clusters found by the possibilistic c-means (PCM) algorithm [3,4]. This work supports the claim [4] that coincident clusters found by PCM can be advantageous. We demonstrate that by over specifying the number of clusters, one can address important different clustering tasks.

II. BACKGROUND

A. Prior Work

Let $U \in M_{hm}$ and $V \in M_{hcn}$ be crisp partitions of O . U and V need not possess the same number of clusters. The four classical combinations for pairs of objects from $O \times O$ in clusters of U and V are: (i) paired in U and V ; (ii) not paired in U nor in V ; (iii) paired in V but not in U ; and (iv) paired in U but not in V [5]. The comparison of U to V with a similarity measure s begins with the $r \times c$ contingency matrix $N = UV^T$ shown in Table 1 that contains counts of the number of occurrences of each of the four types over the $n(n-1)/2$ distinct, unordered pairs in $O \times O$. Entry n_{ij} is the number of objects common to classes U_i and V_j .

Table 1. The contingency matrix N .

		Partition V $V_j = \text{row } j \text{ of } V$				Sums
		V_1	V_2	...	V_c	
Partition U $U_i = \text{row } i \text{ of } U$	U_1	$N = \begin{bmatrix} n_{11} & n_{12} & \cdots & n_{1c} \\ n_{21} & n_{22} & \cdots & n_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ n_{r1} & n_{r2} & \cdots & n_{rc} \end{bmatrix} = UV^T$				$n_{1\bullet}$
	U_2					$n_{2\bullet}$
	$\vdots$					$\vdots$
	U_r					$n_{r\bullet}$
	Sums					$n_{\bullet\bullet} = n$

The building blocks of many similarity measures for $s(U, V)$ are the four equations (2a)-(2d). These four equations simply count the number of occurrences among the $n(n-1)/2$ pairs of each of the four types of unordered pairs.

$$a = \frac{1}{2} \sum_{i=1}^r \sum_{j=1}^c n_{ij}(n_{ij} - 1) \quad (2a)$$

$$b = \frac{1}{2} \left(\sum_{j=1}^c n_{\bullet j}^2 - \sum_{i=1}^r \sum_{j=1}^c n_{ij}^2 \right) \quad (2b)$$

$$c = \frac{1}{2} \left(\sum_{i=1}^r n_{i\bullet}^2 - \sum_{i=1}^r \sum_{j=1}^c n_{ij}^2 \right) \quad (2c)$$

$$d = \frac{1}{2} \left(n^2 + \sum_{i=1}^r \sum_{j=1}^c n_{ij}^2 - \left(\sum_{i=1}^r n_{i\bullet}^2 + \sum_{j=1}^c n_{\bullet j}^2 \right) \right) \quad (2d)$$

Formula (2a) is the number paired in U and V , formula (2d) is the number paired in neither U nor V , formula (2b) is the

number paired in V , not U , and formula (2c) is the number paired in U , not V .

The sums $(a+d)$ and $(b+c)$ are usually interpreted, respectively, as (the total number of) agreements and disagreements between U and V . In [1], we tabulate a [non-exhaustive] list of 14 coefficients that have been proposed for $s(U, V)$ based on functions of a , b , c and d ; Sokal and Sneath [6] list many others. In this note, the only index we consider is Rand's index [7], the classical form of which is

$$s_r(U, V) = (a + d) / (a + b + c + d). \quad (3)$$

B. Generalizing $s_r(U, V)$ when U and/or V are soft partitions

Rand's index first appeared in Sokal and Michener in 1958, where it was called a simple matching coefficient [8]. Rand reintroduced this function in 1971 [7], and the literature has consistently referred to it as "Rand's Index" since then. The resurgence of Rand's index in bioinformatics [9-12] has renewed interest in generalizing it, along with some of the other comparison indices based on the elements in Table 1, to various non-crisp cases. Specifically, we mention the papers (in chronological order) of Campello [13], Frigui et al. [14], Brower [15], Hullermeier and Rifqi [16], and Anderson et al. [1,2]. All of these papers generalize the Rand index to the case of U and/or V being fuzzy partitions of the n objects.

In [1,2], we provide generalizations for 14 comparison indices. It begins with the construction of the contingency matrix as $N = UV^T$. We showed that a modification is needed to accommodate the case when U and/or V are possibilistic. In the possibilistic case, we can have

$$\sum_{i=1}^r n_{i\bullet} > n, \text{ or } \sum_{j=1}^c n_{\bullet j} > n. \quad (4)$$

One or both of the terms $\sum_{i=1}^r n_{i\bullet}^2$ and $\sum_{j=1}^c n_{\bullet j}^2$ can make d in (2d)

relatively large and negative. Depending on b in (2b) and c in (2c), the (soft) Rand index can result in $s_r(U, V) < 0$ or > 1 . To remedy this, we scale N with

$$\phi = n / \sum_{i=1}^r n_{i\bullet} \quad (5)$$

or

$$\phi = n / \sum_{j=1}^c n_{\bullet j}. \quad (6)$$

Since

$$\sum_{i=1}^r n_{i\bullet} = \sum_{j=1}^c n_{\bullet j}, \quad (7)$$

$\phi = \phi$. Our normalized generalization [1] is

$$s_{r*}(U, V) = N^* = \phi UV^T = [n / \sum_{i=1}^r n_{i\bullet}] UV^T. \quad (8)$$

An advantage of this scaling is that when U and V are crisp, fuzzy or probabilistic partitions, $\phi=1$, thus $N^* = N = UV^T$. This

shows that *ANY* index based on only the elements of Table 1 will reduce to the original index when U and V are both crisp partitions of the n objects; and otherwise, they will be valid soft generalizations of those indices. Moreover, for possibilistic partitions, the normalization produces index values in $[0,1]$.

III. COMPARING NON-CRISP PARTITIONS

In [1,2], we explored $A1$, which measures the extent to which U 's in CP recover or retrieve the true clusters in O . Our initial experiments did not contain any analysis of non-crisp to non-crisp partition comparison. In this section, we demonstrate non-crisp experiments, specifically for case $A3$, using the fuzzy c -means (FCM) algorithm [17] and the PCM [3,4]. We discuss the behavior of the measure.

Candidate fuzzy partitions are generated by the FCM using the fcm function from the MATLAB Fuzzy Logic Toolbox with the fuzzifier $q = 2$, maximum number of iterations $MAXIT = 100$, objective function error $EPS = 1e-5$ and random partition initialization. The PCM partitions are generated using our MATLAB implementation of the algorithm with the fuzzifier $q = 2$, $MAXIT = 100$, $EPS = 0.001$ and a FCM-based initialization. As discussed by Krishnapuram and Keller in [3,4], we re-estimated the parameters η_i (i.e. the value that determines the distance at which the membership value of a point in a cluster becomes 0.5) for each cluster at every iteration. The data sets are made up of different numbers of Gaussian clouds. The number of clouds and their parameters are provided with the discussion for each data set.

Example 1. Our analysis begins with data set 1 (D1), the scatterplot of which is shown in the uppermost view in figure 1. The images of the partition matrices shown in figure 1 are color coded, using a jet colormap, to assist visual interpretation. The value displayed, $[0,1]$, ranges from blue to red, and passes through the colors cyan, yellow, and orange.

FCM is applied twice to D1 using different random initializations at $c = 3$, resulting in the terminal partitions $U_{f,1}$ and $U_{f,2}$. The images of these two fuzzy partitions in the middle views of figure 1 suggest that $U_{f,1} = U_{f,2}$, up to a relabeling (the data points are sorted by cluster to aid visual analysis, which is possible because we synthetically generated X). And this is the case. Putting $U = V$ equation (8) we have

$$U = V \Rightarrow s_{r*}(U, U) = s_{r*}(V, V) = s_{r*}(U, V) \quad (9)$$

In view of this, it is not surprising to find that $s_{r*}(U_{f,1}, U_{f,1}) = s_{r*}(U_{f,2}, U_{f,2}) = s_{r*}(U_{f,1}, U_{f,2}) = 0.97$. The converse, which would be a useful result, does not follow. That is, we cannot conclude from $s_{r*}(U_{f,1}, U_{f,1}) = s_{r*}(U_{f,2}, U_{f,2}) = s_{r*}(U_{f,1}, U_{f,2})$ that $U = V$.

Next, the PCM is applied twice with different random initializations and $c = 3$ ($U_{p,1}$ and $U_{p,2}$). The two partition matrices are shown in the bottom view of figure 1. We find $s_{r*}(U_{p,1}, U_{p,1}) = s_{r*}(U_{p,2}, U_{p,2}) = s_{r*}(U_{p,1}, U_{p,2}) = 0.59$. We will explain why s_{r*} is considerably less than 1 after the next example.

Example 2. Data set 2 (D2) is shown in figure 2. The main difference between D2 and D1 is that clusters in D1 are visually more compact and well separated. FCM and PCM are

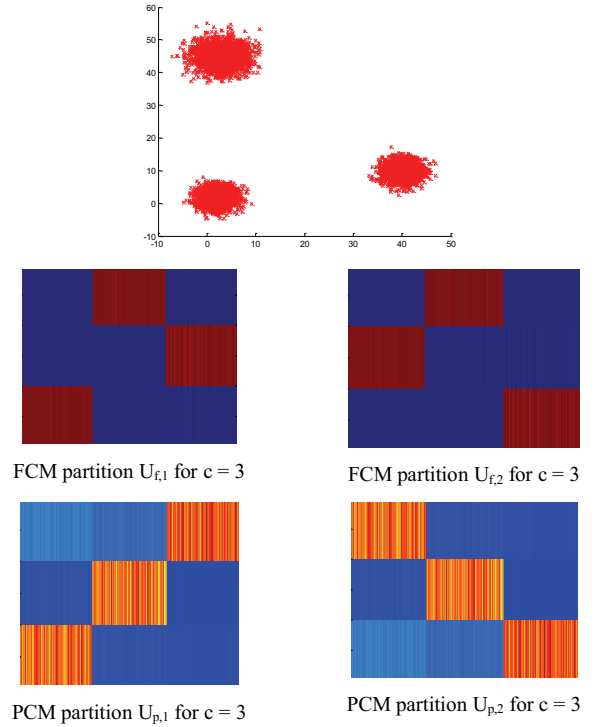


Fig. 1. Data set 1 (D1) and corresponding FCM and PCM partitions. D1 is made up of three Gaussian clouds, each with 2000 samples, means of $[2 \ 2]^T$, $[40 \ 10]^T$, $[3 \ 45]^T$ and standard deviations of 2, 2 and 3 respectively. The four candidate partitions are shown in order to visually compare and contrast D1 to data set 2 (D2), figure 2.

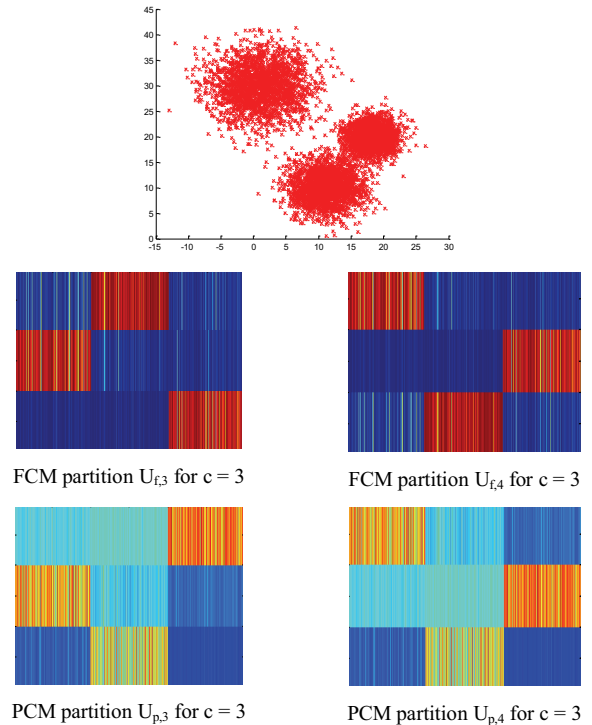


Fig. 2. Data set 2 (D2) and corresponding FCM and PCM partitions. D2 is made up of three Gaussian clouds, each with 2000 samples, means of $[10 \ 10]^T$, $[18 \ 20]^T$, $[1 \ 30]^T$ and standard deviations of 3, 2 and 4 respectively.

each applied twice with random initialization for FCM (and hence, equal initializations for PCM) with $c = 3$. The images of the four terminal partitions discovered by these two algorithms again suggest that the fuzzy partitions are equal, and the PCM partitions are equal, and they are. Thus, in accordance with (9), $s_r^*(U_{f,3}, U_{f,4}) = s_r^*(U_{f,3}, U_{f,3}) = s_r^*(U_{f,4}, U_{f,4}) = 0.80$; and $s_r^*(U_{p,3}, U_{p,4}) = s_r^*(U_{p,3}, U_{p,3}) = s_r^*(U_{p,4}, U_{p,4}) = 0.57$.

Examples 1 and 2 suggest that the *softer* the partition, the lower the $s_r^*(U, V)$ value. For crisp partitions, Rand proved [7] that $s_r(U, U) = 1$, thus $s_r(V, V) = 1$ and $0 \leq s_r(U, V) \leq 1$. Entry n_{ij} in the contingency matrix is the number of objects common to clusters U_i and V_j . We calculated it as follows:

$$n_{ij} = \sum_{k=1}^n (U_{i,k} * V_{j,k}). \quad (10)$$

If the partitions are crisp, then $U_{i,k} * V_{j,k}$ is in $\{0, 1\}$, i.e. a value of 1 when the k^{th} data point belongs to the i^{th} cluster in U and the j^{th} cluster in V . When the values $U_{i,k}$ and $V_{j,k}$ are fuzzy and in particular less than 1, the membership values result in even smaller products. Our formulation of the contingency matrix leads to interesting behavior for the four counting formulae, (2a)-(2d), as well as the Rand ratio $(a+d)/(a+b+c+d)$. The most interesting component is the effect on equations (2b) and (2c), i.e. the disagreements.

In order for the Rand to obtain a value of 1, the disagreement cases (2b) and (2c) must be zero. However, from equation (10), we see that as the partitions become *softer*, the support in agreements decreases and evidence in disagreements rises. This means that the Rand ratio will decrease because the value (2b)+(2c) is increasing. This is what we observe for FCM in D1 for U_1 and U_2 and in D2 for U_5 and U_6 . In D1, the (2a)-(2d) values are: $a=5747176$, $b=249975$, $c=249975$ and $d=11749871$. In D2, we find that $a=4039502$, $b=1997238$, $c=1997242$ and $d=9963017$. What we observe is that the (2a) and (2d) values are less for D2 versus D1 (where D2 produces softer partitions than those for D1). We also notice that the (2b) and (2c) values are higher for D2 versus D1, making the ratio $(a+d)/(b+c)$, i.e. $(\text{agreements})/(\text{disagreements})$, for D1 $(a+d)/(b+c) = 35$ and $(a+d)/(b+c) = 3.51$ for D2.

The following normalization is proposed in order to address this behavior,

$$s_r^*(U, V) = s_r^*(U, V) / \max(s_r^*(U, U), s_r^*(V, V)) \quad (11)$$

This normalization is proposed because $0 \leq s_r^*(U, V) \leq \max(s_r^*(U, U), s_r^*(V, V))$. In the case of a crisp U or V , the normalization is 1 and the scaling reduces to Rand's result.

IV. VISUAL ASSESMENT OF (CLUSTER) TENDENCY

The problem of determining whether clusters are present as a step prior to *actual* clustering is called assessment of cluster tendency (algorithm 1).

The visual assessment of (cluster) tendency (VAT) algorithm [18] can be used to help discover if there are clusters in a given data set. The basic concept behind VAT (algorithm 1) is that objects are reordered and the reordered matrix of pairwise object dissimilarities is displayed as an intensity image.

Algorithm 1: Visual Assessment of (Cluster) Tendency (VAT)

0. Input: An $n \times n$ matrix of dissimilarities $[R_{pq}]$

1. Set $K = \{1, 2, \dots, n\}$; $I = J = \emptyset$; $P[0] = (0, \dots, 0)$.

2. Select $(i, j) \in \arg \max_{p \in K, q \in K} \{R_{pq}\}$

Set $P(1) = i$; $I = \{i\}$; and $J = K - \{i\}$.

3. For $r=2, \dots, n$:

Select $(i, j) \in \arg \min_{p \in I, q \in J} \{R_{pq}\}$

Set $P(r) = j$;

Replace $I \leftarrow I \cup \{j\}$ and $J \leftarrow J - \{j\}$.

Next r .

4. Obtain the ordered dissimilarity matrix $\tilde{R}$ using the ordering array P as: $\tilde{R} = R_{P(i)P(j)}$, for $1 \leq i, j \leq n$.

5. Display the reordered matrix $\tilde{R}$.

Clusters may be indicated by dark blocks of pixels along the matrix diagonal. While VAT is used herein for analysis, clustering in ordered dissimilarity data (CLODD [19] and p-specVAT [20]) can be used to address the partitioning and validity questions by automatically creating partitions from the VAT image and providing a validity measure of each partition. In algorithm 1, R_{pq} is the dissimilarity of objects p and q and the vector P is the permuted indices of the n objects.

V. COMPARING POSSIBILISTIC PARTITIONS

In [21], Barni et al. remarked that the PCM has the undesirable tendency to produce coincidental clusters. In [4], Keller and Krishnapuram responded by providing additional insight into the inner workings of the PCM. In particular, they explained both the importance and behavior of the scaling parameters, η_i , and the fuzzifier, q . They also discussed how PCM can find c good clusters from a data set that may have more than c clusters. On the other hand, if the data set has less than c clusters, PCM can still potentially find c good clusters, some of which may be identical. However, how does one algorithmically discover the number of clusters in O as well as say that two clusters are equal?

Discovering the number of actual clusters. It is sometimes helpful to categorize clustering into three different tasks. In the previous section, the problem of *assessing cluster tendency* was briefly discussed. The FCM and PCM algorithms are examples of the second class of clustering problems, *actual clustering*. The third class is *cluster validity*. First, we may use a validity index, $v(U)$, e.g. Dunn's index [22] or the Davies-Bouldin's index [23]. One generally varies an algorithm parameter, e.g. the number of clusters (c), the fuzzifier (q), etc., and $v(U)$ is used to identify a trend such as the maximum or minimum or

Algorithm 2: *Simultaneous Discovery of the Number of Actual Clusters and Coincident Clusters Found by the PCM*

0. Overspecify the number of clusters (c) and obtain a terminal partition U from the PCM

1. Generate the similarity matrix S ($c \times c$) from U

$$S_{pq} = s'_r(V_p, V_q) \text{ for } \forall p, q \in \{1, \dots, c\}$$

$$V_p = \begin{bmatrix} U_{p,1} & \dots & U_{p,n} \\ \max_{k \neq p}(U_{k,1}) & \dots & \max_{k \neq p}(U_{k,n}) \end{bmatrix} \quad (2 \times n)$$

$$V_q = \begin{bmatrix} U_{q,1} & \dots & U_{q,n} \\ \max_{k \neq q}(U_{k,1}) & \dots & \max_{k \neq q}(U_{k,n}) \end{bmatrix} \quad (2 \times n)$$

2. Generate the dissimilarity matrix D from S via $1 - S_{pq}$

3. Run VAT on D

Visually identify the *actual* number of clusters

4. Clusters in D are coincident clusters in the PCM algorithm

“the big jump”. The second group of methods use comparison indices, $s(U, V)$, that compare pairs of partitions. Examples of this group include the crisp Rand [7] and our generalized Rand index [1-2]. If U and V are both in CP (and hence, both obtained by algorithmic means), $s(U, V)$ is an internal criterion, and otherwise, we call it an external criterion.

Since c is unknown in real clustering situations, it is not generally possible to know that it has been overspecified. However, the problem domain often suggests a maximum plausible value for c , and this knowledge can be used to make a realistic overspecification. For example, if you are segmenting MRI images of the brain with c -means clustering, there will certainly not be more than 8 or 10 meaningful regions, so taking c at, say, 15, is a reasonable guess at overspecification.

However, each approach makes certain assumptions and therefore operates ideally for data sets with particular profiles, e.g. VAT and its relationship to single linkage [24]. Even more critical to PCM, these approaches do not solve the cluster correspondence problem. If one of the above methods is used to discover the number of *actual* clusters, there is no guarantee that PCM will find the c unique clusters (due to coincident clusters).

Since the inception of the PCM algorithm, a few extensions have been presented that attempt to avoid coincident clusters. In [25], Timm et al. add a penalizing term to the PCM objective function in order to help repel cluster prototypes. Additionally, de Caceres et al. present an improved way to estimate the cluster bandwidths, which in return appears to help keep prototypes from converging to similar spots [26].

Cluster correspondence (or registration) problem. A naive approach to cluster correspondence is to *directly* compare clusters, i.e. according to some distance of clustering algorithm parameters (i.e. prototypes, covariance matrices, etc.). It is

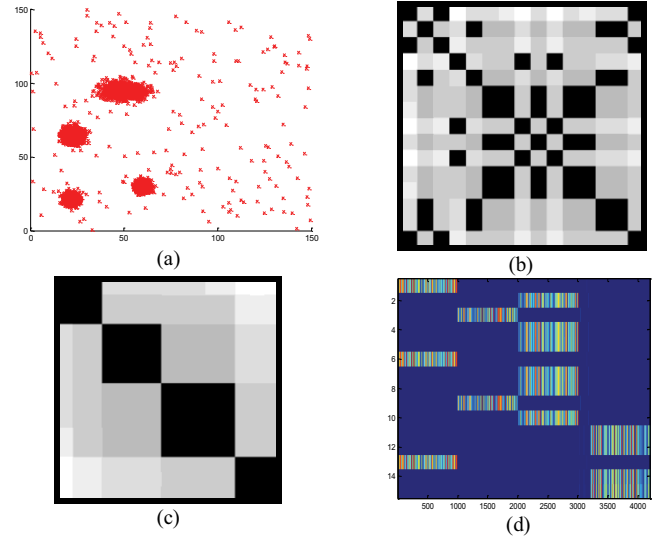


Fig. 3. Data set 3 (D3) and results from algorithm 2. D3 is four Gaussian clouds, each with 1000 samples, means of $[22 \ 22]^T$, $[60 \ 30]^T$, $[23 \ 65]^T$, $[50 \ 95]^T$ and standard deviations of 2, 2, 3 and (6,3) respectively. A set of 200 random points are included. (a) is the object data, (b) is the unordered dissimilarity matrix D for $c = 15$, (c) is the VAT reordering of D showing 4 clusters, and (d) is the possibilistic partition matrix ($15 \times n$) that clearly shows which of the original clusters are coincident. We see that between 3,000 and 3,200 there are 200 objects (noise) that do not belong strongly to any particular cluster.

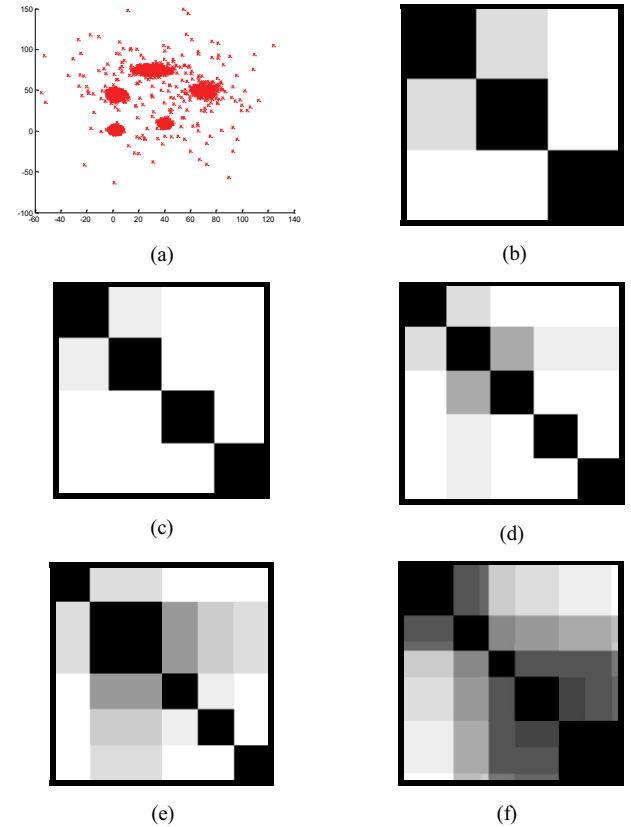


Fig. 4. Data set 4 (D4) and results from algorithm 2. D4 is five Gaussian clouds and noise; $([2 \ 2]^T, 900, 2)$, $([40 \ 10]^T, 500, 2)$, $([3 \ 45]^T, 1000, 3)$, $([30 \ 75]^T, 1200, (6,3))$, and $([70 \ 50]^T, 250, 6)$ for (mean, number of samples, standard deviation). There are 200 random noise points. In (b) $c = 3$, (c) $c = 4$, (d) $c = 5$, (e) $c = 6$, and in (e) $c = 25$.

possible that this approach might work for a limited number of scenarios. One example from the area of computer vision is the color space CIELAB. This space is designed around approximating human perception. Namely, this space makes it possible to interpret distance to a degree and therefore make reasonable working assumptions regarding merging criteria. An example of this is the mean shift clustering algorithm [27]. In [27], CIELAB space is exploited to *merge* mean shift prototypes for image region segmentation and filtering. However, this overly simplistic approach is not theoretically well founded and should not be relied upon in general.

Simultaneous discovery of the number of actual clusters and coincident clusters found by the PCM. Algorithm 2 begins with the overspecification of c . After PCM has terminated, the measure s'_{r*} is used to compare clusters and VAT is used in order to visually discover the number of *actual* clusters and coincident clusters simultaneously. This algorithm supports the claim [4] that coincident clusters are not a shortcoming of PCM. However, s'_{r*} is defined for partitions not clusters. How is it that cluster p can be compared to cluster q in the same partition?

In algorithm 2, a set of possibilistic partitions, $\{V_1, \dots, V_c\}$, are built (where c is the number of clusters in U). The idea is to generate a possibilistic partition ($2 \times n$) for each cluster that contains the membership of every data point in cluster p ($1 \times n$) and the maximum membership value of each data point in the remaining set of clusters ($1 \times n$), i.e. for all $q \neq p$. The value S_{pq} is the similarity, $[0,1]$, between clusters p and q based on the possibilistic partitions V_p and V_q . Next, VAT is used on the dissimilarity matrix D , $D_{pq} = 1 - S_{pq}$, to visually discover the *actual* number of clusters in O and subsequently coincident clusters. Figures 3 and 4 demonstrate this approach.

In figure 3(a), O contains four *actual* clusters and random noise. Figure 3(b) is the dissimilarity matrix D for $c = 15$. Figure 3(c) is the VAT reordering of D , $D_{pq} = 1 - S_{pq}$. Figure 3(c) indicates that there are four *actual* clusters in O and that each *actual* cluster is associated with multiple clusters found by the PCM. Figure 3(d) is the PCM partition matrix. In figure 3(d), rows are clusters and columns are data points (15×4200). The first 1,000 data points belong to clusters $\{1, 6, 13\}$ and the second 1,000 data points belong to $\{3, 9\}$. The next 1,000 data points belong to clusters $\{2, 4, 5, 7, 8, 10\}$. The next 200 data points are the random noise and they do not have a strong membership in any cluster. Finally, the last 1,000 data points belong to clusters $\{11, 12, 14, 15\}$.

In figure 4, O contains five *actual* clusters. When the number of clusters is underspecified, i.e. $c = 3$ in figure 4(b) and $c = 4$ in figure 4(c), PCM finds a subset of the *actual* clusters. However, the VAT image shows that there are three and four clusters respectively and that there are no coincident clusters. The same story is true for figure 4(d), i.e. $c = 5$ and no coincident clusters. However, when the number of clusters is over specified to PCM ($c > 5$), e.g. figures 4(e) and 4(f), algorithm 2 identifies the correct answer, five *actual* clusters. We also discover that each *actual* cluster is accounted for by one or more clusters returned by PCM (coincident clusters). Even one step further, we know which clusters from the PCM map to an *actual* cluster in O .

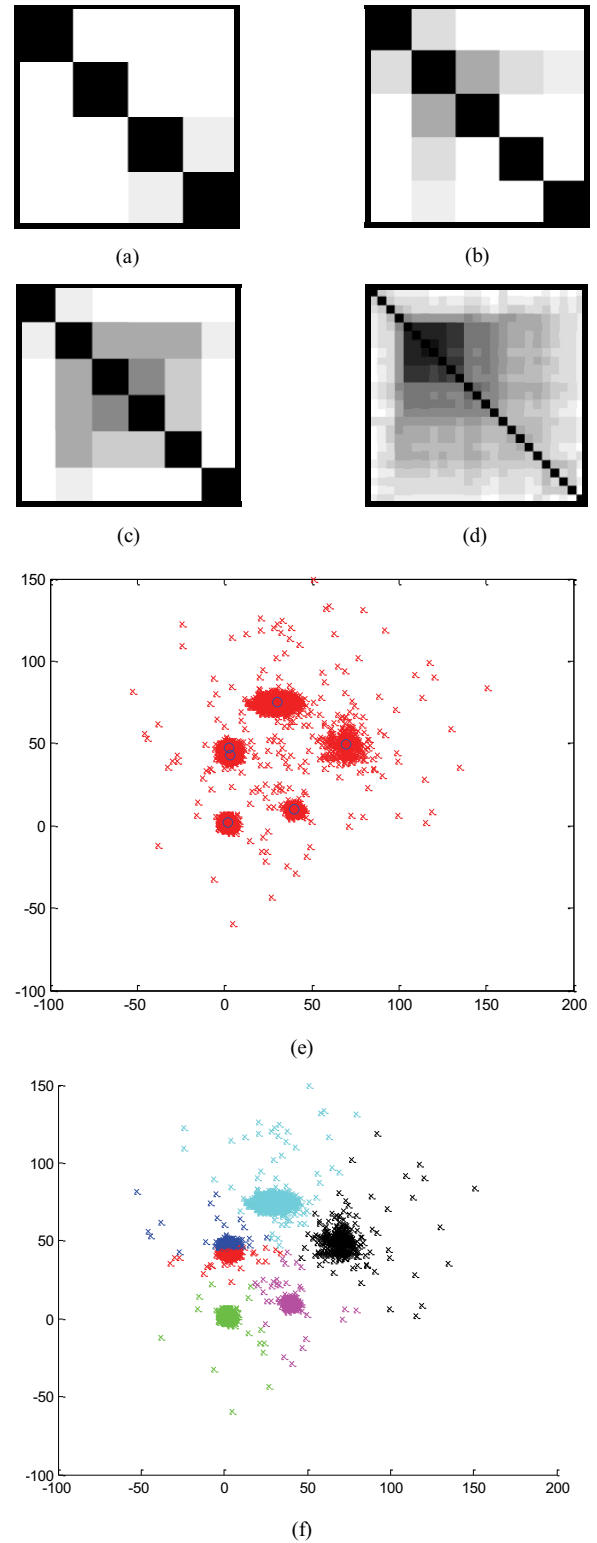


Fig. 5. Data set 5 (D5), which is generated using the same parameters used to generate D4 in figure 4, and the results from algorithm 2. However, in this experiment the FCM algorithm is used in place of the PCM. In (a) $c = 4$, (b) $c = 5$, (c) $c = 6$ and in (d) $c = 25$. In (e), the locations of the prototypes found in (c) are shown as blue circles. In (f), the FCM ($c=6$) membership matrix is hardened and each cluster is color coded.

These experiments demonstrate that when the number of clusters is underspecified to the PCM, algorithm 2 does not identify the correct *actual* number of clusters (instead some value smaller than it). The difficulty is that in a true clustering scenario, we do not know the answer. For example, in figure 4(b), $c = 3$, we would not know if the PCM found the correct number of clusters using algorithm 2. When we attempt $c = 4$, i.e. figure 4(c), we still are unaware if we have found the correct number of clusters. It is not until the specified c is greater than the *actual* number of clusters that we begin to get an indication of the correct number of *actual* clusters. This is the moment in which coincident clusters becomes an advantage. As c is steadily increased, i.e. algorithm 2 is run for $c = 2$, $c = 3$, etc., we eventually notice a *leveling off* in terms of the number of clusters discovered by algorithm 2. Therefore, we recommend testing a range of values for c in algorithm 2 in order to identify the *actual* number of clusters in O . In addition, different initializations can (and often do) lead to different clusterings. Therefore, we recommend running the PCM multiple times, e.g. 10, per c value. For each c , the user should select the largest value for the number of clusters found by algorithm 2.

On a final note, we demonstrate the result of algorithm 2 for FCM instead of PCM (figure 5). We see that when c is less than the number of *actual* clusters, i.e. figures 5(a) and 5(b), the VAT images show that the number of *actual* clusters is c . However, when c is overspecified, i.e. figures 5(c) and 5(d), the VAT image continues to indicate that the number of *actual* clusters is c . We are therefore unable to identify the *actual* number of clusters and the cluster correspondences via algorithm 2. Figure 5(e) is the scatterplot of $D5$ along with the locations of the prototypes. What we see is that when c is overspecified, the FCM tends to splits clusters. This is not the same behavior as the PCM, which produces coincident clusters. That is, coincident clusters in the PCM will have very similar rows in the membership matrix.

VI. CONCLUSIONS

In this article, we extended our prior work by exploring the behavior of the soft Rand for comparing non-crisp partitions. We also demonstrated how the possibilistic Rand and visual assessment of (cluster) tendency (VAT) algorithms can be used to simultaneously discover the number of *actual* clusters and coincident clusters found by the PCM. Our approach supports the claim [4] that coincident clusters are not a shortcoming of the PCM.

In future work, we will investigate how to extend the soft Rand to comparing partitions acquired across different data sets, e.g. samples drawn from the same underlying process. At the moment, our generalization of the Rand requires that U and V share the same object data set, O ; however they need not possess the same number of clusters, $r \neq c$. We will explore next how to extend our approach to cases such as (1) temporal data sets, (2) clustering in unloadable data sets, and (3) the comparison of clustering discoveries found across potentially disjoint (sample) data sets. Namely, case (3) will investigate the question, “how similar are the clustering discoveries in data set A to data set B using clustering algorithms C and D ”?

VII. ACKNOWLEDGEMENT

D. Anderson, J. Keller and M. Popescu would like to acknowledge support from ARO grant 57940-EV and LWI grant LWI-101-022.

REFERENCES

- [1] D. T. Anderson, J. C. Bezdek, M. Popescu and J. M. Keller, “Comparing Fuzzy, Probabilistic, and Possibilistic Partitions,” in *IEEE Transactions on Fuzzy Systems*, 18 (5), 906-918, 2010.
- [2] D. T. Anderson, J. C. Bezdek, J. M. Keller and M. Popescu, “A comparison of five fuzzy Rand indices,” *Intl. Conf. on Info. Processing and Management of Uncertainty in Knowledge-Based Systems*, Germany, 446-454, 2010.
- [3] R. Krishnapuram and J. M. Keller, “A Possibilistic Approach to Clustering,” *IEEE Transactions on Fuzzy Systems*, 1 (2), 98-110, 1993.
- [4] R. Krishnapuram and J. M. Keller, “The Possibilistic C-Means Algorithm: Insights and Recommendations,” *IEEE Transactions on Fuzzy Systems*, 4 (3), 385-393, 1996.
- [5] L. Hubert and P. Arabie, Comparing partitions, *J. Classification*, 2, 193-218, 1985.
- [6] P. Sneath and R. Sokal, *Numerical Taxonomy - The Principles and Practice of Numerical Classification*, W. H. Freeman, San Francisco, CA, 1973.
- [7] W. Rand, “Objective criteria for the evaluation of clustering methods,” *JASA*, 66 (336), 846-850, 1971.
- [8] R. Sokal and C. Michener, “A Statistical Method for Evaluating Systematic Relationships,” *The University of Kansas Scientific Bulletin* 38, 1409-1438, 1958.
- [9] F. Duan and H. Zhang, “Correcting the loss of cell-cycle synchrony in clustering analysis of microarray data using weights,” *Bioinformatics*, 20 (11), 1766-1771, 2004.
- [10] A. Thalamuthu, I. Mukhopadhyay, X. Zheng and G. Tseng, “Evaluation and comparison of gene clustering methods in microarray analysis,” *Bioinformatics*, 22 (19), 2405-2412, 2006.
- [11] D. Wong, F. Wong and R. Wood, “A multi-stage approach to clustering and imputation of gene expression profiles,” *Bioinformatics*, 23 (8), 998-1005, 2007.
- [12] Z. Yu, H. Wong and H. Wang, “Graph-based consensus clustering for class discovery from gene expression data,” *Bioinformatics*, 23 (21), 2888-2896, 2007.
- [13] R. Campello, “A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment,” *Pattern Recognition Letters*, 28, 833-841, 2007.
- [14] H. Frigui, C. Hwang and F. Rhee, “Clustering and aggregation of relational data with applications to image database categorization,” *Pattern Recognition*, 40, 3053 – 3068, 2007.
- [15] R. Brower, “Extending the Rand, adjusted Rand, and Jaccard indices to fuzzy partitions,” *J. Intell. Inf. Systems*, 32, 213-235, 2009.
- [16] E. Hullermeier and M. Rifqi, “A fuzzy variant of the Rand index for comparing clustering structures,” *Proc. IFSA*, Portugal, 1-6, 2009.
- [17] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
- [18] J. Bezdek and R. Hathaway, “VAT: A Tool for Visual Assessment of (Cluster) Tendency,” *Proc. of the Intl. Joint Conf. on Neural Networks*, 2225-2230, 2002.
- [19] T. C. Havens, J. C. Bezdek, J. M. Keller and M. Popescu, “Clustering in ordered dissimilarity data,” *Int. J. Intelligent Systems*, 24 (5), 504-528, 2009.
- [20] L. Wang, X. Geng, J. C. Bezdek, C. Leckie and R. Kotagiri, “Enhanced visual analysis for cluster tendency assessment and data partitioning,” *IEEE TKDE*, 22 (10), 1401-1414, 2010.
- [21] M. Barni, V. Cappellini and A. Mecocci, “Comments on ‘A Possibilistic Approach to Clustering’,” *IEEE Transactions on Fuzzy Systems*, 4 (5), 393-396, 1996.

- [22] J. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact wellseparated clusters," *J. Cybern.*, 3 (3), 32–57, 1974.
- [23] D. Davies and D. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intelligence*, 1, 224–227, 1979.
- [24] T. Havens, J. C. Bezdek, J.M. Keller, M. Popescu and J.M. Huband, "Is VAT really single linkage in disguise?," *Ann. Mathematics and Artificial Intelligence*, 55 (3), 237-251, 2009.
- [25] H. Timm, C. Borgelt, C. Doring, and R. Kruse, "An extension to possibilistic fuzzy cluster analysis", in *Fuzzy Sets and Systems*, 147, 3-16, 2004.
- [26] M. de Caceres, F. Olvia, and X. Font, "On relational possibilistic clustering," in *Pattern Recognition*, 39, 2010-2024, 2006.
- [27] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans. Pattern Anal. Mach. Intelligence*, 24 (5), 603–619, 2002.